

UD1: ARQUITECTURAS Y TECNOLOGÍAS WEB

UD1 - Arquitectura y Tecnologías Web

1. ¿Qué es la Arquitectura Web?	3
2. Modelos de Arquitectura Web	3
2.1. Modelo Punto a Punto (P2P - Peer to Peer)	4
2.2. Modelo Cliente-Servidor	4
2.3. Modelo con servidor de aplicaciones	7
2.4. Modelo con servidor de aplicaciones externo	8
3. Arquitecturas y tecnologías Web	9
3.1. Páginas web estáticas	9
3.2. Aplicaciones web	11
3.3. Aplicaciones web dinámicas del lado cliente	13
3.3.1. Lenguajes embebidos	13
3.3.2. Objetos embebidos	14
3.3.3. AJAX	18
3.4. Aplicaciones web dinámicas del lado servidor	19
3.4.1. CGI	20
3.4.2. Servlets	21
3.4.3. Lenguajes embebidos en el código HTML (PHP, ASP, JSP)	22

1. ¿Qué es la Arquitectura Web?

La Arquitectura Web es la jerarquización de la información en un sitio web. Así como en la arquitectura tradicional se trabaja la estructura de una construcción (sea casa, edificio o salón), la arquitectura web trabaja la estructura de una página web y las categorías que la conforman para su optimización de cara al posicionamiento en buscadores.

También se puede definir como la disciplina y el conjunto de acciones que sirven para organizar y estructurar la información que contiene una web para resultar amigable y rastreable. Para realizar dicho cometido tenemos las siguientes opciones y tecnologías:

- Páginas web estáticas.
 - HTML + CSS
- Aplicaciones web.
 - Contenidos dinámicos. Rich Internet Application (o Aplicación de Internet Enriquecida).
- Aplicaciones web dinámicas del lado cliente.
 - Lenguajes embebidos
 - Objetos embebidos
 - AJAX
- Aplicaciones web dinámicas del lado servidor
 - CGI (Common Gateway Interface). Muy habitual el uso de Perl para crear scripts .
 - Servlets
 - Lenguajes embebidos en el código HTML (PHP, ASP, JSP)

Antes de desarrollar cada uno de estos puntos es importante conocer el concepto de modelo de Arquitectura Web.

2. Modelos de Arquitectura Web

Los modelos de arquitecturas web describen la relación entre los distintos elementos que componen la estructura de funcionamiento de las páginas y aplicaciones web.

2.1. Modelo Punto a Punto (P2P - Peer to Peer).

Una red punto a punto (P2P - Peer to Peer) es un tipo de arquitectura de red descentralizada y distribuida en la que los nodos individuales de la red (peers) actúan tanto como suministradores como clientes de recursos, en contraste con el modelo cliente-servidor (C/S) en el que los nodos clientes acceden a los recursos que proporcionan los servidores centrales.

En una red P2P, las tareas, como realizar un streaming de audio o vídeo, se comparten entre múltiples puntos interconectados que ponen una parte de sus recursos (CPU, almacenamiento, ancho de banda) disponibles directamente por otros puntos de la red, sin la necesidad de disponer de servidores que realicen la coordinación centralizada.

Ventajas:

- **Escalabilidad:** Millones de usuarios potenciales. Cuantos más nodos conectados, mejor será su funcionamiento, al contrario del modelo C/S.
- **Robustez:** Los clientes pueden localizar los recursos sin depender del correcto funcionamiento de un único servidor.
- **Descentralización:** Ningún nodo es imprescindible para el funcionamiento de la red.
- **Distribución de costes entre los nodos:** No hace la carga sobre un único nodo de la red.

Desventajas:

- **Falta de fiabilidad de los recursos:** No hay garantías de que los recursos obtenidos desde un nodo sean realmente los deseados. En el modelo C/S, el recurso obtenido del servidor central es único, y no ha podido ser modificado por otro nodo.
- **Difícil mantenimiento:** La actualización de los recursos compartidos debe realizarse en todos los nodos.

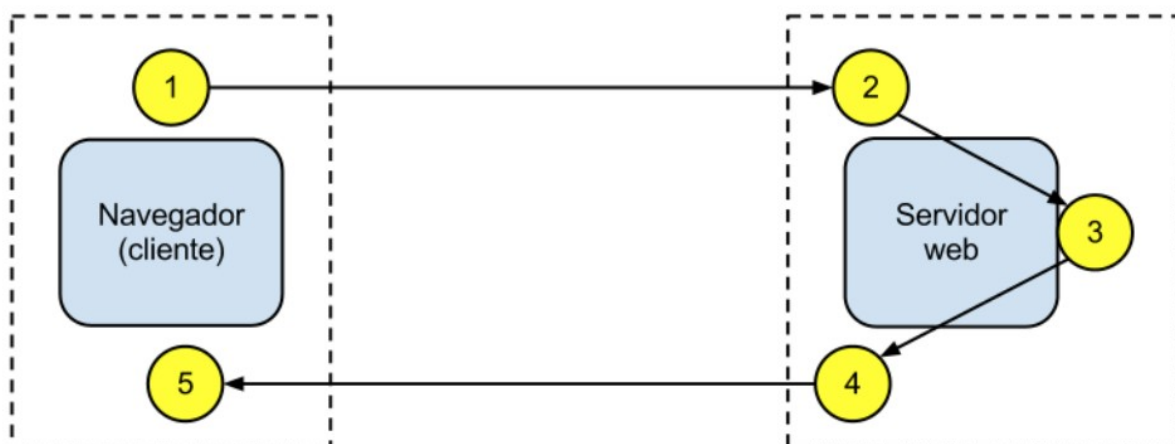
2.2. Modelo Cliente-Servidor.

El modelo cliente-servidor es un modelo informático que actúa como una aplicación distribuida que particiona tareas o cargas de trabajo entre los proveedores de un recurso o servicio, llamados servidores, y los solicitantes del servicio, llamados clientes.

Lo más frecuente es que los clientes y los servidores se comuniquen a través de una red informática, pero ambos pueden residir en la misma máquina. Un servidor es un host que ejecuta uno o más programas servidores que comparten sus recursos con los clientes, aunque no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

El esquema de funcionamiento más básico del modelo cliente-servidor para una arquitectura web está basado en uno o varios clientes que solicitan una página web a un servidor web:

1. Desde el navegador web (cliente) el usuario solicita la carga de una página web indicando su URL.
2. El servidor recibe la petición de la página web
3. Busca en sus sistema de almacenamiento la página solicitada.
4. Envía el contenido de la página web (código fuente) por el mismo medio por el que recibió la petición.
5. El navegador web recibe el código fuente de la página y lo interpreta mostrando al usuario la página web.



El servidor web envía al cliente el recurso solicitado sin hacer ningún tratamiento sobre él. Es decir, envía de manera indiferente una página HTML, una imagen, un archivo de sonido o

cualquier otro tipo de archivo. A continuación se muestran las ventajas y desventajas al respecto:

Ventajas:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de actualizar datos u otros recursos (mejor que en las redes P2P)..
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

Desventajas:

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuanto más nodos hay, mejor es el ancho de banda que se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque

algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.

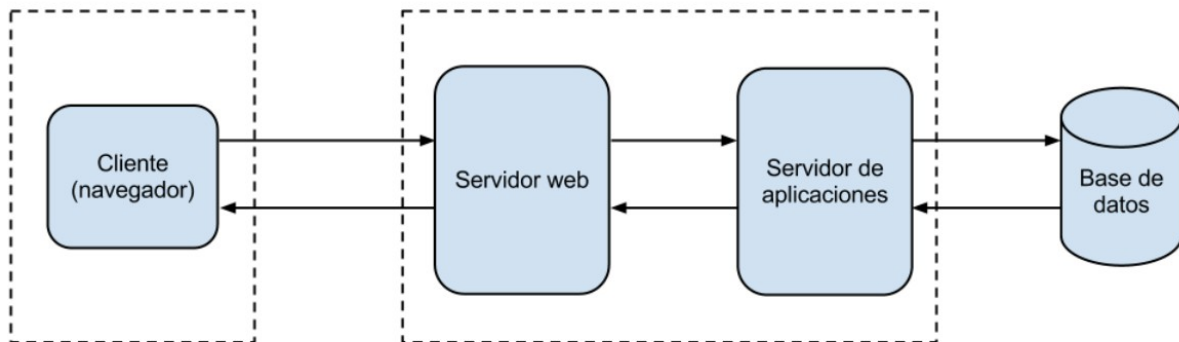
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

2.3. Modelo con servidor de aplicaciones

La función que realiza un servidor de aplicaciones es diferente, ya que los recursos que va a manipular no son archivos estáticos, sino que contienen el código que tiene que ejecutar. Es decir, un servidor web en solitario enviaría al cliente el recurso solicitado tal cual, mientras que el servidor de aplicaciones lo ejecuta y envía al cliente el resultado a través del servidor web.

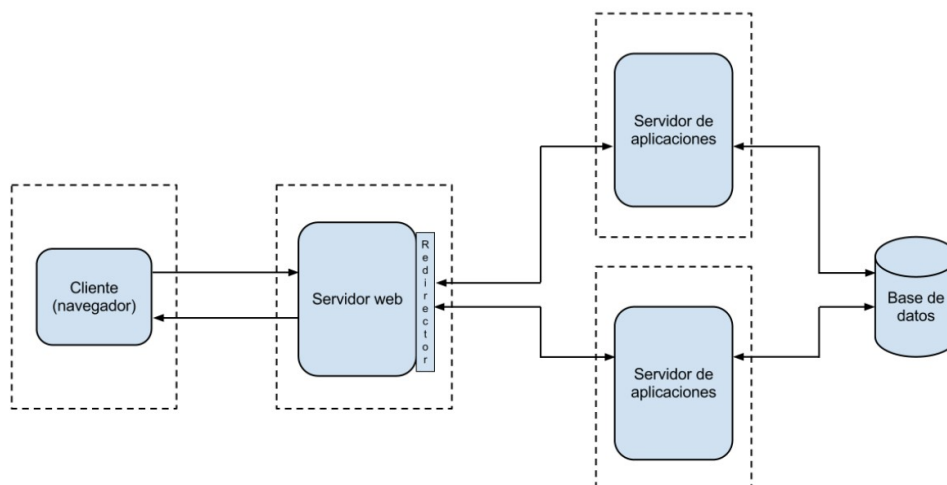
El servidor web y el servidor de aplicaciones pueden residir en una misma máquina como se refleja en el siguiente esquema. El servidor web recibe la petición de un recurso, y si éste corresponde a un recurso dinámico, transfiere la parte correspondiente al servidor de aplicaciones el cual devolverá de nuevo al servidor web el recurso ejecutado. Finalmente será el servidor web el que envíe al cliente el resultado final.

Es muy frecuente que el servidor de aplicaciones deba conectarse con una base de datos para obtener los datos solicitados durante la ejecución del código. Dicha base de datos puede residir en la misma máquina que el servidor web o en otro host conectado en red.



2.4. Modelo con servidor de aplicaciones externo

La parte correspondiente al servidor web suele tener menos carga de trabajo que el servidor de aplicaciones por lo que se puede establecer una estructura en la que el servidor web sea independiente del servidor de aplicaciones, de manera que el servidor web se pueda implantar de forma dedicada precisando menos recursos. Un redirector o proxy será el encargado de transferir al servidor de aplicaciones los elementos que necesiten ser ejecutados, mientras que no pasarán del servidor web los recursos estáticos.



3. Arquitecturas y tecnologías Web.

Las páginas web que puedes visitar en Internet no están todas realizadas siguiendo una misma arquitectura. Algunas de ellas no cambiarán sus contenidos ni su aspecto hasta que su diseñador realice esos cambios. En cambio, otras páginas web pueden ir variando en función de nuestra interacción con esas páginas o en función de otros aspectos externos como el paso del tiempo. Por ello, podemos hacer una diferenciación de las diferentes arquitecturas web en función de la manera en que puedan variar sus contenidos.

3.1. Páginas web estáticas.

Es el primer tipo de páginas web que se crearon desde el origen de la web (~1991) y del que todavía podemos encontrar muchos casos en Internet. Se basa en el lenguaje HTML, el cual es interpretado por los navegadores web para mostrar la información al usuario.

Por ejemplo, el siguiente trozo de código HTML es el contenido de una página web:

```
<HTML>
<HEAD>
  <TITLE>Un Titulo para el Browser de turno </TITLE>
</HEAD>
<BODY>
  <!-- Aqui va todo lo chachi -->
  <H1>Otro título, esta vez más largo. </H1>
  <P> <IMG SRC="/felix.gif" ALIGN= "MIDDLE" ALT= "EL Gato Felix">Hoola.
  <P>Esto es un parrafo con informacion
super importante. Notese que las lineas salen pegadas aun dejando
espacios, saltos de linea, etc. <BR> Si pongo esto
si <STRONG>cambia </STRONG> de linea!</P>
<P>Otro parrafo, esto ya es un poco rollo.</P>
<H3>Pongamos un subtítulo</H3>
<P>Por cierto, que paso con las <A HREF= "#pepe">anclas</A>?</P>
<HR>
<UL>
  <LI> Esto es una lista no ordenada.
  <LI> Las listas quedan mejor si tienen varios elementos.
</UL>
Me voy al <A HREF="http://www.iac.es/home.html">IAC</A>.
<P>Vamos a crear un <EM>ancla </EM>, o lo que es lo mismo, un <A NAME="pepe">anchor.</A></P>
</BODY>
</HTML>
```

Este código será mostrado al usuario, a través del navegador web, de la siguiente manera:



Las páginas web creadas exclusivamente con lenguaje HTML no variarán en su forma ni en su contenido mientras su desarrollador no modifique el código de la página. Como ejemplo, podemos visitar este Tutorial de OpenOffice. Con la utilización de otros lenguajes y tecnologías, se pueden obtener páginas web con contenido dinámico, pero en cualquier caso siempre estarán integradas dentro de un estructura básica de HTML. Es decir, ningún lenguaje de creación de páginas web dinámicas puede existir de forma independiente. Siempre se deben integrar de una manera u otra junto con el lenguaje HTML.

El lenguaje de marcas CSS se puede utilizar junto con el lenguaje HTML para facilitar el diseño del aspecto visual que ofrecen las páginas web, aunque su uso por sí solo no permitirá crear dinamismo en las páginas web.

3.2. Aplicaciones web.

Algunos lenguajes de programación permiten integrar código junto con el lenguaje HTML para ofrecer contenidos dinámicos en las páginas web. De esta manera la web crece enormemente en posibilidades y da origen a las aplicaciones web. Así, los contenidos ofrecidos por las páginas web no son sólo un conjunto de información que se ofrece de forma estática, aunque permiten la navegación entre sus contenidos, sino que se convierten en auténticas aplicaciones informáticas que pueden ser utilizadas desde el navegador web. Se le da el nombre de Rich Internet Application (o Aplicación de Internet Enriquecida) a las aplicaciones web que tienen muchas de las características de las aplicaciones software de escritorio. Normalmente se ofrecen a través de un navegador específico, un complemento para los navegadores habituales, haciendo uso de JavaScript o mediante una máquina virtual. Adobe Flash, JavaFX, y Microsoft Silverlight son actualmente las plataformas más comunes para ofrecer RIA. A continuación se detallan las ventajas y desventajas.

Ventajas:

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.

- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (RIAs).

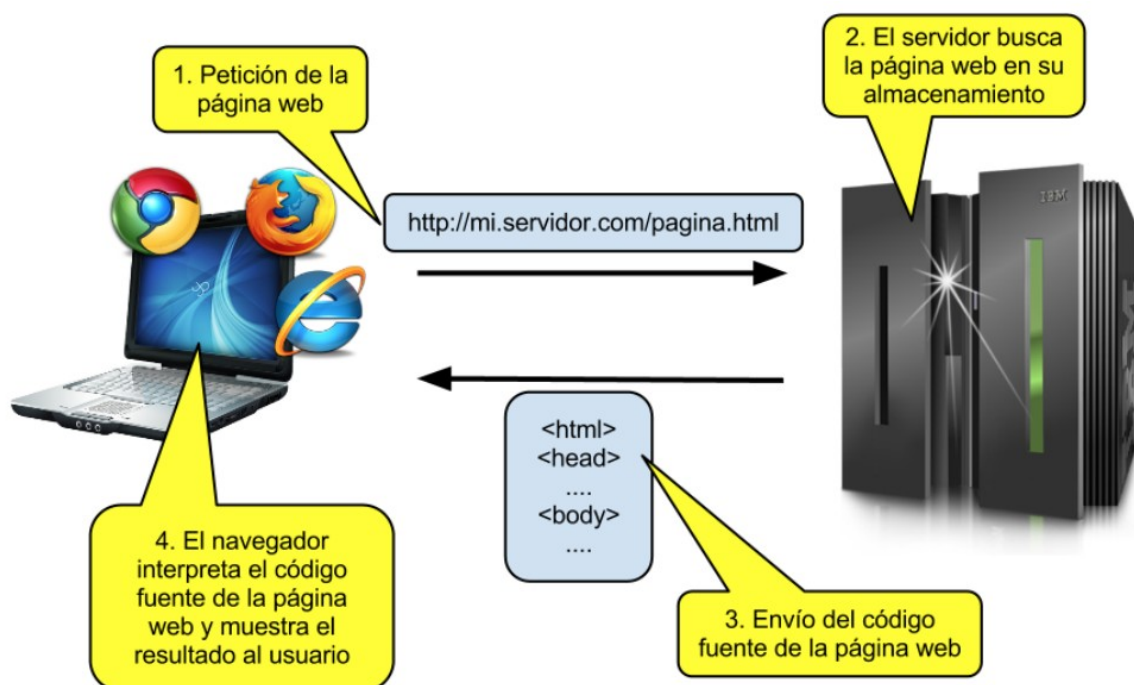
Inconvenientes:

- Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Pero cada vez los navegadores están más preparados para mejorar en este aspecto. La aparición de HTML 5 representa un hito en este sentido. Es posible añadir funcionalidades a estas aplicaciones gracias al uso de Aplicaciones de Internet Ricas.
- La disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.
- A continuación se muestran las principales técnicas de desarrollo de aplicaciones web diferenciando el lugar de ejecución que puede ser en el lado del cliente (navegador) o en el servidor.

3.3. Aplicaciones web dinámicas del lado cliente.

3.3.1. Lenguajes embebidos.

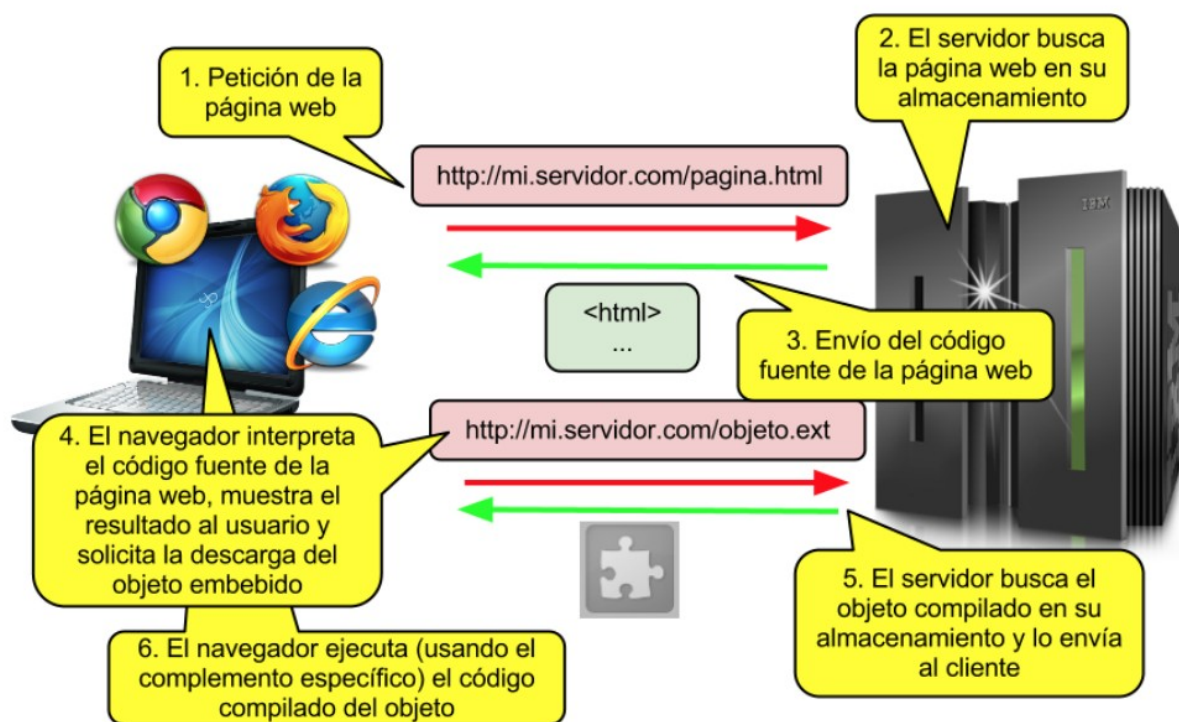
Los casos más sencillos de desarrollo de aplicaciones web son los lenguajes de programación que son interpretados directamente por los navegadores (lenguajes de programación web del lado cliente) y que se encuentran embebidos dentro del código fuente HTML de la página web. Las páginas web que contienen código escrito en ese tipo de lenguajes, viajan a través de la red con todo ese código y el navegador, al recibir el contenido de la página web, será el responsable de interpretarlo para ofrecer el resultado de su ejecución al usuario.



Uno de los casos más utilizados de este tipo de lenguajes es JavaScript. Este lenguaje utiliza una sintaxis similar a C o Java y se integra dentro del código HTML debiendo estar encerrado en la etiqueta `<script>`. Para que el navegador web pueda interpretar el código JavaScript es necesario que lo soporte. Todos los navegadores actuales más frecuentes lo soportan y ofrecen la posibilidad de poder desactivar la interpretación de este lenguaje en algún apartado de la configuración.

3.3.2. Objetos embebidos.

Otra manera de poder ejecutar aplicaciones web en el cliente (navegador web) es mediante objetos o componentes embebidos. En este caso no se dispone del código fuente de la aplicación como ocurría en los casos anteriores para que se ejecute en el navegador, a no ser que el desarrollador haya publicado por separado el código.



Con esta modalidad, el código fuente de la página web que se ha cargado en el cliente solicita al servidor un determinado objeto. Estos objetos consisten en una aplicación compilada que se habrá desarrollado con una de las tecnologías que existen para ello (a continuación se mostrarán algunas de las más importantes). El cliente descarga el objeto desde el servidor que se indique y comienza la ejecución del mismo. Para poder ser ejecutado el objeto es necesario que el navegador disponga de una extensión o complemento (plugin) que lo permita. Por otro lado conviene destacar que no todos los navegadores son totalmente compatibles con algunas de las

tecnologías empleadas para integrar estos tipos de objetos en las páginas web.

- **Flash:** Adobe Flash es una plataforma multimedia usada para añadir animaciones, vídeo e interactividad a las páginas web. Se usa frecuentemente en publicidad, juegos, animaciones, y para el despliegue de audio y vídeo por Internet. Permite capturar las pulsaciones del usuario sobre el ratón y el teclado, así como el micrófono o la cámara. Utiliza un lenguaje de programación llamado ActionScript.
 - Como ejemplo puedes visitar la web juegosjuegos.com donde puedes encontrar gran cantidad de juegos a través de la web que utilizan esta tecnología.
 - El navegador web mostrará una información similar a la siguiente si no se dispone del complemento necesario (Adobe Flash Player):

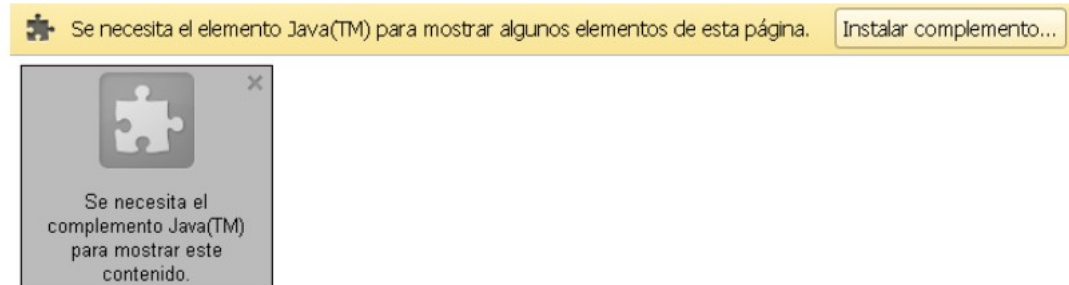


- Este es un trozo de código fuente de una página web que puede ser un ejemplo del código en el que se solicita la carga del objeto Flash:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0" width="100%" height="100%" />
  <param name="movie"
    value="http://cdn.juegosjuegos.com/games14/9396.swf" />
  <param name="quality" value="high" />
  <param name="allowScriptAccess" value="always" />
</object>
```

- **Applet Java:** es un applet (pequeña aplicación) escrito en el lenguaje de programación Java previamente compilado, que puede ser ejecutado por un navegador web que soporte Java, utilizando la máquina virtual de Java que debe estar instalada previamente.

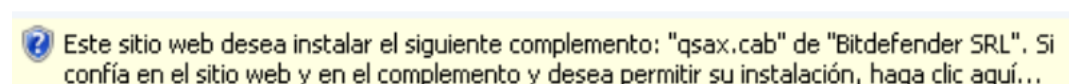
- El navegador web mostrará una información similar a la siguiente si no se dispone del complemento necesario (máquina virtual Java):



- Este es un trozo de código fuente de una página web que puede ser un ejemplo del código en el que se solicita la carga del Applet Java:

```
<applet codebase="." code="Clock.class" width=170 height=150>
  alt="Your browser understands the <APPLET> tag but isn't running the applet, for some reason."
  Your browser is completely ignoring the <APPLET> tag!
</applet>
```

- **ActiveX:** Los controles ActiveX es una tecnología creada por Microsoft basada en pequeños bloques de programas que pueden utilizarse para distribuir aplicaciones a través de Internet mediante navegadores web, al igual que las tecnologías anteriores. Se pueden utilizar para recoger datos, mostrar el contenidos de determinados tipos de archivos, ver animaciones, etc.
 - Ejemplo: Bitdefender QuickScan.
 - Muchas aplicaciones Microsoft Windows como Microsoft Office, Microsoft Visual Studio, y Windows Media Player también usan controles ActiveX para ofrecer sus funcionalidades.
 - El navegador web mostrará una información similar a la siguiente si no se tiene instalado el control ActiveX solicitado:



- Este es un trozo de código fuente de una página web que puede ser un ejemplo del código en el que se solicita la carga del control ActiveX:

```
<object id="CommonDialog1" width=32 height=32 classid="CLSID:F9043C85-F6F2-101A-A3C9-08002B2F49FB"
codebase="http://activex.microsoft.com/controls/vb5/comdlg32.cab
#Version=1,0,0,0">
</object>
```

- **Silverlight:** Microsoft Silverlight es una plataforma de aplicaciones para Internet similar a lo que ofrece Adobe Flash, integrando multimedia, gráficos, animaciones e interactividad. Silverlight es también una de las plataformas de desarrollo de aplicaciones para Windows Phone.
- El navegador web mostrará una información similar a la siguiente si no se tiene instalado el complemento necesario (Microsoft Silverlight):



- Este es un trozo de código fuente de una página web que puede ser un ejemplo del código en el que se solicita la carga del objeto Silverlight:

```
<object id="silverlightBrowser" data="data:application/x-silverlight-2," type="application/x-silverlight-2"
width="100%" height="100%">
  <param name="source" value="ClientBin/SampleBrowser.xap"/>
  <param name="background" value="white" />
</object>
```

- **JavaFX:** Es una tecnología enfocada para el desarrollo de aplicaciones para y escritorio,

soportada por Oracle basándose en el lenguaje de programación Java.

- Un ejemplo (juego BrickBreaker), y desde la web de Oracle puedes descargar otros ejemplos con su código fuente.
- El navegador web mostrará una información similar a la siguiente si no se tiene instalado el complemento necesario (Máquina virtual Java):



- Este es un trozo de código fuente de una página web que puede ser un ejemplo del código en el que se solicita la carga del objeto JavaFX:

```
<script src="http://dl.javafx.com/1.2/dtfx.js"></script>
<script>
  javafx(
    {
      archive: "BubbleBath.jar",
      width: 600,
      height: 500,
      code: "BubbleBath",
      name: "BubbleBath"
    }
  );
</script>
```

3.3.3. AJAX.

La tecnología AJAX (acrónimo de Asynchronous JavaScript and XML) es un grupo de técnicas de desarrollo web usadas en el lado cliente para crear aplicaciones web asíncronas. Con AJAX, las aplicaciones web pueden enviar y recibir datos de un servidor de forma asíncrona (en segundo plano) sin que interfiera en la vista o en el funcionamiento de la página web.

AJAX no es una sola tecnología, sino un grupo de tecnologías. HTML y CSS se usan conjuntamente para dar forma y estilo a la información. El acceso a los elementos que componen la página web (DOM) se realiza a través de JavaScript para mostrar la información de forma dinámica y permitir al usuario interactuar con dicha información presentada.

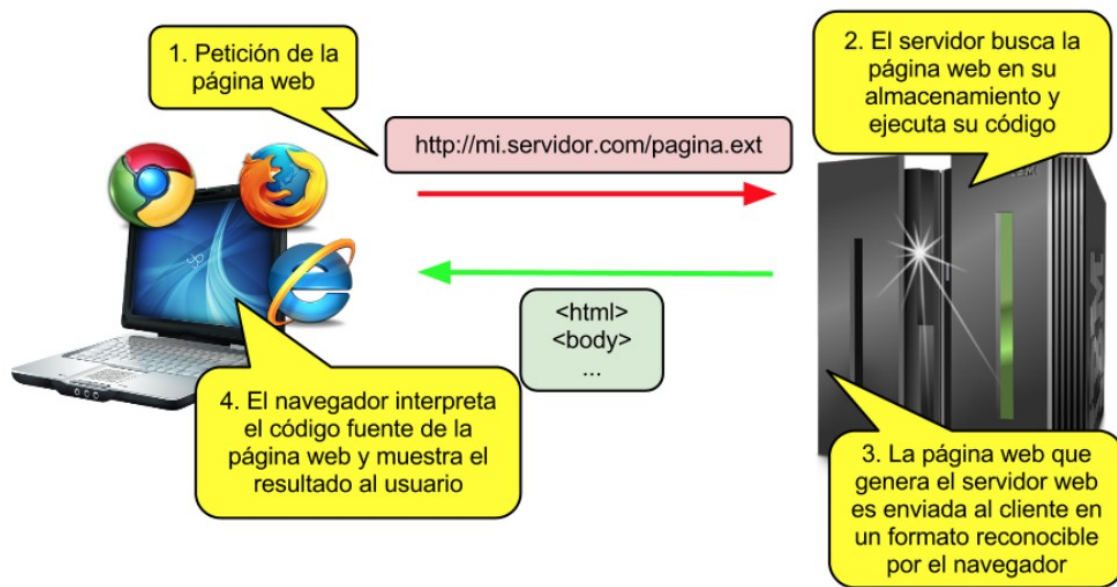
A través de XML, JSON u otros formatos similares, se realiza el intercambio de información entre el navegador y el servidor para evitar así la recarga completa de la página web.

Para ver un ejemplo puedes consultar un Trending Topic de Twitter (o cualquier otro que se actualice con mucha frecuencia) y utiliza alguna herramienta que te permita ver el tráfico de red que se produce sin recargar la página (por ejemplo, el navegador Chrome dispone de la herramienta Inspeccionar elemento > Network).

3.4. Aplicaciones web dinámicas del lado servidor.

No sólo el navegador (lado cliente) puede crear dinamismo en las páginas web. El servidor web puede devolvernos una información u otra en una misma página web en función de la interacción del usuario o de cualquier otra circunstancia.

Para ello se utilizan diversos lenguajes de programación que generan la página web que se enviará finalmente al cliente y en un formato entendible por el navegador (HTML, CSS, JavaScript, etc). La página web que es enviada deberá emplear los lenguajes reconocidos por el navegador, ya que los navegadores web no reconocen los lenguajes de programación que se emplean en los servidores web. En este tipo de aplicaciones es muy frecuente que el servidor web acceda a una base de datos para obtener los datos que finalmente formarán parte de la página web.



A continuación se muestran algunas de las técnicas y lenguajes de programación que se pueden emplear para el desarrollo de aplicaciones web del lado servidor.

3.4.1. CGI.

El Common Gateway Interface (CGI) es un método estándar empleado por servidores web para delegar la generación del contenido web a archivos ejecutables. Dichos archivos son conocidos como scripts CGI, que son aplicaciones escritas generalmente en lenguajes de programación que permiten crear archivos ejecutables como C, C++, Perl, Java, Visual Basic ... aunque es muy habitual el uso de Perl para el desarrollo de scripts CGI por su facilidad en el manejo de cadenas de caracteres.

Las aplicaciones CGI fueron unas de las primeras técnicas empleadas para crear contenido dinámico para las páginas web.

Seguidamente puedes ver un ejemplo de código fuente Perl que crea un script CGI para mostrar la fecha actual en una página web:

```
#!/usr/bin/perl

@months = qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec);
@weekDays = qw(Sun Mon Tue Wed Thu Fri Sat Sun);
($second, $minute, $hour, $dayOfMonth, $month, $yearOffset, $dayOfWeek, $dayOfYear, $daylightSavings) = localtime();
$year = 1900 + $yearOffset;
$time = "$weekDays[$dayOfWeek] $months[$month] $dayOfMonth, $year";

print "Content-type: text/html\n\n";
print <<HTML;
<html>
<head>
<title>A Simple Perl CGI</title>
</head>
<body>
<h1>A Simple Perl CGI</h1>
<p>$time</p>
</body>
HTML
exit;
```

3.4.2. Servlets.

Un Servlet es una tecnología web basada en Java para el lado servidor, basada en recibir las respuestas del cliente y ofrecer una respuesta al servidor web. Es una clase Java de Java EE que utiliza la API Java Servlet, y su código suele tener código HTML embebido. **El contenido generado** por el Servlet suele ser HTML pero puede generar otros tipos de datos como XML.

Para ejecutar un Servlet es necesario usar un contenedor web (también conocido como contenedor de Servlets) que es un componente de un servidor web que interactúa con los Servlets. El contenedor es responsable de manejar el ciclo de vida de los Servlets, mapear una URL a un Servlet en particular y asegurarse de que el solicitante de la URL tiene los permisos de acceso adecuados. Las ventajas de usar Servlets en vez de CGI es que los primeros tienen mejor rendimiento y facilidad de uso combinado con más potencia.

A continuación puedes ver un ejemplo de código fuente de un Servlet que obtiene los datos introducidos por el usuario en un formulario anterior:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ParamServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        // Obtenemos un objeto Print Writer para enviar respuesta
        res.setContentType("text/html");
        PrintWriter pw = res.getWriter();
        pw.println("<HTML><HEAD><TITLE>Leyendo parámetros</TITLE></HEAD>");
        pw.println("<BODY BGCOLOR=\"#CCBBAA\">");
        pw.println("<H2>Leyendo parámetros desde un formulario html</H2><P>");
        pw.println("<UL>\n");
        pw.println("Te llamas " + req.getParameter("NOM") + "<BR>");
        pw.println("y tienes " + req.getParameter("EDA") + " años<BR>");
        pw.println("</BODY></HTML>");
        pw.close();
    }
}
```

3.4.3. Lenguajes embebidos en el código HTML (PHP, ASP, JSP).

A diferencia de los lenguajes comentados anteriormente, cuyo código genera etiquetas HTML, existen otros lenguajes de programación de aplicaciones web del lado servidor cuyas instrucciones se encuentran embebidas en el código HTML. Es el caso de lenguajes como PHP, ASP o JSP. El servidor tiene almacenadas los archivos que contienen el código correspondiente a las páginas web. Dichos archivos contienen etiquetas HTML y código en otro lenguaje que es interpretado por el servidor, ejecutando sus instrucciones con las que se obtiene un código reconocible por el navegador cliente (HTML, CSS, JavaScript, etc). El resultado final obtenido es el que se envía al cliente.

Estas son las características principales de los lenguajes más utilizados en esta categoría:

- **PHP:** Es uno de los primeros lenguajes del lado servidor creados para ser embebido en el código HTML en lugar de llamar a un archivo externo para procesar los datos. El código es interpretado por un servidor web con un módulo procesador de PHP que genera la página web resultante. PHP se puede emplear en la mayoría de los servidores web de los principales sistemas operativos de forma gratuita al ser software libre. Se estima que se utiliza en más de 20 millones de sitios web. Por ejemplo se usa en gestores de contenido

web como Joomla o Drupal, blogs como Wordpress, o aulas virtuales como Moodle.

- **ASP:** Active Server Pages (ASP), también conocido como ASP clásico, fue la primera tecnología de Microsoft para generar páginas web de forma dinámica desde el lado del servidor, lanzándose como un añadido al servidor Internet Information Services (IIS). La mayoría de los programadores crean las páginas ASP usando VBScript, pero se pueden utilizar otros lenguajes.
- **ASP.NET:** Es el sucesor de la tecnología ASP y permite a los programadores escribir código usando cualquier lenguaje .NET soportado, como VB.NET o C#. Sólo funciona sobre el servidor de Microsoft IIS, lo que supone una desventaja respecto a otros lenguajes del lado de servidor que son ejecutables sobre otros servidores más populares.
- **JSP:**
 - JavaServer Pages (JSP) es otra tecnología para crear dinámicamente páginas web de forma similar a PHP pero usando el lenguaje de programación Java. Para utilizarla es necesario disponer de un contenedor de servlets como Apache Tomcat.
 - El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del navegador del usuario.
 - El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.
 - JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.