

2.1. Arquitectura de las aplicaciones Web

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y se ejecuta en una PC o estación de trabajo y envía solicitudes a un servidor para ejecutar operaciones complejas.

Por ejemplo, una estación de trabajo utilizada como cliente puede ejecutar una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos.



Ventajas del Sistema de Dos Capas:

El desarrollo de aplicaciones en un ambiente de dos capas funciona adecuadamente, pero no es necesariamente lo más eficiente. Las

herramientas para el desarrollo con dos capas son robustas y ampliamente evaluadas.

Las técnicas de ingeniería de software de prototipo se emplean fácilmente. Las soluciones de dos capas trabajan bien en ambientes no dinámicos estables, pero no se ejecutan bien en organizaciones rápidamente cambiantes.

Desventajas del sistema de dos capas:

Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.

La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es debido al número de dispositivos con acceso directo al ambiente de esas bases de datos.

Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.

Arquitectura de tres capas

La arquitectura de tres capas es un diseño reciente que introduce una capa intermedia en el proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el

cliente). La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final.

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles.

- El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.
- El segundo nivel está referido habitualmente a algún tipo de programa o script.
- Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución. Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



Las diferentes capas suelen ser:

- **Capa 1**: Cliente de aplicación: Navegador Web
- **Capa 2**: Servidor de Aplicaciones: Apache, Servidor Tomcat con servlet's
- **Capa 3**: Servidor de Datos: Base de datos, servidor SMTP...

Ventajas de la arquitectura de tres capas:

Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.

Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.

El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

Desventajas de las Arquitecturas de Tres Capas y basadas en Web.

Los ambientes de tres capas pueden incrementar el tráfico en la red y requiere más balance de carga u tolerancia a las fallas.

Los exploradores actuales no son todos iguales.

La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.

2.2. Lenguajes de programación del lado del cliente

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero nuestra página no se verá bien si la computadora cliente no tiene instalados los **plug-in** adecuados. El código, tanto del hipertexto como de los **scripts**, es accesible a cualquiera y ello puede afectar a la seguridad.

JavaScript.

JavaScript es un lenguaje de programación que realiza acciones dentro del ámbito de una página web. Su compatibilidad con la mayoría de los navegadores modernos, lo posiciona como el lenguaje de programación del lado del cliente más utilizado. Con JavaScript podemos crear efectos especiales en las páginas y definir interacción con el usuario. El navegador (browser) del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

JavaScript es un lenguaje con muchas posibilidades, es orientado a objetos, con funciones, estructuras de datos complejas, etc.

Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

JavaScript no permite dos de las características clásicas de los lenguajes orientados a objetos (herencia y polimorfismo), pero permite la creación y manipulación de objetos sencillos, y la definición de métodos y propiedades para dichos objetos.

JavaScript soporta el Modelo de Objetos de Documento (DOM, *Document Object Model*). El DOM es el conjunto de objetos predefinidos que nos permite acceder a todos los elementos de una página y a ciertas características específicas del navegador.

<pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /> <title>Ejemplo JavaScript</title> </head> <body> <script> var respuesta = confirm("Aceptame o rechazame") alert ("Has pulsado: " + respuesta) </script> </body> </html></pre>	Ejecutar
---	----------

Visual Basic Script

Es un lenguaje de programación de scripts del lado del cliente, sólo compatible con Internet Explorer. Es por ello se usa poco. Está basado en Visual Basic de Microsoft. Tanto su sintaxis y modo de operación es una versión reducida del primero. El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador.

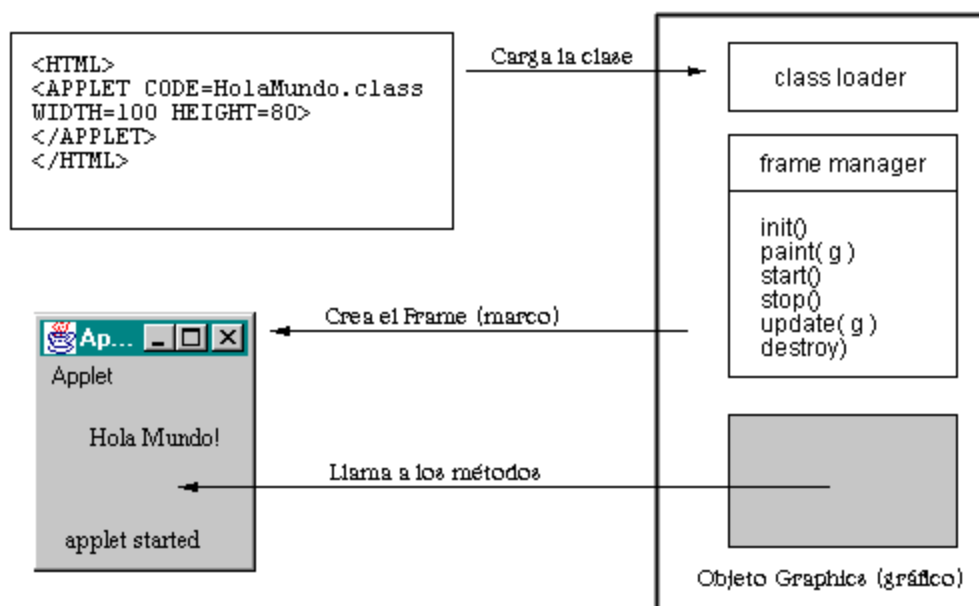
Applets de Java

Son programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página. Los applets son más difíciles de programar que los scripts en Javascript y requerirán conocimientos medios del lenguaje Java. La principal ventaja de utilizar applets consiste en que son independientes del navegador, del sistema operativo y multiplataformas.

Cada sistema operativo dispone de una máquina Virtual de Java que puede interpretar los Byte Codes y transformarlos a sentencias ejecutables en el sistema en cuestión.

Arquitectura de appletviewer

El appletviewer representa la interfaz mínima de navegación. En la figura se muestran los pasos que seguiría appletviewer para presentarnos el resultado de la ejecución del código de nuestra clase.



Esta es una visión simplificada del appletviewer. La función principal de esta aplicación es proporcionar al usuario un objeto de tipo *Graphics* sobre el que dibujar, y varias funciones para facilitar el uso del objeto *Graphics*.

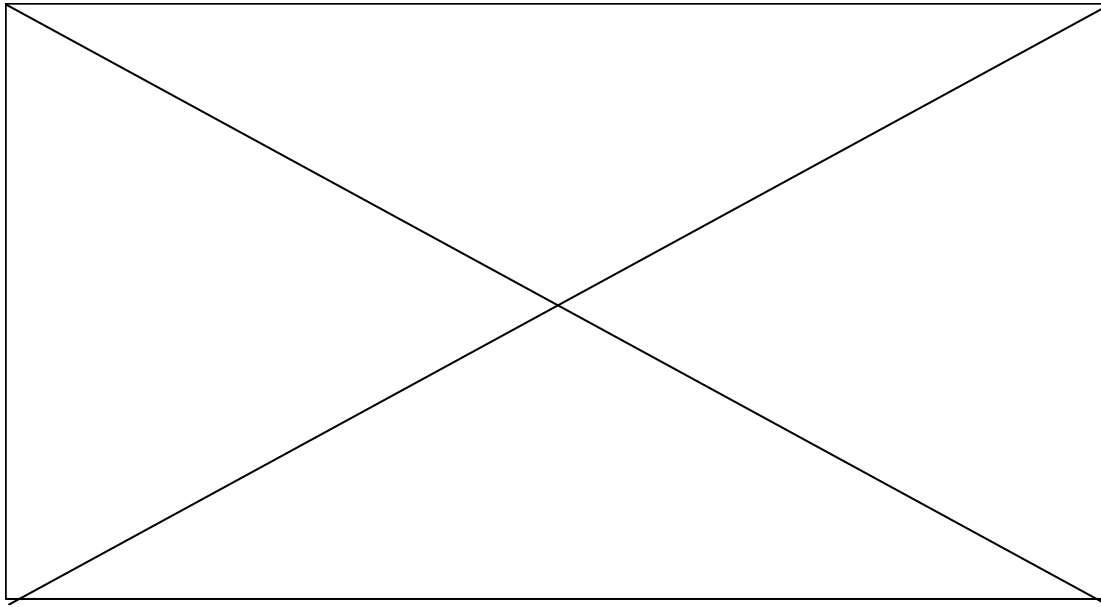
Ciclo de vida de un Applet

Cuando un applet se carga en el appletviewer, comienza su ciclo de vida, que pasaría por las siguientes fases:

- Se crea una instancia de la clase que controla el **applet**. En el ejemplo de la figura anterior, sería la clase **HolaMundo**.
- El applet se inicializa.
- El applet comienza a ejecutarse.
- El applet empieza a recibir llamadas. Primero recibe una llamada *init* (inicializar), seguida de un mensaje *start* (empezar) y *paint* (pintar). Estas llamadas pueden ser recibidas asíncronamente.

Flash

Es un programa multimedia de Adobe que se utiliza para crear presentaciones animadas. Se trata de una aplicación “del lado del cliente” que es leída por los principales navegadores. Las animaciones se realizan a partir de vectores y de imágenes en base a píxeles (llamadas “raster graphics”) y pueden incluir audio y video. Este programa es uno de los más utilizados para animar sitios Web y permitir la interactividad.



Crítica de Steve Jobs

- Flash una tecnología 100% propietaria, argumenta que los estándares web deben de ser completamente abiertos.
- Adobe asevera que los dispositivos que no tienen soporte para Flash, no tienen acceso a toda la web, *“porque más del 75% del vídeo en Internet está basado en Flash”*. Pero lo que no dicen, es que casi todo ese contenido, también está disponible en formatos más modernos como el H.264“.
- Flash ofrece poca seguridad a sus usuarios. Dice que es uno de los principales culpables que las computadoras Mac fallen. Afirma que han tratado de trabajar conjuntamente con Adobe para corregir los problemas, pero que estos persisten después de varios años. *“No queremos reducir la confiabilidad y la seguridad del iPhone, del iPod, o del iPad añadiendo la tecnología Flash”*.
- Flash se desempeña pobremente en los dispositivos portátiles, dado que consume mucha energía, además de que también, consume demasiados recursos aunque el hardware sea capaz de reproducir dicho contenido.

2.3. Lenguajes de programación del lado Servidor

Existe una multitud de lenguajes concebidos o no para Internet. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

Los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, JSP, PERL y PHP.

ASP.NET (Active Server Pages).

Lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

Se desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para su funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Microsoft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores.

El lenguaje ASP consiste en una serie de clases .NET utilizadas para crear aplicaciones Web, tanto del lado cliente (Web Form) como del lado servidor (Web Service). La integración de nativa .NET Framework con el sistema operativo Windows Server 2003 hace que su ejecución sea más estable y rápida que otros lenguajes de programación.

Las páginas creadas con la tecnología ASP.NET funcionan en todo tipo de navegadores – incluyendo Netscape, Safari y Internet Explorer.

Actualizaciones Dinámicas Soporte de servicios web XML Conexiones del tipo DSN, o sin utilización de DSN, para acceder a fuentes de datos ODBC.

Ventajas:

- Completamente orientado a objetos.
- Controles de usuario y personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor velocidad.
- Mayor seguridad.

Desventajas:

- Mayor consumo de recursos.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes.

Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Características:

- Código separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas web.
- Los archivos se encuentran con la extensión (jsp).
- El código JSP puede ser incrustado en código HTML.

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

- **Código:** se puede incrustar código “Java”.
- **Directivas:** permite controlar parámetros del servlet.
- **Acciones:** permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Ejecución rápida del servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.

Desventajas:

- Complejidad de aprendizaje.

PERL

Perl (Practical Extraction and Report Language) es un lenguaje de programación desarrollado por Larry Wall (lwall at netlabs.com) inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell, para la administración de tareas propias de sistemas UNIX. No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos. Lenguaje de programación basado en scripts portable a casi cualquier plataforma. Es muy utilizado para escribir CGIs. Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o Javascript.

Ventajas

- Es un buen lenguaje “pegamento”. Se pueden juntar varios programas de una forma sencilla para alcanzar una meta determinada. Los usuarios de Windows agradecerán esta propiedad ya que normalmente adolecen de un buen lenguaje tipo “script”.
- Es relativamente rápido para un lenguaje tipo “script”.
- Está disponible en múltiples plataformas y sistemas operativos (UNIX, Linux y Windows). Un programa que se escriba teniendo en cuenta la compatibilidad puede ser escrito en una plataforma y ejecutado en otra.
- El desarrollo de aplicaciones es muy rápido.
- Hay una colección enorme de módulos que pueden ser incorporados a cualquier “script” de Perl. Están disponibles en el CPAN (“Comprehensive Perl Archive Network”). En particular existe una extensión para cálculo numérico denominada PDL.

- Perl es gratuito. Mucho más que eso, es “Software Libre”. Esto quiere decir que el código fuente está disponible para que cualquiera lo pueda ver o modificar, y lo que es más importante, siempre lo estará. Aunque nunca pretendas cambiar el código, es importante disponer de la posibilidad de hacerlo, ya que siempre se podrá contratar a una tercera persona para que lo modifique en el caso de que haya un error, y debería ser posible solucionarlo.
- Le otorga al programador mucha libertad para que haga el programa como quiera. Tal como dice el eslogan de Perl “Hay más de una forma de hacerlo”.

Desventajas

- Es lento para algunas aplicaciones, como programación a bajo nivel, escribiendo un “driver” para una aplicación o corriendo modelos numéricos de cálculo intensivo. Si bien se pueden insertar subrutinas FORTRAN o C en Perl, teniendo lo mejor de los dos mundos, pero con algo más de complejidad.
- La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible. Si no se escribe con cuidado puede llegar a ser difícil de leer. De hecho hay un concurso de Perl ofuscado.
- Perl es un lenguaje interpretado. Este tema no es tan crítico como suena, los programas Perl no correrán mucho más rápidos cuando se compilen, la única ventaja está en la desaparición de la fase inicial de compilación al correr la aplicación.
- Utiliza muchos recursos de computó. Esto significa que no es tan ligero como un programa en C, pero en la práctica es ligero comparado con la potencia de procesamiento de las computadoras actuales.

PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

2.4. Ambientes para el desarrollo de aplicaciones Web

Un entorno de desarrollo integrado (**IDE- Integrated Development Environment** -) también conocido como entorno de diseño integrado o el medio ambiente integrado de depuración es una aplicación de software que ofrece servicios integrales a los programadores de computadoras para el desarrollo de software. Un IDE normalmente se compone de:

- Un editor de texto.
- Un compilador.
- Un intérprete.
- Herramientas de automatización.
- Un depurador.
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuario.

IDE para Php

Existen muchos entornos de desarrollo para PHP en software libre tales como: Zend Studio, Open Komodo Project, Eclipse + phpEclipse, entre otros.

Dreamweaver de Adobe

Es en una potente y flexible herramienta de desarrollo que incluye numerosas herramientas para facilitarnos el desarrollo de aplicaciones web.

Dreamweaver permite trabajar con los siguientes modelos de servidor:

- ASP JavaScript
- ASP VBScript
- ASP.NET C#
- ASP.NET VB
- COLDFUSION
- JSP
- PHP MySQL

El concepto de sitios en Dreamweaver es bastante útil ya que nos permite mantener un orden lógico/físico de la estructura de nuestro sitio, enlaces y recursos que este utiliza, tiene el potencial para encargarse completamente de los enlaces relativos (recomendado).

Como se menciona Dreamweaver proporciona soporte para variados modelos de Servidor entregando capacidades de creación de código estándar para una serie de funciones como son la inserción, edición, eliminación y consulta de datos, manejo de ciclos y condiciones, incluso acceso por restringido por password, todo esto de una forma muy básica, pero sin duda es un primer acercamiento para novatos, lo importante de esto es que podemos extender esta capacidad mediante nuestros conocimientos y usar este entorno de desarrollo para facilitarnos la dura tarea de desarrollar para la Web.

Jomla!

Joomla! es un **Sistema de Gestión de Contenidos** (CMS) premiado mundialmente, que le ayuda a construir sitios web y otras aplicaciones online potentes. Lo mejor de todo, es que Joomla es una solución de código abierto y está disponible libremente para cualquiera que desee utilizarlo.

Joomla! se utiliza en todo el mundo para generar desde una simple página web personal hasta complejas aplicaciones web corporativas. Entre los diferentes usos que la gente da a Joomla! están:

- Webs corporativas o portales
- Comercio electrónico
- Pequeños sitios de negocios
- Webs de organizaciones o ONGs
- Aplicaciones gubernamentales
- Intranets y extranets corporativas
- Webs de escuelas o agrupaciones
- Páginas personales o familiares
- Portales de comunidades
- Revistas y periódicos

Joomla se puede usar para gestionar fácilmente cualquiera de los aspectos de un sitio web, desde la introducción de contenidos e imágenes hasta la actualización de un catálogo de productos o la realización de reservaciones online.

WordPress

WordPress es un sistema de gestión de contenido enfocado a la creación de blogs (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL y código modificable, tiene como fundador a Matt Mullenweg.

WordPress fue creado a partir del desaparecido b2/cafeelog y se ha convertido junto a Movable Type en el CMS más popular de la blogosfera.

Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

2.5 Metodologías para el desarrollo de aplicaciones Web.

El creciente desarrollo del comercio electrónico así como el desplazamiento de las distintas organizaciones hacia la Web ha traído en la actualidad una constante evolución de las aplicaciones Web. Cada día se incrementan las transacciones financieras, la transferencia de información confidencial y ejecución de procesos online, entre otros, las cuales exigen funcionalidad, confiabilidad, usabilidad y eficiencia por mencionar algunas características de calidad. Esta relevancia de la economía genera grandes desafíos en las aplicaciones Web que son los de controlar y mejorar su calidad.

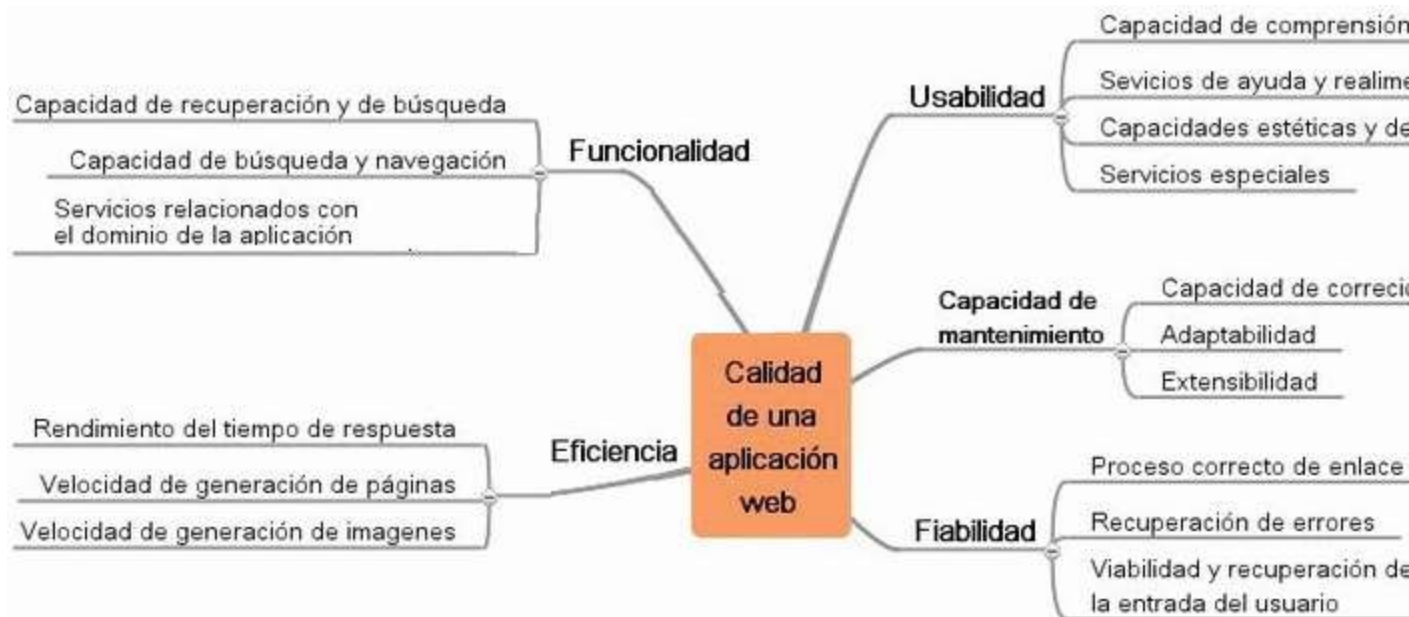
Aunque las aplicaciones Web están creciendo rápidamente tanto en uso como en aceptación, su desarrollo tiende a ser ad hoc, resultando en aplicaciones de pobre calidad. La mayoría de los desarrolladores Web ponen poca atención en la elicitación y análisis de requisitos, así como en las metodologías y procesos de desarrollo. Además los desarrolladores de aplicaciones confían excesivamente en el conocimiento y experticia de los desarrolladores individuales y sus prácticas de desarrollo individual más bien que en las prácticas estándar. No obstante, son las mismas metodologías de desarrollo las que no tratan de manera adecuada y profunda los atributos de calidad. Estas situaciones traen como consecuencia que los atributos de calidad de los sistemas basados en la Web tales como la funcionalidad, confiabilidad, mantenibilidad, usabilidad y portabilidad no se les da la debida consideración que se merecen durante el proceso de desarrollo.

En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizarlas y mantenerlas. Algunas aplicaciones son: los webmails, wikis, weblogs, tiendas en línea, intranet.

Ingeniería Web es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad. Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar que tipo de proceso es más adecuado en función de las características del mismo.

Un empleo sistemático y disciplinado de modelos, métodos y modelos de ingeniería de software en la Web para el aseguramiento y control de la calidad favorece a la comprensión y análisis y potencialmente mejorar los desarrollos producidos.



El Proceso de Ingeniería Web

Características como inmediatez y evolución y crecimiento continuos, nos llevan a un proceso incremental y evolutivo, que permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita.

Según Pressman, las actividades que formarían parte del marco de trabajo incluirían las tareas abajo enumeradas. Dichas tareas serían aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma.

Las actividades que forman parte del proceso son: formulación, planificación, análisis, modelado, generación de páginas, test y evaluación del cliente.

- La **formulación** identifica objetivos y establece el alcance de la primera entrega.
- La **planificación** genera la estimación del costo general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.
- El **análisis** especifica los requerimientos e identifica el contenido.
- La **Modelado** se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.
- En la **generación de páginas** se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de las aplicación, las páginas.
- El **Test** busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que interoperen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.
- Finalmente, el resultado es sometido a la **evaluación del cliente**.

Control y Garantía de la Calidad

Una de las tareas colaterales que forman parte del proceso es el Control y Garantía de la Calidad (CGC). Todas las actividades CGC de la ingeniería software tradicional como son: establecimiento y supervisión de estándares, revisiones técnicas formales, análisis, seguimiento y registro de informes, etc, son igualmente aplicables a la Ingeniería Web. Sin embargo, en la Web toman especial relevancia para valorar la calidad aspectos como:

- Usabilidad
- Funcionabilidad
- Fiabilidad
- Seguridad

- Eficiencia
- Mantenibilidad

Control de la Configuración

Establecer mecanismos adecuados de control de la configuración para la Ingeniería Web es uno de los mayores desafíos a los que esta nueva disciplina se enfrenta. La Web tiene características únicas que demandan estrategias y herramientas nuevas. Hay cuatro aspectos importantes a tener en cuenta en el desarrollo de tácticas de control de configuración para la Web.

- **Contenido:** Considerando la dinamicidad con la que el contenido se genera, es tarea compleja organizar racionalmente los objetos que forman la configuración y establecer mecanismos de control.
- **Personal:** Cualquiera realiza cambios. Hay mucho personal no especializado que no reconoce la importancia que tiene el control del cambio.
- **Escalabilidad:** Es común encontrar aplicaciones que de un día para otro crecen considerablemente. Sin embargo, las técnicas de control no escalan de forma adecuada.
- **Política:** ¿Quién posee la información? ¿Quién asume la responsabilidad y coste de mantenerla?

La Gestión del Proceso

En un proceso tan rápido como es el proceso de Ingeniería Web, donde los tiempos de desarrollo y los ciclos de vida de los productos son tan cortos, ¿merece la pena el esfuerzo requerido por la gestión? La respuesta es que dada su complejidad es imprescindible. Entre los aspectos que añaden dificultad a la gestión destacamos:

- Alto porcentaje de contratación a terceros
- El desarrollo incluye una gran variedad de personal técnico y no técnico trabajando en paralelo
- El equipo de desarrollo debe dominar aspectos tan variados como, software basado en componentes, redes, diseño de arquitectura y navegación, diseño gráfico y de interfaces, lenguajes y estándares en Internet, test de aplicaciones Web, etc, lo que hace que el proceso de búsqueda y contratación de personal sea arduo.

¿Qué marca la diferencia?

A modo de breve resumen enumeramos las siguientes diferencias:

- Confluencia de disciplinas: Sistemas de información, ingeniería software y diseño gráfico que requiere equipos multidisciplinares y polivalentes. Ciclos de vida y tiempo de desarrollo muy cortos
- Cambio continuo:
- Necesidad de soluciones que permitan flexibilidad y adaptación conforme el proyecto cambia.
- Requisitos fuertes de seguridad, rendimiento y usabilidad.

¿Por qué es necesaria?

La Web evoluciona y crece sin diseño alguno. Prácticas tan pobres de calidad pueden introducir defectos que dejen al efecto 2000 como un juego de niños. Es deber de todos proporcionar cimientos firmes a una tecnología que “mágicamente” nos permite acceder a cualquier hora a cualquier punto del planeta para obtener bienes tan valiosos como son los **servicios** y la **información**.

Conclusiones

La aplicación de principios de ingeniería pueden evitar el caos potencial al que nos enfrentamos, y poner bajo control el desarrollo de las aplicaciones Web, minimizando riesgos y mejorando el mantenimiento y calidad.

2.6. Aspectos de seguridad.

Hoy en día las aplicaciones web son uno de los servicios más utilizados ya sea accediendo a un servidor web a través de internet o de una intranet.

Los lenguaje de programación web son versátiles, sencillos de usar y permiten crear soluciones de gran envergadura.

Las aplicaciones escritas en estos lenguajes están expuestas a diversas amenazas si no se toman las medidas adecuadas para evitarlo, por este motivo se resalta la importancia de la formación en seguridad que deberían de recibir todos los programadores.

Si no se procede de forma correcta, las aplicaciones web desarrolladas en estos lenguajes pueden servir como puerta de entrada de un intruso a la red de la empresa, por lo que la seguridad de este tipo de aplicaciones tiene que ser tomada como prioritaria para no exponer la seguridad de su negocio.

Esta formación está orientada a que los equipos de programadores conozcan y utilicen las buenas prácticas en el diseño y desarrollo de aplicaciones para evitar vulnerabilidades de seguridad.

El objetivo del mismo es conocer como proteger la aplicación contra los fallos más comunes (SQL Injection, XSS, Command Execution, File Inclusion ...).

Las fallas de seguridad más comunes son:

- Cross Site Scripting (XSS). Las vulnerabilidades de XSS originalmente abarcaban cualquier ataque que permitiera ejecutar código de "*scripting*", como VBScript o JavaScript, en el contexto de otro sitio web (y recientemente esto se podría clasificar más correctamente como "distintos orígenes").
- Injection Flaws
- Insecure Remote File Include
- Insecure Direct Object Reference
- Cross Site Request Forgery (CSRF)
- Information Leakage and Improper Error Handling
- Broken Authentication and Session Management
- Insecure Cryptographic Storage
- Insecure Communications
- Failure to Restrict URL access

3.1. Procesamiento del lado del servidor.

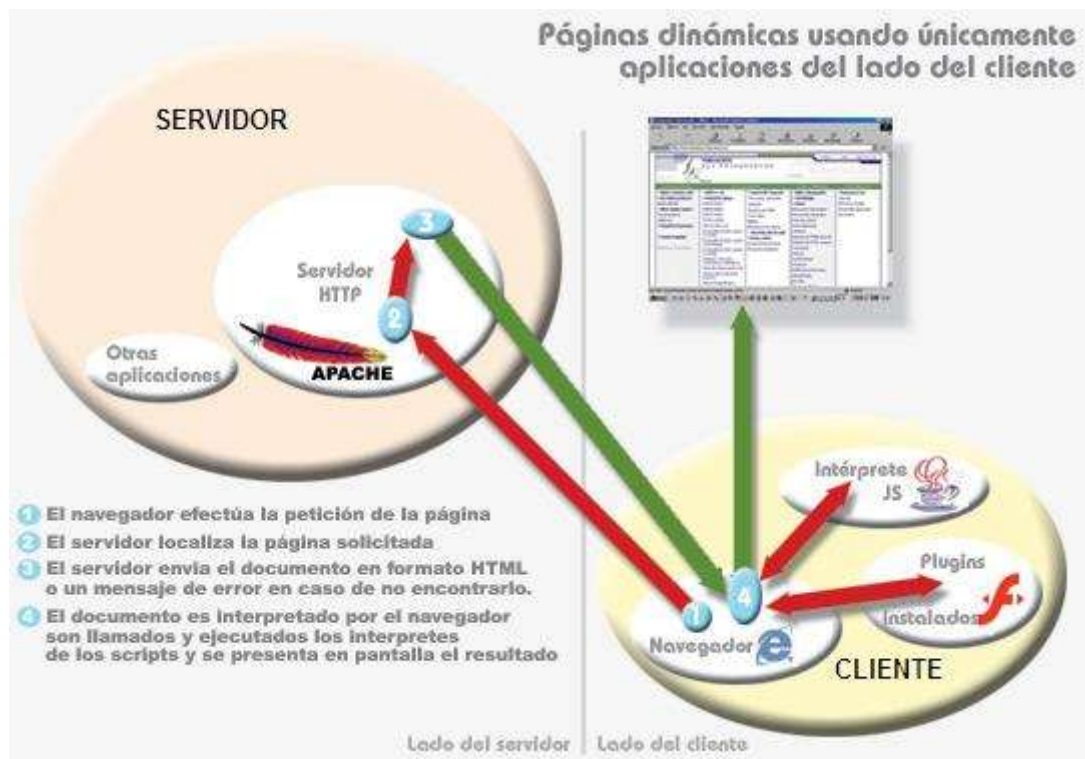
La Programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

Todo lo que suceda dentro del servidor es llamado procesamiento del lado del servidor, o **server-side processing**. Cuando tu aplicación necesita interactuar con el servidor (por ejemplo, para cargar o guardar datos), ésta realiza una petición del lado del cliente (**client-side request**) desde el navegador, a través de la red usando invocaciones remotas a métodos (**remote procedure call, RPC**). Mientras se está procesando una llamada RPC, tu servidor está ejecutando código del lado del servidor.

La utilización de las diferentes aplicaciones o servicios de Internet se lleva a cabo respondiendo al llamado *modelo* **cliente-servidor**.

Cuando se utiliza un servicio en Internet, como consultar una base de datos, transferir un archivo o participar en un foro de discusión, se establece un proceso en el que entran en juego dos partes. Por un lado, el usuario, quien ejecuta una aplicación en procesador local: el denominado *programa* **cliente**. Este programa **cliente** se encarga de ponerse en contacto con el **procesador remoto** para solicitar el servicio deseado. El **procesador remoto** por su parte responderá a lo solicitado mediante un programa que esta ejecutando. Este último se denomina *programa* **servidor**. Los términos **cliente** y **servidor** se utilizan tanto para referirse a los programas que cumplen estas funciones, como a las computadoras donde son ejecutados esos programas.

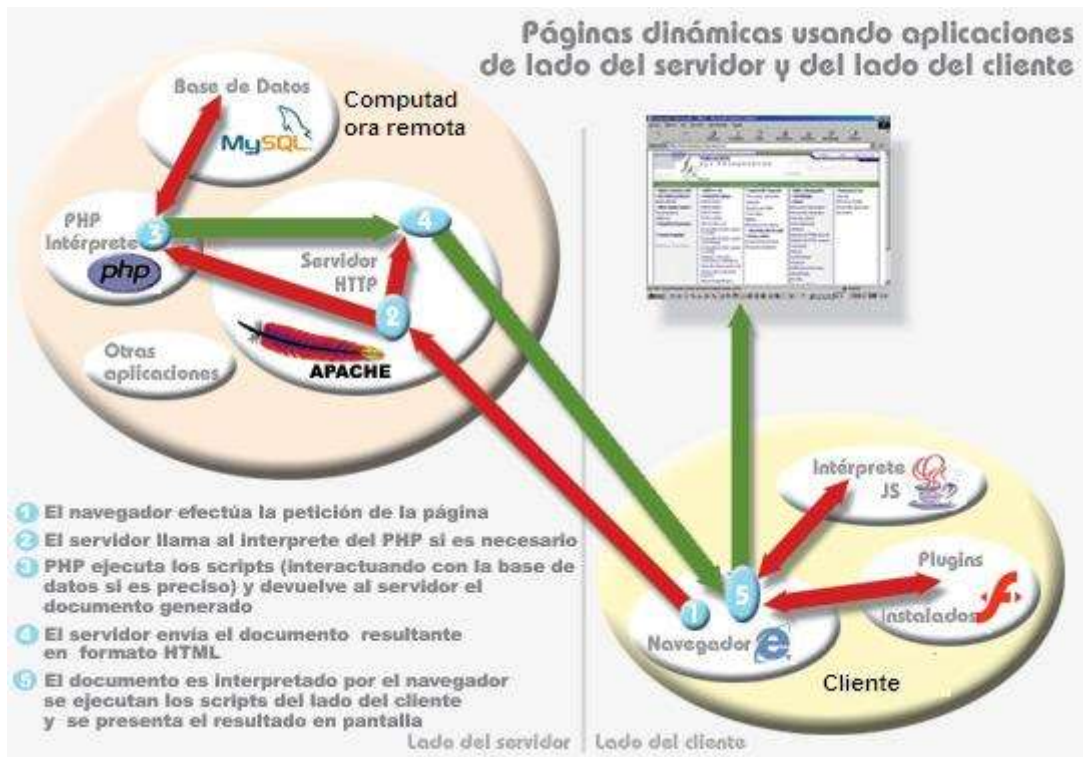
El programa o los programas cliente que el usuario utiliza para acceder a los servicios de Internet realizan dos funciones distintas. Por una parte, se encargan de gestionar la comunicación con el computador servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste; y por otra, es la herramienta que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las prestaciones que ofrece el servidor.



Cuando nosotros seleccionamos un enlace hipertexto, en realidad lo que pasa es que establecemos una petición de un archivo HTML residente en el servidor (una computadora que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

Así pues, podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él, por ejemplo: ASP, PHP, JSP.

Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.



Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas.

Inversamente, un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.