

**2º Desarrollo de Aplicaciones Web**

**IES Doñana**

**Curso: 2019 - 2020**



## **Despliegue Aplicaciones Web**

**Tema 1 - Implantación de arquitecturas Web**

**Profesor: Miguel Ángel García**



# ÍNDICE DE CONTENIDOS

<b>1</b>	<b><i>Introducción</i></b>	<b>3</b>
<b>2</b>	<b><i>Aspectos generales de arquitecturas web</i></b>	<b>3</b>
2.1	Evolución de los servicios web	4
2.2	Tecnologías asociadas a las aplicaciones web	5
2.3	Arquitecturas web. Modelos	6
2.4	Servidores web y de aplicaciones.	7
2.4.1	Servidores web	7
2.4.2	Servidores de aplicaciones	7
2.4.3	Servidores de bases de datos	8
2.4.4	Servidores de ficheros	8
2.4.5	Servidores de directorio	8
<b>3</b>	<b><i>Tipos de aplicaciones web</i></b>	<b>8</b>

**Resultados de aprendizaje y criterios de evaluación** asociados a esta unidad:

-  **RA1:** Implanta arquitecturas Web analizando y aplicando criterios de funcionalidad

**Fuentes usadas para la elaboración del tema:**

-  Wikipedia
-  Recursos del IOC

## 1 Introducción

La arquitectura de aplicaciones en entornos web difiere bastante de la de aplicaciones de escritorio, en la que un programa se ejecuta directamente sobre la máquina en la que trabaja el usuario.

El modelo de arquitectura básico que hay en toda aplicación web es el **modelo** llamado **cliente-servidor**, en el que entran en juego varias máquinas o plataformas, cada una de las cuales desarrolla un rol diferenciado en la ejecución de la aplicación. Según las necesidades y la complejidad de la aplicación, este modelo básico de arquitectura se puede complicar más o menos con el fin de conseguir una mejor distribución de tareas, mejor rendimiento, fiabilidad, aumento de la capacidad de proceso, etc.



### Cliente-servidor

En el modelo cliente-servidor hay dos tipos de componentes:

- 🚦 **Cientes:** hacen peticiones de servicio. Normalmente los clientes inician la comunicación con el servidor.
- 🚦 **Servidores:** proveen servicios. Normalmente los servidores esperan recibir peticiones. Una vez han recibido una petición, la resuelven y devuelven el resultado al cliente.

Los servidores pueden ser con estado o sin estado. Un **servidor sin estado** no mantiene ninguna información entre peticiones, mientras que un **servidor con estado** puede recordar información entre peticiones. Por ejemplo, un servidor sin estado podría ser aquel que contiene páginas web estáticas. En cambio, un servidor que tenga una página web con contenido dinámico sería un ejemplo de servidor con estado.

El modelo cliente-servidor básico de la figura anterior es válido para aplicaciones web pequeñas, sencillas y que no tengan una gran carga de trabajo, es decir, un número pequeño de clientes conectados simultáneamente.

Un servidor también puede ser cliente de otros servidores. Por ejemplo, los servidores web y los otros servicios disponibles en Internet son clientes del servicio de resolución de nombres ( DNS ).



En entornos reales, es habitual que no se den estas tres características y, por tanto, deba implementar una arquitectura más compleja basada en el modelo cliente-servidor pero que puede presentar diferencias o ampliaciones al modelo para garantizar un buen rendimiento de las aplicaciones web, su fiabilidad y/o la capacidad de atender un número elevado de peticiones de los clientes de forma simultánea en aplicaciones web de tamaño medio o grande y de un nivel de complejidad medio/alto.

En este primer tema, vamos a tratar de hacer un resumen claro y conciso del concepto de **arquitecturas web** en donde pretendemos abarcar como mínimo los siguientes puntos: **evolución de los servicios web, tecnologías asociadas a las aplicaciones web, tipos de aplicaciones web, arquitecturas web**, etc.

También se pretende incluir en otro de los puntos el **servidor web Apache**, abarcando los siguientes apartados: instalación, configuración básica e inicio de Apache, así como los mismos puntos para el servidor Tomcat.

## 2 Aspectos generales de arquitecturas web

La **arquitectura World Wide Web (WWW)** de Internet, provee un modelo de programación sumamente poderoso y flexible. Las aplicaciones y los contenidos son presentados en formatos de datos estándar y son localizados por aplicaciones conocidas como "web browsers", que envían requerimientos de objetos a un servidor y éste responde con el dato codificado según un formato estándar.

Los estándares WWW especifican muchos de los mecanismos necesarios para construir un ambiente de aplicación de propósito general, por ejemplo:

- 🚦 **Modelo estándar de nombres:** todos los servidores, así como el contenido de la WWW se denominan según un **Localizador Uniforme de Recursos** (Uniform Resource Locator: URL).
- 🚦 **Contenido:** a todos los contenidos en la WWW se les especifica un determinado tipo permitiendo de esta forma que los browsers (navegadores) los interpreten correctamente.
- 🚦 **Formatos de contenidos estándar:** todos los navegadores soportan un conjunto de formatos estándar, por ejemplo HTML, JavaScript, etc.

- ✚ **Protocolos estándar:** éstos permiten que cualquier navegador pueda comunicarse con cualquier servidor web. El más usado en WWW es **HTTP** (Protocolo de Transporte de HiperTexto), que opera sobre el conjunto de protocolos TCP/IP.

Esta infraestructura permite a los usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros.

También permite a los desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

Los aspectos generales a destacar en una arquitectura web son los siguientes:

- ✚ Escalabilidad.
- ✚ Separación de responsabilidades.
- ✚ Portabilidad.
- ✚ Utilización de componentes en los servicios de infraestructura.
- ✚ Gestión de las sesiones del usuario.
- ✚ Aplicación de patrones de diseño.

El esquema de funcionamiento de los servicios web requiere de tres elementos fundamentales:

- ✚ **Proveedor del servicio web**, que es quien lo diseña, desarrolla e implementa y lo pone disponible para su uso, ya sea dentro de la misma organización o en público.
- ✚ **Consumidor del servicio**, que es quien accede al componente para utilizar los servicios que éste presta.
- ✚ **Agente del servicio**, que sirve como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

De forma genérica podríamos decir que la arquitectura web es un modelo compuesto de tres capas:

- ✚ Capa de Base de Datos, donde estaría toda la documentación de la información que se pretende administrar mediante el servicio web y emplearía una plataforma del tipo MySQL, PostgreSQL, etc.
- ✚ En una segunda capa estarían los servidores de aplicaciones web, ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.
- ✚ En una tercera capa estarían los clientes del servicio web al que accederían mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

## 2.1 Evolución de los servicios web

La evolución del uso de Servicios web en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios. Entre los factores que han impulsado el uso de los servicios web se encuentran:

- ✚ **El contenido se está volviendo más dinámico:** Los sitios web actuales proporcionan contenidos "instantáneos". Un Servicio web debe ser capaz de combinar contenido proveniente de fuentes muy diferentes.
- ✚ **El ancho de banda es menos costoso:** Actualmente un Servicio web puede entregar tipos variables de contenidos como vídeo o audio. A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenidos.
- ✚ **El almacenamiento es más barato:** Un Servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.
- ✚ **El éxito de la computación extendida** se está volviendo más importante: Con cientos de millones de dispositivos como teléfonos móviles, tablets, ..., existentes actualmente, estamos llegando a un momento en el cual las computadoras están dejando de ser el dispositivo más común en Internet. A medida que las plataformas se hacen más diversas, tecnologías como XML se volverán más importantes. Un servicio web no puede exigir que los usuarios ejecuten, por ejemplo, un navegador web tradicional en alguna versión de Microsoft Windows; por el contrario, los servicios web deben servir a todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.

Estos factores, unidos a los beneficios proporcionados por los servicios web en la organización y los buenos productos disponibles para su desarrollo, han hecho que su utilización se extienda sin mayores obstáculos.

En términos generales, cuando se empiezan a utilizar servicios web en una organización, estos se desarrollan e implementan como servicios simples, que poco a poco se van integrando hasta llegar a servicios web mucho más complejos.

En los orígenes del mundo web nos situábamos ante un entorno estático, con páginas en formato HTML que raramente sufrían modificaciones o actualizaciones y en las que apenas había interacción con el usuario.

La **Web 2.0** es la transición que se ha dado desde las aplicaciones tradicionales hacia aplicaciones que funcionan a través de la web y que están fuertemente enfocadas al usuario final. En este nuevo entorno existen una serie de nuevas tecnologías que, en general, tienen como objetivo:

- ✚ Transformar software de escritorio hacia la web.
- ✚ Separar hojas de estilo.
- ✚ Potenciar el trabajo colaborativo y la utilización de redes sociales.
- ✚ Dar control total a los usuarios en el manejo de su información.

## 2.2 Tecnologías asociadas a las aplicaciones web

Las aplicaciones web emplean páginas dinámicas, éstas se ejecutan en un servidor web y se muestran en el navegador de un equipo cliente que es el que ha realizado previamente la solicitud. Cuando una página web llega al navegador, es posible que también incluya algún programa o fragmento de código que se deba ejecutar. Ese código, normalmente en lenguaje JavaScript, lo ejecutará el propio navegador. Es por ello que en este apartado nos centraremos en las tecnologías asociadas a las aplicaciones web que se ejecutarán tanto del lado del servidor como del cliente, especificando lo que corresponda en cada uno de los casos.

- ✚ **ASP (Active Server Pages):** Las "Páginas Activas" se ejecutan del lado del servidor, de este modo se forman los resultados que luego se mostrarán en el navegador de cada equipo cliente que ha realizado la solicitud. Un buen ejemplo de ello son los buscadores, donde un usuario realiza una petición de información y el servidor nos entrega un resultado a medida de nuestra petición.  
Existen versiones de ASP para Unix y Linux, a pesar de que fue una tecnología desarrollada por Microsoft para la creación dinámica de páginas web ofrecida junto a su servidor IIS.
- ✚ **CGI (Common Gateway Interface):** La "Interface Común de Entrada" es uno de los estándares más antiguos en Internet para trasladar información desde una página a un servidor web. Este estándar es utilizado para bases de datos, motores de búsqueda, formularios, generadores de email automático, foros, comercio electrónico, rotadores y mapas de imágenes, juegos en línea, etc.  
Las rutinas de CGI son habitualmente escritas en lenguajes interpretados como Perl o por lenguajes compilados como C.
- ✚ **CSS (Cascading Style Sheets):** Las "Hojas de Estilo en Cascada" se usan para formatear las páginas web; se trata de separar el contenido de un documento de su presentación. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS.
- ✚ **Java:** Este es un lenguaje que trabaja en el cliente, es decir: se ejecuta en el navegador del equipo cliente y no en el servidor. Es un lenguaje eficiente y muy poderoso, que se caracteriza por:
  - Una misma aplicación puede funcionar en diversos tipos de ordenadores y sistemas operativos: Windows, Linux, Solaris, MacOS, etc., así como en otros dispositivos inteligentes.
  - Los programas Java pueden ser aplicaciones independientes (que corren en una ventana propia) o "applets", que son pequeños programas interactivos que se encuentran incrustados en una página web y pueden funcionar con cualquier tipo de navegador: Explorer, Netscape, Ópera, etc.
  - Se trata de un lenguaje "orientado a objetos". Esto significa que los programas se construyen a partir de módulos independientes, y que estos módulos se pueden transformar o ampliar fácilmente. Un equipo de programadores puede partir de una aplicación existente para extenderla con nuevas funcionalidades.
  - Desarrollado por la empresa Sun Microsystems, pero posteriormente liberado bajo licencia GNU GPL, con lo cual es un software libre.

- ✚ **JavaScript:** Lenguaje que se interpreta y se ejecuta en el cliente. Útil para realizar tareas como mover imágenes por la pantalla, crear menús de navegación interactivos, utilizar algunos juegos, etc. En las páginas web suele preferirse JavaScript porque es aceptado por muchos más navegadores que VBScript (creado por Microsoft)
- ✚ **PHP (Hypertext Preprocessor):** Este lenguaje es, como ASP, ejecutado en el lado del servidor. PHP es similar a ASP y puede ser usado en circunstancias similares. Es muy eficiente, permitiendo el acceso a bases de datos empleando servidores como MySQL y, por lo tanto, suele utilizarse para crear páginas dinámicas complejas.
- ✚ **VBScript (Visual Basic Scripting):** La respuesta de Microsoft a JavaScript. VBScript es una buena herramienta para cualquier sitio destinado a ser mostrado exclusivamente en el navegador Microsoft Internet Explorer. El código en VBScript puede, además, estar diseñado para su ejecución en el lado del cliente o en el del servidor, la diferencia es que un código que se ejecuta en el lado del servidor no es visible en el lado del cliente. Éste recibe los resultados, pero no el código.

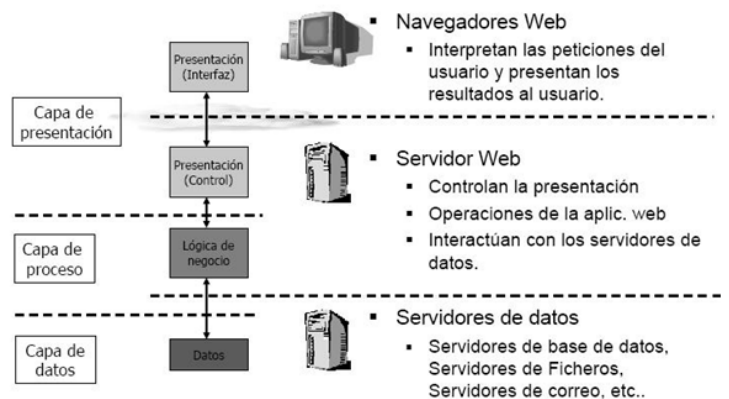
## 2.3 Arquitecturas web. Modelos

En este apartado vamos a explicar brevemente los diversos tipos de modelos de arquitecturas web que se han utilizado a lo largo del tiempo, veremos las características que presenta cada uno de ellos y podremos distinguir sus ventajas e inconvenientes.

Se puede establecer que la arquitectura de un sitio web comprende los sistemas de organización y estructuración de los contenidos junto con los sistemas de recuperación de información y navegación que provea el sitio web, con el objetivo de servir de ayuda a los usuarios a encontrar y manejar la información.

Centraremos el estudio de los modelos de arquitectura web relacionados, en función de cómo implementan cada una de las capas establecidas en una aplicación web:

1. **Capa de presentación** es la encargada de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.
2. **Capa de negocio** es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifican que las reglas establecidas se cumplen.
3. **Capa de acceso a datos** es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio. Generalmente es un gestor de bases de datos relacionales (SGDB) pero pueden ser ficheros de texto plano, ficheros XML, etc. Una opción cada vez más usada es la creación de ficheros XML a partir de datos almacenados en una base de datos y su presentación mediante alguna de las opciones vistas el curso pasado como por ejemplo XSLT.



La evolución experimentada por los medios informáticos en los últimos años ha convivido con otra evolución paralela, la evolución de la arquitectura de las aplicaciones web, que permite aprovechar las nuevas características que éstas ofrecen. De esta forma, el modelo arquitectónico de las aplicaciones de Internet ha sufrido dos grandes saltos, con algún paso intermedio, desde la aparición de los primeros portales web. Los distintos modelos de aplicación sobre los que se ha ido desarrollando, según diversos autores, se podrían clasificar del siguiente modo:

- ✚ **Modelo 1:** En este caso las aplicaciones se diseñan en un modelo web CGI, basadas en la ejecución de procesos externos al servidor web, cuya salida por pantalla era el HTML que el navegador recibía en respuesta a su petición. Presentación, negocio y acceso a datos se confundían en un mismo script perl.
- ✚ **Modelo 1.5:** Aplicado a la tecnología java, se da con la aparición de las JSP y los servlets. En este modelo, las responsabilidades de presentación recaen en las páginas JSP, mientras que los beans incrustados en las mismas son los responsables del modelo de negocio y acceso a datos.

- ✚ **Modelo 2:** Como evolución del modelo anterior, con la incorporación del patrón MVC (Model - View - Controller) en este tipo de aplicaciones, se aprecia la incorporación de un elemento controlador de la navegación de la aplicación. En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.  
Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés.
- ✚ **Modelo 2X:** Aparecen con el objetivo de dar respuesta a la necesidad, cada vez más habitual, de desarrollar aplicaciones multicanal, es decir, aplicaciones web que pueden ser atacadas desde distintos tipos de clientes remotos. Así, una aplicación web multicanal podrá ejecutarse desde una PDA, desde un terminal de telefonía móvil, o desde cualquier navegador HTML estándar. El medio para lograr publicar la misma aplicación para distintos dispositivos es emplear plantillas XSL para transformar los datos XML.

## 2.4 Servidores web y de aplicaciones.

Durante las fases de desarrollo, de puesta en producción y de mantenimiento de una aplicación web podemos encontrarnos con varios tipos de servidores que realizan tareas concretas dentro del funcionamiento global.

### 2.4.1 Servidores web

Un **servidor web** es un servidor que permite el acceso a recursos mediante el **protocolo HTTP** (HyperText Transfer Protocol ) de Internet.

La definición original y estricta del concepto de servidor HTTP hace referencia a aquellos servidores capaces de dar acceso y permitir la gestión de un conjunto de recursos estáticos como respuesta a peticiones recibidas por los clientes. Es decir, que permiten consultar, cargar y eliminar recursos del servidor. Estos recursos suelen ser documentos de HTML o variantes de este formato y contenidos adjuntos o relacionados con estos documentos, como pueden ser imágenes, videos, etc.

Estos recursos suelen estar guardados en forma de archivos en dispositivos de almacenamiento propios del servidor. El concepto original de servidor web no contempla la posibilidad de generar de forma dinámica los contenidos a partir de la ejecución de código como respuesta de las peticiones. Pero, en la actualidad, la mayoría de servidores web admiten la instalación de módulos que permiten que se generen contenidos dinámicos a partir de la ejecución de programas escritos en varios lenguajes de programación (PHP, Javascript, Python, Perl, etc .), aunque esta característica es más propia de los servidores de aplicaciones.

Algunos ejemplos de servidores web son **Apache HTTP Server**, para sistemas operativos Linux, y **Microsoft Internet Information Server**, para Windows.

### 2.4.2 Servidores de aplicaciones

Un servidor de aplicaciones en general es un servidor que ofrece a los clientes un servicio de ejecución de aplicaciones. Si nos centramos en las aplicaciones web, un servidor de aplicaciones es un software que controla la ejecución de programas. Los clientes, desde un navegador (usando el protocolo HTTP), acceden a una interfaz web desde donde ejecutarán la aplicación. Normalmente, los servidores de aplicaciones se utilizan en aplicaciones web con un grado de complejidad elevado.

Un servidor de aplicaciones web se puede entender como un servidor orientado a la ejecución de programas que puede recibir las peticiones de servicio y devolver los resultados utilizando los mismos protocolos (HTTP) y formatos de datos que los servidores web (HTML). Si el mismo servidor no tiene la capacidad de interactuar con estos protocolos puede trabajar conjuntamente con el apoyo de un servidor web que haga de intermediario entre el servidor de aplicaciones y el cliente. Los servidores de aplicaciones, además, suelen proporcionar un amplio conjunto de servicios complementarios orientados a la persistencia de datos, la seguridad, el control de transacciones y concurrencia, entre otros.

Algunos ejemplos de servidores de aplicaciones son GlassFish (servidor Java EE, Oracle) o Microsoft Internet Information Server (servidor .NET).



### 2.4.3 Servidores de bases de datos

Una base de datos se utiliza para almacenar, recuperar y administrar los datos de una base de datos. El servidor gestiona las actualizaciones de datos, permite el acceso simultáneo de muchos servidores o usuarios web y garantiza la seguridad y la integridad de los datos.

Entre sus funciones básicas, el software de servidores de bases de datos ofrece herramientas para facilitar y acelerar la administración de bases de datos. Algunas funciones son la exportación de datos, la configuración del acceso de los usuarios y el soporte de datos.

Algunos ejemplos de servidores de bases de datos son Oracle Database, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB o Firebase.

### 2.4.4 Servidores de ficheros

Un servidor de archivos es un servidor que permite gestionar a través de red la carga, descarga, actualización y eliminación de ficheros almacenados en sus dispositivos desde ordenadores cliente.

En el ámbito de las aplicaciones web, los servidores de archivos se utilizan principalmente para desplegar las aplicaciones sobre el servidor donde se ejecutarán. El desarrollo de una aplicación web sobre los servidores de producción conlleva habitualmente la carga de grandes cantidades de archivos sobre estos servidores. Como el desarrollo y mantenimiento de estas aplicaciones se hace en las máquinas de los programadores, hay algún sistema de transferencia de archivos cada vez que se quiere actualizar la versión de producción de una aplicación.

Uno de los protocolos más usados para la transferencia de ficheros en el desarrollo de aplicaciones web es el protocolo FTP (File Transfer Protocol), con sus variantes FTPS y SFTP para adaptarse a las necesidades actuales de seguridad.

Algunos ejemplos de servidores de transferencia de archivos son ProFTPD o vsftpd, para sistemas operativos Linux, y Microsoft Internet Information Server, para Windows.

### 2.4.5 Servidores de directorio

Un servidor de directorio es un servidor que permite gestionar información administrativa respecto al entorno de una aplicación web, como pueden ser, por ejemplo, los usuarios autorizados con sus roles o permisos, etc.

La utilidad principal de los servidores de directorio es facilitar la gestión de información relativa a la explotación de aplicaciones web. La ventaja de gestionar esta información mediante este tipo de servidores es la centralización de datos y la facilidad de acceso mediante protocolos estándar como LDAP.




Algunos ejemplos de servidores de directorio son OpenLDAP, para Linux, y Active Directory, para Windows.

## 3 Tipos de aplicaciones web

Cualquier proyecto que se quiera desarrollar en Internet, bien sea comercio electrónico, reservas de billetes de vuelo on-line, información meteorológica, registro de usuarios, simuladores de hipotecas, etc, conlleva el desarrollo de una aplicación web. En definitiva, una aplicación web es una plataforma orientada a automatizar los procesos de servicios que se quieran ofrecer a usuarios.

Establecer una clasificación de los tipos de aplicaciones web es una tarea compleja debido a la dificultad existente para poder establecer algún parámetro en función del cual establecer dicha clasificación, junto con la innumerable cantidad de aplicaciones existentes en el actual entorno web 2.0.

En función de cómo se presenta la aplicación web junto con el contenido que pretende mostrar, se ha establecido la siguiente clasificación:

-  **Página web Estática.** Están implementadas en HTML y pueden mostrar en alguna parte de la página objetos en movimiento tales como banners, GIF animados, vídeos, etc.
-  **Página web Animada.** Se realizan con la tecnología FLASH; ésta permite que una página web presente el contenido con ciertos efectos animados continuados. El uso de esta tecnología permite diseños más vanguardistas, modernos y creativos.
-  **Página web Dinámica.** Existen muchos lenguajes de programación que son la base para la mayoría de páginas web dinámicas. Los que destacamos aquí son los lenguajes PHP y ASP. Estos lenguajes permiten una perfecta estructuración del contenido. Por una parte crearíamos la estructura de las páginas web y por otra, almacenaríamos el contenido en determinados archivos. A partir de ahí, crearíamos el código de llamada, que insertaría el contenido en la propia página web estructurada. Este es el principio básico que siguen los



lenguajes de programación. A partir de aquí se desarrollan aplicaciones para poder gestionar el contenido a través de un panel de control.

✚ **Gestores de contenidos (CMS - Content Management System):** Son aplicaciones que permiten a los usuarios crear sitios web usando un navegador sin que sea necesario escribir código HTML (ya que lo generan dinámicamente). El contenido de la página se actualizará mediante el uso de un panel de gestión por parte del administrador del sitio. Este panel de gestión suele ser muy intuitivo y fácil de usar. En aquellas páginas web que requieran una actualización constante, se suele incorporar este panel de gestión para que la web pueda controlarse día a día por parte del cliente.

Dentro de este tipo de aplicaciones web, podemos hacer una segunda clasificación:

- **Gestores de contenidos genéricos:** Permiten un uso genérico para organizar y publicar cualquier tipo de información. Ejemplo: Drupal Joomla MojoPortal
- **Blogs:** Permiten crear sitios Web que se actualizan periódicamente y cuyo contenido se ordena cronológicamente (Posts). Ejemplo: Wordpress, LifeType, ...
- **Foros:** Para crear sitios donde se publican mensajes o noticias sobre cualquier tema por los usuarios dados de alta que podrán ser consultados por cualquier usuario. Ejemplo: PhpBB, SMF, ...
- **Wikis:** Para crear sitios web que se pueden editar por múltiples usuarios al mismo tiempo y compartir dichos contenidos: diccionarios, enciclopedias y documentación de consulta, .... Ejemplo: DokuWiki, MediaWiki (Wikipedia)
- **Gestores de imágenes:** Usados para crear galerías fotográficas. Ejemplos: Coppermine PhotoGallery, Gallery, ...
- **Entornos e-learning o de aprendizaje:** Usados para crear plataforma de aprendizaje online. Ejemplo: Moodle, ...
- **Gestores de comercio electrónico.** Sitio web que permite publicar los productos de una tienda en Internet. Permite la compra on-line a través de tarjeta de crédito, domiciliación bancaria o transferencia bancaria en general. Ofrece al administrador un panel de gestión para poder subir los productos, actualizarlos, eliminarlos, etc. Ejemplo: OsCommerce, Prestashop, ...

✚ **Gestores de Archivos.** Son aplicaciones informáticas que permiten, una vez instaladas en el servidor Web, el almacenamiento, clasificación y recuperación de documentos a través de un navegador. Ejemplos: DropBox, GoogleDrive, LogicalDOC,...

✚ **Aplicaciones web de escritorio.** Son aplicaciones informáticas que permiten a los usuarios usar los programas de oficina habituales (procesadores de texto, hojas de cálculo, agendas), a través de un navegador. Dentro de este tipo de aplicaciones podremos hacer nuevamente la siguiente clasificación:

- **Aplicaciones de correo Web:** Son aplicaciones que permiten acceder y manipular el correo electrónico a través del navegador, sin necesidad de instalar ni de configurar ningún cliente de correo en el ordenador. Ejemplos: Gmail, RoundCube, SquirrelMail, Outlook Web Access
- **Aplicaciones de calendario Web:** Permiten utilizar mediante un navegador una especie de agenda: añadir citas, tareas y otras acciones en una determinada fecha y hora. Ejemplos: Google Calendar WebCalendar Plans
- **Aplicaciones de ofimática Web.** Permiten utilizar mediante un navegador una suite de ofimática: Procesador de texto, hoja de cálculo, programa de presentaciones, sistema de gestión de base de datos, cliente de correo, Gestor de agenda y contactos, ... Ejemplos de este tipo de aplicaciones pueden ser: Acrobat.com, GoogleDocs, ThinkFree Online, Etherpad, ...
- **Sistemas operativo Web.** También se conoce como Escritorio Web. Es un sistema de aplicación de red para integrar aplicaciones web en un espacio de trabajo basado en web, es decir, es un escritorio virtual en la web, corriendo en un navegador de web como software. Los escritorios web a menudo están caracterizados por un entorno similar al de Windows, Mac, o Linux. Los beneficios típicos incluyen la habilidad de guardar trabajo y configuraciones en Internet en vez de hacerlo en un escritorio local. Ejemplos: Glide OS Cloudo Icloud eyeOS

**Nota:** No se consideran realmente S.O. porque no controlan los recursos físicos de nuestro equipo al 100%.