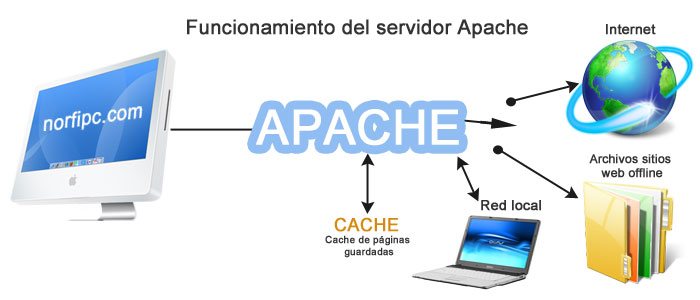
Tema :2.2 :Instalación básica del servidor web Apache en Windows



Un **servidor web c**omo su nombre lo indica, es un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización.  
Existen varios tipos de servidores web, Apache es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows.  
Al instalarlo en nuestra PC dispondremos de un servidor completo, con todos los requisitos para ejecutarlo de forma local.

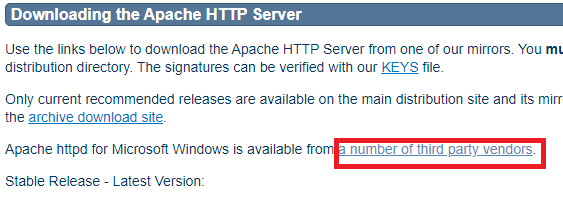
**¿Cómo instalar el Servidor Apache?**

En la práctica Apache se instala de varias formas:

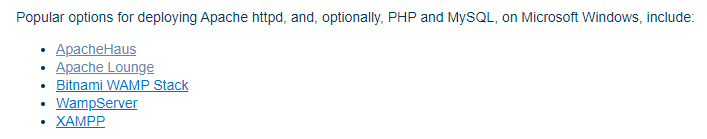
* El programa exclusivo, al que después se le agregan manualmente distintos módulos.
* El paquete XAMPP que incluye Apache, la base de datos MySQL, PHP y Perl.
* Otras distribuciones como LAMP o MAMP, wampserver

La instalación de la aplicación es sencilla

La instalación del [servidor web Apache (Apache HTTP Server)](http://httpd.apache.org/) debe comenzar por la descarga de la aplicación desde la [página de descargas](http://httpd.apache.org/download.cgi) (http://httpd.apache.org/download.cgi) de dicho proyecto. Puedes observar que hay varias versiones, todas ellas con el código fuente de la aplicación.



Pulsando sobre este enlace:

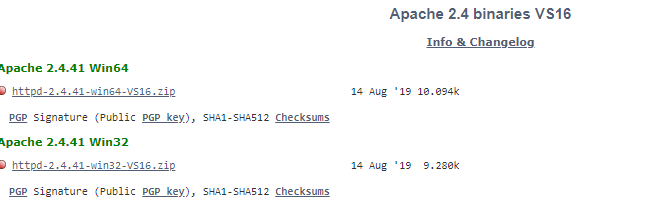


Podremos elegir entre una serie de opciones.

Esta descarga de Apache Lounge nos permite construir Apache siempre con las dependencias actualizadas y los compiladores más recientes que han sido probados de forma completa. Los binarios son referenciados por ASF, Microsoft, PHP, etc. y cada vez más software se empaqueta dentro de estos binarios y módulos.

Estos binarios son construidos con las fuentes de ASF en httpd.apache.org, y contienen los últimos parches y las últimas dependencias como zlib, openssl, etc. Los binarios no se ejecutan en Windows XP y 2003 y se pueden ejecutar en: 7 SP1, Vista SP2, 8 / 8.1, 10, Server 2008 SP2 / R2 SP1, Server 2012 / R2, Server 2016.

Allí podremos elegir la versión de 32 o 64 bits en base a la arquitectura que dispongamos. Una vez descarguemos el archivo .zip procedemos a su extracción:



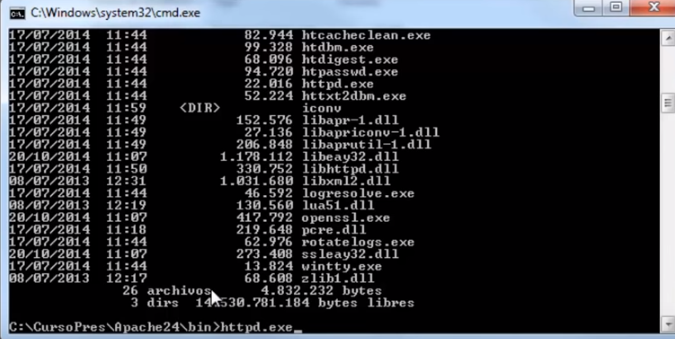
Lo descargamos y lo descomprimimos, copiamos la carpeta Apache24 al directorio raíz C, en:

El siguiente paso consiste en descargar e instalar **Microsoft Visual C ++ 2017, 2012,2015** Runtime el cual es necesario para ejecutar Apache en Windows Environment.

Este podrá ser descargado desde el siguiente enlace:

**https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads**

y dentro de la carpeta **bin**, existirá un archivo que se llama **httpd.exe**, el cual ejecutaremos para lanzar el servidor apache, esto mismo se puede hacer con la consola de comandos.

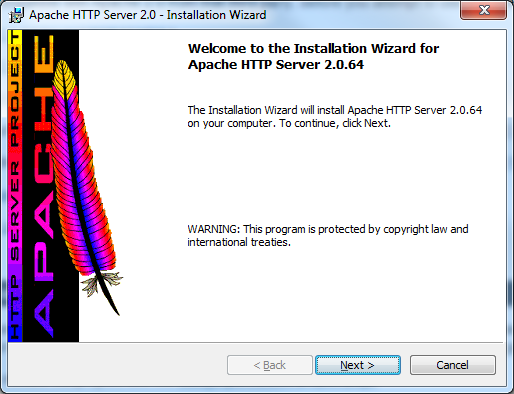


Pero para **hacer más sencilla la instalación desde Windows**, debemos buscar una versión que disponga de una compilación en binario para Windows.

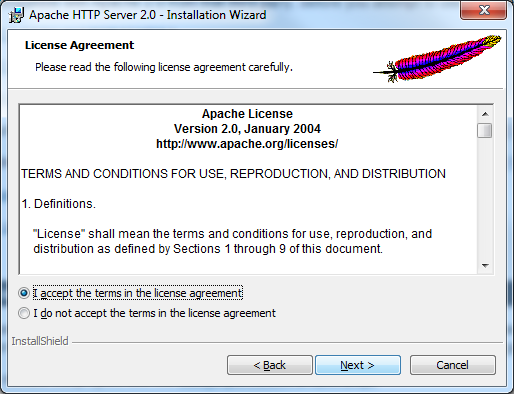
En primer lugar instalaremos Apache en su versión actual 2.4.28 VC15 desde el siguiente enlace:

<https://www.anindya.com/apache-http-server-2-4-2-x86-and-x64-windows-installers/>

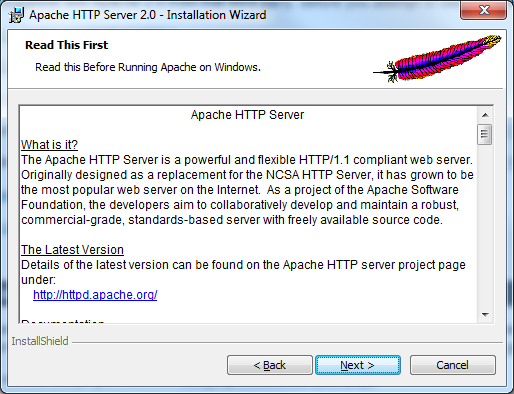
Al iniciar la instalación, ejecutando el archivo descargado, aparecerá la primera pantalla de bienvenida.



En principio deberás pulsar el botón Next (Siguiente) en cada una de las pantallas para hacer una instalación básica. En la siguiente pantalla te pide la aceptación de los términos de la licencia.



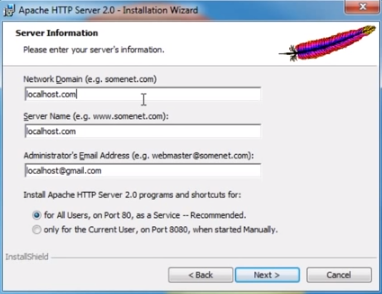
 Seguidamente te ofrece alguna información de interés sobre la instalación.



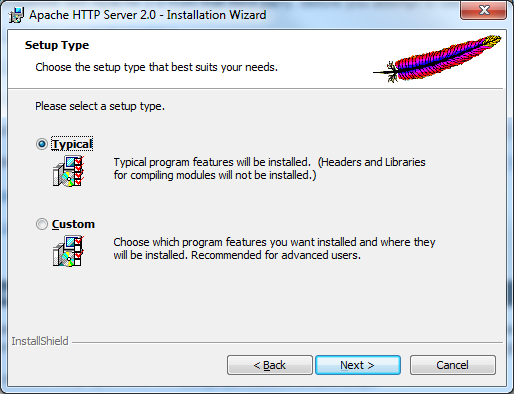
Y en la siguiente pantalla debes indicar alguna información sobre el servidor que vas a instalar. Si la instalación que estás realizando no es para un entorno productivo, sólo para algunas pruebas, puedes rellenar la siguiente información con cualquier dato. Por ejemplo, con los datos que te indica de ejemplo. Si va a ser un servidor con uso de un dominio propio deberías indicar los valores correspondientes al **dominio en el que se va a instalar el servidor**.

Además se pregunta si quieres instalar el servidor para todos los usuarios (for All Users) del sistema de Windows que hay en el PC, o si sólo va a ser para el usuario actual (only for the Current User). Ten en cuenta que **si sólo lo instalas para el usuario actual**, el servidor debe **iniciarse manualmente desde los iconos de la aplicación**. Si se instala **para todos** los usuarios, el servidor se inicia **automáticamente** como un servicio al iniciar Windows (es la opción recomendada).

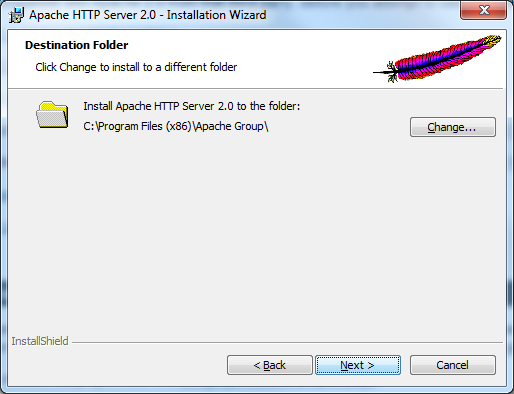
En función de la opción que elijas deberás recordar el número de puerto que indica, ya que para conectarte al servidor posteriormente desde un navegador web deberás indicar el puerto de conexión si no se trata del puerto 80. Observa que, en caso de que hagas la instalación sólo para el usuario actual, el puerto que utilizará el servidor es el **8080**.



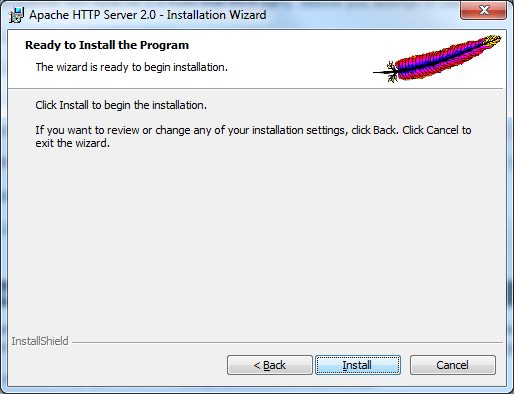
Ahora es el momento de determinar si quieres hacer una instalación típica o seleccionar manualmente las características que quieres usar del servidor. Se recomienda utilizar la **instalación típica**.



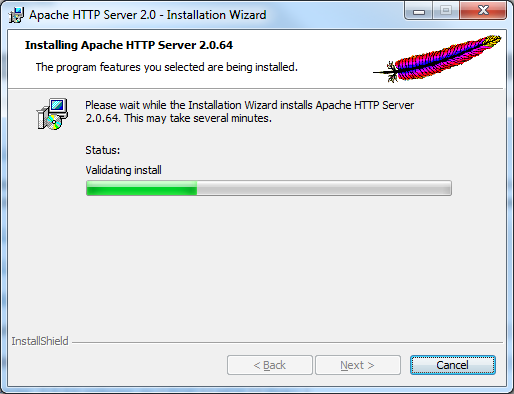
 Puedes elegir en la siguiente pantalla la **carpeta en la que se realizará la instalación**. Se recomienda dejar la que ofrece por defecto.



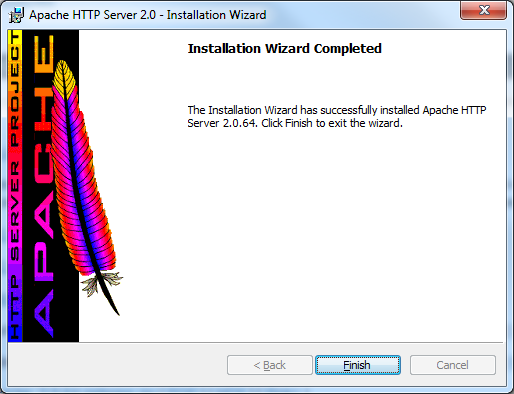
**Comienza** la instalación.



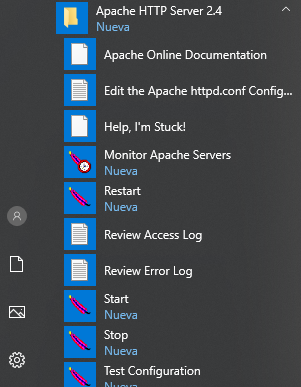
 Durante un tiempo, la barra de progreso irá indicando el **proceso de la instalación**.



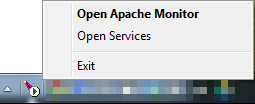
 Y una vez finalizado tan sólo nos queda hacer clic en el **botón Finish**.



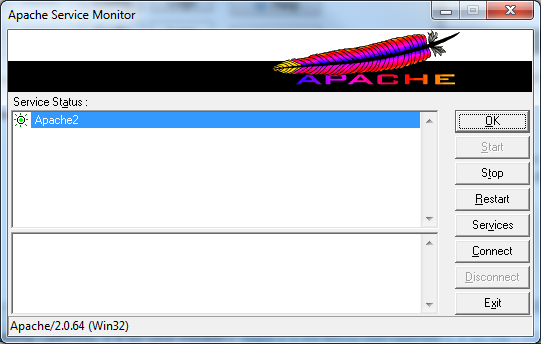
A partir de ese momento podrás ver en el **menú de Inicio** que se ha creado una **nueva carpeta Apache HTTP Server** con una serie de accesos a distintas utilidades del servidor. La principal de todas ellas podría decirse que es el monitor del servidor (**Monitor Apache Servers**), que puedes encontrar dentro de la subcarpeta Control Apache Server.



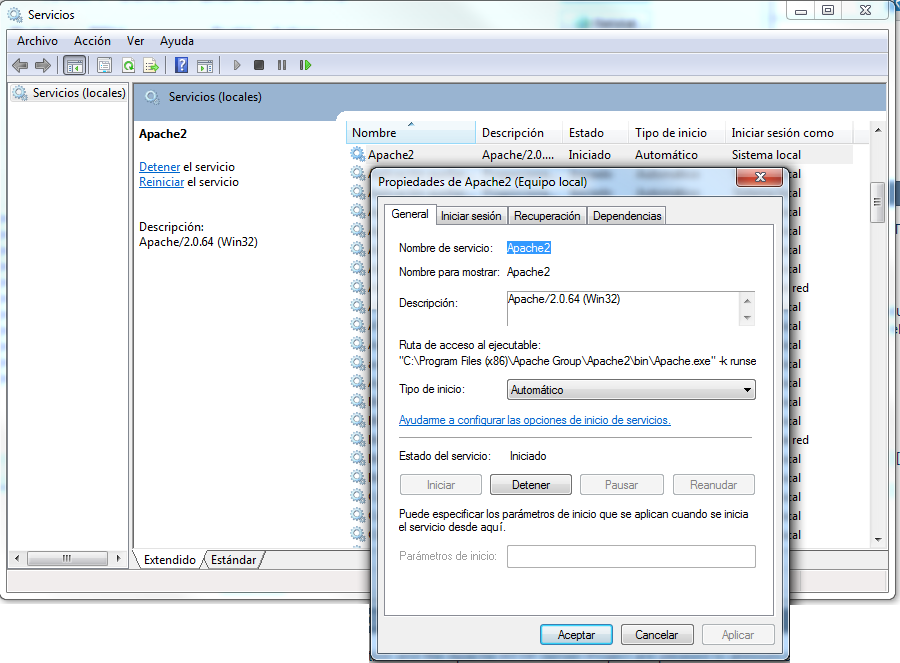
También puedes acceder al monitor desde el menú contextual que ofrece el icono del servidor Apache en la barra de tareas.



El monitor del servidor web Apache indica el estado del servidor. Si aparece en la parte izquierda con una **luz verde** se puede interpretar que está **funcionado**, y si aparece con luz roja nos está indicando que el servidor está **parado**. En la parte derecha dispones de una serie de botones para ocultar el monitor (OK), iniciar el servidor (**Start**), pararlo (**Stop**), reiniciarlo (**Restart**), acceder a los servicios de Windows (Services), conectar a un servidor que se encuentra en otra máquina para monitorizarlo (**Connect**), desconectarse del control de otro equipo remoto (**Disconnect**), y salir del monitor (**Exit**).



Como se ha comentado, desde el monitor del servidor web Apache puedes acceder directamente a los servicios de Windows. Recuerda que también puedes hacerlo desde el Panel de control, buscando por ejemplo la palabra **Servicios** o poniendo **services.msc**. En esta herramienta puedes ver una lista de los servicios registrados en tu equipo, donde aparecerá Apache2 si has realizado la instalación para todos los usuarios. Desde las propiedades (menú Acción > Propiedades) puedes ver la información detallada de este servicio. Desde ahí también puedes Iniciar, Detener, Pausar o Reanudar el servicio si lo necesitaras. Además, puedes desactivar la opción de que se arranque automáticamente al iniciar Windows, seleccionando en la lista desplegable Tipo de inicio la opción Manual.



Teniendo el servidor web en funcionamiento, debes **comprobar** que está todo correcto accediendo a la página web de prueba del servidor. Para ello, desde la misma máquina abre el **navegador web** e indica la dirección **http://localhost**. Al aceptar esa dirección debe aparecer la página que se muestra a continuación.



Si dispones de una red de ordenadores deberías comprobar que tienes acceso a la misma página web desde otros equipos. En ese caso debes indicar, en el navegador web de otro equipo de la red, la dirección IP del equipo en el que se encuentra instalado el servidor, en lugar de localhost.

En caso de que la instalación la hayas realizado sólo para el usuario actual de Windows, recuerda que el servidor está funcionando en el puerto 8080, por lo que la comprobación del funcionamiento la debes realizar indicado ese puerto después de la dirección del servidor, separado por el carácter dos puntos. Por ejemplo: [**http://localhost:8080**](http://localhost:8080/).

Tema 2.2.1- Configuración Apache Web Server

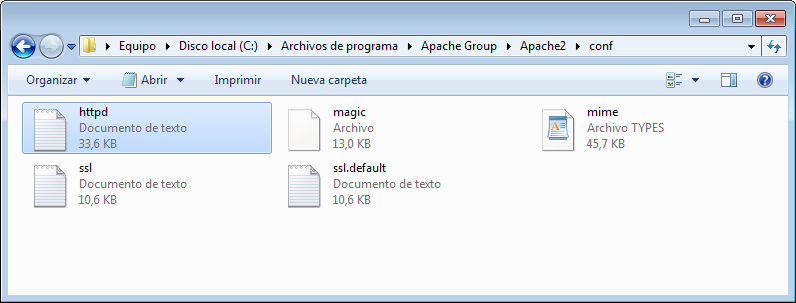
La configuración del servidor web de Apache debe realizarse **a través de archivos de texto**, en el que se incluyen una serie de directivas que establecen la configuración. Se dispone de un archivo principal y otros secundarios. Puedes consultar las directivas que pueden utilizarse en la [Guía Rápida de Referencia de Directivas](https://httpd.apache.org/docs/2.2/mod/quickreference.html).

**Localización de los archivos de configuración**

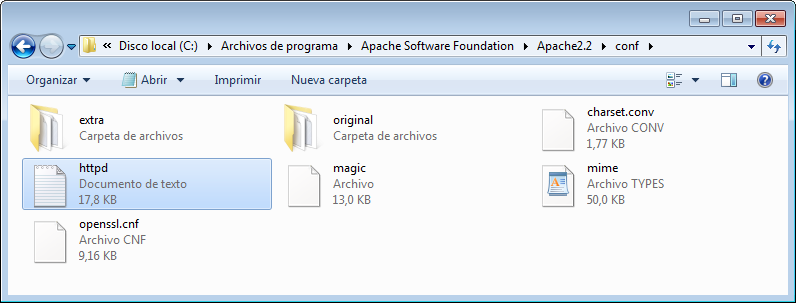
El **archivo principal** de configuración tiene el nombre ***httpd.conf*** en Windows y se encuentra en la ruta: **C:\Apache24\conf\httpd.conf**.

En **Ubuntu** se mantiene ese fichero, aunque podrás comprobar que está vacío, pero se utiliza en su lugar el archivo ***apache2.conf***.

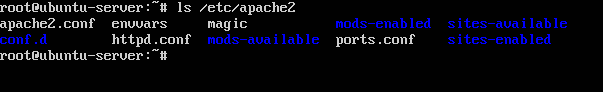
El archivo de configuración de Apache en **Windows** se encuentra en la **carpeta *conf***, dentro de la carpeta en la que se encuentre instalado. Recuerda que para la versión 2.0 la instalación se hace por defecto en la carpeta *C:\Archivos de programa\Apache Group\Apache2*.



En la versión 2.2 de Windows se hace en C:\Archivos de programa\Apache Software Foundation\Apache2.2 .

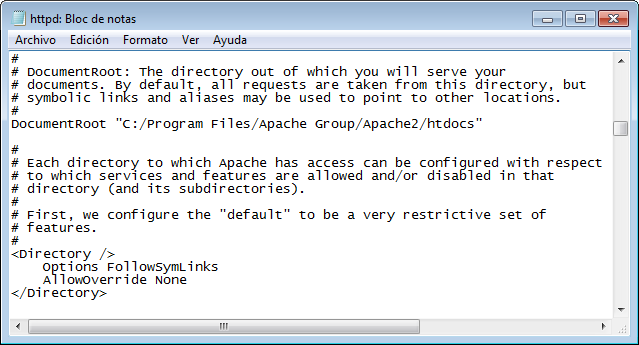


En Ubuntu puedes encontrar los archivos de configuración en la carpeta /etc/apache2 .



**Contenido de los archivos de configuración**

El contenido del archivo de configuración se organiza a través de **directivas**, directivas en forma de **etiquetas XML que contienen otras directivas** y líneas de comentarios que empiezan por el carácter **"#**".



La **configuración de Apache** se realiza a través de **archivos de texto** mediante **directivas** que permiten escoger las **distintas opciones disponibles**.

Aunque las directivas son las mismas sea cual sea el sistema operativo, **la localización de los archivos de configuración es diferente.**

Requisitos de las directivas

* ***Una directiva por línea***. Para indicar que una directiva continúa en la siguiente línea se puede poner una barra invertida \ como último carácter.
* Las directivas **no son sensibles a mayúsculas o minúsculas** pero muchos argumentos si.
* Los argumentos se separan por espacios en blanco. Si un argumento contiene espacios debe ponerse entre comillas.
* Los **comentarios** comienzan con el carácter **#**, y no pueden estar en la misma línea que una directiva.
* Las **líneas en blanco y los espacios a principio de línea se ignoran**. Sólo sirven para facilitar la lectura de los ficheros.

**Carpeta principal visible en servidor web Apache (DocumentRoot)**

Las páginas webs que se publican a través del servidor web Apache se encuentran alojadas en una carpeta del mismo equipo. La ruta a dicha carpeta debe aparecer en el archivo de configuración usando la **directiva**[**DocumentRoot**](https://httpd.apache.org/docs/2.2/mod/core.html#documentroot).

La **ruta por defecto** que se establece en una instalación limpia depende del sistema utilizado para la instalación de apache (XAMP, Wamserver, Apache,..)es:

1. Windows: *<CarpetaInstalacion>/****htdocs***
2. Ubuntu: */var/www*

La *directiva* ***DocumentRoot*** la puedes **encontrar en el archivo**:

* **Windows**: *<CarpetaInstalacion>/conf/****httpd.conf***
* Ubuntu: */etc/apache2/sites-enabled/000-default*

Como puedes comprobar, para usar esta directiva tan solo hay que especificar tras ella la ruta a la carpeta raíz a partir de la cual se van a guardar las páginas web que se van a servir.

Indica el **directorio desde el que apache va a servir archivos**. El servidor añade la ruta indicada en la URL a este directorio. Todos los directorios que va a servir apache deben tener permiso de lectura y ejecución para todos los usuarios, y todos los archivos que sirva permiso de lectura.

Ejemplo:

//Si la directiva es:

DocumentRoot /var/www/html

//Si la URL solicitada es:

http://www.miservidor.com/prueba/index.html

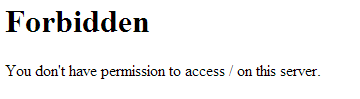
//Apache servirá el fichero:

index.html

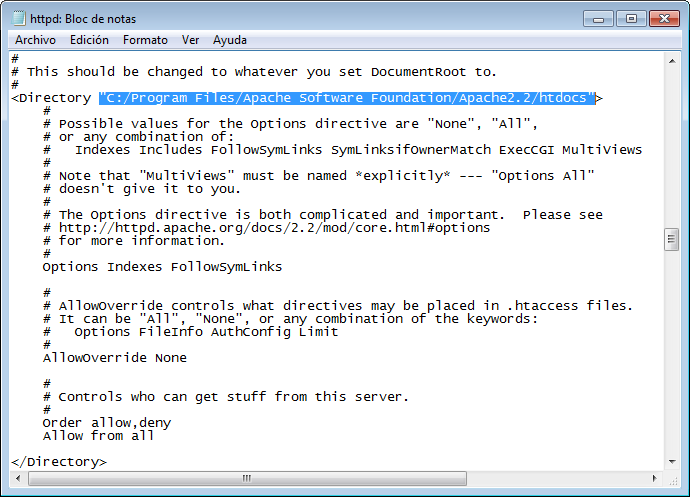
//Que estará en el directorio:

//var/www/html/prueba

En esa directiva se encuentran los permisos de acceso a esa carpeta, por lo que si no se hace este cambio se obtendrá un mensaje de error de permiso de acceso denegado.



Por tanto, en el mismo archivo de configuración que has utilizado para cambiar la directiva DocumentRoot, debes buscar la directiva <Directory> que contiene la misma ruta. Ten en cuenta que **puedes encontrar varias directivas** <**Directory**> por lo que debes asegurarte que estás cambiando aquella que contiene la misma ruta que había en DocumentRoot. Dentro de la etiqueta <Directory> debes sustituir la ruta anterior por la nueva.



Para que los cambios tengan efecto, recuerda guardar los cambios del archivo de configuración y reiniciar el servidor web Apache

**Configuración de directorios**

Los directorios que son publicados a través del servidor web Apache pueden ser configurados a través de la **directiva**[<**Directory**>](http://httpd.apache.org/docs/2.2/mod/core.html#directory) con la posibilidad de especificar una serie de opciones y otras directivas que se puedan aplicar al contexto de directorios, como los permisos de acceso a dicho directorio. Esta directiva debe incluirse **en el archivo de configuración** del servidor web Apache.

**Directory**

Las directivas que se incluyan dentro de Directory **sólo se aplican a ese directorio**, **los subdirectorios y los contenidos**. Debe **indicarse** con la **ruta absoluta**.

Ejemplo:

<Directory /**var**/**www**/**html**/**cursos**>

//Aquí van las directivas

</Directory>

## Directivas de accesos autorizados (Allow, Deny y Order)

La **directiva**[***Allow***](http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#allow) permite controlar los equipos que **pueden acceder al directorio**. El formato de uso es indicar ***Allow******from*** seguido de una de las opciones que se comentan a continuación:

Si se indica **Allow from all**, **se autoriza el acceso al directorio** desde cualquier equipo:

**Allow** **from all**

Para establecer límites en el acceso se puede indicar una **lista de direcciones IP o nombres de dominio** que indicará los equipos que pueden acceder al directorio. En este formato, se puede indicar sólo la parte final de un nombre de dominio, la parte inicial de una dirección IP, o indicar la máscara de red para indicar un rango más amplio de equipos. A continuación se muestran varios ejemplos:

Allow from apache.org   
Allow from .net example.edu   
Allow from 10.1.2.3   
Allow from 192.168.1.104 192.168.1.205   
Allow from 10.1   
Allow from 10 172.20 192.168.2   
Allow from 10.1.0.0/255.255.0.0   
Allow from 10.1.0.0/16

También se pueden utilizar **variables de entorno** como puedes ver en la documentación del servidor web Apache.

De manera similar se puede utilizar la **directiva**[**Deny**](http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#deny)para **denegar el acceso** a los equipos que se indiquen, siguiendo el mismo formato que el usado con la directiva Allow.

Las directivas **Allow** y **Deny** **se pueden colocar en cualquier orden**, y se pueden añadir varias líneas de cada tipo.

Es la **directiva**[**Order**](http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#order)la que **establece el orden en el que se aplicarán**. **Por defecto**, el orden es el siguiente:

**Order** **Deny**,**Allow**

Es decir, en primer lugar se consideran todas las directivas de denegación de acceso al directorio, y luego se tienen en cuentas las directivas de aceptación de permiso.

Por ejemplo, para **permitir únicamente el acceso a los equipos del dominio apache.org**:

**Order** **Deny**,**Allow**  
**Deny** from all  
**Allow** from apache.org

En cambio, si se desea dar permisos únicamente a los equipos del dominio apache.org excepto a los equipos del dominio foo.apache.org se debe usar la siguiente configuración, cambiando el orden a Allow,Deny:

**Order** **Allow**,**Deny**   
**Allow** from apache.org   
**Deny** from foo.apache.org

Observa que en este último caso, si se hubiera indicado el order contrario (Order Deny,Allow) se daría acceso a todo el mundo excepto al dominio foo.apache.org.

Más ejemplos

Ejemplo 1

Order deny,allow Acceso permitido por defecto.

deny from all Se deniegan todos los equipos.

allow from daw.net Se permite el acceso a los equipos del dominio daw.net .

**Resultado**: sólo los equipos de \*.daw.net son permitidos.

Ejemplo 2

Order allow,deny Acceso denegado por defecto.

allow from daw.net Se permite el acceso a los equipos del dominio daw.net .

deny from bbdd.daw.net Se deniega el acceso a los equipos del dominio bbdd.daw.net .

**Resultado**: se permite el acceso a todos los equipos de \*.daw.net excepto a los de \*.bbdd.daw.net .

## Directiva AllowOverride

Cuando el servidor encuentra un **archivo**.htaccess en cualquiera de sus carpetas publicadas, las directivas que incluye dicho archivo son **tenidas en cuenta por encima de** **las directivas establecidas en la configuración del servidor**.

## ¿Qué es el archivo .htaccess?

El archivo .htaccess (hypertext Access) es un texto en formato ASCII que permite definir los parámetros de un directorio en una página web sin tener que editar el archivo de configuración del servidor completo.

Normalmente, los archivos .htaccess funcionan para **poner restricciones de seguridad a un sitio web** ya que desde él, se pueden bloquear IPs desconocidas, bots y también modificar las direcciones para hacerlas más amigables y/o seguras.

Si **no se quiere** que se tenga en cuenta este archivo, se debe utilizar la **directiva**[AllowOverride](http://httpd.apache.org/docs/2.2/mod/core.html#allowoverride) con el parámetro None.

AllowOverride None

**Por defecto, se permite el uso del archivo .htaccess**, pero si quieres especificar literalmente que se considere dicho archivo dentro de una directiva Directory, debes indicar el parámetro All:

AllowOverride All

Además de los parámetros None y All, puedes usar el **tipo de directivas que deseas permitir** dentro de los archivos .htaccess. Puedes elegir entre las siguientes directivas: AuthConfig, FileInfo, Indexes, Limit, Options. En la documentación del servidor web Apache puedes encontrar infromación sobre su uso.

## ServerRoot

En **ServerRoot** se encuentra especificada la **localización del directorio raíz donde está instalado apache**. Esta directiva no se debe cambiar a menos que se cambie la carpeta de instalación de apache a otro directorio.

Ejemplo:

ServerRoot /**etc**/**apache2**

## Directiva DirectoryIndex

La **directiva**[DirectoryIndex](http://httpd.apache.org/docs/2.2/mod/mod_dir.html#directoryindex) permite indicar la lista de archivos que se buscarán cuando el **cliente accede al servidor web sin indicar un nombre de archivo** específico.

**Por defecto**, se buscará el archivo index.html**.**

**Si quieres** que también se busque el archivo **index.php** debes indicar lo siguiente:

<Directory /directorio/mi/web>  
 **DirectoryIndex** **index.html** **index.php**  
</Directory>

Esta directiva se puede usar dentro de una directiva Directory o en la configuración global del servidor.

## Listen

La **directiva**[**Listen**](http://httpd.apache.org/docs/2.2/mod/mpm_common.html#listen) permite indicar los **puertos** a través de los cuales el servidor web va a permitir **admitir solicitudes** de páginas web. Opcionalmente, también se pueden indicar las **direcciones IP** correspondientes a los adaptadores de red del servidor a través de los que se admitirán las solicitudes.

La **opción por defecto** que trae el servidor web Apache es escuchar **por el puerto 80** a través de cualquier dirección IP. Esto lo puedes comprobar en los archivos de configuración, donde aparece:

Listen 80

Para hacer que el servidor acepte conexiones en los puertos 80 y 8080 escribiremos la directiva:

Listen 80

Listen 8080

Para atender dos IP distintas, con distintos puertos escribiremos:

Listen 192.168.255.5:80

Listen 192.168.255.3:8080

## ErrorLog

La directiva ErrorLog indica **qué fichero tiene el registro de errores**. Si la ruta que se le indica no es absoluta, se considera relativa al ServerRoot. Por defecto se encuentra en el directorio logs dentro de ServerRoot

## Proteger sitios web con contraseñas en el servidor web Apache

Aunque puedes utilizar las directivas Allow y Deny para restringir el acceso mediante direcciones IP o nombres de dominio a los sitios web publicados por el servidor web de Apache, una medida de seguridad más estricta es **solicitar al usuario una contraseña para acceder a los directorios publicados**.

Por muchos motivos puede resultar muy útil tener controlado el acceso a **determinados directorios de un servidor web** o simplemente a contenidos de nuestra página o blog mediante un servicio de autenticación basado en usuario y contraseña. Hablo de algo como la **típica ventana** que **nos aparece cuando deseamos entrar en una web con servicios restringidos**.

Para realizar esta configuración, es necesario disponer de un **archivo de contraseñas** donde se almacenarán las parejas de nombres de usuarios con sus contraseñas correspondientes (encriptadas). Este archivo puede crearse en cualquier carpeta, aunque es **recomendable tenerlo fuera de las carpetas publicadas** por el servidor web.

La creación del archivo se realiza con la siguiente instrucción:

htpasswd -c /ruta/al/archivo/nombre\_archivo nombre\_usuario

En Windows puedes encontrar la aplicación **htpasswd** en la **carpeta** **bin** de la instalación del servidor web Apache.

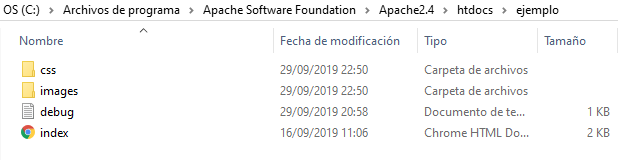
Ahora debes indicar en la configuración del servidor que quieres utilizar este tipo de seguridad en un determinado directorio, por lo que debes **añadir las líneas siguientes** dentro de la **sección Directory** correspondiente al directorio que deseas proteger, o bien dentro de un **archivo .htaccess** **que se almacene en la carpeta a proteger**.

### Comenzamos.

Como algunos ya deben saber, por defecto los sitios disponibles de Apache se almacenan en el directorio **/var/www/** y para este ejemplo crearé un directorio con una plantilla HTML e intentaré blindarlo con .htpasswd.

El **directorio a proteger** **htdocs/ejemplo/** donde he colocado la siguiente página.

En el servidor



En el navegador



### 1. Vamos a crear el .htpasswd.

## Cómo crear el fichero .htaccess en Windows

El fichero **.htaccess** no es especial en cuanto a su forma de creación. No se trata más que de un fichero de texto en modo ASCII que, simplemente, no tiene nombre. Al menos tal y como estamos acostumbrados a ver: nombre.extensión

El hecho de que **tenga un punto delante** indica que se trata de un **fichero invisible**, que no será visualizado si se genera un listado de ficheros de una carpeta concreta.

Para crearlo bastará con un editor de textos (notepad, Sublime,…) y guardarlo como **htaccess.txt**

Una vez hecho esto se puede subir al servidor vía FTP y allí se renombra a .htaccess También existe la posibilidad de crearlo directamente desde el panel del hosting.

Si lo prefieres, también puedes emplear cualquier plantilla de las muchas que hay en la Web para crearlo. No tendrás más seleccionar del formulario las opciones que quieres configurar, darle a aplicar y listo. Entre las plantillas que más me gustan:

* [.htaccess Generator](http://www.htaccessredirect.net/)
* [Htaccess generator](http://approsto.com/htaccess-generator/)
* [.htaccessEditor](http://www.htaccesseditor.com/es.shtml)

## Generadores de htpasswd

De igual forma no obligo a nadie a utilizar la terminal cuando existen servicios en Internet para ello:

<https://www.web2generators.com/apache-tools/htpasswd-generator>

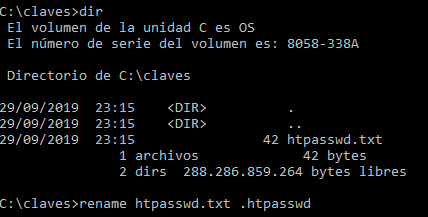
<http://www.htaccesstools.com/htpasswd-generator/>

Genere el contenido de un archivo **.htdigest** para que su [servidor web](https://websistent.com/category/web-servers/) realice [la autenticación de resumen HTTP](http://en.wikipedia.org/wiki/Digest_access_authentication)

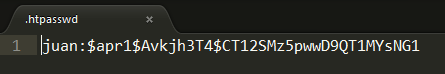
<https://websistent.com/tools/htdigest-generator-tool/>

El resultado los copiamos en un archivo de texto y con el nombre **.htpasswd** si así lo desean. Con respecto a este método no dejen de leer lo que comento en el siguiente párrafo.

En nuestro caso veremos el archivo se ha generado donde los esperábamos /home/krel/.htpasswd. En la imagen he resaltado que con este método el archivo pertenece al grupo y usuario de apache, que en Ubuntu se le llama www-data. Esto es importante porque si hemos generado de alguna otra forma el .htpasswd.



Bien , ahora veremos cómo son sus tripas:



Como pueden ver, está el usuario (juan) pero la contraseña está cifrada.

De esta forma ya hemos creado y diseñado .htpasswd para el usuario juan, al final no es más que un **contenedor para la contraseña**. “Todo con un comando” sería un buen eslogan.

### 2. Configurar Apache de forma que lo utilice y permita ese servicio en el sitio.

Editaremos el fichero de configuración y lo hacemos tal como aparece la imagen a continuación y adecuada a nuestro ejemplo:

<Directory "C:\xampp\htdocs\ejemplo">

AuthType Basic

AuthName "Sitio restringido"

AuthUserFile "C:/claves/.htpasswd"

Require valid-user

Require user juan

</Directory>

Explico las líneas:

<Directory “C:\xampp\htdocs\ejemplo”>

Hay que poner el directorio a proteger.

AuthType Basic

Es el tipo de negociado de usuario y contraseña con el navegador, en este caso es Basic, sin cifrar.

Existe otro tipo de negociado: **Digest**, este si cifra el usuario y la contraseña sin embargo la última palabra la tiene el navegador en cuanto a si soporta dicho servicio. **Digest** se puede decir que es una implementación un poco más robusta ante sniffers. **Les invito a profundizar en esto**.

AuthName “Sitio Restringido”.

Es mensaje que le saldrá al usuario en el navegador con el formulario de login y puede ser el que queramos.

AuthUserFile "C:\claves\**htpasswd.txt**"

Ruta el archivo **.htpasswd**. Por esa razón es que digo que ni siquiera es necesario nombrarlo .htpasswd, simplemente poniendo la ruta funcionaría igual.

Require valid-user

Esa línea está diseñada para múltiples usuarios con contraseñas individualizadas. En caso que sea un único usuario puede resumirse como la siguiente variante.

Require user el\_que\_sea

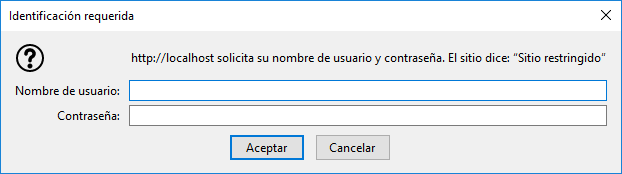
En mi ejemplo podría ser: Require user Luke

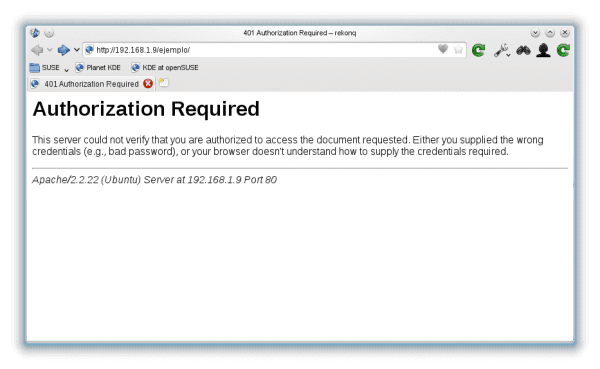
Y finalmente encerramos todo con </Directory>

Reiniciamos apache:

# service apache2 restart

Probemos.

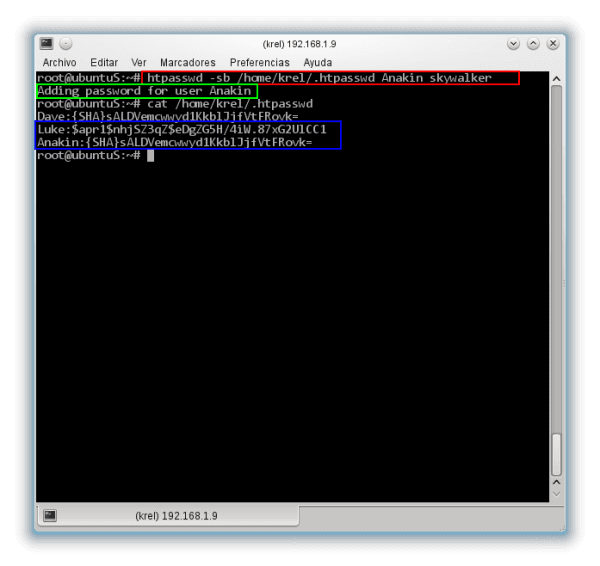


Si no nos autenticamos correctamente nos saldrá un Error 401.

En el lado opuesto, si ponemos el usuario correcto y la contraseña nos dejaría pasar sin mayores inconvenientes.



**Tips adicionales.**

* Para añadir **más usuarios** a un archivo htpasswd ya creado 

Más información.

**htpasswd – -help**

Si quieren profundizar más en este tema siempre es bueno RTFM!!!  
<http://httpd.apache.org/docs/2.0/es/howto/auth.html>

<http://httpd.apache.org/docs/2.2/misc/password_encryptions.html>

ACTIVIDADES A ENTREGAR OBLIGATORIAMENTE

**ACTIVIDAD 1 : CONTROL DEL ACCESO AL SERVIDOR**

Indica los resultados de las siguientes configuraciones:

**Configuración 1**

Order allow,deny

allow from 100.100.200.0/24 deny from www.daw.com

**Configuración 2**

Order allow,deny

allow from 192.168.1.0/16 deny from all

**ACTIVIDAD 2: CONTROL DE ACCESO MEDIANTE IP**

1. Inicia sesión en con un usuario con privilegios de administrador.
2. Crea el directorio www/alumnoDaw . Dentro, crea un archivo llamado ***alumno.html*** con el contenido que quieras.
3. Edita el archivo de configuración. Utilizaremos la directiva <Directory> junto con las directivas Order , Allow y Deny para denegar el acceso al directorio a todos los equipos excepto al local
4. Reinicia el servidor para que los cambios tengan efecto.
5. Comprueba que se puede acceder a alumnoDaw desde local y no desde otro host. adjunta un pantallazos.

**Autenticación y autorización Basic y Digest .**

Autenticación y autorización son términos distintos. La autenticación es el proceso de verificar que alguien es realmente quien dice ser, mientras que la autorización es el proceso por el que se permite a alguien hacer o acceder a algo que quiere.

Hay varios tipos de autenticación:

* **Basic**: la lleva a cabo el módulo **mod\_auth\_basic** .
* **Digest**: la lleva a cabo el módulo **mod\_auth\_digest** .

Nos vamos a centrar en las dos primeras. En la autenticación **basic** la contraseña es enviada por el cliente en **texto plano**, mientras que en la autenticación **digest** se envía **cifrada** (con cifrado débil) por el cliente.

**ACTIVIDAD 3: AUTENTICACIÓN HTTP BASIC**

Esta autenticación guarda los usuarios y sus contraseñas **encriptadas en un archivo**. Los usuarios y contraseñas se tienen que ir metiendo uno a uno. Este módulo de apache viene activado por defecto

Vamos a configurar en Apache la autenticación HTTP Basic sobre el directorio /var/www/ alumnoDaw para que sólo puedan acceder los usuarios alumnoDaw1 y alumnoDaw2.

1. Inicia sesión con un usuario con privilegios de administrador. Comprueba, consultando el directorio: C:\Apache24\conf\httpd.conf que el **módulo auth\_basic** está habilitado. Adjunta un pantallazo.
2. Para poder usar la autenticación **basic** tenemos que crearnos un fichero accesible por Apache en el que se guarden los usuarios y sus contraseñas. Para crear este fichero usaremos el comando **htpasswd o** mediante un generador de claves

<http://www.htaccesstools.com/htpasswd-generator/>

<https://www.web2generators.com/apache-tools/htpasswd-generator>

1. Edita el fichero de configuración y permite el acceso al directorio /var/www/alumoDaw a los usuarios alumnoDaw1 y alumnoDaw2, tal y como se puede ver en la imagen.

<Directory /var/www/alumoDaw>

completarlo

</Directory>

1. Reinicia el servidor para que los cambios surtan efecto.

Accede con los usuarios alumnoDaw1 y alumnoDaw2. Inténtalo con otro usuario. ¿Es posible? ¿Qué error aparece? Haz otro pantallazo donde se aprecie esto último.

No he conseguido que me de error

**ACTIVIDAD 4: AUTENTICACIÓN HTTP DIGEST**

Vamos a configurar la autenticación digest sobre el directorio /var/www/daw para que sólo puedan acceder los usuarios adminDaw1 y adminDaw2 .

1. Crea el directorio /var/www/ daw. Dentro, crea un fichero daw.html con el contenido que quieras.
2. El módulo auth\_digest no está habilitado por defecto (puedes comprobarlo). Habilítalo
3. Reinicia el servidor para que los cambios tengan efecto.
4. Para usar la autenticación digest hay que crear un fichero accesible por Apache en el que se guardarán los usuarios y sus contraseñas asociados a un dominio ( realm ). Para crear este fichero usaremos el comando htdigest o utilizando alguna utilidad que nos proporciona internet como esta página

<https://websistent.com/tools/htdigest-generator-tool/>

1. Edita el fichero de configuración y permite el acceso al directorio /var/www/departamento a los usuarios adminDaw1 y adminDaw2, según puedes ver en la figura.

<Directory "c:/var/www/departamento">

       completarlo

</Directory>

7. Reinicia el servidor para que los cambios surtan efecto.