

## CAPÍTULO II.



## 2. SISTEMAS GESTORES DE CONTENIDO.

### 2.1 Generalidades.

### 2.2 ¿Qué es un Gestor de Contenido?

### 2.3 Tipos de Gestores de Contenido.

### 2.4 Características de los Gestores de Contenido.

### 2.5 Estructura de los Gestores de Contenido.

### 2.6 Evolución de los Gestores de Contenido.

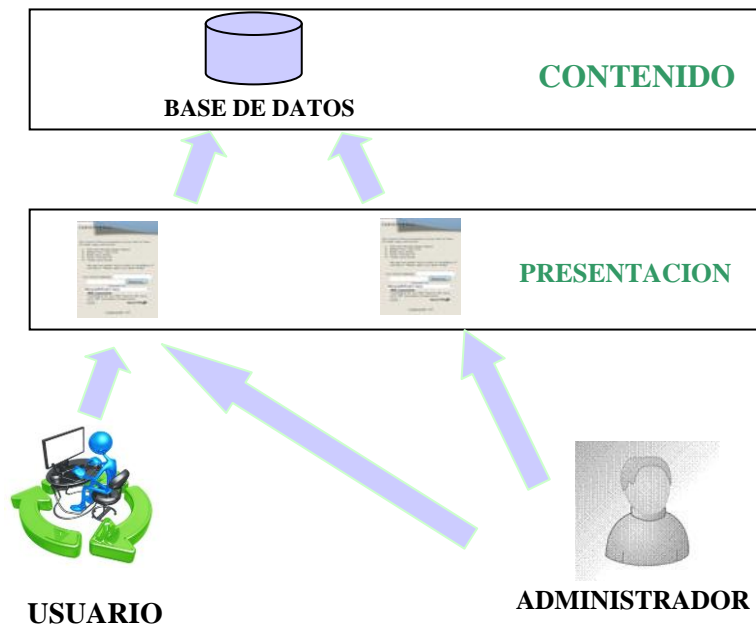
### 2.7 Ventajas de los Gestores de Contenido.

## **2.1. GENERALIDADES.**

El concepto de gestores de contenido no era conocido a inicios de la década de los 90, todas las aplicaciones de este tipo eran realizadas con editores de texto y bases de datos desarrolladas a la medida pero para el año de 1994 la organización *Illustra Information Technology(IIT)* comenzó la utilización de bases de datos como repositorio de contenido con el fin de poder reutilizarlos, no tuvo el éxito deseado pero el traspaso hacia AOL la parte Web y hacia Informix una parte de la base de datos fue este acontecimiento el comienzo de su impulso.

El encargado de iniciar el desarrollo de gestores de contenido fue la empresa **RedDot** que al final de 1994 presentó una versión de **CMS** que incluía base de datos, luego aparecieron los **CMS** de código libre de los cuales sobresale **Typo3** que fue presentado en 1997 y el que publicitó el uso de este tipo de herramientas fue **PHPNuke** utilizado para comunidades de usuarios y su desarrollo empezó en el 2000 y fue el resultado de la reescritura de **Thatware**.

Con esto debemos tener en cuenta antes del surgimiento de los gestores de contenido para la realización de un proyecto Web se presentaban muchas vicisitudes al no tener en cuenta todas las herramientas para su realización, ya que en el inicio se tenía a disposición una serie de editores que con el paso del tiempo se modificaron para tener una buena organización del proyecto Web pero esto no era suficiente por que solo se tomaba en cuenta aspectos de desarrollo en vez de aspectos futuros como es el mantenimiento y la actualización. Es por estos aspectos que se crearon los sistemas gestores de contenidos (**CMS**), teniendo en cuenta que el manejo de corrección de errores es una de sus principales fortalezas en su uso, que además su flexibilidad de manejar el contenido mediante la Web lo definen como una herramienta clave para el desarrollo de sitios Web.



**Fuente:** <http://www.slideshare.net/zarmath/creacion-de-sitios-web-a-medida-con-cms>

**Figura 2.1:** Funcionamiento de un CMS.

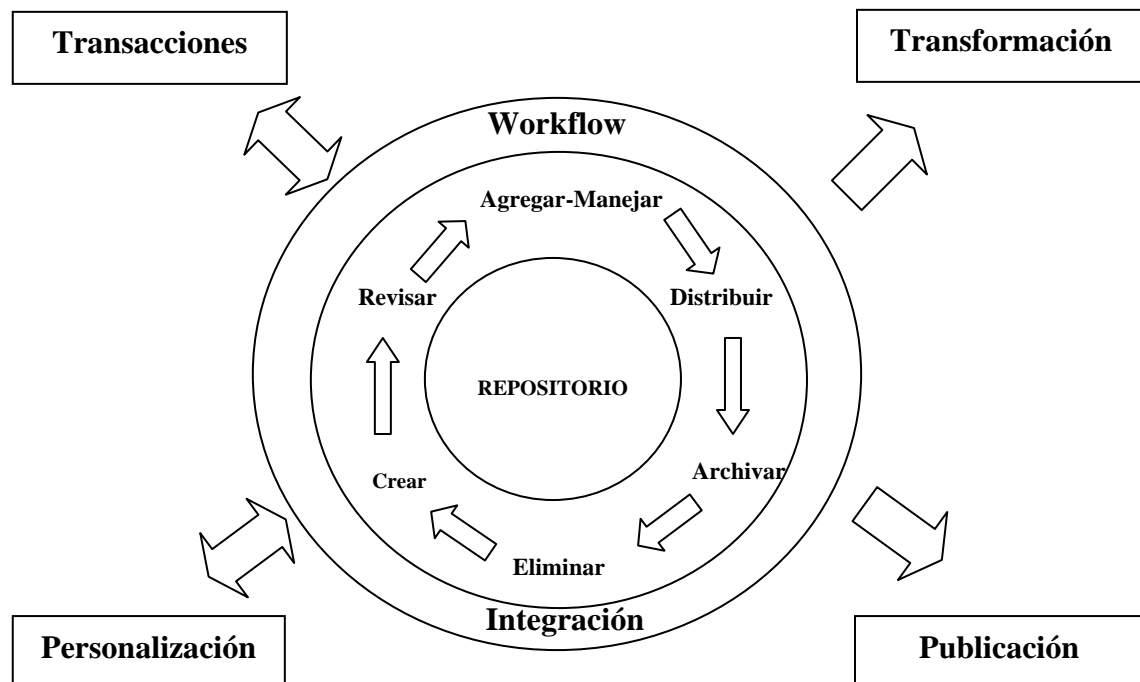
En la **Figura 2.1** podemos ver que el Administrador y los Usuarios mediante el interfaz que posee el gestor de contenido crea contenido que es recolectado por un repositorio en este sentido podemos ver la separación de la presentación con respecto al contenido.

La tecnología utilizada por estos sistemas es la cliente-servidor, en la cual los usuarios que solicitan la información que son los llamados clientes y los servidores que son los computadores que proporcionan dicha información y atienden los requerimientos de los clientes, en esta tecnología se realiza una transferencia de información pero en este caso no se instala ninguna aplicación en el cliente ya que el usuario para poder acceder a este servicio abre simplemente el explorador y el servidor es el encargado de tener actualizada la página ya que esta sujeto a conexión.

## 2.2. ¿QUÉ ES UN GESTOR DE CONTENIDOS (CMS)?

*“Un sistema de gestión de contenidos (en inglés Content Management System, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.” [WWW 11]*

Con esto podemos decir que un **CMS** es una aplicación dedicada a la administración de sitios Web aunque su fin no sea solo ese, teniendo sus principales funcionalidades la creación, gestión, publicación y presentación de contenido, este aplicativo genera páginas dinámicas mediante DHTML (HTML dinámico), en el cual este se conecta al servidor de la página por la petición de un usuario cualquiera y el contenido de esta petición es extraído de una o varias bases de datos de dicho servidor y además el contenido es manejado de manera independiente con respecto a la presentación.



**Fuente:** <http://infonautica.net/santiago/gestcont.pdf>

**Figura 2.2:** Ciclo de vida del contenido en un Gestor de Contenidos.

Para poder comprender y utilizar este tipo de herramientas se presenta la **Figura 2.2** en la cual podemos ver que el contenido guardado en el repositorio puede ser: creado, revisado, agregado, distribuido, archivado y eliminado, además se puede utilizar un sistema de comunicación para realizar un flujo de trabajo en grupo para realizar transacciones y también para transformar, personalizar y publicar el contenido.

Las funciones principales de un gestor de contenido son:

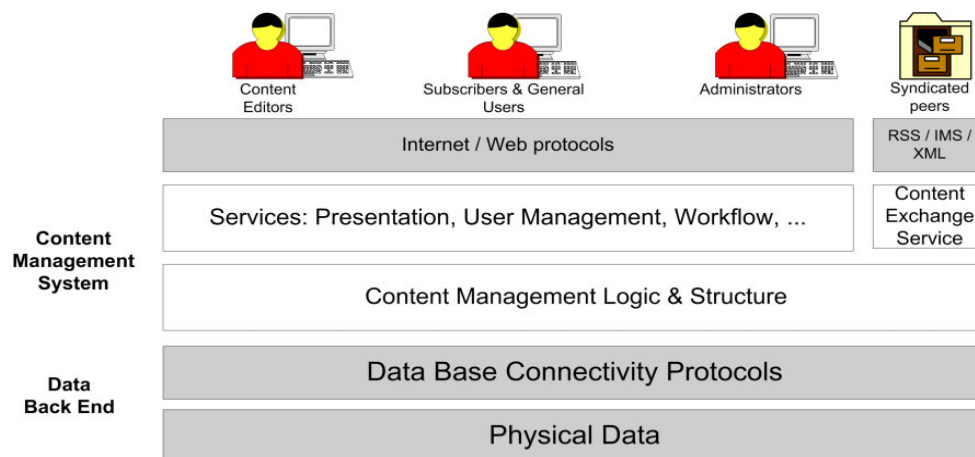
- La **creación de contenido** es un aspecto esencial de un **CMS** ya que un usuario sin conocimientos técnicos de programación puede realizar una publicación mediante un editor de texto **WYSIWYG** y así concentrarse únicamente en la realización del texto verificando el resultado final con dicho editor que no se basa únicamente en él ya que se puede integrar otras herramientas de edición de documentos **XML**.
- El **manejo del contenido** se basa en la manipulación de la base de datos en donde se encuentran los documentos creados como también todos los datos relacionados con la Web, permitiendo organizar la misma y editarla para poder culminar el proceso de trabajo(*workflow*) en la realización de publicaciones.
- Las **publicaciones** de las páginas se las realiza por secciones las cuales tienen una fecha de inicio y otra de caducidad de la publicación, teniendo en cuenta que al pasar este periodo queda para futuras consultas en el sitio. El contenido esta separado al aspecto de la Web permitiendo separar los dos para así no hacer cambios en dicho contenido.
- La **Presentación** es un aspecto esencial que es manejado por los **CMS** ya que al basarse a las normas internacionales de accesibilidad **WAI**<sup>16</sup>, puede

---

<sup>16</sup> **WAI**.- La **Web Accessibility Initiative** (WAI) o **Iniciativa para la Accesibilidad Web** es una rama del World Wide Web Consortium que vela por la accesibilidad de la Web.

adaptarse al perfil del usuario como también a las plataformas existentes e idioma. Permite gestionar módulos internos como externos y la capacidad de añadir enlaces automáticos, menús, etc.

Para ver la estructura básica de un gestor de contenidos presentamos la siguiente figura:



Fuente: <http://www.recercat.net/bitstream/2072/63749/1/Pascual.pdf>

**Figura 2.3:** Estructura general de un CMS.

En la **Figura 2.3** podemos ver que se trata de un modelo de tres capas que se compone de tres elementos principales:

- **Capa de base de Datos.-** Es donde se almacena el contenido que generalmente es una base de datos relacional (Relational Database Management Systems, RDBMS).
- **Capa de Lógica.-** Aquí se encuentra las funcionalidades que se ofrecen a los usuarios y administradores. En esta capa se localiza la estructura lógica de nuestro contenido y se manipula los elementos del contenido para organizarlos antes de su presentación, con esto se maneja el contenido de manera centralizada.

- **Capa de cliente.-** Es la interfaz ofrecida para visualizar el contenido y otros servicios que se los representa en un navegador.

### **2.3.TIPOS DE CMS.**

Existen dos formas de agrupar a los sistemas de gestión de contenido y se basa en su utilización, su costo y el lenguaje en que se encuentran estructurados.

#### **2.3.1. Según su utilización.**

Dada la multitud y diferentes **CMS** existentes podemos agruparlas en las siguientes categorías:

- **Portales (CMS).-** Este tipo de sistemas llamados **WCMS** (*Web Content Management Systems*) son comúnmente utilizados para manejar sitios con actualizaciones periódicas y gran cantidad de información y que por lo general son utilizados por usuarios sin conocimientos de programación. Estos sistemas tienen un *back* donde se administra el sitio y un *front* donde se ve el resultado de la realización del contenido que va dirigido a atender las necesidades de sus usuarios en búsquedas de contenido, construcción de comunidades virtuales, etc. Ejemplos claros de este tipo de **CMS** son **Drupal, Joomla, Phpnuke**, etc.

Existen dos tipos de portales:

- **Horizontales.-** Son sitios que brindan información de todo tipo ya que son de carácter universal y son de propósito general y a la vez sus usuarios se identifican por tener perfiles muy variados y acceder a diferentes secciones del sitio.
- **Verticales.-** Son sitios que dedican sus funcionalidades en contenidos especializados y solo para cierto tipo de perfiles de público en este caso la información es de utilidad para personas

que cumplan con dicho perfil, es el caso de los sitios de especialidades académicas.

- **Blogs.-** Las bitácoras con sistemas de gestión de contenido son aplicaciones que nos permiten manejar entradas de texto, fotos y audio que tienden a tratar un asunto en particular. La naturaleza de los blogs es individual pero existen comunidades dirigidas a varias personas como también grandes comunidades virtuales basadas en blogs, algunos ejemplos de estos **CMS** son **Serendepity, Movabletype**.

En estos sistemas el contenido es organizado de forma cronológica y en categorías es decir en la página principal de una bitácora se tendrán los últimos contenidos enviados organizados en su respectiva categoría es decir semiestructurada.

- **Wikis.-** La naturaleza de estos sistemas gestores es permitir la actuación de los usuarios y que estos aporten con sus conocimientos. Este tipo de sistemas fueron publicitados por Wikipedia que es la esencia de trabajo en colaboración, pero que por su sencillez es utilizado para la creación de simples páginas. Estos sitios que el contenido es editable por los usuarios, estos sistemas permiten establecer plantillas e integrar funcionalidades logrando sitios muy dinámicos pero con la debilidad que solo se puntualiza en la edición del contenido y no en la apariencia. En la mayoría de estos sistemas el historial de edición de los documentos y sus metadatos es guardado para poder recuperar versiones anteriores por posibles daños, además la información es organizada en nodos que son nada más que los documentos enlazados mediante enlaces que siguen un formato por ejemplo el uso de corchetes [] para crear enlaces explícitos y enlaces externos si el sistema lo permitiese. Las colaboraciones pueden ser esporádicas y espontáneas por la posibilidad de dejar la Web abierta



para que los colaboradores corrijan posibles errores o mejoren el contenido. Un ejemplo de estas aplicaciones son **MediaWiki** (PHP CMS) que permite que todos puedan modificar el contenido y **TikiWiki** (PHP).

- **e-Commerce.-** Estos sistemas de gestión de contenido se encargan de realizar transacciones comerciales electrónicas en donde los contenidos creados son los artículos que están siendo comercializados, y así realizar las compras por Internet. Un ejemplo de estas tiendas en línea es **OsCommerce**<sup>17</sup>.
- **Foros.-** Con estas aplicaciones podemos participar libremente en debates de un asunto en concreto, con el objetivo de intercambiar conocimientos, información, conversar, etc. Esta es la prueba más sencilla del funcionamiento de un CMS ya que aquí se puede manejar el contenido de forma rápida y sencilla poniendo como aspecto primordial la participación, un ejemplo de este tipo de sistemas es **phpBB**<sup>18</sup>.
- **E-learning.-** Estas aplicaciones se encargan de brindar cursos a distancia y permiten a los docentes llevar a cabo exámenes, foros, tutorías, salas de trabajo en común, chat, etc. Las funcionalidades de estos CMS incrementan el grado de conocimiento y el desempeño de los integrantes mediante el uso de las tecnologías de Internet. **Moodle**<sup>19</sup> es la herramienta que al momento es un ejemplo claro de este tipo de sistemas.
- **Galería de Imágenes.-** Estos sistemas se utilizan para manejar imágenes con muchas opciones de configuración que

---

<sup>17</sup> [WWW12], <http://www.oscommerce.com/>

<sup>18</sup> [WWW 13], <http://www.phpbb.com/>

<sup>19</sup> [WWW 14], <http://moodle.org/>

fundamentalmente se preocupan de organizar estas imágenes en carpetas, subcarpetas, series, subseries, etc. Convirtiéndose en un repositorio gráfico, un ejemplo de este tipo de sistemas es **Gallery**<sup>20</sup> con muchas opciones útiles.

- **Gestores de Documentos (DMS).**- Estos se encargan de manejar gran cantidad de documentos electrónicos, que pueden ser utilizados libremente por los usuarios. Dentro de los DMS, no solo tenemos que situar a las herramientas que nos permiten gestionar archivos digitales, también las que permiten gestionar las colecciones de bibliotecas, hemerotecas, fonotecas. Algunos ejemplos de este tipo de sistemas son **Bitcubo**<sup>21</sup>, **DocShare**<sup>22</sup>, etc.

### 2.3.2. Según la propiedad del código.

El código fuente de la aplicación es propiedad del desarrollador. No permite desarrollos de terceras partes, ajenas o sin permiso.

#### 2.3.2.1. CMS Comerciales y de código abierto.

Esta división se la realiza por el tipo de licencia que posee la aplicación, ya que las empresas protegen sus activos y no permiten que los usuarios especializados puedan manipularlo y por otro lado tenemos los de código abierto y gratuitos que son realizados por usuarios o grupos de usuarios que permiten el acceso total al código fuente.

Antes de seguir con esta categorización se presenta las diferencias que existen entre Software libre y Open source:

---

<sup>20</sup> [WWW 15], <http://gallery.menalto.com/>

<sup>21</sup> [WWW 16], <http://www.bitcubo.com/>

<sup>22</sup> [WWW 17], <http://www.docshare.es/>

- El Software libre tiene una visión moral en la cual “el software debería ser libre”, mientras que el *Open source* tiene una visión práctica en la cual “*el software es mejor si su código es abierto*”.
- El Software libre no significa “software gratis” y sus bases morales permiten copiar, distribuir, estudiar y modificar sin ninguna restricción y si no se puede realizar esto se considera inmoral, mientras que el Open source se refiere al acceso al código fuente para ser promovido para mejorar su calidad. *Open Source* opina que se puede utilizar software *Open Source* y *no Open Source* cuando uno desee conveniente.

El acceso al código fuente hace posible la personalización como también la corrección de errores y el desarrollo de nuevas funcionalidades, así logrando estar en constante actualización que va más allá de la existencia de las empresas creadoras.

Existen algunas empresas que permiten acceder al código fuente con previo pago de una licencia y que en la mayoría de casos sin derechos de modificación en dicho código fuente.

En este sentido los **CMS** de código abierto se caracterizan por su flexibilidad y por tratar como en los comerciales de ser estables y coherentes al ser manejados por grupos o empresas. Estas herramientas de código abierto son utilizadas por no tener que pagar ninguna licencia, aunque en algunos casos esto no podría ocurrir.

Los CMS de carácter comercial por lo general el soporte en caso de actualizaciones y problemas presentados tiene un costo elevado, mientras tanto en los de código abierto este mismo soporte puede ser realizado por las distintas comunidades de usuarios que intercambian información, pero también existen empresas que ofrecen soporte a **CMS** de código abierto. La documentación en los

**CMS** de código abierto por lo general es mínima o en algunos casos su sustentación documental ha sido mal realizada.

Los módulos realizados por terceros es una de los problemas urgentes a solucionar ya que pueden dar las funcionalidades deseadas pero la documentación presentada no es una guía para poder implementar claramente estos módulos. Algunos ejemplos de gestores de código abierto son **Drupal** y **Joomla**, mientras que los ejemplos representativos de gestores comerciales son **Vignette**, **CMPortalSolution** y **Websphere EIP**.

Código Abierto - Comerciales	
Open Source	Closed Source
• Uso gratuito	• Licencias de uso
• Libre acceso al código.	• Acceso restringido al código.
• Comunidad de usuarios	• Una empresa controladora.
• Intercambio de información.	• Soporte profesional
• Documentación a veces escasa	• Documentación correcta

**Fuente:** Adaptación personal.

**Tabla 2.1:** Gestores de código abierto y comercial.

### 2.3.3. Por el lenguaje de programación empleado.

Los gestores de contenido pueden ser agrupados de acuerdo al lenguaje de programación en el cual fueron escritos y algunos ejemplos pueden ser los siguientes:

#### **Java:**

- **jAPS**(<http://www.japsportal.org>)
- **Liferay**([www.liferay.com/](http://www.liferay.com/))
- **DSpace** (<http://www.dspace.org/>)

- **Fedora**(<http://www.fedora-es.com/>)
- **Nuxeo EP**(<http://www.nuxeo.org/>)
- **Magnolia**(<http://www.magnolia-cms.com/>)
- **Hippo CMS**(<http://www.onehippo.com/>)
- **Calenco**(<http://ww.calenco.com/es/home.html>)
- **Polopoly**(<http://www.atex.com/>)
- **IBM Lotus Web Content Management** (<http://www-01.ibm.com>)
- **Day Communiqué WCM**(<http://www.day.com/day/en.html>)
- **Jarimba**([www.jarimba.org/](http://www.jarimba.org/))
- **Vignette**(<http://www.vignette.com/es>)

**PHP:**

- **Drupal**(<http://drupal.org/>)
- **CMS Made Simple**(<http://www.cmsmadesimple.org/>)
- **Joomla!**( <http://www.joomla.org/>)
- **Mambo**(<http://www.mamboserver.com/>)
- **PHP-Nuke**(<http://phpnuke.org/>)
- **TikiWiki**(<http://info.tiki.org/Tiki+Wiki+CMS+Groupware>)
- **TYPO3**(<http://typo3.com/>)
- **WordPress**(WordPress)
- **Xoops**(<http://www.esxoops.com/>)
- **Zikula**(<http://zikula.org/>)
- **Jadu**(<http://www.jadu.co.uk/>)
- **ExpressionEngine**(<http://expressionengine.com/>)
- **Accrisoft Freedom**(<http://www.accrisoft.com/>)
- **CMS 10**(<http://www.cms10.net/>)
- **Dim Works CMS**(<http://dimworks.org/dim/>)
- **Content-SORT**(<http://www.content-sort.com/>)
- **Prodigia Easy Site Manager**([www.easysitemanager.com](http://www.easysitemanager.com))
- **PipePS**(<http://iagofg.com/pipeps.htm>)

#### **ASP.NET:**

- **DotNetNuke Community Edition** (<http://www.dotnetnuke.com/>).
- **Umbraco** (<http://umbraco.org/>).
- **mojoPortal** (<http://www.mojoportal.com/>).
- **Kentico CMS** (<http://www.kentico.com/home.aspx>).
- **SharePoint Server** (<http://office.microsoft.com/es-es/servers/>).
- **Telligent Community** ([http://telligent.com/products/telligent\\_community/](http://telligent.com/products/telligent_community/)).
- **Ektron CMS400.NET** (<http://www.ektron.com/Products/Web-CMS/Web-Content-Management/>).
- **Quantum Art QP7** (<http://www.quantumart.com/>).
- **webControl CMS**  
(<http://soluciones.grupointermark.com/webcontrol/portal.do>).

#### **2.4.CARACTERÍSTICAS DE LOS CMS.**

La diferencia de un **CMS** a sus predecesores es su facilidad de utilización y sus características principales son:

- **Accesibilidad.-** La posibilidad de poder acceder desde cualquier computadora de la red, sin necesidad de instalar programas adicionales.
- **Seguridad.-** Es una de las características principales que debe tener un **CMS** ya que en Internet se debe tener planes de seguridad específicos para posibles ataques ya que estos sistemas manejan nuestros documentos, archivos, cuentas, etc. El factor de seguridad de los **CMS** trata de corregir posibles fallos de seguridad tanto para los administradores, redactores y catalogadores pero también para los usuarios que diariamente se sirven de estos sistemas. Todas las deficiencias de seguridad necesitan atención ya que al integrar nuevos módulos o extensiones también se extiende una puerta de entrada a ataques.

- **Escalabilidad y extensibilidad-** Inclusión de nuevas funcionalidades en el Web. Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un **CMS** eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la Web. El sistema puede crecer y adaptarse a las necesidades futuras.

La flexibilidad y escalabilidad que tienen los **CMS** hacen que sean una herramienta muy solicitada que además al tener un cúmulo de herramientas de completa gratuidad permiten su proliferación y mejoramiento.

- **Separación del Contenido y del Diseño.-** Estos sistemas tienen absoluta independencia de la presentación con respecto al contenido esto quiere decir que los cambios que se realizan no presentan la necesidad de cambiar o editar el contenido esto se logra con los estándares CSS, WAI, XML, RSS.
- **Administración y perfiles de usuarios.-** Los **CMS** manejan a los usuarios restringiendo sus capacidades según el rol que estos desempeñen. Para el ejercicio de las funciones de estos gestores se deben establecer roles en los cuales cada rol tiene permisos para realizar una función en particular con esto ponemos los roles principales que se deberían tener:
  - **Administrador.-** Es el rol que tiene acceso absoluto a todas las funcionalidades y módulos del sitio, que puede estar a cargo de una persona que no tiene conocimientos de programación.
  - **Redactor.-** Es un rol que esta dirigido a las personas que crean el contenido y a veces se encargan de su catalogación.

- **Catalogador.-** Este rol trata de la persona o personas que analizan el contenido que es creado por los usuarios o redactores y con esto indicar a que categoría pertenecen los datos o metadatos.
- **Usuarios registrados.-** En este rol se integran los usuarios anónimos que son moderados por los redactores que se encargan de catalogar la información obtenida de estos usuarios y sus atributos serán asignados por el administrador.
- **Intuitivo.-** Su interfaz gráfica debe ser fácil de entender y usar, sin complicadas opciones y así manejar gran volumen de contenido en forma ágil aumentando los tiempos de servicio y así incrementado la productividad cuando se navega en el sitio Web.

*“Navegar por un sitio Web, la navegación global y por las secciones de un sitio Web debe dar a los visitantes la seguridad de que saben donde se encuentran y cómo ir donde quieren llegar. La navegación pronto se hace intuitiva si se utiliza un tratamiento, colocación y comportamiento consistentes.” [Libro 11]*

- **Documentado.-** Uno de los aspectos más importantes es tener un soporte legible y que este disponible a los usuarios, sabiendo que el usuario se siente seguro al formar parte de una comunidad que esta continua actualización y brinda apoyo a sus participantes.
- **Cumple los estándares.-** Los sistemas de gestión deben cumplir los estándares establecidos para sitios Web ya que deben ser accesibles a la mayoría de usuarios y navegadores que se encuentren en el mercado.
- **Motor de búsqueda.-** Facilidad de hacer búsquedas en todo el sitio.
- **Metadatos.-** Asignar metadatos, sea automática o manualmente, a los contenidos para una mejor recuperación.



- **Lenguaje.-** Posibilidad de personalizar el sistema en el lenguaje nativo de los usuarios.
- **Trabajo en grupo.-** Colaboración entre los integrantes de la organización, sincronización de trabajo y actividades.
- **Gestión de documentos.-** Acceso a documentos, tanto internos como externos, por medio de enlaces, autoría, fecha de publicación, etcétera.
- **Flujo de trabajo.-** Mantiene el control de los contenidos, ingreso, alta, baja, perfiles de usuarios, entre otros aspectos.
- **Base de datos.-**Facilidad de organización, almacenamiento, búsqueda y recuperación de la información del sitio.
- **Licencia:** Los términos de uso.
- **Costo:** Comercial o gratuita.

## **2.5. ESTRUCTURA DE UN CMS.**

La estructura de un sistema de gestión de contenidos se divide en 3 partes que vemos a continuación:

- **Base de Datos.-** Es donde se almacena todo el contenido y también todos los parámetros que competen a la configuración en cuanto a categorías, organización, usuarios, claves, etc. Las bases de datos más utilizadas en este tipo de sistemas son **MySql** y **Postgres**.
- **Programación.-** Esta parte se encuentra en el directorio de la Web y estos archivos al ser ejecutados solicitan información mediante el navegador y visualizan esta información en los lugares predeterminados, comúnmente el lenguaje de programación es **Php**.
- **Diseño.-** Esta parte también se encuentra en el directorio del sitio y es la maqueta en la cual se inserta el contenido que es traído de la base de datos

por la programación, para esto se utiliza el estándar **CSS** que es complementado con otros como **jquery**, **javascript** y **Ajax**. Esta parte se distingue por disponer de diferentes diseños de plantillas que se cambian con facilidad.

- *CSS (Cascading Style Sheets)*.- Es una herramienta para poder diseñar la presentación e incluso la pronunciación por medio de un aparato de lectura permitiendo así tener control total del estilo y del formato del contenido.

Se puede dar estilo de presentación a documentos HTML y XML, y se puede separar por parte del desarrollador el estilo de múltiples páginas vinculadas, es decir los cambios que se pudieran realizar se los realiza a todas estas páginas.

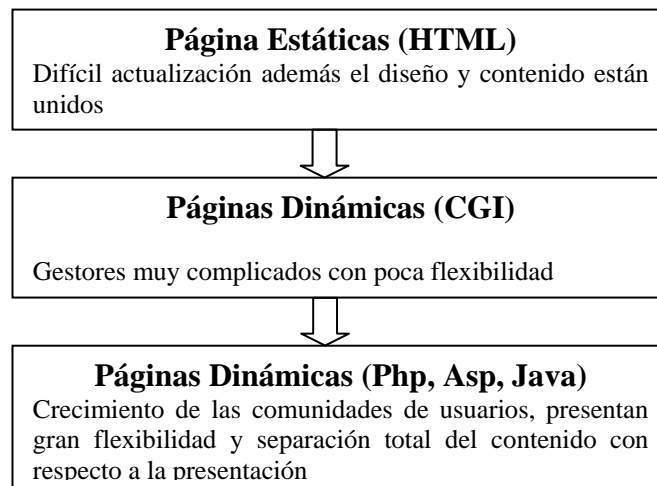
El funcionamiento básico de CSS es de definir una sintaxis especial para la presentación y esta sintaxis la aplicaremos en:

- Todas las páginas que conforman el sitio con una sola definición.
- En un documento HTML o página definiendo la forma de presentación en la cabecera.
- En partes de documentos, aplicando estilos visibles en una parte de la página.

Todas estas partes son imperceptibles para su distinción ya que observamos el trabajo final de su interacción esto es lo que se denomina renderización. Se puede editar cada una de estas partes sin ningún problema por ejemplo se puede instalar una nueva plantilla de diseño con el mismo contenido que se tenía antes.

## 2.6. EVOLUCIÓN DE LOS CMS.

Al inicio cuando se dio a conocer la navegación se tubo *páginas estáticas* que eran construidas en lenguaje HTML, y que presentaban pocas funcionalidades solo enlaces ya que se trataba de textos planos que iban acompañados de contenido multimedia como pueden ser videos y sonidos.



**Fuente:** <http://www.slideshare.net/delineas/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms>

**Figura 2.4:** Evolución de los CMS.

El *Common Gateway Interface* CGI protocolo de comunicación entre el servidor Web y el navegador. Se utiliza habitualmente para contadores, formularios, chats, buscadores. Se utilizo realizar páginas dinámicas de servidor que tubo su declive por la pesada carga que daba a los servidores, el lenguaje Perl fue la base de estos sistemas pero también fueron desarrollados en otros lenguajes como C, C++ y Visual Basic.

El funcionamiento básico de un CGI es el siguiente:

1. Petición HTTP, en este requerimiento pueden venir datos a través de la URL o por un formulario.
2. El servidor ejecuta el CGI y compromete los recursos requeridos.

3. Se construye la salida de los datos requeridos mediante el CGI construyendo etiquetas HTML.

El problema de los CGI es que el código es de difícil comprensión debido a la mezcla de código y también por que cada CGI ocupa su propio espacio de memoria.

Luego de los CGI surgieron las páginas dinámicas que incluían efectos especiales y funcionalidades que se logran con la utilización de otros lenguajes de programación de mayor complejidad y versatilidad. HTML es más que un lenguaje descriptivo que solo permite dar formato y enlaces al contenido y no permite realizar ningún tipo de cálculo o interactuar con una base de datos, es por eso que surgió la necesidad de utilizar otros lenguajes que aportaban a estas carencias y podían responder de manera más inteligente las exigencias de los navegadores y con esto automatizar tareas que con HTML serían tediosas y algunos casos imposibles de realizar.

### **2.6.1. Lenguajes de Programación de Páginas Dinámicas.**

Los principales lenguajes de programación para la realización de páginas dinámicas son:

#### **2.6.1.1. JavaScript.**

*“JavaScript es un lenguaje de guiones desarrollado por Netscape y diseñado para que sea estructuralmente similar al lenguaje de programación Java” [Libro 09]*

Este lenguaje no necesita compilación, es similar a Java pero con la peculiaridad de no ser orientado a objetos, el código de este lenguaje es integrado a la página con un estándar llamado **DOM** que es el *Modelo de Objetos del Documento*.

Sus principales características son:

- Claro y confiable en su interpretación
- Usado como complemento con otros lenguajes mas elaborados puede ser de gran refuerzo.
- Presenta problemas de seguridad por estar el código visible.
- Es ejecutado en la parte del cliente que esto conlleva a tener un servidor con menor carga, pero si se lo realiza en el lado del servidor puede ser un inconveniente por la cantidad de usuarios.
- Debe completarse su descarga para su utilización que presenta una ventaja considerable.

#### **2.6.1.2.Lenguaje PHP.**

*“PHP es un lenguaje de programación diseñado para funcionar con HTML pero, al contrario que HTML, tiene prestaciones de procesamiento de datos” [Libro 08]*

En la actualidad es el lenguaje que más se utiliza para la creación de sitios Web, es un acrónimo recursivo que significa “PHP Pre-procesador de Hipertexto”, surgió en el año de 1995 y es un lenguaje de script que es interpretado en el lado del servidor y que sirve para generar páginas dinámicas y que se encuentra incrustado en páginas HTML. Este lenguaje no necesita ser compilado antes de su ejecución y la mayoría de su sintaxis fue tomada de **C, Java y Perl**.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como **UNIX** (y de ese tipo, como *Linux* o *Mac OS X*) y *Windows*, y la creación de sitios Web con sistemas de gestión de contenidos (**CMS**) con un

aspecto personalizado puede interactuar con los servidores de Web más populares ya que existe en versión **CGI**<sup>23</sup>, módulo para **Apache**<sup>24</sup> e **ISAPI**<sup>25</sup>.

La diferencia que existe entre **CGI** e **ISAPI** es que en **CGI** los procesos son manejados por separado pero esto se vuelve un problema si existen un gran número de peticiones, mientras que los dll (Dinamic Link Librerries) de **ISAPI** pueden desenvolverse en el mismo espacio del proceso y además estos dll se comunican con los componentes de extensión en vez de variables de ambiente como lo hace **CGI**.

### Sintaxis:

La sintaxis utilizada para incorporar código PHP es la siguiente:

```
<?
    $mensaje = "Hola";
    echo $mensaje;
?>
```

*También puede usarse:*

```
<?php
    $mensaje = "Hola";
    echo $mensaje;
?>
```

Las principales características de este lenguaje son:

- La asimilación del lenguaje es relativamente fácil y rápida.
- Soporta orientación a objetos.

---

<sup>23</sup> **CGI.**- es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

<sup>24</sup> **Apache.**- Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

<sup>25</sup> **ISAPI.**- Es una interfaz de programación de aplicaciones (API) para el servidor web de Microsoft, IIS (Internet Information Server).

- Se lo puede utilizar en las diferentes plataformas.
- Se conecta a la mayoría de gestores de base de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, etc.
- Es modular.
- Suficiente información.
- Libre de pago.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- Para su utilización debe estar instalado en el servidor.
- El trabajo en ejecución es realizado por el servidor que conlleva una ineficiencia si el número de usuarios aumenta considerablemente.
- La comprensión del código se dificulta por estar incrustado en HTML.
- Al estar instalado en el servidor puede realizar diversas tareas que esto presenta una deficiencia de seguridad si no se toma un buen planeamiento en este ámbito.

### **2.6.1.3.Lenguaje ASP.**

*“Active Server Pages (ASP), es una tecnología propietaria de Microsoft. Se trata básicamente de un lenguaje de tratamiento de textos (scripts), basado en Basic, y que se denomina VBScript (Visual Basic Script). Se utiliza casi exclusivamente en los servidores Web de Microsoft (Internet Information Server y Personal Web Server). Los scripts ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y Javascript para realizar tareas interactivas y en tiempo real con el cliente.” [WWW 18]*

#### **Sintaxis:**

```
<% Response.Write("Hola mundo") %>
```

Las principales características de este lenguaje son:

- Es de fácil comprensión ya que utiliza Visual Basic Script.
- Se comunica de manera óptima con el gestor de base de datos SQL Server.
- Se puede complementar con JScript de Microsoft.
- No se puede organizar la codificación.
- La realización de funciones demandan muchas líneas de código.
- Su utilización representa pago al propietario.
- Costo de hosting elevado.

#### 2.6.1.4.Lenguaje ASP.NET.

Es el sucesor de ASP y trata de solucionar las limitaciones que presentaba su predecesor y poder realizar además de sitios Web pequeños también realizar grandes aplicaciones en las que también se puede utilizar C#, VB.NET o J#, el cambio de extensión es aspx y para su normal funcionamiento necesita igualmente IIS, pero adicionalmente necesita un Framework .Net.

*“ASP.NET, los servicios de plataforma que permiten programar aplicaciones web y Servicios Web en cualquier lenguaje .NET, con cualquier característica de la librería de clase de .NET.” [Libro 07]*

#### Sintaxis:

```
%@ Page Language="VB" %>
<html>
  <body>
    <form runat="server">
      </form>
    </body>
  </html>
```



Las principales características de este lenguaje son:

- Es su totalidad es orientado a objetos.
- Esta dividida la capa de diseño con el código.
- Mantenimiento ágil para aplicaciones de gran tamaño.
- Mejores tiempos de respuesta por parte del servidor.
- Aspectos de seguridad más elaborados.
- Inconvenientes por mayor carga de recursos.

#### 2.6.1.5.Lenguaje JSP (Java Server Pages).

*“Para hacer la creación de contenido Web dinámico todavía más fácil, Java ha introducido las Páginas de Servidor también llamadas **páginas JSP**. Mientras que los Servlets pueden requerir un conocimiento bastante extenso de Java para su creación.” [Libro 10]*

*“Los Servlets, por ellos mismos, no son aplicaciones autónomas; se cargan en la memoria mediante un contenedor de servlet. Este contenedor de servlet funciona entonces como un servidor Web, recibiendo peticiones HTTP de los navegantes Web y pasándolas a los servlets.” [Libro 10]*

JSP es un lenguaje multiplataforma que sirve para la realización de páginas dinámicas al ser realizado en Java y se ejecuta en lado del servidor su elaboración estuvo a cargo de Sun Microsystems en este sentido posee características de ASP.Net con la particularidad de tener un motor basado en los servlets de Java y para que funcione necesita la instalación del servidor Tomcat.

#### Sintaxis:

```
<%= new java.util.Date() %>
```

Las principales características de este lenguaje son:

- Separación de la lógica de programación.
- Las páginas deben ser compiladas en la primera petición de la misma.
- Se puede separar la parte estática de la dinámica en la creación de páginas.
- Se puede incrustar el código en páginas HTML.
- Se puede controlar los parámetros del servlet mediante las directivas.
- Se puede desviar el flujo de la ejecución de la página mediante las acciones.
- El inconveniente principal surge por lo complejo del aprendizaje para su utilización.

## **2.7. VENTAJAS DE LOS GESTORES DE CONTENIDO.**

La importancia de estos sistemas se centra en que pueden administrar un entorno Web de cualquier tamaño con información creciente de manera acelerada, es por consecuencia de esto que la utilización de este tipo de sistemas permiten tener sitios flexibles y escalables ya que su orientación hacen posible la administración de cualquier tipo de Web.

Otro aspecto para su utilización es que existen varios sistemas de valiosa calidad que son gratuitos y con gran número de módulos aportados para diferentes funcionalidades para diversos tipos de usuarios y que por consiguiente son la razón por la cual las páginas estáticas por no tener estos valores van perdiendo sentido en la creación de sitios.

Para calificar la utilización de estos sistemas definimos algunos puntos importantes que identifican el uso de un **CMS**:

- **Nuevas Funcionalidades.-** Para implementar nuevas funciones a un sitio con páginas estáticas se tendría que realizar una minuciosa revisión de todas las páginas y además construir código para realizar dichas funcionalidades. Pero con el uso de un **CMS** se incluirá

simplemente un módulo aportado por terceros sin tener que realizar cambios dramáticos al sitio que además esto dice de la flexibilidad que tienen estos sistemas de adaptarse y crecer al ritmo de nuevas necesidades que se presenten.

- **Mantenimiento de gran volumen de páginas.-** Para la administración de sitios con abundante información estos sistemas poseen la capacidad de delegar el trabajo de creación, edición y mantenimiento mediante permisos a las diferentes secciones que poseen una variedad de metadatos de todos los documentos. Los metadatos de estos documentos son administrados por estos sistemas ya sea por su versión, fecha de publicación, caducidad del documento, etc.
- **Reutilización de objetos o componentes.-** Estos sistemas al tener un repositorio permite la recuperación y reutilización de los documentos como también de los metadatos que han sido publicados o almacenados en el sitio, pudiendo solicitarse cualquier tipo de información almacenada.
- **Páginas interactivas.-** Las páginas dinámicas no existen en el servidor como sucede en las páginas estáticas, su ventaja radica que son construidas de acuerdo a las necesidades de los usuarios y esto es el resultado de la utilización de un repositorio donde se concentran todos los datos y estos son consultados según los requerimientos de los usuarios. Este repositorio contiene todos los datos concernientes a la Web.
- **Diseño modificable.-** Existe una separación marcada del contenido con respecto al diseño y esto presenta una gran ventaja ya que no se tiene que realizar una exhaustiva revisión de páginas para su posterior adaptación al nuevo diseño sino que mediante la utilización del

estándar CSS (Cascading Style Sheets u hojas de estilo en cascada), se logra conseguir independencia .

- **Consistencia de la Web.-** En este sentido no se quiere decir que hay semejanza absoluta en todas las páginas sino que existe un orden en el aspecto visual que no presenta un caos en la presentación. Los usuarios perciben estos aspectos ya sea por semejanza con las demás páginas o por la disposición y los objetos o incluso por los cambios al navegar por dicha Web. Al utilizar CSS (Cascading Style Sheets u hojas de estilo en cascada) se puede asignar una misma estructura mediante diferentes patrones de página.
- **Control de acceso.-** En este aspecto se deben tener en cuenta los permisos de los usuarios y sus roles a fin de que cada usuario acceda a su área con su rol determinado.