Administración de Sistemas de Información

4. *Hyper-Text Transfer Protocol (HTTP)*

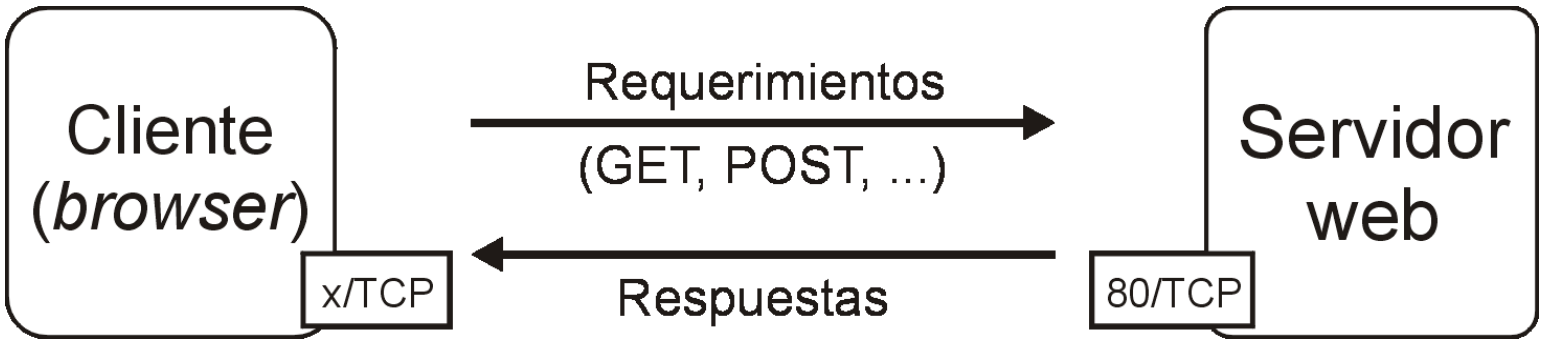
Curso 2023 / 24

Prof: Teodoro Rojo

UNIVERSIDAD SAN PABLO – CEU
ESCUELA POLITÉCNICA SUPERIOR
Departamento de Tecnologías de la Información

HTTP (Hyper-Text Transfer Protocol)

- Protocolo cliente-servidor para el intercambio de documentos hipertexto. RFC 2616 (ver 1.1, 1999 – ver 2.0, 2012, push, mpx).
- El cliente (navegador o *browser*), establece una conexión TCP con el servidor (puerto 80 por defecto) y envía por ella una petición mediante un comando HTTP con parámetros. El servidor procesa la petición y devuelve una respuesta (cabecera con código de respuesta y datos adicionales). La conexión TCP se cierra o se mantiene abierta para nuevas peticiones (*keep-alive*)



HTTP. Ejemplo de petición

GET /index.php HTTP/1.1

Host: www.ceu.es

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Referer: http://www.ceu.es/ceu/mision-historia.php

Accept-Encoding: gzip, deflate, sdch

Accept-Language: es,en-US;q=0.8,en;q=0.6

Cookie: _gat=1; _ga=GA1.2.1092392610.1480594414; cb-enabled=accepted

HTTP. Ejemplo de respuesta

HTTP/1.1 200 OK

Content-Type: text/html; charset=iso-8859-15

Connection: keep-alive

ETag: "56f00305-6f"

Server: Microsoft-IIS/7.5

X-Powered-By: PHP/5.4.0

X-Powered-By: ASP.NET

Last-Modified: Mon, 21 Mar 2016 14:19:49 GMT

Date: Thu, 02 Mar 2017 09:28:09 GMT

Content-Length: 34975

Otros códigos de respuesta: 3XX (redirección), 4XX (error del cliente), 5XX (error del servidor)

Servidor web Apache

- Proyecto mantenido por la ASF, *The Apache Software Foundation* (www.apache.org). Primera versión en 1995. Actualmente versión **2.4.56** (marzo de 2023)
- Robusto, seguro y de alto rendimiento.
- Característica más destacada: arquitectura modular.
- Módulos DSO (*Dynamic Shared Objects*). Son cargados en tiempo de ejecución. Están disponibles infinidad de módulos estándar: intérpretes de scripts (perl, Python, lua, …), páginas activas (php, asp, …), generadores de índices, servidores virtuales, proxy, etc., siendo posible desarrollar e integrar en Apache cualquier otra funcionalidad.

Configuración de Apache

- Se establece en el archivo **httpd.conf** (en /etc/httpd/conf/) o **apache2.conf** (/etc/apache2). Los elementos de configuración se establecen mediante directivas, que se agrupan en tres secciones:
  - **Entorno global**. Describe el funcionamiento general del servidor
  - **Servidor Principal**. Configura el servidor Principal, que atiende todas las peticiones que no van dirigidas a ningún servidor virtual
  - **Servidores virtuales**. Describe los distintos servidores virtuales que se mantienen en el sistema y sus parámetros de configuración

Normalmente la configuración se extiende a otros ficheros **.conf**, enlazados con el principal mediante directivas **include** (directorios *-enabled de Debian, conf.d y conf.modules.d de RedHat)

Directivas básicas (I)

- ## 1. Entorno global

ServerRoot (directorio raíz de la aplicación), PidFile, StartServers, MaxClients, MaxConnectionsPerChild, LoadModule (módulos DSO), Listen (direcciones y puertos de escucha), User y Group (usuario y grupo que ejecutan el demonio, sin apenas privilegios), …

```
ServerRoot      "/usr/local/apache"
StartServers    5
MaxClients      150
LoadModule      mimodulo    libexec/mimodulo.so

Listen 80
Listen 192.170.2.1:8080
```

Nota: Muchas directivas de Apache se incluyen con clausulas condicionales de módulo DSO (<IfModule> … </IfModule>)

Directivas básicas (II)

- **2. Servidor principal**

  DocumentRoot (raíz de documentos del servidor), DirectoryIndex (página de inicio por defecto), Alias (nombres alternativos para archivos o directorios), Redirect (redirección de una petición a otra ubicación), Directory (establece propiedades de los directorios accesibles: indexes, Require, …), AccessLog, ErrorLog, …

```
DocumentRoot /var/www/html
Alias /iconos/ /var/www/icons/
Redirect /docs http://www.otroservidor.com/docs

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

Options [+|-]option [[+|-]option] ...

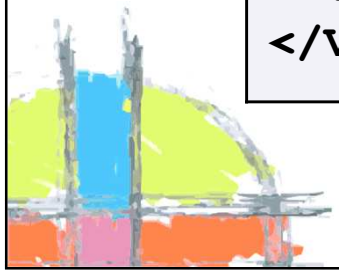
Directivas básicas (III)

- ## 3. Servidores virtuales

Los servidores virtuales (directiva VirtualHost) permiten atender las peticiones de usuario de forma diferenciada en un solo sistema. Pueden configurarse servidores virtuales por puerto de escucha, dirección IP y/o nombre del host.

```
Listen 8080
<VirtualHost *:8080>
  DocumentRoot /var/web1
  ...
</VirtualHost>
```

```
<VirtualHost 192.168.1.200:80>
  DocumentRoot /var/web2
  ...
</VirtualHost>
```



Directivas básicas (y IV)

- Ejemplo: Servidores basados en nombre

```
NameVirtualHost *:80

<VirtualHost *:80>
  ServerName www.miempresa.com
  ServerAlias miempresa.com
  DocumentRoot /var/miempresa.com
  ...
</VirtualHost>

<VirtualHost *:80>
  ServerName www.otraempresa.com
  ServerAlias otraempresa.com
  DocumentRoot /var/otraempresa.com
  ...
</VirtualHost>
```

Verificar sintaxis: apachectl configtest

Utilidades en Apache2 (Debian y variantes)

- Para activar o desactivar elementos de la **configuración**, **módulos** DSO y los **servidores virtuales**, contamos con los directorios **available** y **enabled** de cada una de estas secciones. Sólo formará parte de la configuración de Apache el contenido de los archivos .conf ubicados en los directorios **enabled**.
- Lo que se espera con este modelo es que en available tengamos los archivos .conf "reales", con las directivas de configuración, y los directorios enabled tengan **enlaces simbólicos** a los primeros.
  → Usamos los symlinks para activar/desactivar una función.
- Utilidades para crear o borrar los enlaces simbólicos:
  a2enconf / a2disconf <nombre de la configuración>, a2ensite / a2dissite <nombre del sitio>, a2enmod / a2dismod <modulo>

Autorización (I)

- **Require**: Establece condiciones para el acceso al servidor o a directorios concretos. Por direcciones IP, nombres de dominio, propiedades de la petición HTTP, usuarios, ...
- El valor por defecto se fija con all (granted o denied). Se pueden identificar direcciones o bloques de direcciones IP (con ip) o nombres de dominio (con host).
- También se pueden usar negaciones (con not)

```
Require all granted
Require not ip 10.80.24.100
Require not host gov
```

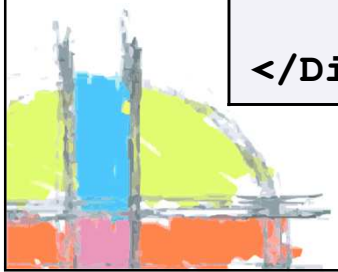
- Nota: Las directivas <limit> order, allow, deny </limit> están en deshuso (hasta v2.2):

```
Order allow,deny
  Allow from all
```

Autorización (y II)

- Puede estructurarse: RequireAll, RequireAny, RequireNone

```
<Directory "/www/mydocs">
    <RequireAll>
        <RequireAny>
            Require user superadmin
            <RequireAll>
                Require group admins
                Require ldap-group cn=Administrators,o=Airius
                <RequireAny>
                    Require group sales
                    Require ldap-attribute dept="sales"
                </RequireAny>
            </RequireAll>
        </RequireAny>
        <RequireNone>
            Require group temps
            Require ldap-group cn=Temporary Employees,o=Airius
        </RequireNone>
    </RequireAll>
</Directory>
```



Archivo .htaccess - Autentificación

- Es posible delegar elementos de la configuración a otros archivos ubicados en directorios visibles vía web.
- El nombre de estos archivos se fija con AccessFileName (por defecto, **.htaccess**). Para activar su procesamiento: AllowOverride (All / None / AuthConfig / Limit /…).
- Un uso habitual: restricción de acceso a directorios a usuarios o grupos mediante contraseñas. Un archivo auxiliar contendrá los usuarios y contraseñas válidos (directiva AuthUserFile)
- Creación del archivo contraseñas: utilidad htpasswd archivo usuario (-c para crear un nuevo archivo)

Ejemplo de archivo .htaccess

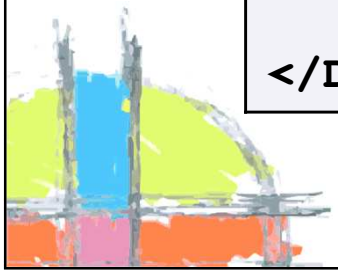
```
AuthUserFile /home/www/seguro/.htpasswd
AuthGroupFile /dev/null
AuthName "Directorio Restringido"
AuthType Basic
<Limit GET POST>
   require valid-user
</Limit>
```

- Para crear el fichero de contraseñas (para el usuario juan):

```
htpasswd -c /home/www/seguro/.htpasswd juan
```

- Para habilitar el uso de los ficheros .htacces

```
<Directory /home/www/seguro>
   AllowOverride AuthConfig
</Directory>
```

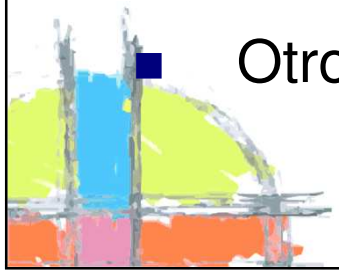


Servidor nginx

- Nginx es un servidor web y proxy inverso multiplataforma, ligero y de alto rendimiento. También incluye un **proxy** para protocolos HTTP/HTTPS y correo electrónico (SMTP/IMAP/POP3).
- Autor: Igor Sysoev. Lanzamiento en 2004. En 2011 se crea nginx Inc. Versión con soporte: nginx Plus. Actual versión estable: 1.23.3.
- Muy eficiente sirviendo contenidos estáticos. Opera orientado a eventos, con gestión asíncrona. Se puede configurar como balanceador de carga para distintos sitios y/o contenidos.
- Diseño modular. Además del *core*, módulos de aceleración de aplicaciones, proxy web inverso, proxy de correo, cifrado TLS, Gestión de ancho de banda, balanceo de carga y *streaming* de video.

Configuración nginx (I)

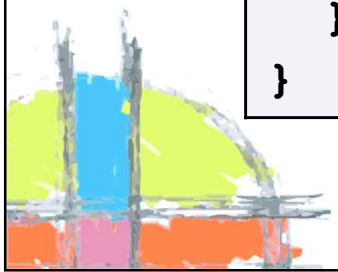
- El archivo de configuración (nginx.conf) se organiza en contextos.
- El contexto principal no está delimitado por llaves y contiene parámetros generales. Los restantes contextos se delimitan con llaves y pueden contener nuevos contextos.
- Contextos básicos:
  - events (modo en el que nginx maneja las conexiones)
  - http (configuración de los servicios web y proxy inverso)
  - server (declarado dentro de http, para cada servidor virtual mantenido. Pueden ser múltiples)
  - location (propiedades de las ubicaciones a las que se accede). Se ubican en contextos de servidor y pueden estar anidados.
- Otros contextos: upstream (proxies), mail, if, limit_except, …



Configuración nginx (II)

- La configuración del servidor virtual se suele almacenar en un archivo .conf, que se añade al contexto html mediante una directiva Include

```
server {
  listen       80;
  server_name  pruebas.com www.pruebas.com;
  location / {
    root   /var/www/pruebas.com/html;
    index  index.html index.htm;
    try_files $uri $uri/ =404;
  }
  error_page   500 502 503 504  /50x.html;
  location = /50x.html {
    root   /var/www/error;
  }
}
```



Configuración nginx (y III)

- nginx como balanceador de carga

```
http {
    upstream serv1 {
        least_conn;
        server srv1.pruebas.com;
        server srv2.pruebas.com;
        server srv3.pruebas.com;
    }
    server {
        listen 80;
        location / {
            proxy_pass http://serv1;
        }
    }
}
```

- Opciones de balanceo: round-robin (por defecto; es posible asignar pesos con weight), least_conn (al de menor número de conexiones), ip_hash (todas las peticiones de una conexión al mismo servidor),