

## A1\_T1. Descripció de components: Servidor web, SGBD i aplicacions

### Características y alternativas

Una aplicación web que reside en uno o varios servidores normalmente está estructurada en tres capas: servidor web, lógica de la aplicación y un sistema gestor de bases de datos (en adelante SGBD). El primero de ellos mantiene la comunicación con el navegador del usuario que es quién proporciona la interfaz de acceso al servicio web. La lógica de la aplicación está formada por los scripts o programas que se ejecutan en el servidor para atender las solicitudes del usuario, y el SGBD es el repositorio en el que se guardan los datos con los que trabaja el servicio. Las tres capas pueden estar montadas sobre una o varias máquinas dependiendo del volumen de carga que tenga que soportar el servicio, y todo corriendo sobre algún sistema operativo como GNU/Linux.

Aunque existen muchas alternativas, una de las más frecuentes está compuesta por el servidor web Apache, PHP como intérprete para la lógica de la aplicación y MySQL como SGBD, todo ello corriendo sobre GNU/Linux. A esta combinación se la llama LAMP para abreviar.



*Pila LAMP: **L**inux, **A**pache, **M**ySQL y **P**HP (o **P**erl o **P**ython)*

## Servidor web

El **servidor web** se encarga la comunicación a través de la red con el navegador del usuario. Normalmente escucha en el puerto **TCP 80** cuando se trata de una conexión **HTTP**, y en el **TCP 443** cuando se trata de una **HTTPS**, aunque este parámetro es configurable.

Cuando recibe una petición del usuario puede atenderla de manera estática cuando contesta con un fichero que está en el sistema de archivos (la descarga de una imagen, un fichero css o html) o de manera dinámica cuando se ejecuta algún programa y se envía al cliente el resultado de dicha ejecución.

El servidor web evidentemente es capaz de atender muchas peticiones de forma concurrente pudiendo realizar diferentes funciones habituales entre las que se encuentran:

- Registro de actividad y errores
- Control de acceso basado en la dirección del cliente, contenido o usuario/contraseña
- Virtual Hosts, para mantener diferentes webs (por ejemplo `http://elpuig.xeill.net` y `http://blog.elpuig.xeill.net`)
- Proxy, para reenviar las peticiones a otro servidor
- Reescritura de URLs
- Alias o mapeados de rutas

Aunque existen diferentes alternativas Apache es el servidor web más utilizado en Internet. Se trata de una aplicación libre robusta y modular con una impresionante colección de funciones, pudiendo servir tanto para contenido estático como para prácticamente cualquier contenido dinámico.

Otras alternativas pueden ser:

### Cherokee

Un servidor web más moderno que Apache con la pretensión de ser eficiente y fácil de configurar (tiene una interfaz web de administración). Está bajo la GPL.

### NGINX

Orientado a sitios con una gran carga de trabajo en los que es necesario un gran rendimiento.

### Tomcat

Contenedor de servlets para aplicaciones web escritas en Java.

### Zope

Servidor de aplicaciones escrito en Python.

## Capa de aplicaciones

En cualquier página web dinámica se ejecutan programas que determinan qué se debe responder a las solicitudes del usuario. Estos programas pueden estar escritos en una gran variedad de lenguajes de programación, una posible clasificación es: lenguajes compilados y lenguajes interpretados o scripts.

Los primeros obligan a compilar el código fuente antes de obtener un ejecutable, en líneas generales priman el rendimiento en tiempo de ejecución y cuando una aplicación se debe utilizar en diferentes plataformas debe ser compilada para cada una de ellas. Ejemplos típicos son los lenguajes C, C++ y Java.

Los lenguajes interpretados, o scripts, no necesitan ser compilados pues un programa llamado intérprete realiza las operaciones indicadas por el código fuente. En líneas generales permiten un desarrollo más rápido aunque no alcanzan en velocidad de ejecución a los lenguajes compilados. Ejemplos típicos son Bash, Perl, PHP, Python y Ruby.

## **SGBD - Sistema Gestor de Bases de Datos**

Una aplicación sencilla puede trabajar únicamente con ficheros, pero en cuanto el servicio deja de ser trivial aparece la necesidad de utilizar un SGBD que organice la información con la que se trabaja.

La historia de las bases de datos es larga, pero en cuanto a lo que una introducción a aplicaciones web se refiere se puede resumir en diferentes modelos:

### SGBD relacional:

Las más extendidas y con una larga vida por delante. Guardan la información en registros de tablas y mediante SQL se realizan consultas u operaciones para manipular los datos. Dependiendo de la herramienta utilizada hay grandes diferencias en cuanto a sus capacidades y administración, pero en general, para las aplicaciones web prima la velocidad sobre las características.

Ejemplos de uso: MySQL y PostgreSQL

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A1: INSTAL·LACIÓ DE SERVIDORS D'APLICACIONS WEB



#### SGBD orientado a objetos:

En lugar de almacenar registros directamente guardan y recuperan objetos. De manera que cuando se utiliza un lenguaje de programación orientado a objetos nos ahorramos el trabajo de adaptar las entidades con las que trabajamos al modelo relacional. Ejemplos de uso: DB4O y Zope Object Database

#### SGBD NoSQL:

Aquí se agrupan un conjunto de técnicas diferentes que tienen en común que no cumplen con los requisitos ACID y/o no estructuran la información en tablas como las bases de datos relacionales. La ventaja es que consiguen romper algunas de las limitaciones de los sistemas relacionales en cuanto a escalabilidad y rendimiento permitiendo crear ingentes bases de datos distribuidas.

Ejemplos de uso: Apache Cassandra y Apache CouchDB

#### **Enlaces de interés:**

[Informe de Netcraft sobre el uso de servidores web](#)