



TEMA



Tema 1. Servidores de aplicaciones web

Sistemas Microinformáticos y Redes

Aplicaciones web

Tema 1: Servidores de aplicaciones web

¿Qué aprenderás?

- Reconocer la evolución de internet.
- Identificar los tipos de servidores web.
- Instalar un entorno lamp.
- Crear hostings virtuales.

¿Sabías que...?

- MySQL es el gestor de base de datos más utilizado para el desarrollo de aplicaciones web.
- Nginx es el servidor web más utilizado en la actualidad.
- Phpmyadmin es un entorno web pensado para administrar una base de datos.
- XAMPP es un paquete integrado para instalar de forma automática, apache, mariadb, php y phpmyadmin.

1. Servidores de aplicaciones web

1.1. Introducción

Desde los inicios de Internet en 1966, las páginas web han ido evolucionando tanto en la forma de utilizarla como en las tecnologías usadas para su desarrollo.

La Web 1.0 era un conjunto de páginas escritas en lenguaje HTML que ofrecían información estática a sus visitantes. Con un proceso de actualización complicado y con una interacción mínima reducida sobre todo a formularios de contacto o a suscripciones.

Con la Web 2.0, a partir de 2004, la web se convierte en colaborativa, los usuarios se transforman en productores de contenidos. Aparecen los blogs, las redes sociales, las wikis y las páginas dinámicas. La información está en permanente cambio.



Con el término Web 3.0 conocemos a la web semántica, se trata de una versión más inteligente y próxima a entender el lenguaje humano, con un mayor uso de los dispositivos móviles y las herramientas en la nube. Aparecen nuevos lenguajes sobre todo relacionados con la inteligencia artificial, se ofrece publicidad selectiva en función de nuestros hábitos de navegación y preferencias.

La web no se detiene y la versión 4.0 va a suponer una importante evolución. En este modelo la web será capaz de mostrar soluciones completas a una determinada necesidad con la mínima interacción. Por ejemplo solicitar un taxi, el dispositivo será capaz de conectar directamente con el servicio de taxis y enviarlo a la geolocalización del usuario.

Cada vez más, las aplicaciones informáticas que se instalaban localmente en un ordenador tienen su equivalente en la web. Es decir existe la tendencia a desarrollar aplicaciones en un entorno web con funcionalidades similares a las que ya existen en versión local, como el correo electrónico, los calendarios o los paquetes ofimáticos.

Se conoce como aplicación web el software que tiene como entorno de trabajo un espacio web, a la que se accede utilizando un navegador web y está alojada en servidores de Internet o de una Intranet.

1.2. Servidor Web

Un servidor web es un software que forma parte de los servidores y tiene como misión atender las peticiones de los clientes (navegadores) y devolver la respuesta normalmente en forma de página web.

Los servidores web tienen como principal función almacenar (hosting) todos los archivos propios de una página web (textos, documentos, videos, imágenes, etc.) y transmitirlos utilizando el protocolo HTTP (Hypertext Transfer Protocol) a los clientes a través de un navegador web.

El servidor web puede realizar diferentes funciones entre las que se encuentran:

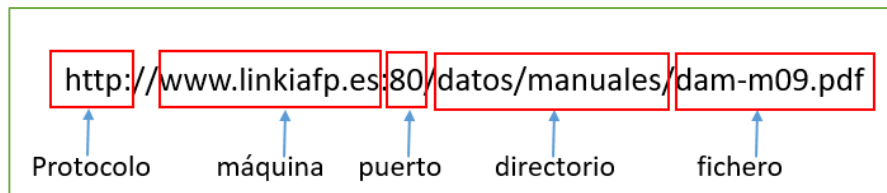
- Registro de actividades y errores.
- Control de acceso por: dirección del cliente, contenido, o usuario y contraseña.
- Virtual Hosts: mantener diferentes webs en el mismo servidor.
- Proxy: para reenviar peticiones a otro servidor.
- Reescritura y Mapeo de URLs

La comunicación entre el navegador y el servidor se realiza utilizando tres protocolos:

- **TCP (Transmission Control Protocol)**. Es el encargado de que los mensajes lleguen al destino sin errores.
- **IP (Internet Protocol)**. Es el responsable de hacer que el mensaje encuentre el camino hasta el servidor.
- **HTTP (Hypertext Transfer Protocol)**. Es el protocolo que utiliza el cliente para pedir un recurso al servidor. En la dirección URL se indica que protocolo utilizaran el cliente y el servidor para intercambiar los datos.

Una URL es la dirección que se asigna a cada uno de los recursos disponibles en la red con la finalidad de que podamos encontrar aquello que estamos buscando en la red: una página, un sitio, un archivo, un documento, etc.

En general una URL se divide en varias partes:



1.3. Cliente Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida. Las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Un cliente realiza peticiones a otro programa, y el servidor, le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajoso en una red de ordenadores.

El cliente es el que inicia las solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación. Espera y recibe las respuestas del servidor. Puede conectarse a varios servidores a la vez. Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

El servidor al iniciarse, espera a que le lleguen las solicitudes de los clientes. Desempeña entonces, un papel pasivo en la comunicación. Siempre está a la espera de que llegue alguna petición. Tras la recepción de una solicitud, la procesa y luego envía la respuesta al cliente. Por lo general, acepta conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).



Como ventajas de este sistema podemos destacar:

- Centralización del control: El acceso, los recursos y los datos son controlados por el servidor impidiendo accesos no autorizados al sistema.
- Escalabilidad: Se pueden aumentar la cantidad de clientes y servidores por separado. Cualquier elemento puede ser mejorado en cualquier momento.
- Fácil mantenimiento: Al estar distribuidas las funciones entre varios elementos es posible reemplazar, actualizar o trasladar un servidor, sin que los clientes se afecten o se afectaran mínimamente.
- La existencia de tecnologías lo suficientemente desarrolladas que aseguran que el entorno sea fácil de usar, amigable y seguro en las transacciones de información.

Como inconvenientes del modelo cliente servidor tenemos:

- La congestión del tráfico ha sido siempre el principal problema, esto se produce cuando una gran cantidad de clientes envían peticiones a un mismo servidor.
- Cuando un servidor está caído las peticiones de los clientes no pueden ser satisfechas, esto no pasa en otros modelos como por ejemplo en las redes P2P donde los elementos desempeñan al mismo tiempo el rol de cliente y de servidor.
- El hardware y el software del cliente y del servidor son muy determinantes en el sistema.

1.4. Arquitectura de 3 capas

Una aplicación almacenada en uno o varios servidores normalmente está estructurada en una arquitectura de tres capas:

- El cliente, el que solicita los recursos, utilizando el navegador web ejecuta la interfaz de acceso al servicio web.
- El servidor de aplicaciones, está formado por los scripts o programas que se ejecutan en el servidor para atender las solicitudes del usuario.
- El servidor de datos (SGDB) Donde se almacenan los datos con los que trabaja la aplicación o el servicio.



Dependiendo del volumen de carga que tenga que soportar el servicio, las tres capas se pueden implementar sobre una o varias máquinas. En el mercado existen muchas alternativas, aunque una de las más frecuentes está compuesta por el servidor web apache, PHP como capa de aplicación y MySQL como SGDB.

1.5. Tipos de servidor Web

Los servidores web más usados en Internet son los siguientes:

- **Apache.**

Nacido en Abril de 1996 es el servidor web de referencia para internet y hoy en día sigue vigente. Durante más de 25 años ha sido el líder indiscutible de internet, ahora está perdiendo popularidad sobre todo por el auge de IIS y Nginx (en noviembre de 2007 servía casi el 51% de todos los sitios web). Se utiliza tanto para aplicaciones web estáticas como dinámicas.

Como ventajas podemos citar: que es de código abierto, multiplataforma (Windows, Linux, Unix) y gratuito. El auge de las aplicaciones realizadas en PHP a partir del año 2000 popularizó mucho el paquete LAMP (Linux, Apache, Mysql y PHP).

Su principal desventaja es su bajo rendimiento cuando recibe miles de peticiones simultáneas.

- **Nginx**

Nginx (pronunciado en inglés “engine ex”), aparecido a finales de 2004, es un servidor web de código abierto y gratuito, aunque también existe una versión comercial. Destaca sobre todo por su alto rendimiento consumiendo muy pocos recursos, ideal para entornos de trabajo con muchas peticiones simultáneas. Es el sistema usado por un larga lista de sitios web conocidos como: Wordpress, GitHub, SourceForge y algunas partes de Facebook.

Incorpora también un servidor proxy, balanceador de carga, POP3 y IMAP, y está disponible para Windows, Linux y Unix.

- **Microsoft IIS**

Internet Information Server, conocido con el nombre de IIS, es un servidor web creado por Microsoft para su plataforma de sistemas operativos Windows. Se integró de forma nativa a partir de Windows Server 2003.

Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Tiene una estructura modular que le permite ejecutar diferentes tipos de páginas web, sobre todo ASP y ASP.NET. Aunque también puede procesar páginas PHP y Perl. Actualmente se integra con Microsoft Azure.

Su principal inconveniente es que es un servidor propietario y exclusivo de Microsoft, por tanto carece de integración con muchas tecnologías y lenguajes de programación.



- **GWS**

Google Web Server es un servidor web privado escrito en C++ utilizado por google para la mayoría de su infraestructura web, no está disponible para el público y no puede descargarse desde ningún sitio.

- **LiteSpeed**

Es uno de los servidores más rápidos de la actualidad debido sobre todo a su arquitectura. La principal diferencia con Apache es que LiteSpeed trabaja con eventos y Apache con procesos que consumen muchos más recursos. Lo que permite ejecutar los procesos de PHP mucho más rápido.

Ha sido integrado en diferentes proveedores de hosting ya que se integra con los paneles de control CPanel y Plesk. Incluye también un sistema de cache para las aplicaciones WordPress y Magento para que su acceso sea mucho más rápido.

Una de sus principales inconvenientes es que la versión completa es de pago.

- **Cherokee**

Creado en 2001 por Álvaro López Ortega, se trata de un servidor web open source y surgió como una alternativa a Apache. Fácil de configurar y puede utilizarse en entornos de alta demanda. Integrado con PHP, CGI, SSL y HTTP Proxy. También es multiplataforma pero en la actualidad casi no se usa.

- **NodeJS**

Es un entorno de desarrollo open source para Javascript utilizando como base el motor V8 de Javascript desarrollado por Google Chrome. Aunque muchos opinan que no es un servidor web, dispone de un módulo Node HTTP module que permite ejecutar Javascript desde el lado del servidor.

- **Sun Java System Web Server**

Conocido también como SJSAS es un tipo de servidor destinado sobre todo a aplicaciones realizadas en Java, soporta integración nativa con Netbeans, Java Studio Creator y Java Studio Enterprise.

- **NCSA HTTPd**

Era un servidor web desarrollado originalmente por Robert McCool en el National Center for Supercomputing Applications su desarrollo se suspendió en 1998. Pero su código sobrevivió gracias al Proyecto Apache. Fue el servidor web más utilizado hasta 1996.

En la página web de NetCraf podemos obtener cada mes la lista de servidores web más utilizados. En Noviembre de 2020, el más utilizado es Nginx (33,7%), seguido de Apache (26,7%) y IIS con el 7,9%.



1.6. Instalación de un servidor LAMP en Ubuntu.

Instalar un servidor web en nuestro ordenador nos va a permitir probar todos nuestros desarrollos web en local sin necesidad de contratar un hosting.

LAMP es el acrónimo usado para describir el siguiente conjunto de herramientas:

- L: Linux Sistema Operativo
- A: Servidor Web Apache
- M: Base de datos, MySQL o MariaDB
- P: Lenguaje de programación: PHP



1.6.1. Instalación del servidor web Apache

- Empezamos instalando el servidor web apache con el siguiente comando:

```
root@joan-ubuntu:~# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

- Una vez instalado abrimos los puertos 80 y 443 en el firewall y recargamos el firewall con los siguientes comandos:

```
root@joan-ubuntu:~# ufw allow 80/tcp
Reglas actualizadas
Reglas actualizadas (v6)
root@joan-ubuntu:~# ufw allow 443/tcp
Reglas actualizadas
Reglas actualizadas (v6)
root@joan-ubuntu:~# ufw reload
```

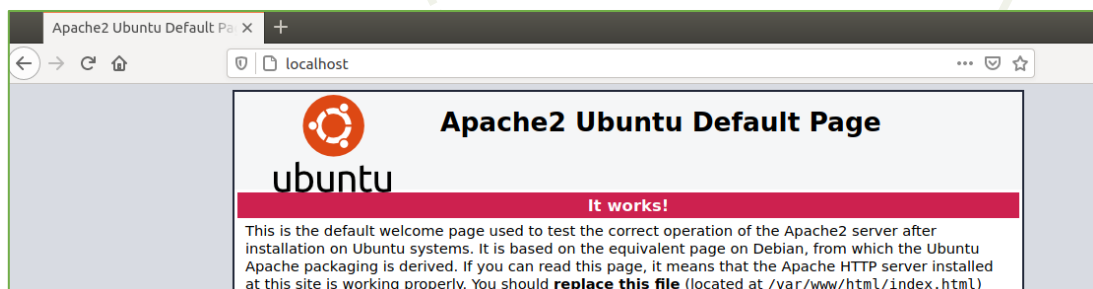
- Con esto ya tenemos instalado el servidor web. Para ver el estado debemos introducir el siguiente comando:

systemctl status apache2

Deberá salir como **Active (running)**.

```
root@joan-ubuntu:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Sun 2020-12-06 15:56:30 CET; 2min 36s ago
   Main PID: 22803 (apache2)
     Tasks: 55 (limit: 4552)
    CGroup: /system.slice/apache2.service
            └─22803 /usr/sbin/apache2 -k start
              └─22805 /usr/sbin/apache2 -k start
                └─22806 /usr/sbin/apache2 -k start
```

- También podemos ejecutar la dirección localhost en el navegador, para comprobar el funcionamiento.



- Para iniciar o parar el servidor web utilizamos los comandos:

Detener el servidor: **systemctl stop apache2**

Iniciar el Servidor **systemctl start apache2**

Reiniciar el servidor: **systemctl restart apache2**

1.6.2. Instalación del servidor de base de datos MySQL

- Una vez está funcionando correctamente el servidor web instalamos el servidor de base de datos MySQL.

```
root@joan-ubuntu:~# apt-get install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Debemos confirmar la descarga de aproximadamente 160 MB de ficheros. En las instalaciones actuales para acceder al Shell de Mysql debemos introducir la contraseña root del sistema. En versiones anteriores pedía crear una cuenta de usuario.

- Con la instrucción **systemctl status mysql** podemos comprobar su estado.

```
root@joan-ubuntu:~# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Sun 2020-12-06 16:03:41 CET; 1min 20s ago
   Main PID: 24842 (mysqld)
     Tasks: 27 (limit: 4552)
    CGroup: /system.slice/mysql.service
            └─24842 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

dic 06 16:03:40 joan-ubuntu systemd[1]: Starting MySQL Community Server...
dic 06 16:03:41 joan-ubuntu systemd[1]: Started MySQL Community Server.
```

- Ejecutando la consola de Mysql podemos comprobar su funcionamiento.

```
root@joan-ubuntu:~# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql>
```

- Por defecto la instalación de mysql no viene con seguridad. Para añadirla deberemos ejecutar este comando:

```
root@joan-ubuntu:~# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG  Length >= 8, numeric, mixed case, special characters and dictionary
le

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
```

Nos pedirá validar el password (respondemos Y) y luego su nivel de seguridad (0-2). Como estamos en un entorno de pruebas escogemos 0 bajo, solo números.

Deberemos escoger una contraseña de acuerdo a estos criterios. Nos pide algunas configuraciones más. Debemos tener en cuenta que es un servidor de pruebas y no un servidor en producción.

- Borrar a los usuarios anónimos->Sí
- Deshabilitar al usuario root en remoto -> No
- Borrar la base de datos de test->No
- Recargar las tablas de privilegios->Sí

```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n
... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
```


1.6.3. Instalación del servidor de base de datos MySQL

En el siguiente paso instalamos la capa de aplicación, es decir los módulos de php necesarios para que se pueda leer el lenguaje de programación php en el servidor Web. Se pueden instalar muchos módulos pero nosotros lo vamos a hacer con estos básicos para que pueda funcionar:

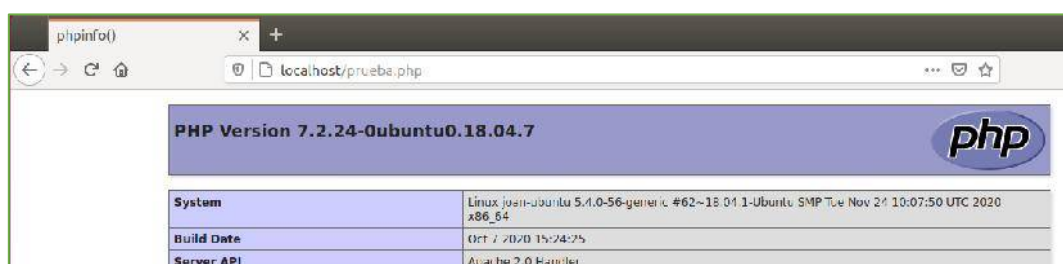
```
apt-get install php libapache2-mod-php php-mysql php-cli
```

```
root@joan-ubuntu:~# apt-get install php libapache2-mod-php php-mysql php-cli
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios
```

Para comprobar el funcionamiento correcto de php podemos escribir un pequeño fichero en el directorio /var/www/html, raíz del servidor apache con la siguiente información.

```
<?php
phpinfo()
?>
```

El fichero lo almacenamos con el nombre prueba.php y lo ejecutamos desde el navegador con el siguiente comando.

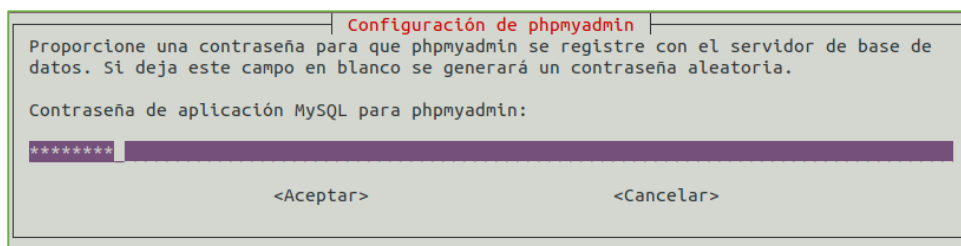


1.6.4. Instalación de phpmyadmin

La herramienta phpmyadmin nos va a permitir para administrar de manera gráfica mysql, se ejecuta en un entorno web. Es muy recomendable para administrar bases de datos de forma remota.

```
root@joan-ubuntu:~# apt-get install phpmyadmin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Seleccionamos que la instalación la tenemos sobre el servidor web apache2. Ahora nos pide la instalación de la propia base de datos para que funcione phpmyadmin. Marcamos SI para configurar la base de datos para phpmyadmin con dbconfig-common. En el último paso especificamos la contraseña con la que hemos instalado Mysql.



Para acceder a phpmyadmin debemos escribir en el navegador localhost/phpmyadmin. El nombre del usuario es **phpmyadmin** y la contraseña la que hemos elegido en el proceso de instalación.



1.7. Creación de un host virtual en Apache

Cuando usas un servidor web basado en Apache para alojar varios sitios, lo recomendable es crear uno o más “hosts virtuales” para organizar y proteger la configuración de cada sitio o dominio.

Ubuntu 18.04 con Apache y de manera predeterminada ya viene con un host habilitado por defecto, lo puedes localizar en “/var/www/html”. Esta listo para servir un sitio.

Como ejemplo vamos a crear un dominio asix-m09.edu para almacenar la información del curso.

- Creamos el directorio donde se almacena la información.

```
root@joan-ubuntu:~# mkdir -p /var/www/html/asix-m09.edu/html
```

- Asignamos la propiedad del directorio al usuario que entre en el sistema.

```
root@joan-ubuntu:~# chown -R $USER:$USER /var/www/html/asix-m09.edu/html
```

- Le damos permisos a la raíz del directorio.

```
root@joan-ubuntu:~# chmod -R 755 /var/www/html/asix-m09.edu
```

- Creamos el fichero index.html con un determinado código de prueba.

```
root@joan-ubuntu:~# nano /var/www/html/asix-m09.edu/html/index.html
```

```
<html>
<head>
  <title>Bienvenido a ASIX-M09</title>
</head>
<body>
  <h3>El Host Virtual asix-m09.edu funciona correctamente</h3>
</body>
</html>
```

- Para que apache pueda utilizar el nuevo sitio web debemos modificar la configuración del servidor. La configuración la encontramos en el fichero:

```
/etc/apache2/sites-available/000-default.conf
```

- Creamos una copia del fichero y modificamos los parámetros que nos interesan.

```
root@joan-ubuntu:~# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/asix-m09.edu.conf
```

- Editamos el fichero y modificamos las variables ServerAdmin, ServerName, ServerAlias y DocumentRoot.

```
ServerAdmin webmaster@asix-m09.edu
ServerName asix-m09.edu
ServerAlias www.asix-m09.edu
DocumentRoot /var/www/html/asix-m09.edu/html
```

- En el último paso desactivamos la configuración por defecto y activamos la nueva configuración de host virtual y reiniciamos el servidor.

```
root@joan-ubuntu:~# a2dissite 000-default.conf
```

```
root@joan-ubuntu:~# a2ensite asix-m09.edu.conf
```

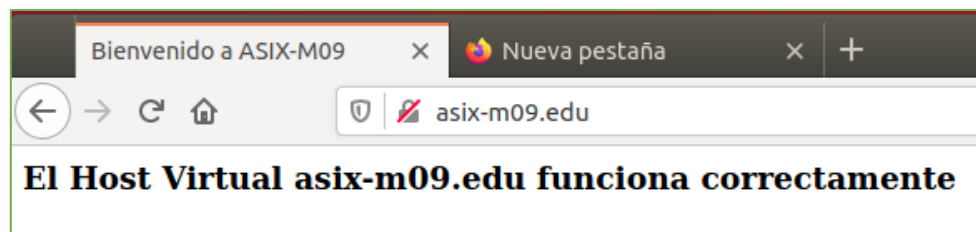
```
root@joan-ubuntu:~# systemctl restart apache2
```

- Editamos el fichero hosts para que identifique correctamente el dominio.

```
root@joan-ubuntu:~# nano /etc/hosts
```

- Añadimos una línea para direccionar la ip local al dominio.

```
127.0.0.1 localhost
127.0.1.1 joan-ubuntu
127.0.0.1 asix-m09.edu
```



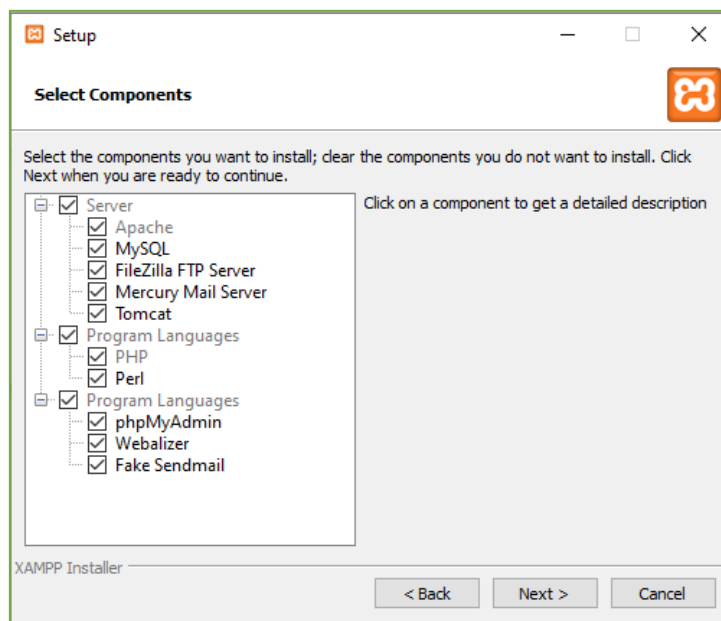
1.8. XAMPP

Es una herramienta que nos permite instalar de forma automática y muy simple todo lo que hemos realizado en el paso anterior. Xampp incorpora el servidor web Apache, la base de datos MariaDB, el servidor FTP Filezilla, el servidor de correo Mercury. Como lenguajes de programación PHP y Perl. El entorno phpmyadmin y otras utilidades que podemos escoger durante el proceso de instalación.

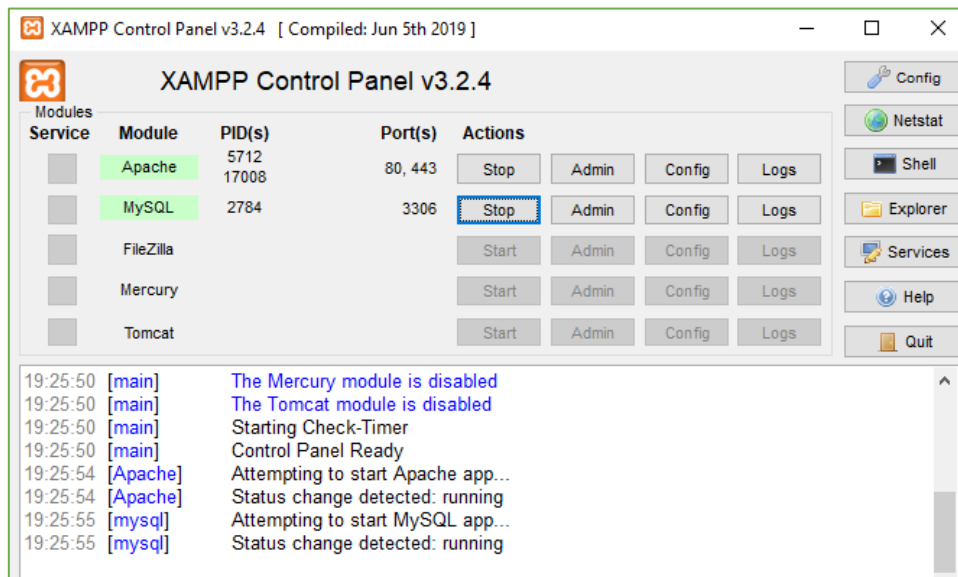
Existen versiones para Windows, Linux, OS X y Solaris (versión del 2011). Está disponible en la siguiente URL. <https://www.apachefriends.org/es/index.html>



La instalación en Windows se realiza por defecto en el directorio c:\xampp y la raíz del sitio web es c:\xampp\htdocs



Si en el proceso de instalación no hemos marcado como servicio las diferentes opciones, disponemos de un panel de control donde podemos arrancar y parar los diversos programas de forma manual.



[Video](#): instalación entorno xampp en Windows.

Recursos y enlaces

- [Documentación Apache](#)



- [Estadística de uso de servidores web](#)



- [Descargar XAMPP](#)



Conceptos clave

- **Arquitectura de tres capas:** Compuesta por uno o varios clientes, un servidor de aplicaciones (programación) y un servidor de datos.
- **Servidor Web:** Atiende las peticiones de los clientes (navegadores) y devuelve la respuesta normalmente en forma de página web.
- **LAMP:** Paquete formado por un sistema operativo Linux, un servidor web apache, una base de datos MySql y un lenguaje de programación PHP.
- **Nginx:** Servidor web más utilizado en la actualidad.

Ponlo en práctica

Actividad 1

Crea en apache un host virtual para almacenar los ejercicios de clase, el host virtual se llama asix-ejer.edu.