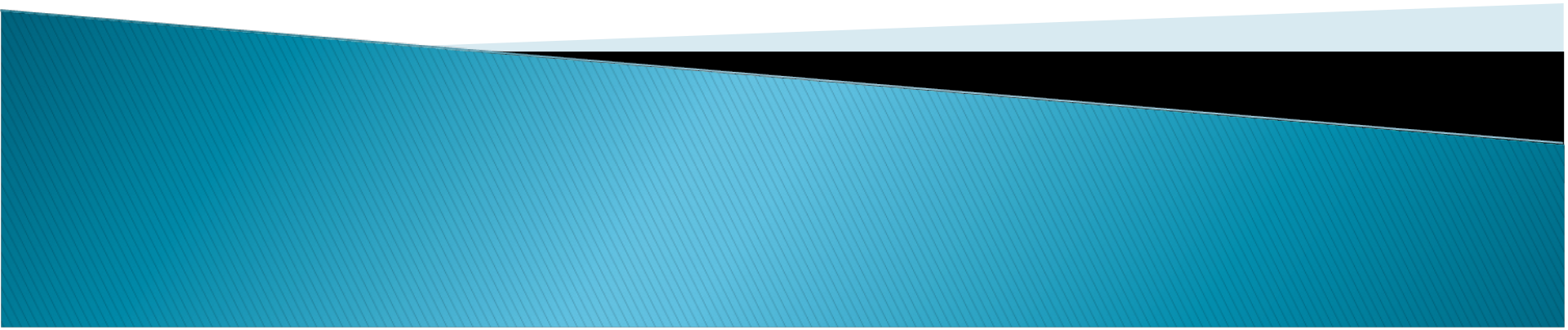
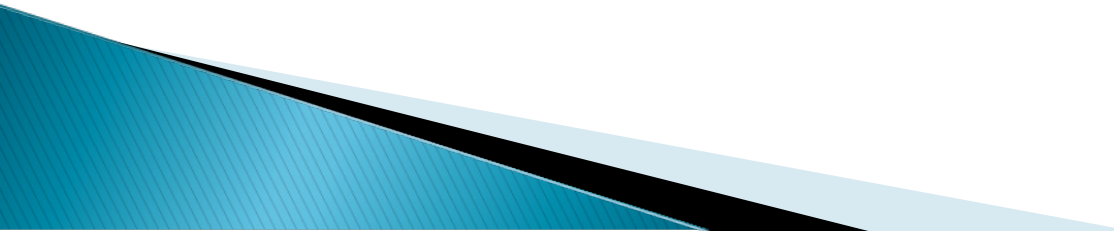


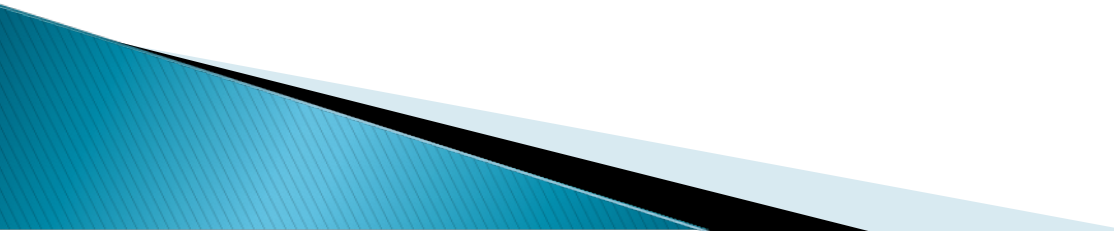
# SERVICIO HTTP



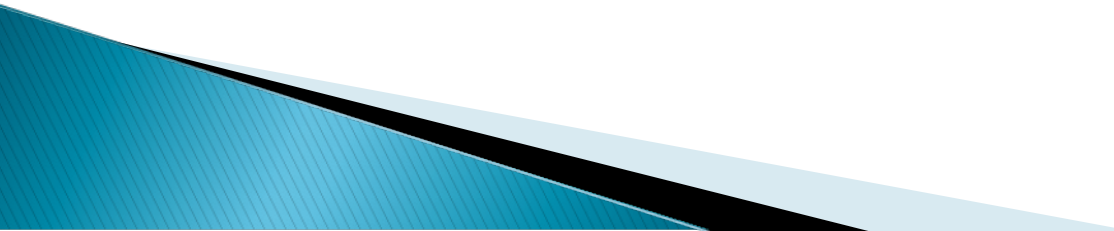
# Objetivos de aprendizaje

- ▶ Analizar aspectos generales de arquitecturas web, sus características, ventajas e inconvenientes.
  - ▶ Describir los fundamentos y protocolos en los que se basa el funcionamiento de un servidor web.
  - ▶ Realizar la instalación y configuración básica de servidores web.
  - ▶ Realizar pruebas de funcionamiento de los servidores web y de aplicaciones.
  - ▶ Reconocer los parámetros de administración más importantes del servidor web.
- 


# Objetivos de aprendizaje

- ▶ Ampliar la funcionalidad del servidor mediante la activación y configuración de módulos.
  - ▶ Crear y configurar sitios virtuales.
  - ▶ Configurar los mecanismos de autenticación y control de acceso del servidor.
  - ▶ Establecer mecanismos para asegurar las comunicaciones entre el cliente y el servidor.
  - ▶ Realizar pruebas de funcionamiento y rendimiento del servidor web.
  - ▶ Elaborar documentación relativa a la configuración, administración segura y recomendaciones de uso del servidor.
  - ▶ Realizar los ajustes necesarios para la implantación de aplicaciones en el servidor web.
- 

# Servidor web

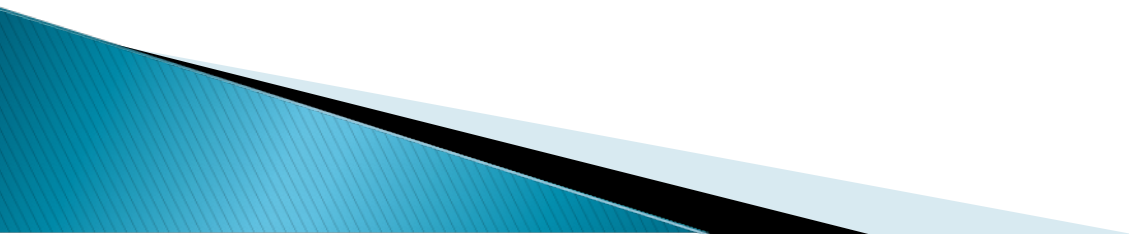
- ▶ Es un programa que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente.
  - ▶ El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.
  - ▶ Está diseñado para servir el Contenido de HTTP.
- 

# Protocolo HTTP

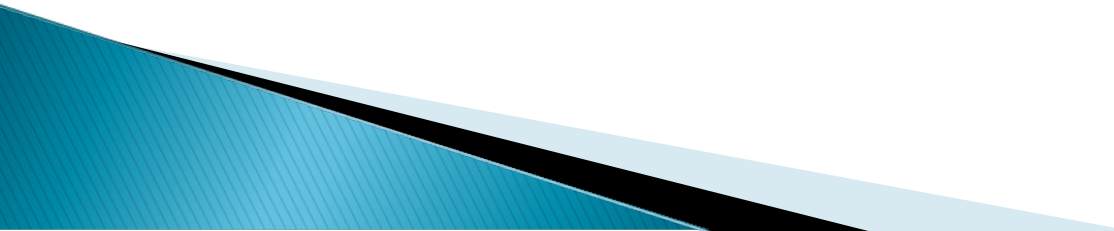
- ▶ HTTP (HyperText Transfer Protocol) es un protocolo de transferencia de hipertexto.
  - ▶ El método más habitual de transferir páginas webs en la World Wide Web (www).
  - ▶ Se creó en el CERN (Ginebra, Suiza) en 1990, para compartir datos científicos entre las distintas naciones de forma muy rápida y a un precio bajo.
  - ▶ HTTPS: realiza transferencia de páginas de forma segura, usando métodos de cifrado.
  - ▶ El servicio HTTP es la base sobre la que asienta todo el entramado que hace posible la comunicación a escala intercontinental.
- 

# Protocolo HTTP

- ▶ El modelo que sigue la www es de cliente-servidor.
- ▶ El cliente HTTP es el navegador web.
- ▶ El cliente no tiene por qué saber donde se aloja el sitio web al que desea acceder, simplemente dispone del nombre del sitio.



# Protocolo HTTP

- ▶ El protocolo http utiliza por defecto el puerto 80.
  - ▶ El navegador cliente es capaz de interpretar los distintos lenguajes en los que están escritas las páginas, como HTML, javascript, etc.
  - ▶ Un cliente puede solicitar una página al servidor tecleando en el navegador del cliente la IP del servidor.
- 

# Protocolo HTTP

## Petición GET:

- ▶ Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición **GET**, en el que el recurso se solicita a través de la url al servidor Web.

GET /index.html HTTP/1.1 HOST: www.host.com

- ▶ En la barra de URL de un navegador cualquiera, la petición anterior sería análoga a la siguiente dirección Web:

www.host.com/index.html






# Protocolo HTTP

- ▶ El navegador, permite al usuario realizar una o varias peticiones Web. Una petición Web no sólo puede ser realizada mediante un navegador, sino con cualquier herramienta habilitada para tal fin, como una consola de comandos Telnet.
- ▶ Elementos del entorno de usuario más comunes en navegadores :


Nombre	Descripción
Link	Es una porción de contenido Web, texto, imagen y otros elementos, que enlaza con una dirección Web. Al pulsar un hipervínculo, el navegador genera una petición GET automática a la dirección URL de dicho link.
Formulario web	Al realizar el envío satisfactorio de los datos de un formulario, el navegador Web genera una petición GET o POST (comúnmente POST) automática a la par que envía los datos al servidor.
Barra de direcciones	Todos los navegadores incluyen una barra de direcciones mediante la cual puede accederse manualmente a cualquier dirección URL, de modo que el navegador generará una petición GET automática a dicha URL cada vez que el usuario lo desee.
Script	Cualquier aplicación Javascript tiene acceso al estado del navegador, cómo puede modificar los datos que describen tal estado, de forma pasiva (sin medio de la intervención del usuario) o de forma activa (mediante alguna acción del usuario).

# Protocolo HTTP

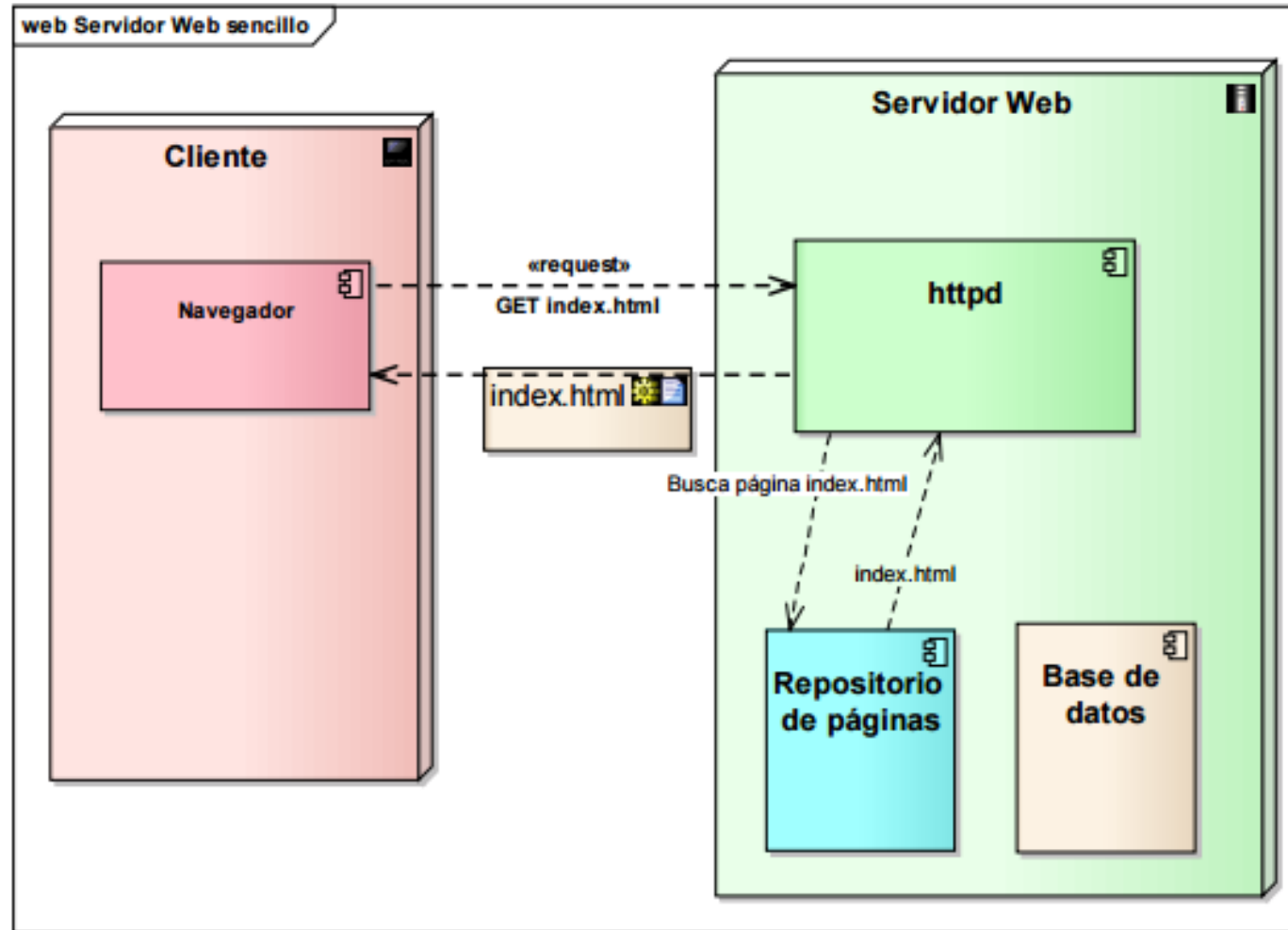
## Petición POST:

- ▶ Más utilizado.
  - ▶ Los datos a enviar al servidor se incluyen en el cuerpo de la misma petición con las cabeceras HTTP asignadas correspondientemente respecto al tipo de petición.
  - ▶ Generalmente se asocia con los formularios web en los que los datos suelen ser cifrados para enviarlos de manera segura al servidor.
- 

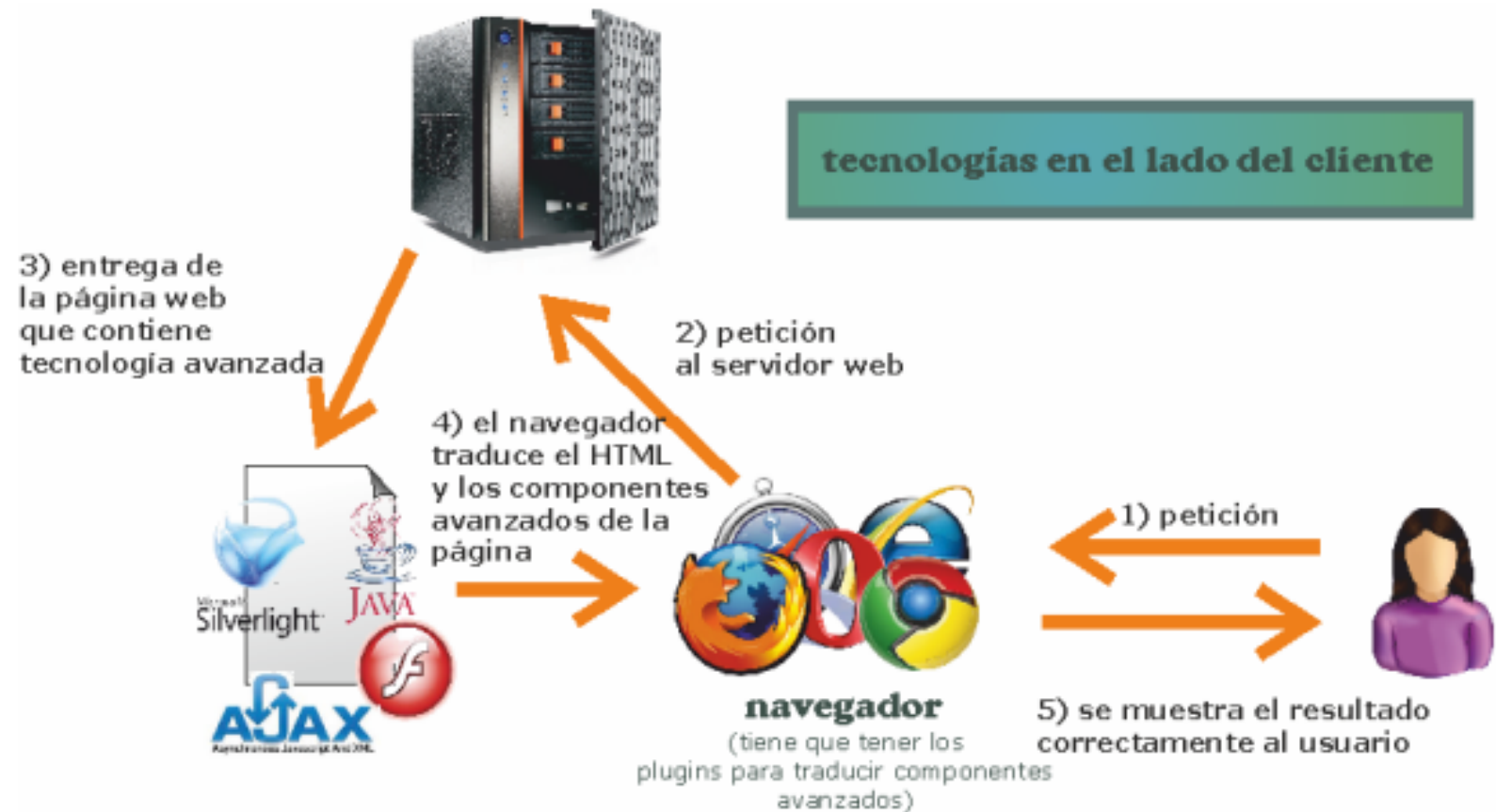
# Protocolo HTTP

- ▶ Para acceder a una página se puede proporcionar la URL (Uniform Resource Locator o Localizador Uniforme de Recursos) al navegador.
  - ▶ Si la dirección web no existe, es el servidor DNS el que envía el mensaje de error al navegador del cliente.
  - ▶ Si la dirección web es correcta (el servidor web existe), pero no es correcto el fichero o la ruta (dentro del servidor web no existe) el mensaje de error lo envía el servidor web.
- 

# Funcionamiento

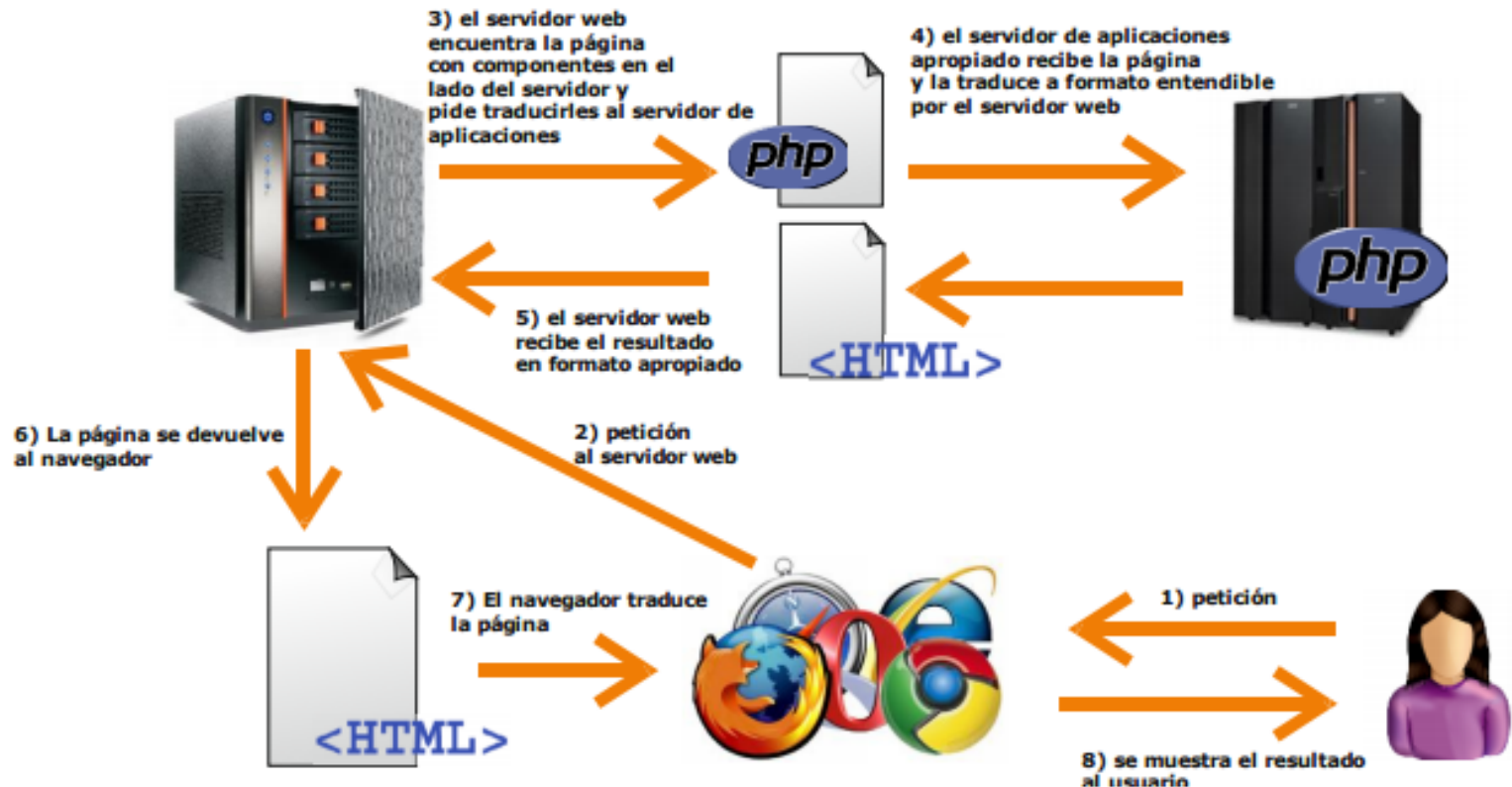


# Tecnologías lado cliente

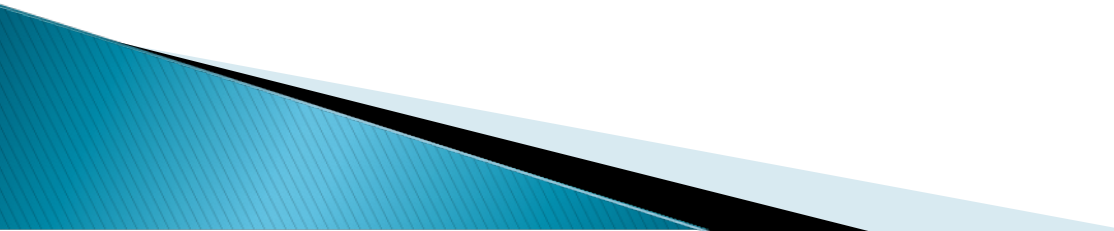


# Tecnologías lado servidor

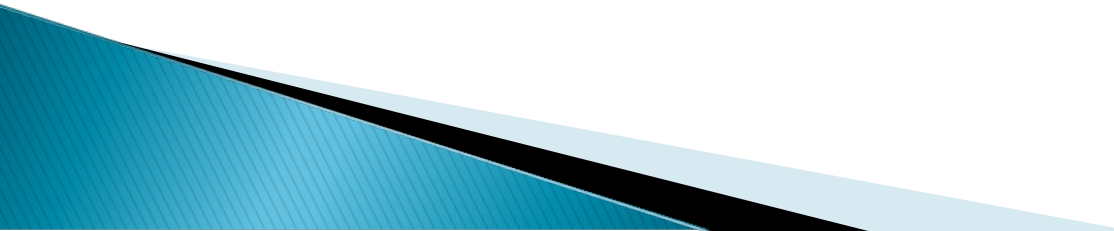
## tecnologías en el lado del servidor



# El servidor...

- ▶ Espera las peticiones
  - ▶ Envía archivos
  - ▶ Ejecuta las respuestas a las peticiones y envía los resultados
  - ▶ Establece conexión a Sistemas de Bases de Datos
  - ▶ Actúa de puerta de enlace para servicios como el correo, ftp, etc
- 

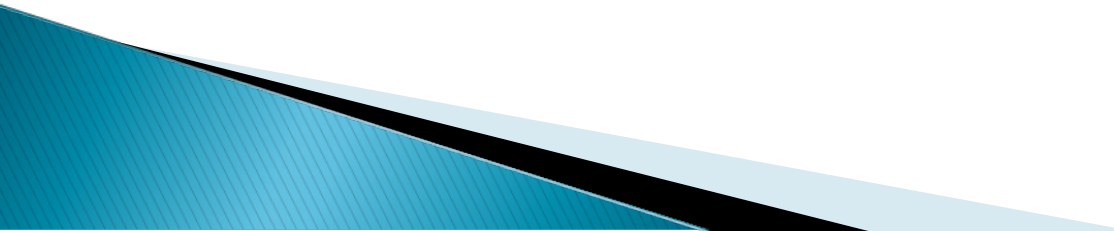
# El cliente...

- ▶ Realiza las peticiones
  - ▶ Interpreta el código HTML que recibe.
  - ▶ Interpreta y ejecuta scripts “del lado del cliente” como javascripts.
  - ▶ Arranca aplicaciones externas.
  - ▶ Controla aspectos del formato del documento.
- 



# Programas

Algunos servidores web importantes son:

- ▶ Nginx
  - ▶ Apache
  - ▶ Internet Information Services
  - ▶ Cherokee
  - ▶ Tomcat
- 

# Actividades



[1-11]

# MiME


***Multipurpose Internet Mail Extensions*** o en español "extensiones multipropósito de correo de internet.

Son una serie de especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.

Está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.



# MIME

- ▶ Inicialmente las páginas web contenían texto y enlaces (hypertext) por lo que inicialmente el protocolo HTTP te permitía enviar únicamente texto.
  - ▶ Actualmente HTTP nos permite enviar todo tipo de documentos gracias a los tipos MIME.
  - ▶ Utiliza encabezados que describen el contenido del propio mensaje mediante codificación MIME.
- 

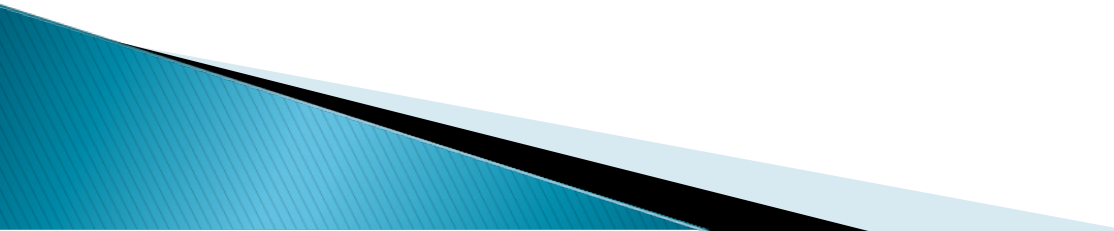
# MIME

- ▶ En 1991 la IETF comenzó a desarrollar esta norma y desde 1994 todas las extensiones MIME están especificadas de forma detallada en diversos documentos oficiales disponibles en Internet.
- ▶ MIME está especificado en seis *Request for Comments* o RFC.

# MiME

IANA es el organismo internacional que define los tipos MIME.

Podemos ver los diferentes tipos MIME que existen en el fichero de linux **/etc/mimes.types** o (C:\Windows\system32\drivers\etc\mimes.types).



# MIME

En sentido general las extensiones de MIME van encaminadas a soportar:

- Texto en conjuntos de caracteres distintos de US-ASCII;
- Adjuntos que no son de tipo texto;
- Cuerpos de mensajes con múltiples partes (multipart);
- Información de encabezados con conjuntos de caracteres distintos de ASCII.

# Tipos de MIME

Los tipos MIME se utilizan en los siguientes casos:

- ▶ Para informar al navegador del tipo de datos que esta recibiendo del servidor. Se define usando: **Content-type** dependiendo de si el navegador puede interpretarlo puede: visualizar el documento (por ejemplo text/html), llamar a una aplicación externa (application/pdf al descargar un documento pdf lo leemos con un lector de pdf) o preguntar al usuario que hacer si no sabe con que abrir su contenido (application/abb).
- ▶ Para permitir o denegar un contenido: Por ejemplo, **Allow: application/doc**
- ▶ Para encapsular uno o mas objetos dentro del cuerpo del mensajes. El más conocido son los tipos multipart (multipart/form-data, que se suelen utilizar junto al método POST para hacer un upload de una imagen al servidor.



# MIME Headers

## MIME version:

- ▶ Indica que el mensaje utiliza el formato MIME. Su valor es típicamente igual a "1.0":

MIME-Version: 1.0

## Content-Type

- ▶ Indica el tipo de medio que representa el contenido del mensaje, consiste en un tipo: *type* y un subtipo: *subtype*, por ejemplo:

Content-Type: text/plain



# MIME Headers

**Content-Transfer-Encoding:** indica el método que ha sido usado:

Content-transfer-encoding: base64;

- **7 bits:** formato de texto de 7 bits (para mensajes sin caracteres acentuados);
- **8 bits:** formato de texto de 8 bits;
- **QP:** formato QP, recomendado para mensajes que usan un alfabeto de 7 bits (como cuando hay acentos);
- **base 64:** Base 64, recomendado para enviar archivos binarios como adjuntos;
- **binario:** formato binario; no recomendado.

# Tipos de MIME

**texto:** texto de datos legible text/rfc822 [RFC822]; text/plain [RFC2646]; text/html [RFC2854].

**imagen:** datos binarios que representan imágenes digitales: image/jpeg, image/gif, image/png.

**audio:** datos de sonido digital: audio/basic, audio/wav

**video:** datos de vídeo: video/mpeg

**aplicación:** Otros datos binarios: application/octet-stream, application/pdf

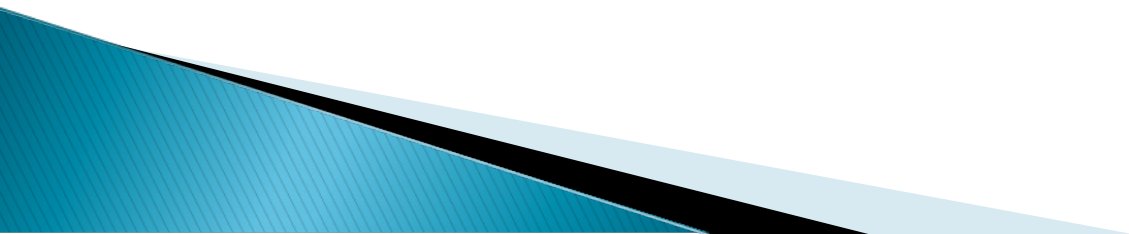
# Actividades



[12-17]

# El comando TELNET

- ▶ Telnet es un programa basado en texto que sirve para conectarse a otro equipo por medio de Internet.
- ▶ Podrá obtener acceso a programas y servicios que se encuentren en el equipo remoto como si estuviese sentado frente a él.



# El comando TELNET

- ▶ Se puede emplear el comando telnet seguido del nombre o la dirección IP de la máquina en la que desea trabajar: telnet purpura.micolegio.edu.com
- ▶ Para conectarse con la dirección IP 1.2.3.4 deberá utilizar telnet 1.2.3.4.
- ▶ Una vez conectado, podrá ingresar el nombre de usuario y contraseña remoto para iniciar una sesión en modo texto a modo de consola virtual .

# Actividades




[18]

# Instalación Apache Ubuntu

Instalación:


```
sudo apt-get install apache2
```

Archivos de configuración:

- `apache2.conf`: el archivo de configuración principal de Apache2.
  - `conf-available`: esta carpeta contiene los archivos de configuración disponibles.
  - `envvars`: archivo en el que se establecen las variables de entorno de Apache2.
- 



# Instalación Apache Ubuntu

- `mods-available`: este directorio contiene los archivos de configuración para cargar módulos y configurarlos.
  - `ports.conf`: almacena las directivas que determinan los puertos TCP por los que Apache2 está escuchando.
  - `magic`: instrucciones para determinar el tipo MIME en base a los primeros bytes de un archivo.
- 

# Instalación Apache Ubuntu

Para reiniciar Apache: `sudo /etc/init.d/apache2 restart`

Comprobar instalación:

Ir a: `http://localhost :`

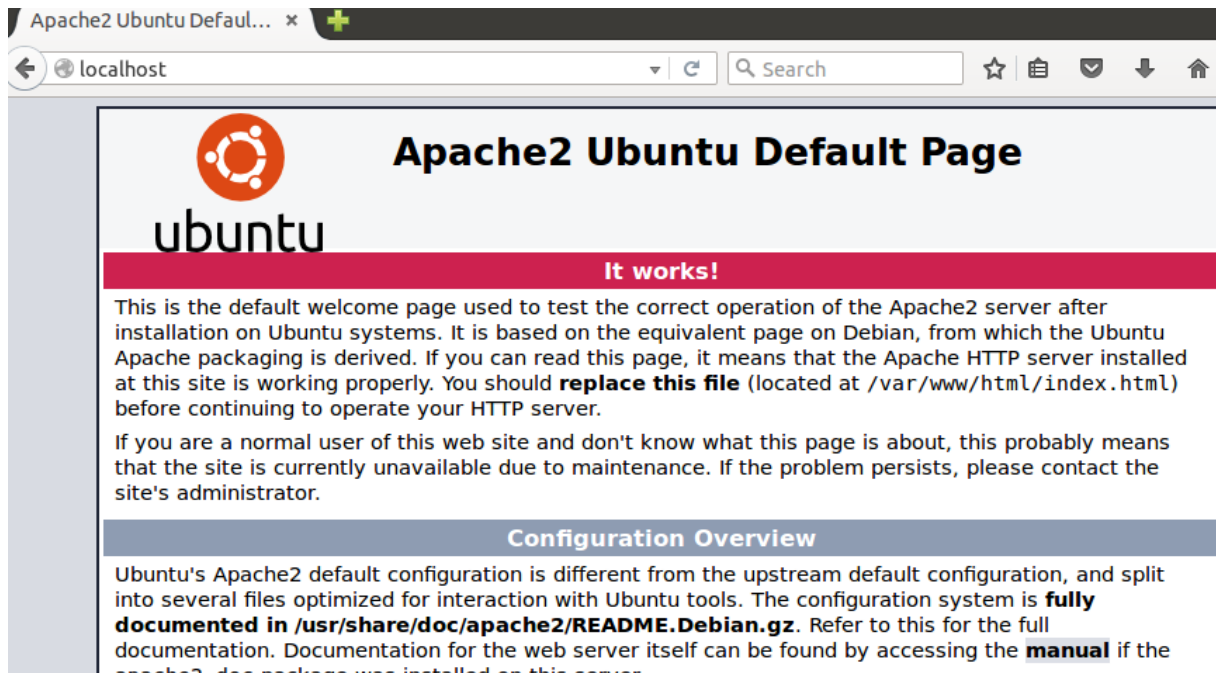
Si lees "It works!" → funciona

Es el contenido del fichero:

`/var/www/index.html`



# Instalación Apache Ubuntu



# Instalación Apache Ubuntu

Si no ha salido:

```
sudo nano /etc/apache2/conf-  
available/fqdn.conf
```

Añadimos: ServerName localhost y guardalo.

```
sudo a2enconf fqdn
```



# Añadir más páginas

Copia la página predeterminada para tener una referencia con el nombre que queramos:

```
sudo cp /etc/apache2/sites-available/000-  
default.conf /etc/apache2/sites-  
available/mysite.conf
```

Edita el nuevo fichero de configuración:

```
gedit /etc/apache2/sites-available/mysite.conf
```



# Añadir más páginas

Cambia el DocumentRoot para que apunte a un nuevo directorio. Por ejemplo:

`/home/user/public_html/`

En el archivo: `/etc/apache2/apache2.conf`,  
cambia el directorio:

`<Directory /var/www/>` to `<Directory  
/home/user/public_html/>`



# Añadir más páginas

Tenemos que desactivar el viejo y activar la nueva utilizando estos comandos: a2ensite (apache2enable site) and a2dissitesuso (apache2disable site).

```
$ sudo a2dissite 000-default && sudo a2ensite  
mysite
```

```
$ sudo /etc/init.d/apache2 restart
```



# Securizar Apache

```
$ gedit /etc/apache2/ports.conf
```

Cambialo para que contenga:

```
Listen 127.0.0.1:80
```

Guarda y reinicia Apache.

Ahora solo escuchara desde el puerto 80 a tu localhost.

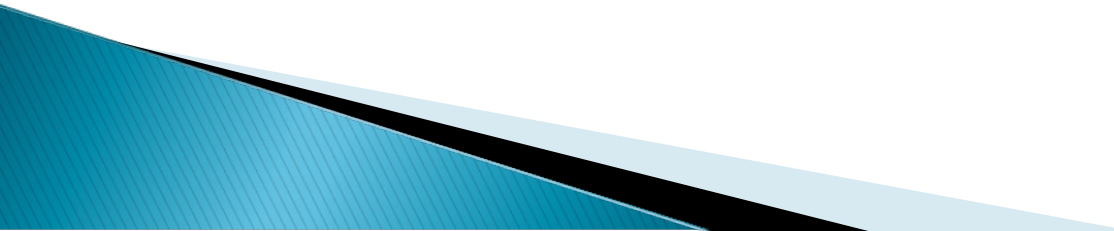




# Autenticación y autorización de usuarios

Cuando un servidor apache recibe una petición de una página web, lleva a cabo varias acciones para verificar que la petición esta autorizada.

Las distintas acciones para verificar la validez de la aplicación:

- Autenticación.
  - Autorización.
  - Control de Acceso.
- 


# Autenticación y autorización de usuarios

## La autenticación:

Se verifica la identidad de una persona.

Se puede llevar a cabo mediante un nombre de usuario y una contraseña.

Se pueden llegar a utilizar otros métodos para validar la identidad de una persona, como mediante el uso de certificados, tarjetas etc...



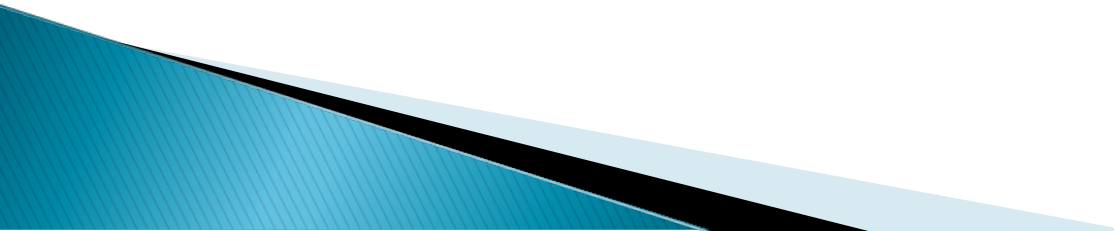
# Autenticación y autorización de usuarios

## La autenticación:

Puede estar gestionada por distintos módulos, dependiendo de la forma de implementación.

Si decide llevarla a cabo gestionando ficheros con listas de usuarios y contraseñas (encriptadas), deberá utilizar el módulo `mod_auth`.

Sin embargo, si decide llevarla a cabo mediante base de datos, deberá utilizar los módulos `mod_auth_dbm`.




# Autenticación y autorización de usuarios

## La autorización:

Se verifica que un usuario con una identidad conocida, tiene acceso al recurso solicitado.

Se suelen utilizar listas de permisos en las cuales se enumeran cada una de las acciones que puede realizar un usuario, o las que no puede hacer.

En apache la autorización a recursos es gestionada o bien mediante la directiva <directory> en el fichero principal de configuración, o bien mediante la configuración de la carpeta a través de ficheros .htaccess.



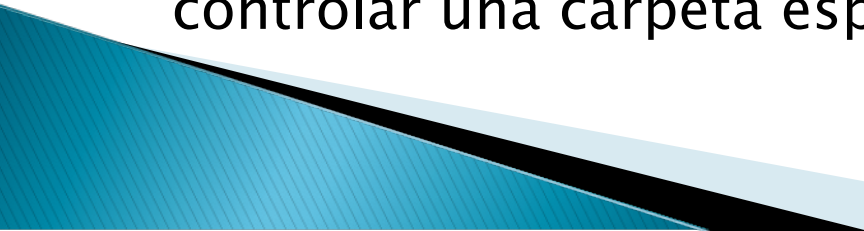
# Autenticación y autorización de usuarios

## El control de acceso

Se verifica que la máquina desde la que se ha hecho la petición, tiene acceso al recurso.

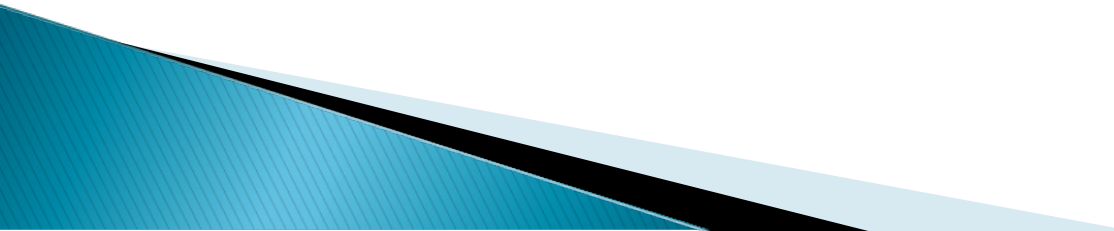
Se utilizan para limitar y controlar las máquinas que tienen acceso a un recurso independientemente del usuario que accede, ya que estos controles se llevan a cabo antes de que se realice el proceso de autenticación.

En apache, el control de acceso se puede llevar a cabo mediante las directivas `<directory>`, `<files>` y `<location>`, o a través del fichero de configuración `.htaccess` para controlar una carpeta específica.



# Autenticación y autorización de usuarios

En todo caso y para poder llevar a cabo la configuración de las tres es necesario tener la directiva **AllowOverride** con el valor **AuthConfig**, para así permitir el uso de las distintas directivas de autenticación.

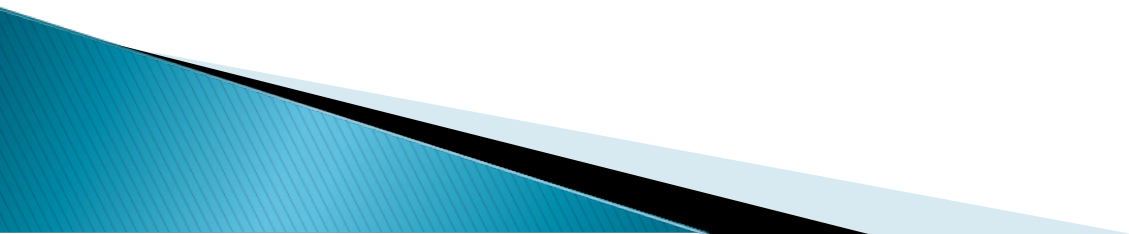


# Autenticación y autorización de usuarios

Para autenticar a los usuarios y verificar la autorización del mismo al recurso solicitado:

Crear un fichero con usuarios

Definir las directivas en el fichero de configuración o mediante un fichero .htaccess



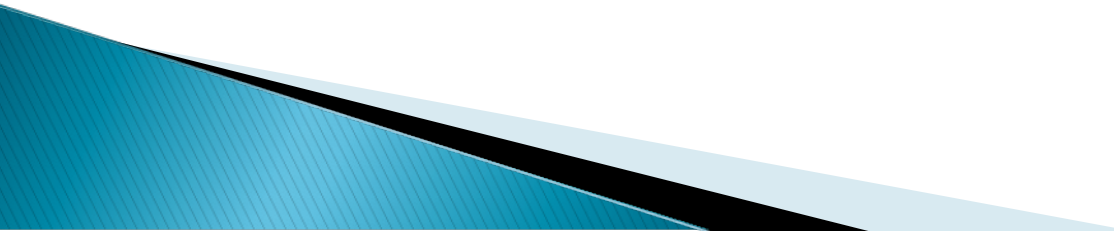
# Autenticación y autorización de usuarios

En los ficheros de usuarios de apache, en cada línea se especifica un usuario, escribiendo el nombre de usuario separado de dos puntos, seguido de la contraseña encriptada con MD5.



# Autenticación y autorización de usuarios

Es recomendable que los ficheros de usuarios se encuentren almacenados fuera de los directorios publicados, para que de esta forma nadie pueda descargarlos. Asimismo, solo el usuario root debería estar autorizado a escribir en él, mientras que solo el usuario que ejecuta el servicio web, debería estar autorizado para leerlo.



# Autenticación y autorización de usuarios

Para crear un fichero de usuario se utilizará la siguiente sintaxis:


```
htpasswd -c ruta/passwords usuario
```

El flag `-c` se utiliza para crear un fichero nuevo, por lo que sólo se deberá poner la primera vez que se crea el fichero, sino lo borrará.

# Autenticación y autorización de usuarios

Los módulos que intervienen en la autenticación y autorización son los de core y mod\_auth.

Las directivas de mod\_auth necesarias para configurar la autenticación y autorización son las siguientes:

- AuthUserFile: sirve para especificar la ruta donde se almacenará el fichero de usuarios.
  - AuthGroupFile: sirve para especificar la ruta donde se almacenará el fichero de grupos.
- 

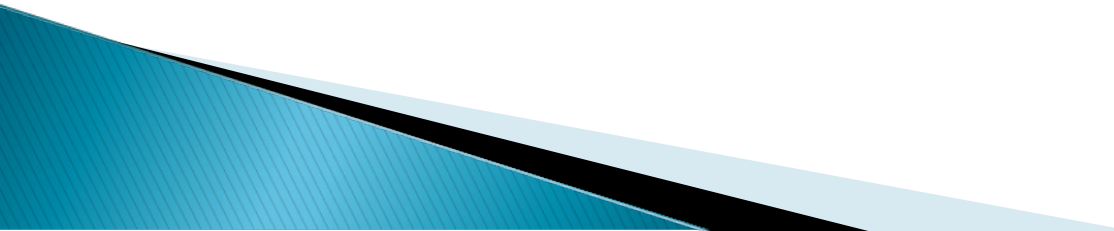
# Autenticación y autorización de usuarios

Las directivas necesarias para complementar la configuración del módulo son:

**AuthType:** selecciona el tipo de autenticación de usuarios que se utilizará para autenticar a un usuario. Puede variar por directorio. Los valores posibles son Basic y Digest. Con Basic, la transferencia de las claves se hará sin cifrar, y con digest se harán cifradas.

**AuthName:** especifica un nombre del dominio para el cual se solicita el acceso, este nombre figurará en la pantalla donde se pide la clave, y a su vez servirá para que el cliente identifique la contraseña que debe utilizar al enviar una petición a un área determinada.

**Require:** selecciona los usuarios que pueden acceder a un área determinada, los usuarios se pueden determinar a través de nombres o grupos.



# Autenticación y autorización de usuarios

Htpasswd -c /home/egibide/passwords alazne

Gedit /etc/apache2/sites-available/mysite.conf

```
</VirtualHost>
<Directory /home/egibide/public_html>
AllowOverride None
Order allow,deny
allow from 127.0.0.1 localhost
AuthName "Acceso privado"
AuthType Basic
AuthUserFile /home/passwords
Require user alazne
</Directory>
```

# Actividades




[20,21,22]

# Cómo configurar y personalizar la página de error 404

Las páginas de error 404 son tan típicas que es posible que todos las conozcamos sin hablar más de ellas.

En principio, las dos posibilidades que veremos son independientes una de la otra, por lo que podremos elegir la que más nos convenga.

Las dos maneras pasan por configurar la variable de sistema "ErrorDocument", que sirve para indicar el documento HTML que tiene que entregarse cuando surja un error de página no encontrada.



# Cómo configurar y personalizar la página de error 404

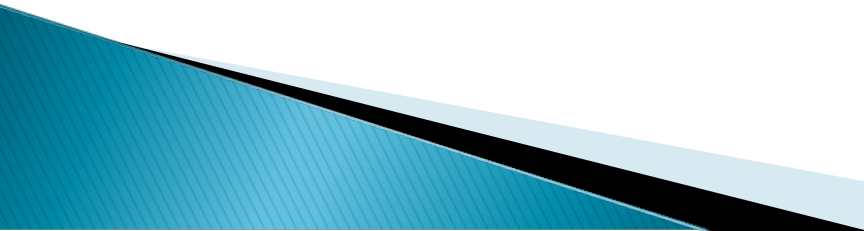
## Mediante httpd.conf:

Agregar una línea donde indiquemos la ruta del archivo que enviar al producirse un error de recurso no encontrado, por medio de la variable "ErrorDocument".

```
ErrorDocument 404 /mipagina404.html
```

Se indicaría bajo un error 404 que se debe enviar el documento que está en el archivo "mipagina404.html" (o cualquier otro documento que queramos utilizar como página de error).

La ruta indicada hacia el archivo comienza por "/", por lo que Apache entenderá que el mismo está en la raíz del dominio.



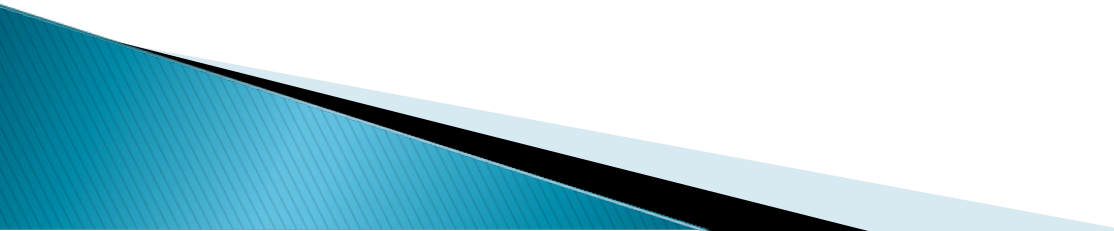


# Cómo configurar y personalizar la página de error 404

## Mediante .htaccess:

En algunos casos, el proveedor de alojamiento no nos permitirá editar el `httpd.conf`.

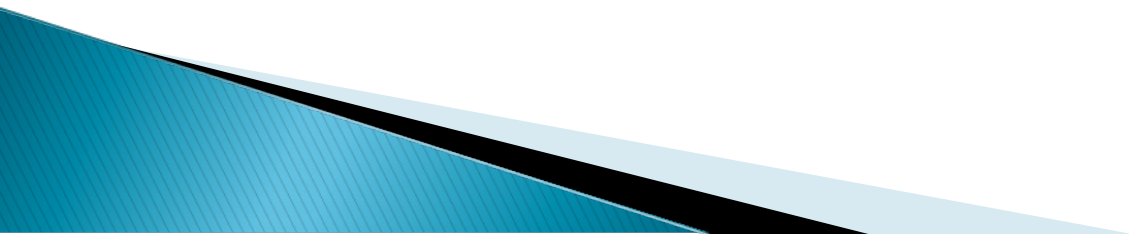
Para esto hay que crear un archivo que se llame `.htaccess` (el archivo empieza por ".", como si no tuviera nombre y sólo se indica esa extensión), donde colocaremos la directiva de configuración "ErrorDocument".



# Cómo configurar y personalizar la página de error 404

El archivo .htaccess se colocará en la raíz del dominio y tendrá el siguiente contenido.

```
ErrorDocument 404 /error404.html
```



# Actividades



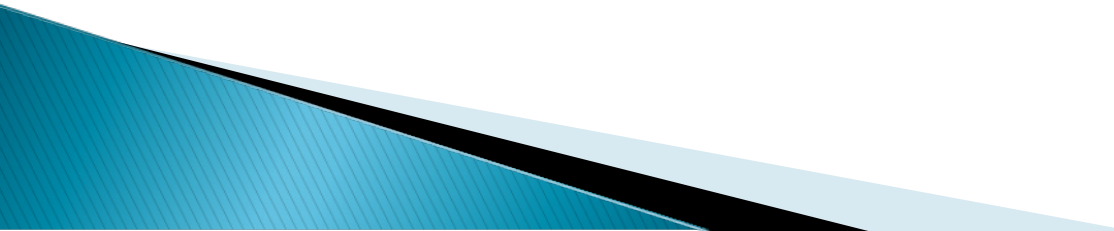
[23]

# Módulos Apache

El servidor Apache es un software que esta estructurado en módulos.

La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo.

Los módulos del Apache se pueden clasificar en tres categorías:



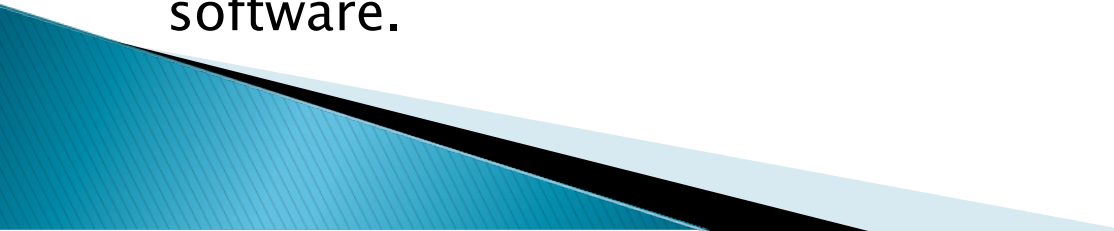
# Módulos Apache

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones.


Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar.

Para añadir un conjunto de utilidades al servidor, hay que añadirle un módulo, no es necesario volver a instalar el software.

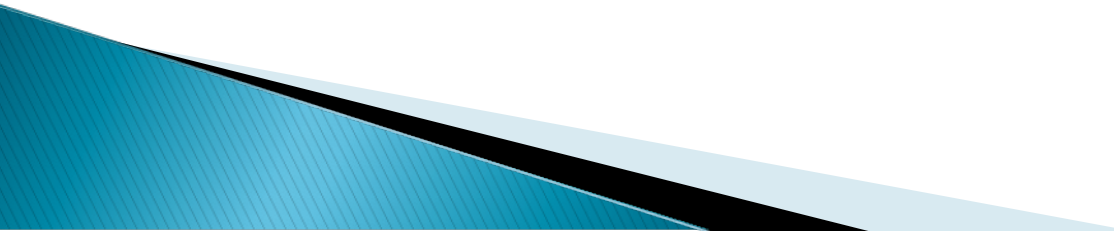



# Módulos Apache

- Módulos Base: Módulo con las funciones básicas del Apache.
  - Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
  - Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.
- 

# Módulos Apache

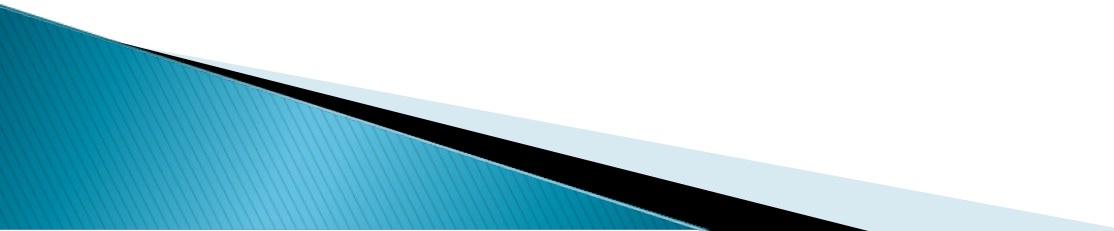
## Módulos Base y Módulos Multiproceso:

- core: Funciones básicas del Apache que están siempre disponibles.
  - prefork: Implementa un servidor sin hilos.
  - mod\_access: proporciona control de acceso basándose en el nombre del host del cliente, su dirección IP u otras características de la petición del cliente.
- 

- mod\_asis: envío de ficheros que tienen sus propias cabeceras http
  - mod\_auth: autenticación de usuario utilizando ficheros de texto.
  - mod\_auth\_anon: permite a usuarios anónimos acceder a áreas autenticadas.
  - mod\_auth\_digest: autenticación de usuario utilizando MD5.
  - mod\_deflate: comprime el contenido antes de ser enviado al cliente
- 



# Módulos Apache

- mod\_headers: personalización de las peticiones HTTP y las cabeceras de las respuestas.
  - mod\_info: proporciona una visión comprensiva de la configuración del servidor.
  - mod\_log\_config: registro de las peticiones hechas al servidor.
  - mod\_mime\_magic: determina el tipo MIME de un fichero mirando unos pocos bytes del contenido.
- 

# Module info

`sudo a2enmod info`

Para configurarlo añade lo siguiente a tu archivo `access.conf`.

```
<Location /server-info>  
SetHandler server-info  
</Location>
```

Una vez configurado la información es obtenida accediendo

A `http://your.host.example.com/server-info`



# Status Module

Modulo que permite ver el estado de nuestro servidor.

Puedes editar el fichero de configuración en “mods-available” con el siguiente comando:

```
sudo nano /etc/apache2/mods-  
available/status.conf
```

# Status Module

Para ver el estado del servidor:

.../server-status

Resultado :

Apache Server Status for 192.241.167.189

Server Version: Apache/2.2.22 (Ubuntu)

Server Built: Jul 12 2013 13:37:15

Current Time: Thursday, 08-Aug-2013 16:36:48 UTC

Restart Time: Thursday, 08-Aug-2013 16:04:59 UTC

Parent Server Generation: 2

Server uptime: 31 minutes 49 seconds

Total accesses: 3 – Total Traffic: 0 kB

CPU Usage: u0 s0 cu0 cs0

...



# Actividades



[24,25]

# Servidor Apache Tomcat

Apache Tomcat (también llamado Jakarta Tomcat) funciona como un contenedor de servlets.

Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP).

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.

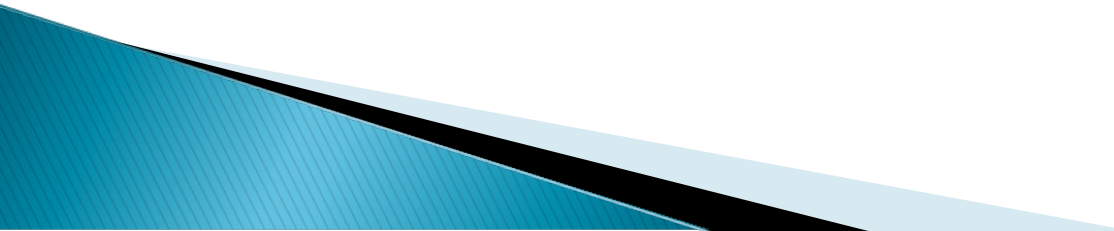


# Servidor Apache Tomcat

## Estado de su desarrollo:

Tomcat es desarrollado y actualizado por miembros de la Apache Software Foundation y voluntarios independientes.

Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License.




# Servidor Apache Tomcat

## Entorno:

El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones.

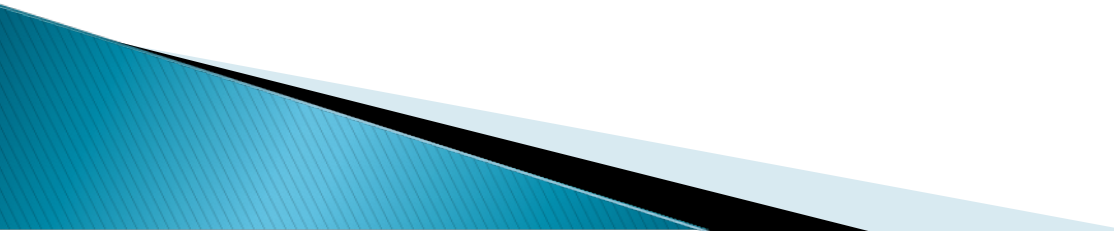
Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.





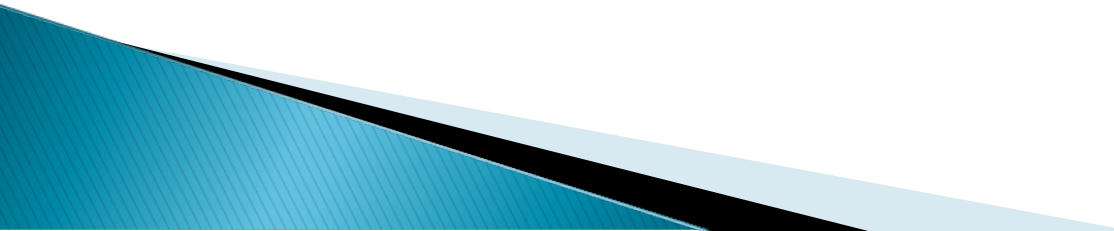
# Servidor Apache Tomcat

## Estructura de directorios:

- Bin: arranque, cierre, y otros scripts y ejecutables.
  - Common: clases comunes que pueden utilizar Catalina y las aplicaciones web.
  - Conf: ficheros XML y los correspondientes para la configuración de Tomcat.
  - Logs: logs de Catalina y de las aplicaciones.
- 

# Servidor Apache Tomcat

## Estructura de directorios:

- Bin: arranque, cierre, y otros scripts y ejecutables.
  - Common: clases comunes que pueden utilizar Catalina y las aplicaciones web.
  - Conf: ficheros XML y los correspondientes para la configuración de Tomcat.
  - Logs: logs de Catalina y de las aplicaciones.
- 

# Actividades



[26]

# Servidor Apache Tomcat

## Instalación:

1. En primer lugar, debemos tener instalado el compilador de Java, que se puede descargar de

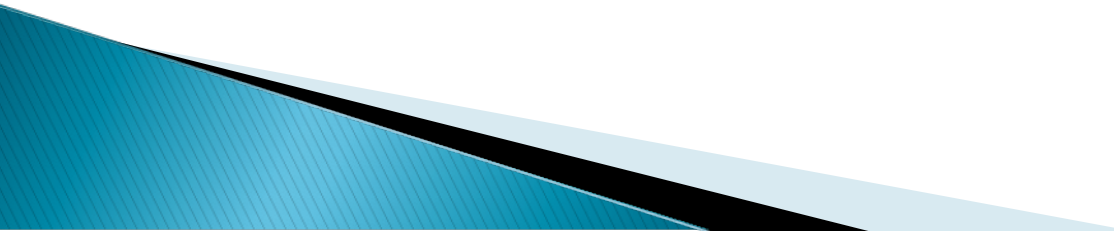
<http://java.sun.com/javase/downloads/index.jsp>

2. Hay que descargarse Tomcat de  
<http://tomcat.apache.org/>

La versión a elegir depende de la versión de Servlet que elijamos.

La última de Servlet, según

<http://java.sun.com/products/servlet/> es la 2.5 que corresponde con Tomcat 6.x, así que bajamos esa.



# Servidor Apache Tomcat

## Instalación:

3. En <http://tomcat.apache.org/download-60.cgi> elegimos hacia el final de la página la distribución binaria Windows Service Installer .

# Servidor Apache Tomcat

4. Ejecutamos el ejecutable que nos acabamos de descargar:

- Sale un presentación. Pulsamos "next"
- Sale la licencia. Pulsamos "I Agree".
- Pregunta qué queremos instalar. Dejamos la opción por defecto: "Normal" y opciones marcadas por defecto. Aquí podríamos elegir si queremos que se instalen los ejemplos.
- Pregunta en qué directorio. Ponemos uno a nuestro gusto o dejamos el de defecto.
- Pregunta puerto de conexión 8080, lo dejamos. Ponemos un administrador y password.
- Pregunta el path del jre de java. Si no tienes instalado hay que instalar:

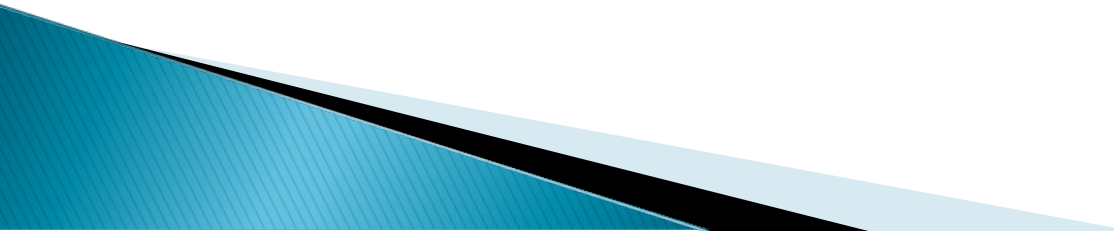
<http://www.oracle.com/technetwork/es/java/javase/downloads/jre7-downloads-1880261.html>

- Le damos a "install" y esperamos.

# Servidor Apache Tomcat

5. Cuando termina nos pregunta si queremos arrancar apache tomcat. Le decimos que sí. Luego, para probar, arrancamos el navegador y ponemos la siguiente dirección `http://localhost:8080`.

Si todo está bien instalado deberíamos ver en el navegador la página de bienvenida de Apache Tomcat.



# Actividades



[27, 28]

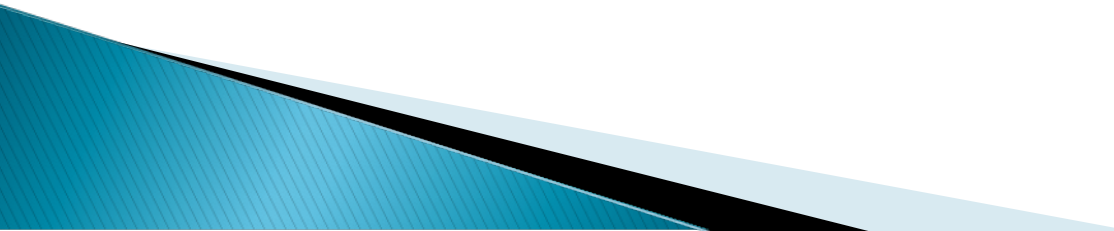


# Configurar Tomcat con un servidor Apache

Tomcat está pensado para servir páginas html, jsp y Servlets.

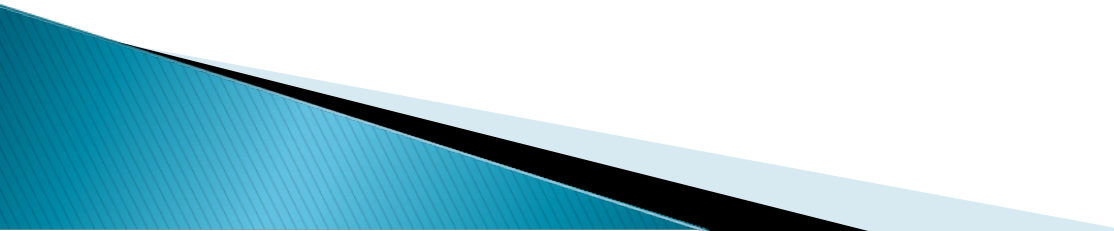
Suele ser más eficiente montar un servidor Apache normal y configurar Tomcat y Apache para que trabajen conjuntamente.

El navegador se conectará contra el servidor Apache para la mayoría de las cosas. Cuando el navegador pida una página jsp o un Servlet, Apache redirigirá la petición a Tomcat y será este el que nos sirva dicha página jsp o Servlet.



# Configurar Tomcat con un servidor Apache

Suponemos que ya tenemos instalado el Tomcat y el servidor Apache.

1. Descargarse el conector de apache con tomcat.
  2. Podemos bajarnos el binario si lo hay para nuestro sistema operativo de <http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/>  
¡Importante descargar el que corresponda a vuestra versión de apache.
- 

# Configurar Tomcat con un servidor Apache

3. Lo guardaremos con el nombre `mod_jk.so` en la carpeta de nuestro servidor apache `/modules`
4. Abriremos el fichero de configuración de apache `conf/httpd.conf` e incluiremos la siguiente línea al final del fichero:

```
#Carga el mod_jk module
LoadModule jk_module modules/mod_jk.so
# Lo configura workers.properties
JkWorkersFile conf/workers.properties
#Dónde poner los logs logs
JkLogFile logs/mod_jk.log
#Nivel de log [debug/error/info]
JkLogLevel info
#Formato del log
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
#Manda toda url que contenga /misjsp/ al tomcat ajp13w
JkMount /miaplicacion/* miservidor
```

# Configurar Tomcat con un servidor Apache

5. En la misma carpeta se crea el fichero worker.properties:

```
worker.list=miservidor
```

```
worker.miservidor.port=8009
```

```
worker.miservidor.host=localhost
```

```
worker.miservidor.type=ajp13
```

```
worker.miservidor.socket_timeout=3600
```

6. En Apache, quitale el comentario a esta linea en el archivo httpd:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

7. En el archivo httpd, incluye las siguientes lineas al final, pero antes de la configuración de jk:

```
#Activación
```

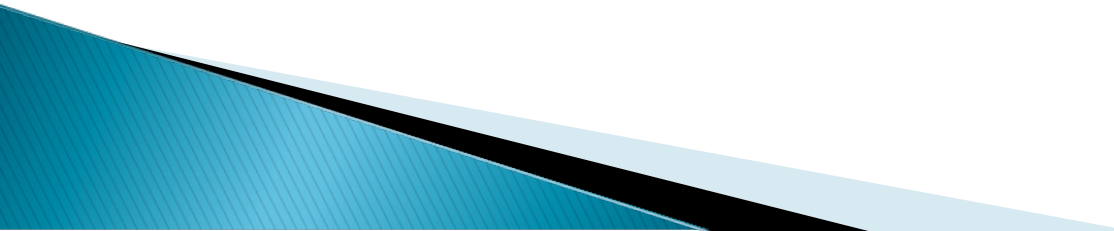
```
RewriteEngine on
```

```
#Configuración
```

```
RewriteRule ^(.*) (png|jpg|gif|css|js)$ /$1$2 [NC,L]
```

```
RewriteRule ^/(.*)$ /miaplicacion/$1 [PT,QSA,NC,L]
```

# Configurar Tomcat con un servidor Apache

8. Reiniciamos tanto tomcat como apache.
  9. Metemos los archivos jsp dentro de webapps/root/misjsp
  10. entramos en localhost desde el puerto de apache.
  11. Vemos que la pagina esta visible.
- 

# Actividades



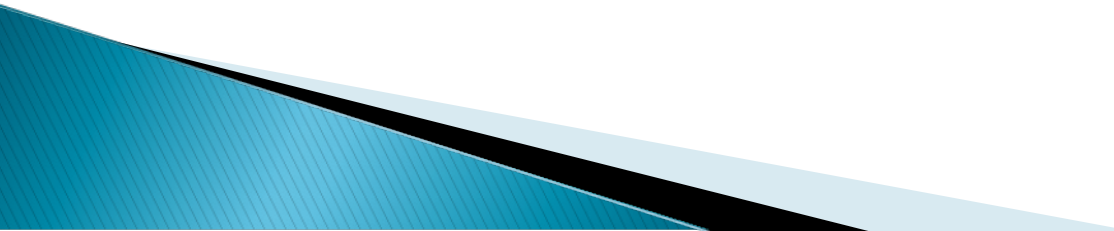
[29, 30]

# Servidor IIS

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.

Este servicio convierte a un PC en un servidor web.

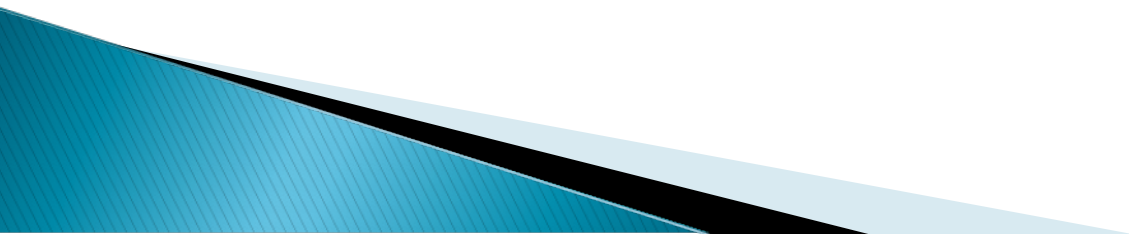
Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.



# Servidor IIS

De forma predeterminada, IIS 7.5 no está instalado en Windows Server® 2008 R2.

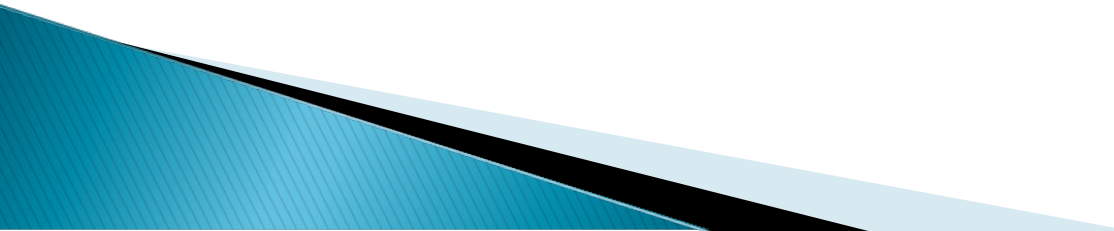
Se puede instalar utilizando el asistente para Agregar roles en el Administrador de servidores.





# Servidor IIS

Utilizar la interfaz de usuario

1. Haga clic en Inicio, señale a Herramientas administrativas y, a continuación, haga clic en Administrador del servidor.
  2. En Resumen de roles, haga clic en Agregar roles.
  3. Utilice el Asistente para agregar roles para agregar el rol del servidor web.
- 

# Actividades



[31, 32]