

Introducción

- El HTTP es la base arquitectura que conocemos como Web. Algunas fechas:
 - HTTP surgió como idea en el CERN en 1989.
 - El primer prototipo estuvo operativo en 1990.
 - Su primera exposición pública fue en 1991.
 - El primer interfaz gráfico fue Mosaic, en 1993.
- La dificultad principal de la arquitectura reside en el servidor.
- El cliente es, desde un punto de vista de administración, muy sencillo.

El protocolo HTTP (I)

- El protocolo HTTP consiste en:
 - Un solicitud ASCII.
 - Una respuesta MIME RFC 822.
- Los comandos son sensibles al contexto.

<u>Método</u>	<u>Descripción</u>
OPTIONS	Solicita información sobre las opciones de comunicación.
GET	Solicita recibir una página Web.
HEAD	Solicita leer la cabecera de una página Web.
POST	Añade información a un recurso nombrado.
PUT	Solicita almacenar una página Web.
DELETE	Elimina una página Web.
TRACE	Invoca la devolución del mensaje de solicitud.

El protocolo HTTP (II)

- La respuesta MIME RFC 822 consiste en:
 - Una línea de estado formada por:
 - La versión del protocolo.
 - El código de respuesta.
 - Texto aclaratorio del código.
 - Líneas con una página Web, etc.

<u>Código</u>	<u>Descripción</u>
1xx	Informativo. No utilizado, reservado para usos futuros.
2xx	Éxito. La acción fue recibida y aceptada.
3xx	Redirección. Se necesita una acción adicional para llevar a cabo la solicitud.
4xx	Error del cliente. La solicitud contiene sintaxis errónea o no se puede conceder.
5xx	Error del servidor. El servidor no puede atender una solicitud aparentemente correcta.

El método GET (I)

- Es utilizado para solicitar un página Web al servidor.
- Existen dos tipos de solicitudes:
 - Sencillas:
 - Una única línea con GET <nombre página>
 - Completas:
 - Varias líneas:
 - Una primera con GET <nombre página> <versión protocolo>
 - Una segunda con Host: <nombre del servidor>
 - Líneas (opcionales) con información sobre el cliente, los tipos MIME soportados, etc.

El método GET (II)

- Una solicitud sencilla recibe como respuesta:
 - Una página Web sin línea de estado, etc.

- Ejemplo:

```
GET /home.html
```

- Respuesta:

```
<!doctype html public "-//w3c//dtd html 4.0  
transitional//en">
```

```
<html>
```

```
<head>
```

```
...
```

```
</body>
```

```
</html>
```

El método GET (III)

- Una solicitud completa recibe como respuesta:
 - Una línea de estado.
 - Información sobre la página que va a recibirse.
 - La propia página.

- Ejemplo:

```
GET /home.html HTTP/1.1
```

```
Host: robotica.uv.es
```

```
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US;  
rv:1.0.1) Gecko/20021003
```

```
...
```

El método GET (IV)

- Respuesta:

HTTP/1.1 200 OK

Date: Sun, 17 Dec 2006 17:47:37 GMT

Server: Apache/2.0.52 (Red Hat)

Last-Modified: Wed, 03 Nov 2004 16:42:35 GMT

ETag: "97864-2c0-3e7fd542b54c0"

Accept-Ranges: bytes

Content-Length: 704

Connection: close

Content-Type: text/html; charset=iso-8859-1

...

El cliente HTTP

- Un cliente Web es un navegador.
- El navegador localiza las páginas Web mediante una URL, con formato:

`http://<nombre del ordenador>[:puerto]
[/<página web solicitada>]`

- Los dos navegadores más usados son:
 - Internet Explorer.
 - Mozilla Firefox.

El servidor HTTP (I)

- El servidor Web es el programa que atiende las peticiones de los clientes y sirve las páginas Web solicitadas.
- Utiliza, de forma general, el puerto 80 TCP.
- El servidor más utilizado es Apache.
 - Tiene versiones para:
 - Linux.
 - Windows.
 - Solaris.
 - IRIX.

El servidor HTTP (II)

- El servidor Apache es el programa `/usr/sbin/httpd`.
- Sus opciones más frecuentes son:
 - `-d <directorio>`: Indica el directorio raíz donde se encuentran las páginas Web del servidor.
 - `-f <fichero>`: Indica el fichero de configuración.
 - `-t`: Comprueba si el fichero de configuración es correcto o no.
 - `-D DUMP_VHOST`: Muestra información de los servidores virtuales.
 - `-D DUMP_MODULES`: Muestra información de los módulos.

El servidor HTTP (III)

- Los ficheros de configuración por defecto se encuentran dentro del directorio `/etc/httpd`.
 - Subdirectorio `conf`: Configuración del servidor.
 - Subdirectorio `conf.d`: Configuración de las extensiones.
 - Subdirectorio `conf.modules.d`: Configuración de los módulos dinámicos del servidor, incluidos los módulos de las extensiones.

El servidor HTTP (IV)

- Las extensiones permiten al servidor:
 - Utilizar SSL (fichero ssl.conf).
 - Consultar bases de datos y crear páginas Web (fichero php.conf).
 - Etc.

El servidor HTTP (V)

- Fichero `conf.modules.d/10-php.conf`:

```
<IfModule prefork.c>
    LoadModule php5_module modules/libphp5.so
</IfModule>
<IfModule !prefork.c>
    LoadModule php5_module modules/libphp5-zts.so
</IfModule>
```

- Fichero `php.conf`:

```
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
AddType text/html .php
DirectoryIndex index.php
php_value session.save_handler "files"
php_value session.save_path "/var/lib/php/session"
```

El servidor HTTP (VI)

- El modo de funcionamiento del servidor se decide en la compilación del mismo.
 - prefork: El proceso padre ejecuta procesos hijo para atender las peticiones de los clientes.
 - worker: El proceso padre ejecuta procesos hijo que lanzan hilos para atender las peticiones de los clientes.
- El modo de compilación puede obtenerse con:
`httpd -V|grep "Server MPM:"`
 - En UNIX/Linux generalmente es prefork.

El servidor HTTP (VII)

- El fichero de configuración del servidor Web es `httpd.conf (/etc/httpd/conf/httpd.conf)`.
- Tiene tres secciones:
 - Configuración global.
 - Configuración general del servidor.
 - Configuración de los servidores virtuales.
- Toda línea que comienza por `#` es un comentario.

Configuración global (I)

ServerRoot "/etc/httpd"

- Directorio raíz del servidor donde se encuentran los ficheros de:
 - Configuración.
 - Error.
 - Log.
- No puede añadirse una / al final.

Configuración global (II)

`Listen 147.156.222.65:80`

- Indica la dirección IP y puerto en que se escuchan las conexiones.
 - Si no se pone dirección IP (o se pone *) se suponen todas las direcciones IP del servidor.
- Pueden existir varias directivas Listen para indicar distintos interfaces y/o puertos.

Configuración global (III)

```
Include conf.modules.d/*.conf
```

- Indica que se incluyan los módulos dinámicos que permiten el funcionamiento de las extensiones del servidor.

```
User apache
```

```
Group apache
```

- Indica como que usuario y grupo se ejecutará el servidor una vez arrancado.

Configuración general del servidor (I)

- Define el comportamiento de:
 - El servidor por defecto.
 - Los servidores virtuales, si se utilizan, para aquellos valores no definidos explícitamente en los mismos.

ServerAdmin webmaster@robotica.uv.es

- Dirección de correo del administrador.

ServerName irtic.uv.es

- ServerName: Nombre (y puerto) con el que debe responderse a las peticiones.

- Si no se indica se utiliza DNS para averiguarlo.

Configuración general del servidor (II)

```
<Directory />  
    AllowOverride None  
    Require all granted  
</Directory>
```

- Especifica opciones de control de acceso a las páginas web.

```
/var/www/html
```

- Directorio raíz a partir del cual se encuentran las páginas web.

Configuración general del servidor (III)

```
<Directory "/var/www">  
    AllowOverride None  
    Require all granted  
</Directory>
```

```
<Directory "/var/www/html">  
    Options Indexes FollowSymLinks  
    AllowOverride None  
    Require all granted  
</Directory>
```

- Especifican opciones de control de acceso a las páginas web.

Configuración general del servidor (IV)

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

- Nombre y orden por defecto de las páginas a buscar si no se solicita ninguna.

```
<Files ".ht*">
    Require all denied
</FilesMatch>
```

- Regla para evitar el acceso a determinadas páginas web de control de acceso a directorios del servidor.

Configuración general del servidor (V)

`ErrorLog "logs/error_log"`

- Indica donde se escribirán los mensajes de error.

`LogLevel warn`

- Indica el nivel de detalle de los mensajes de error que se guardan.

Configuración general del servidor (VI)

```
<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common
    <IfModule logio_module>
        LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>
    CustomLog logs/access_log common
</IfModule>
```

- Definición de formatos de log y fichero y tipo de formato de log a guardar.

Configuración general del servidor (VII)

```
<IfModule alias_module>
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

- La entrada ScriptAlias indica un camino absoluto para un camino relativo al raíz del servidor: /var/www/html/cgi-bin/ -> /var/www/cgi-bin/.
- La entrada siguiente indica condiciones de acceso a ese directorio.

Configuración general del servidor (VIII)

```
<IfModule mime_module>
  TypesConfig /etc/mime.types
  AddType application/x-compress .Z
  AddType application/x-gzip .gz .tgz
  AddType text/html .shtml
  AddOutputFilter INCLUDES .shtml
</IfModule>
```

- Define los tipos de ficheros que conoce el servidor web y añade la definición de algunos tipos de ficheros en función de su extensión.

Configuración general del servidor (IX)

AddDefaultCharset UTF-8

- Conjunto de caracteres por defecto.

```
<IfModule mime_magic_module>  
    MIMEMagicFile conf/magic  
</IfModule>
```

- Fichero que contiene la especificación de tipos MIME en función de los códigos MIME.

EnableSendFile on

- Indica al servidor que si el fichero es estático indique al kernel que lo envíe por la conexión sin necesidad de leerlo el servidor.

Servidores virtuales (I)

- Un servidor virtual es una entrada que permite atender peticiones a URLs que no corresponden con el nombre del ordenador.
- Permiten alojar en su solo ordenador múltiples dominios de páginas web.
- En nuestro caso el servidor virtual posee dos dominios:
 - irtic.uv.es
 - www.cdlibre.org

Servidores virtuales (II)

```
<VirtualHost 147.156.222.65:80>
  ServerName irtic.uv.es
  DocumentRoot /var/www/html
  ServerAdmin webmaster@irtic.uv.es
  ServerSignature email
  DirectoryIndex home.html index.html
</VirtualHost>
<VirtualHost 147.156.222.65:80>
  ServerName www.cdlibre.org
  ServerAlias cdlibre.org *.cdlibre.org
  DocumentRoot /home/barto/public_html
  ServerAdmin barto@glup.uv.es
  ServerSignature email
  DirectoryIndex home.html index.html
  ErrorLog /home/barto/error.log
  TransferLog /home/barto/transfer.log
</VirtualHost>
```

Servidores virtuales (III)

```
<VirtualHost nombre[:puerto]>
```

```
...
```

```
</VirtualHost>
```

- Nombre o dirección IP y puerto donde escucha este servidor virtual.

```
ServerName www.cdlibre.org
```

```
ServerAlias cdlibre.org *.cdlibre.org
```

- Nombre del servidor que solicita el cliente y otros nombres (alias) por los que también es conocido.

Servidores virtuales (IV)

- Aspectos de los servidores virtuales:
 - Los valores no especificados toman los valores indicados en el servidor principal.
 - El servidor principal queda anulado, por lo que se debe incluir como una entrada como servidor virtual para que de servicio.
 - Si una URL hace referencia a un servidor virtual que no existe, responde el primer servidor virtual definido:
 - Colocar en primer lugar un servidor virtual que tan solo muestre una página diciendo que ese servidor no existe.

Control de acceso por IP (I)

- Se realiza mediante las entradas:

```
<Directory "directorio">
```

```
...
```

```
</Directory>
```

```
<Location "URL">
```

```
...
```

```
</Location>
```

```
<Files "fichero">
```

```
...
```

```
</Files>
```

Control de acceso por IP (II)

- Directory: Se aplican al directorio y subdirectorios especificados por “directorio”, excepto que exista una entrada para algún subdirectorio.

```
<Directory "/">
```

```
...
```

```
</Directory>
```

```
<Directory "/var/www/html">
```

```
...
```

```
</Directory>
```

Control de acceso por IP (III)

- Location: Se aplican a la URL especificada. Debe tenerse en cuenta el valor de DocumentRoot.

```
<Location "/">
```

```
...
```

```
</Location>
```

```
<Location "/monitor">
```

```
...
```

```
</Location>
```

- Si varias URLs especifican el mismo elemento debe limitarse para todas ellas.

Control de acceso por IP (IV)

- Files: Se aplican a los ficheros cuyo nombre se especifica, independiente de su directorio, URL, etc.

```
<Files "privado.html">
```

```
...
```

```
</Files>
```

- Todas las entradas anteriores tienen sus equivalentes con expresiones regulares:
 - DirectoryMatch.
 - LocationMatch.
 - FileMatch.

Control de acceso por IP (V)

- El orden de evaluación de las entrada es:
 1. Directory.
 2. DirectoryMatch.
 3. Files y FilesMatch.
 4. Location y LocationMatch.

Control de acceso por IP (VI)

- Ejemplo:

```
<Location "/">
```

```
D
```

```
</Location>
```

```
<Files "fichero.html">
```

```
C
```

```
</Files>
```

```
<DirectoryMatch "^.*b$">
```

```
B
```

```
</DirectoryMatch>
```

```
<Directory "/">
```

```
A
```

```
</Directory>
```

- Se evaluarán en el orden A, B, C y D.

Control de acceso por IP (VII)

- Las directivas que controlan el acceso dentro de esas entradas son:

`Require all granted`

`Require all denied`

`Require [not] host {nombre|dominio}`

`Require [not] ip {ip|subred}`

- `Require all granted`: Acceso permitido a todos los ordenadores.
- `Require all denied`: Acceso denegado a todos los ordenadores.

Control de acceso por IP (VIII)

- Las directivas que controlan el acceso dentro de esas entradas son:

Require all granted

Require all denied

Require [not] host {nombre|dominio}

Require [not] ip {ip|subred}

- Require all granted: Acceso permitido a todos los ordenadores.
- Require all denied: Acceso denegado a todos los ordenadores.
- Require [not] host {nombre|dominio}: El requisito se cumple si host es el nombre o dominio (o no).
- Require [not] ip {ip|subred}: El requisito se cumple si la ip es ip o subred (o no).

Control de acceso por IP (IX)

- La directiva Require se puede combinar con:

`<RequireAll>...</RequireAll>`

`<RequireAny>...</RequireAny>`

`<RequireNone>...</RequireNone>`

- `<RequireAll>...</RequireAll>`: Se cumple si ninguna de las directivas falla y al menos una es cierta.
- `<RequireAny>...</RequireAny>`: Se cumple si al menos una de las directivas es cierta.
- `<RequireNone>...</RequireNone>`: Se cumple si todas las directivas no son ciertas.

Control de acceso por IP (X)

- Ejemplos:

```
<RequireAll>
```

```
    Require all granted
```

```
    Require not ip 147.156.0.0/16
```

```
</RequireAll>
```

```
<RequireAny>
```

```
    Require all denied
```

```
    Require ip 147.156.222.0/23
```

```
    Require host www.uv.es
```

```
</RequireAny>
```

```
<RequireNone>
```

```
    Require all granted
```

```
    Require not ip 147.156.0.0/16
```

```
</RequireNone>
```

Control de acceso por IP (XI)

- Importancia del orden de evaluación:

```
<Location "/">
```

```
    Require all granted
```

```
</Location>
```

```
<Directory "/var/www/html">
```

```
    Require all denied
```

```
</Directory>
```

- La entrada Directory limita el acceso, pero la entrada Location elimina esa limitación.

Control de acceso por IP (XII)

- Las funcionalidades adicionales se indican mediante:
Options <funcionalidad> [... funcionalidad]

<u>Valor</u>	<u>Descripción</u>
None	Ninguna funcionalidad adicional está activa
All	Todas las funcionalidades adicionales menos <i>MultiViews</i> .
ExecCGI	Permite utilizar scripts CGI.
FollowSymLinks	Permite seguir los enlaces simbólicos.
SymLinksIfOwnerMatch	Permite seguir los enlaces simbólicos solo si el fichero o directorio final tiene el mismo dueño que el enlace.
Includes	Permite incluir SSI.
IncludesNoExec	Permite incluir SSI pero excluyendo aquellos que ejecutan comandos o CGIs.
Indexes	Muestra un listado con el contenido del directorio si no existen los archivos especificados en <i>DirectoryIndex</i> .
MultiViews	Permite la negociación del contenido, por ejemplo, el uso de distintos lenguajes en la respuesta.

Control de acceso por IP (XIII)

```
<Directory "/var/www/html">  
  Options Indexes Multiview  
  <RequireAll>  
    Require all granted  
    Require ip 147.156.0.0/16  
  </RequireAll>  
</Directory>
```

Control de acceso por usuario (I)

- Se realiza con la directiva `AllowOverride`:
 - Debe especificarse siempre dentro de una entrada `Directory`.
 - Indica que directivas están permitidas en los ficheros de control de acceso por usuario.

Control de acceso por usuario (II)

<u>Valor</u>	<u>Descripción</u>
All	Permitir todas las directivas.
None	No permitir ninguna directiva.
AuthConfig	Permitir directivas de autenticación de usuarios.
FileInfo	Permitir directivas de control del tipo de documentos.
Indexes	Permitir directivas de indexado de directorios.
Limit	Permitir directivas que controlan el acceso por dirección IP del cliente.
Options	Permitir directivas que controlan funcionalidades de los directorios.

- Ejemplo:

AllowOverride Indexes

- Que permite directivas de indexado de directorios.

Control de acceso por usuario (II)

<u>Valor</u>	<u>Descripción</u>
All	Permitir todas las directivas.
None	No permitir ninguna directiva.
AuthConfig	Permitir directivas de autenticación de usuarios.
FileInfo	Permitir directivas de control del tipo de documentos.
Indexes	Permitir directivas de indexado de directorios.
Limit	Permitir directivas que controlan el acceso por dirección IP del cliente.
Options	Permitir directivas que controlan funcionalidades de los directorios.

- Ejemplo:

AllowOverride Indexes

- Que permite directivas de indexado de directorios.

Control de acceso por usuario (III)

- Para el control de acceso por usuario los valores que debe tomar AllowOverride son:
 - All: Todas las directivas.
 - AuthConfig: Control de acceso por usuario.

Control de acceso por usuario (IV)

- El fichero de control de acceso contiene las entradas:
 - AuthType: Tipo de autenticación {Basic|Digest}.
 - AuthName: Nombre a utilizar en la autenticación.
 - AuthUserFile: Fichero con los usuarios/contraseñas.
 - AuthGroupFile: Fichero con los grupos y los usuarios que los forman.

<nombre del grupo>: <usuario1> <usuario2> ... <usuarioN>

- Require {user|group|valid-user} <lista>:

```
Require user usuario1 [...usuarioN]
```

```
Require group grupo1 [...grupoN]
```

```
Require valid-user
```

Control de acceso por usuario (V)

- Ejemplos:

```
AuthType Basic
```

```
AuthName "Acceso Restringido"
```

```
AuthUserFile /var/www/users
```

```
Require user usuario1 usuario2
```

```
AuthType Basic
```

```
AuthName "Acceso Restringido"
```

```
AuthUserFile /var/www/users
```

```
AuthGroupFile /var/www/groups
```

```
Require group grupo1 grupo2
```

Control de acceso por usuario (VI)

- El fichero con la relación usuario/contraseña se crea/maneja con htpasswd:

```
htpasswd -c [-p | -d | -m | -s] <fichero> <usuario>
```

```
htpasswd [-p | -d | -m | -s] <fichero> <usuario>
```

```
htpasswd -D <fichero> <usuario>
```

- Opciones:
 - -c: Crear fichero.
 - -D: Borrar usuario.
 - -p: Cifrado en texto plano (sin cifrar).
 - -d: Cifrado con crypt (por defecto).
 - -m: Cifrado MD5.
 - -s: Cifrado SHA.

La extensión SSL (I)

- Es la que permite el acceso al servidor mediante una conexión segura.
- Se configura en el fichero `conf.d/ssl.conf`.

`Listen 147.156.222.65:443`

- IP y puerto en que se escucha.
- Modo de solicitud de la contraseña si la clave privada está protegida.

`SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog`

- `builtin`: Solicitarla al usuario.
- `exec:<programa>`: Ejecutar el programa indicado.

La extensión SSL (II)

SSLSessionCache shmcb:/var/cache/mod_ssl/scache(512000)

SSLSessionCacheTimeout 300

- Tipo de cache y timeout.

SSLMutex default

- Tipo de semáforo a utilizar.

SSLRandomSeed startup file:/dev/urandom 256

SSLRandomSeed connect builtin

- Generador de números aleatorios a utilizar.

SSLCryptoDevice builtin

- Permite utilizar tarjetas criptográficas del sistema.

La extensión SSL (III)

```
<VirtualHost 147.156.222.65:443>
  ErrorLog logs/ssl_error_log
  TransferLog logs/ssl_access_log
  LogLevel warn
  SSLEngine on
  SSLProtocol all -SSLv2
  SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
  SSLCertificateFile /etc/pki/tls/certs/server.crt
  SSLCertificateKeyFile /etc/pki/tls/private/server.key
</VirtualHost>
```

- Servidor virtual que atiende las solicitudes SSL.
 - SSLEngine: Habilita (on) o deshabilita (off, valor por defecto) el uso de SSL.
 - SSLProtocol: Protocolos validos.
 - SSLCipherSuite: Tipos de cifrado OpenSSL soportados en la negociación.
 - SSLCertificateFile: Fichero con la clave pública.
 - SSLCertificateKeyFile: Fichero con la clave privada.

La extensión SSL (IV)

- Hasta la versión 2.2.12 de Apache solo era posible definir un servidor virtual SSL en la misma dirección IP y puerto.
 - SSL negocia la comunicación antes de recibir/enviar datos, luego no conoce el servidor virtual que solicitará el cliente y negocia siempre con el primero.
- A partir de la versión 2.2.12 se añadió la extensión Server Name Indication.
 - Permite enviar el nombre del servidor que se solicitará antes de que SSL establezca la negociación, luego puede elegirse el certificado adecuado.
- El uso de SNI se activa con:
`SSLStrictSNIVHostCheck On`

La extensión SSL (V)

```
SSLStrictSNIVHostCheck On
<VirtualHost 147.156.222.65:443>
NameVirtualHost irtic.uv.es:443
DocumentRoot /irtic
    ...
    SSLCertificateFile /etc/pki/tls/certs/irtic.crt
    SSLCertificateKeyFile /etc/pki/tls/private/irtic.key
    ...
</VirtualHost>
<VirtualHost 147.156.222.65:443>
NameVirtualHost www.cdlibre.org:443
DocumentRoot /cdlibre
    ...
    SSLCertificateFile /etc/pki/tls/certs/cdlibre.crt
    SSLCertificateKeyFile /etc/pki/tls/private/cdlibre.key
    ...
</VirtualHost>
```

La extensión SSL (VI)

- Internet Explorer 7 o superior en Windows Vista o superior.
- Internet Explorer 8 o superior en Windows XP.
- Firefox 2.0 o superior.
- Opera 8.0 o superior (requiere que la versión 1.1 del protocolo TLS este habilitada).
- Chrome 5.0.342.1 o superior en Mac OS X 10.5.7 o superior.
- Chrome 6 o superior en Windows XP o superior.
- Safari 2.1 o superior en Mac OS X 10.5.6 o superior o Windows Vista o superior.
- Konqueror/KDE 4.7 o superior.

Reenvío de páginas web (I)

- En ciertas ocasiones deseamos reenviar una petición:
 - Cambio en la URL de las páginas.
 - Conexión no segura a segura.
- Se carga el módulo con el fichero `conf.modules.d/00-ssl.conf`.
- Los reenvíos requieren las directivas:
 - `RewriteEngine {On|Off}`
 - `RewriteCond <cadena de texto> <condición>`
 - `RewriteRule <patrón> <sustitución> [[banderas]]`

Reenvío de páginas web (II)

- RewriteEngine:
 - Habilita (on) o deshabilita (off, valor por defecto), el reenvío de páginas web.
- RewriteCond:
 - Indica la condición que debe cumplir la cadena de texto para que se aplique la regla de reenvío.
- RewriteRule:
 - Indica la sustitución que debe hacerse si se cumple el patrón indicado. Banderas indican acciones adicionales.

Reenvío de páginas web (III)

- En RewriteCond la condición puede ser una variable:

<u>Cabeceras HTTP</u>			
HTTP_USER_AGENT	HTTP_REFERER	HTTP_COOKIE	HTTP_HOST
HTTP_ACCEPT	HTTP_FORWARDED	HTTP_PROXY_CONNECTION	
<u>Variables internas</u>			
DOCUMENT_ROOT	SERVER_ADMIN	SERVER_NAME	SERVER_ADDR
SERVER_PORT	SERVER_PROTOCOL	SERVER_SOFTWARE	
<u>Conexiones y requerimientos</u>			
REMOTE_ADDR	REMOTE_HOST	REMOTE_PORT	REMOTE_USER
REMOTE_IDENT	REQUEST_METHOD	SCRIPT_FILENAME	PATH_INFO
QUERY_STRING	AUTH_TYPE		
<u>Fecha y hora del sistema</u>			
TIME_YEAR	TIME_MON	TIME_DAY	TIME_HOUR
TIME_MIN	TIME_SEC	TIME_WDAY	TIME
<u>Otras variables</u>			
API_VERSION	THE_REQUEST	REQUEST_URI	IS_SUBREQ
HTTPS	REQUEST_FILENAME		

Reenvío de páginas web (IV)

<u>Texto</u>	
.	Un carácter simple.
[caracteres]	Cualquier carácter indicado en “caracteres”.
[^caracteres]	Cualquier carácter no indicado en “caracteres”.
texto1 texto2	Textos alternativos
<u>Cuantificadores</u>	
?	0 o 1 ocurrencias del texto anterior.
*	0 o N ocurrencias del texto anterior (N>0).
+	1 o N ocurrencias del texto anterior (N>1).
<u>Agrupación</u>	
(texto)	Agrupación de textos en uno solo.
<u>Delimitadores</u>	
^	Comienzo de delimitador de línea.
\$	Final de delimitador de línea.
<u>Escape</u>	
\carácter	Escape del significado especial del carácter.

Reenvío de páginas web (V)

- Ejemplos:

```
RewriteEngine On
```

```
RewriteCond %{HTTP_HOST} ^dominio_viejo.com$
```

```
RewriteRule ^(.*)$ http://dominio_nuevo.com$1 [L,R=301]
```

```
RewriteCond %{HTTP_HOST} ^www.dominio_viejo.com$
```

```
RewriteRule ^(.*)$ http://dominio_nuevo.com$1 [L,R=301]
```

```
RewriteEngine On
```

```
RewriteCond %{SERVER_PORT} !^443$
```

```
RewriteRule ^(.*)$ https://%{SERVER_NAME}$1 [L,R=303]
```

Reenvío de páginas web (VI)

```
<Directory "/var/www/html/directorio">  
    RewriteEngine On  
    RewriteCond %{SERVER_PORT} !^443$  
    RewriteRule ^(.*)$ https://%{SERVER_NAME}/directorio/$1  
    [L,R=303]  
</Directory>
```

```
RewriteEngine On  
RewriteCond %{REQUEST_METHOD} GET  
RewriteRule ^(.*\.\.iso)$ ftp://ftp.servidor.com/$1
```