



Implantación de arquitecturas web

Tema 1
Despliegue de aplicaciones web
Curso 2024-25

Índice

Introducción de conceptos:

- Tecnologías
- Tipos
- Modelos

Servidor web Apache

- Principales características
- Instalación
- Configuración básica

Comparativa entre servidor y servidor de app web

- Tomcat

Estructura y despliegue de app web

- Archivos WAR
- Descriptor del despliegue

1. Aspectos generales de arquitecturas web



Arquitectura WWW

O

World Wide Web

- Modelo de programación poderoso y flexible.
- Las aplicaciones y los contenidos:
 - formatos de datos estándar.
 - Localizados por web browsers.
- Permite:
 - usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros.
 - desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

Estándares WWW

Especifican muchos de los mecanismos necesarios para una app de propósito general.

Modelo estándar de nombres:

- Todos los servidores, así como el contenido de la WWW se denominan según un Localizador Uniforme de Recursos (**Uniform Resource Locator: URL**).

Contenido:

- Todos los contenidos en la WWW **se les especifica un determinado tipo** permitiendo de esta forma que los browsers (navegadores) los interpreten correctamente.

Formatos de contenidos estándar:

- Todos los navegadores soportan un conjunto de formatos estándar, por ejemplo **HTML, ECMA, JavaScript, etc.**

Protocolos estándar:

- Permiten que cualquier navegador pueda comunicarse con cualquier servidor web.
- El más común es **HTML** (Protocolo de Transporte de HiperTexto), que opera sobre el conjunto de protocolos TCP/IP.

Aspectos generales de una arquitectura web

Escalabilidad

Separación de responsabilidades

Portabilidad

Utilización de componentes en los servicios de infraestructura

Gestión de las sesiones del usuario

Aplicación de patrones de diseño

Modelo compuesto de tres capas:

Capa de Base de Datos

- Documentación de la información que se pretende administrar
- Empleando una plataforma del tipo MySQL, PostgreSQL, etc.

Servidores de aplicaciones web

- Ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.

Clients del servicio web

- Acceden mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

Elementos necesarios de los servicios web

Proveedor del servicio web

- Lo diseña
- Lo desarrolla e implementa
- Lo pone disponible para su uso, dentro de la misma organización o en público.

Consumidor del servicio

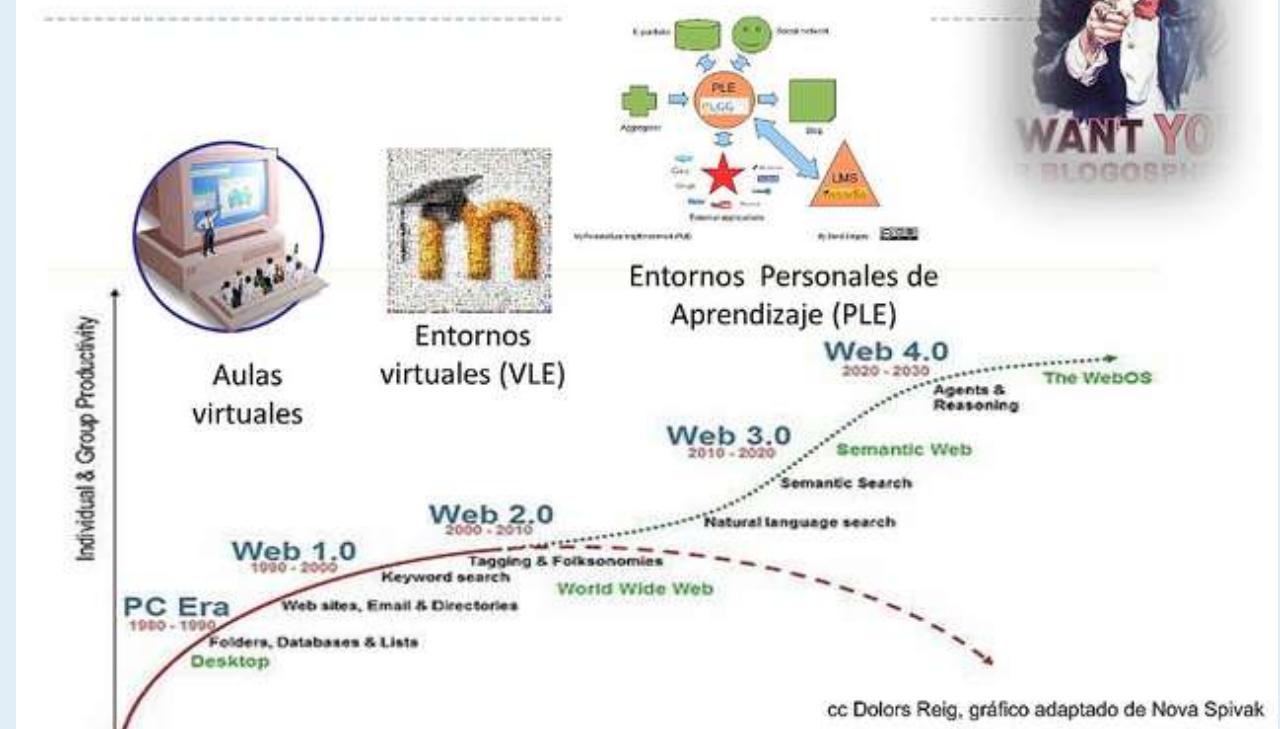
- Quien accede para utilizar los servicios

Agente del servicio

- Enlace entre proveedor y consumidor para:
 - Publicación
 - Búsqueda
 - Localización del servicio.

1.1.- Evolución de los servicios web

Web – Web Social – Web Personal



¿Cuál es el origen de internet?

¿Habéis oído hablar alguna vez
de la web 1.0, 2.0, 3.0 o 4.0?



Evolución de la web

- https://youtu.be/hi-J6PjhX1I?si=F_gjUX-SBsCfoJ3



Factores que han impulsado el uso de los servicios web

El contenido se está volviendo más dinámico:

El ancho de banda es menos costoso:

El almacenamiento es más barato:

- Un Servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.

El éxito de la computación extendida se está volviendo más importante:

1.2.- Tecnologías asociadas a las aplicaciones web



Páginas dinámicas

Se ejecutan en un servidor web

Se muestran en el navegador de un equipo cliente

- que es el que ha realizado previamente la solicitud.

Cuando una página web llega al navegador

- posible que también incluya algún programa o fragmento de código que se deba ejecutar.
- Ese código, normalmente JavaScript, lo ejecutará el propio navegador.

ASP
(Active Server
Pages)

CGI
(Common Gateway
Interface)

CSS
(Cascading Style
Sheets)

Java

JavaScript

PHP (Hypertext
Preprocessor)

VBScript (Visual
Basic Scripting)

¿Cuáles de las anteriores tecnologías se utilizan en el lado del cliente y cuales en el lado del servidor?

¿Cuáles de las anteriores tecnologías se utilizan en el lado del cliente y cuales en el lado del servidor?

Cliente

- HTML
- CSS
- Java
- JavaScript
- VBScript

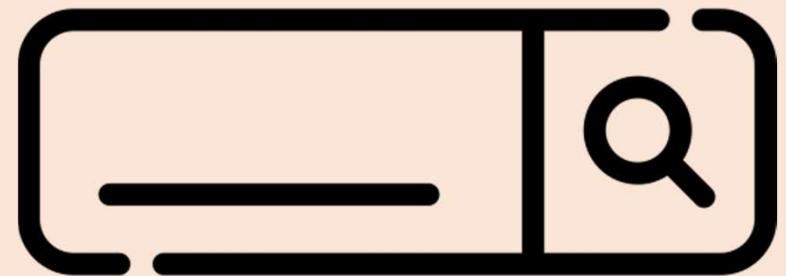
Servidor

- CGI
- ASP
- PHP
- Java
- JavaScript

ASP (Active Server Pages) o “Páginas activas”

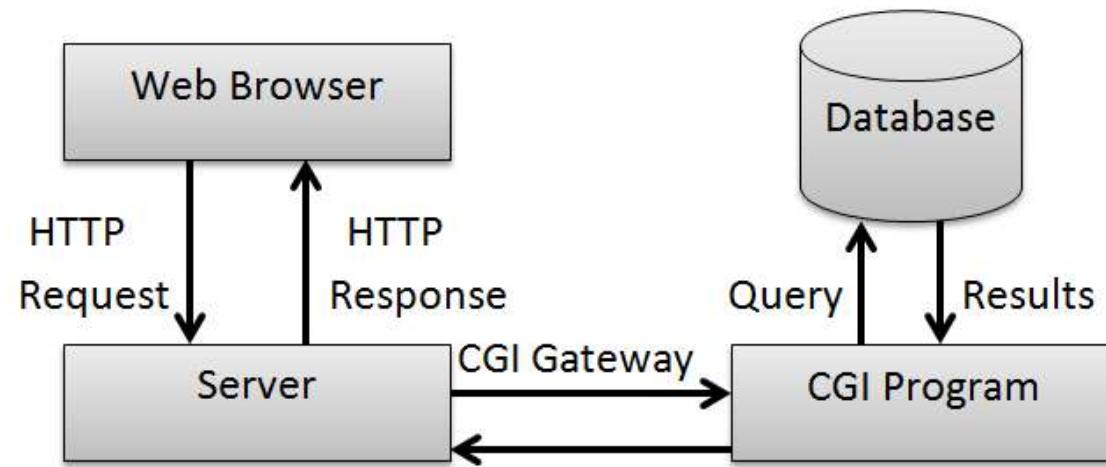
- Tecnología desarrollada por **Microsoft** para crear **páginas web dinámicas** junto con el servidor IIS, pero también existen versiones para Unix y Linux.
- Se ejecutan del **lado del servidor**.
 - Se forman los resultados que luego se mostrarán en el navegador de cada equipo cliente que ha realizado la solicitud.

¿Ejemplo?



CGI (Common Gateway Interface) o Interfaz Común de Entrada

- Uno de los estándares más antiguos en Internet para **trasladar información desde una página a un servidor web**.
- Utilizado para bases de datos, motores de búsqueda, formularios, generadores de email automático, foros, comercio electrónico, rotadores y mapas de imágenes, juegos en línea, etc.
- Normalmente en C o PERL.





```
bottom: 0;  
width: 65%;  
padding-left: 12.5px;  
padding-top: 100px;  
padding-bottom: 0;
```

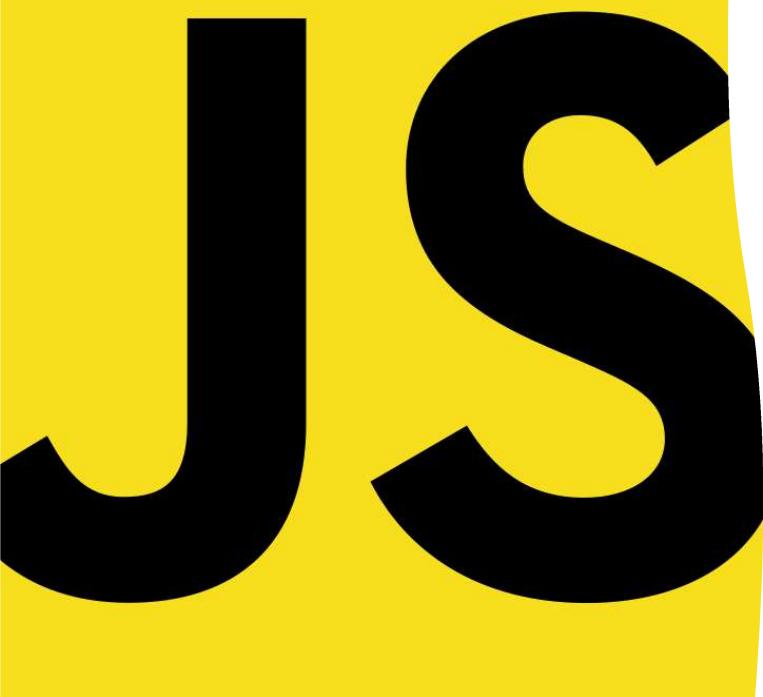
CSS (Cascading Style Sheets) u “Hojas de Estilo en Cascada”

- Para **formatear las páginas web**
- Sepa el contenido de un documento, de su presentación.
- Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS.

- Desarrollado por la empresa Sun Microsystems, liberado bajo licencia GNU GPL.
- Este es un lenguaje que **trabaja en el cliente**.
- Lenguaje eficiente y muy poderoso, caracterizado por:
 - Una misma aplicación puede funcionar multiplataforma.
- Los programas Java pueden ser **aplicaciones independientes** (que corren en una ventana propia) o "applets".
- Lenguaje "**orientado a objetos**".



JavaScript



- Lenguaje que se **interpreta y se ejecuta en el cliente.**
- Útil para realizar tareas como mover imágenes por la pantalla, crear menús de navegación interactivos, utilizar algunos juegos, etc.
- En las páginas web suele preferirse JavaScript porque es aceptado por muchos más navegadores que VBScript.



PHP (Hypertext Preprocessor)

- Ejecutado en el **lado del servidor**.
- **Similar a ASP** y puede ser usado en circunstancias similares.
- Es muy eficiente, permitiendo el acceso a bases de datos empleando servidores como MySQL.
- Suele utilizarse para crear **páginas dinámicas complejas**.



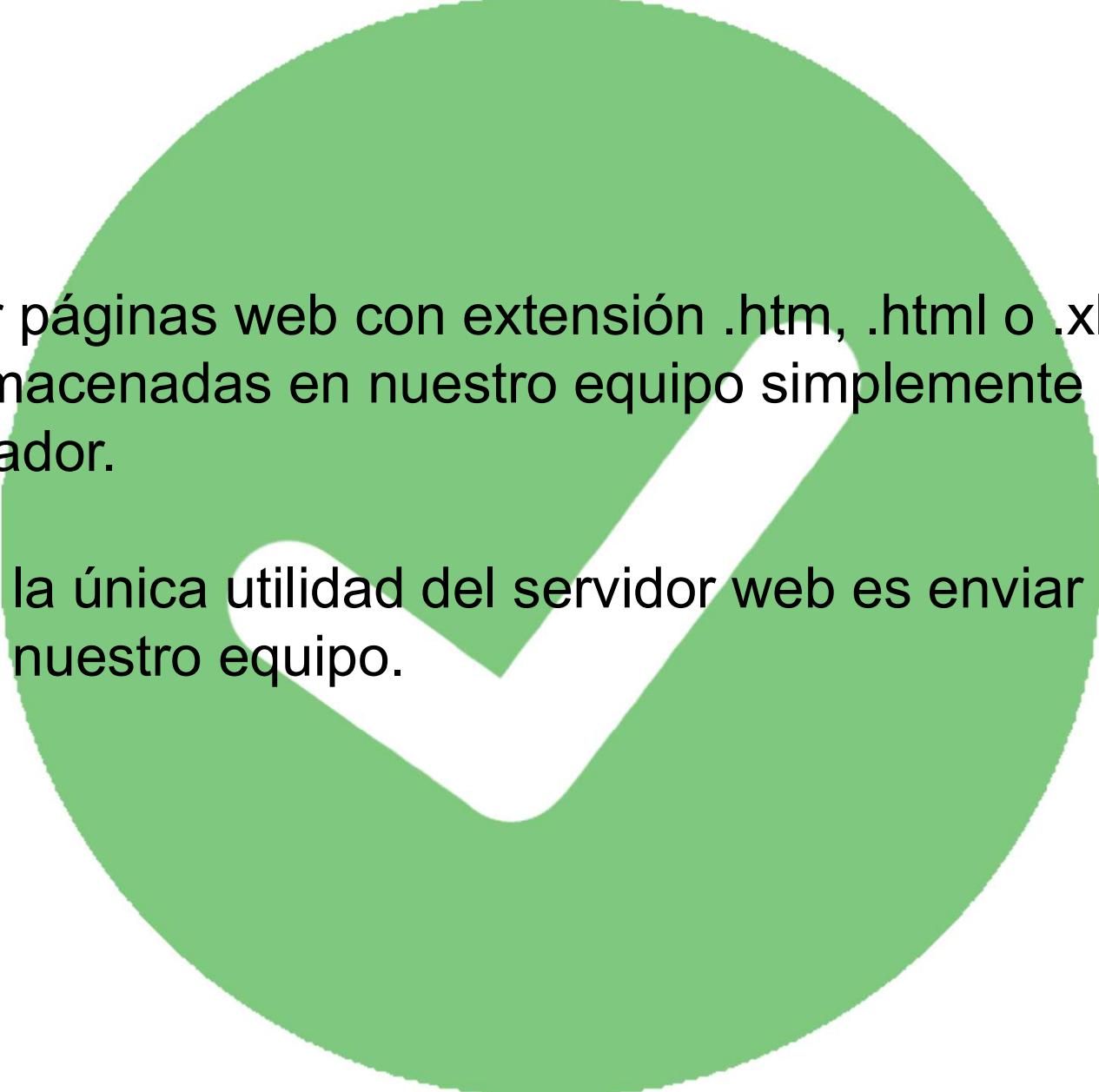
VBScript (Visual Basic Scripting)

- La respuesta de Microsoft a JavaScript.
- **Buena** herramienta para cualquier sitio destinado a ser mostrado **exclusivamente** en el **navegador Microsoft Internet Explorer**.
- El código en VBScript puede, además, estar diseñado para su **ejecución en el lado del cliente o en el del servidor**, la diferencia es que un código que se ejecuta en el lado del servidor no es visible en el lado del cliente. Éste recibe los resultados, pero no el código.

¿Podemos ver una página web sin que intervenga un servidor web?

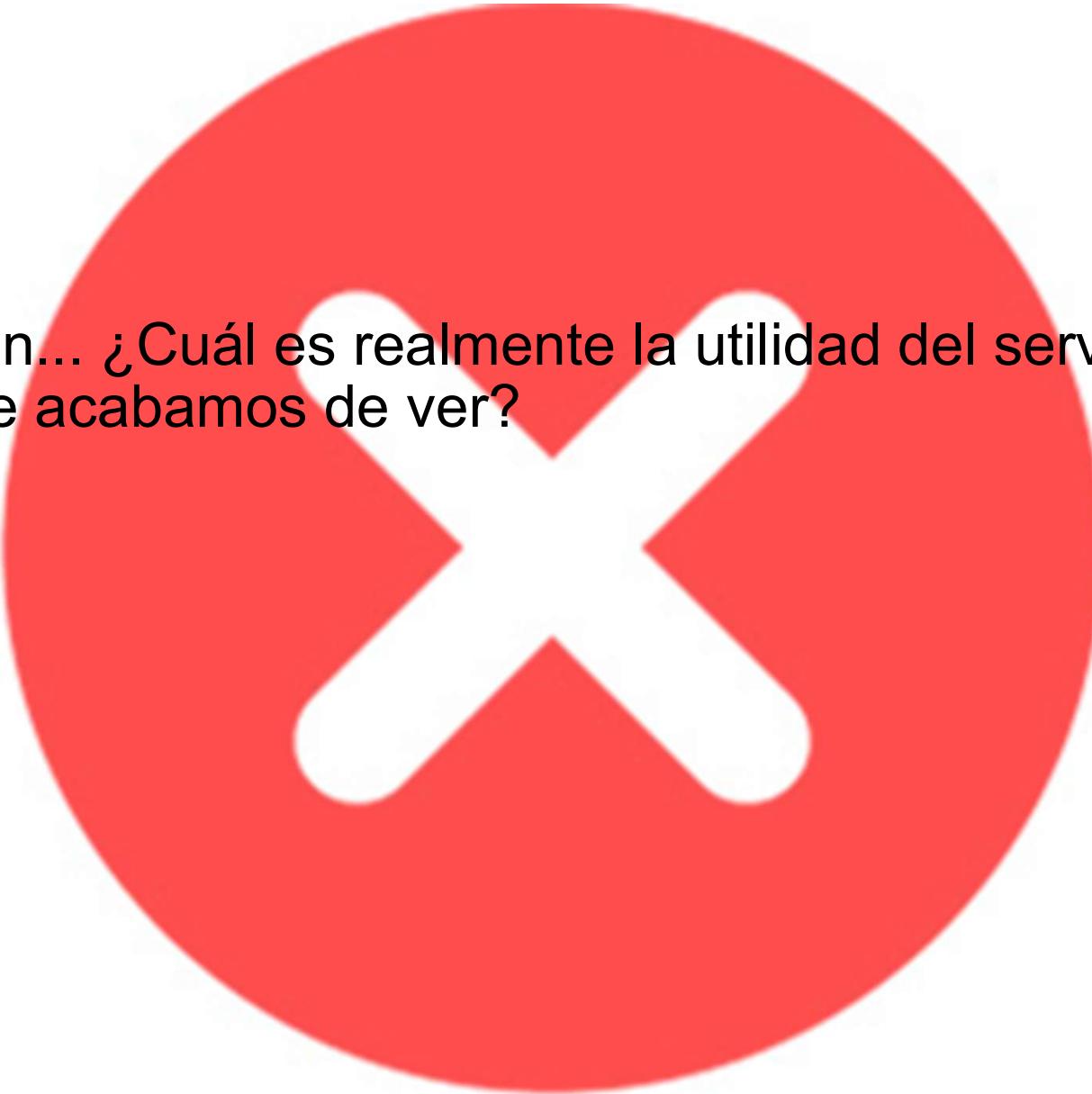
SI

NO



Podemos ver páginas web con extensión .htm, .html o .xhtml que tengamos almacenadas en nuestro equipo simplemente abriéndolas con el navegador.

En este caso la única utilidad del servidor web es enviar la página que solicitemos a nuestro equipo.



Piénsalo bien... ¿Cuál es realmente la utilidad del servidor web, según lo que acabamos de ver?

1.3.- Tipos de aplicaciones web

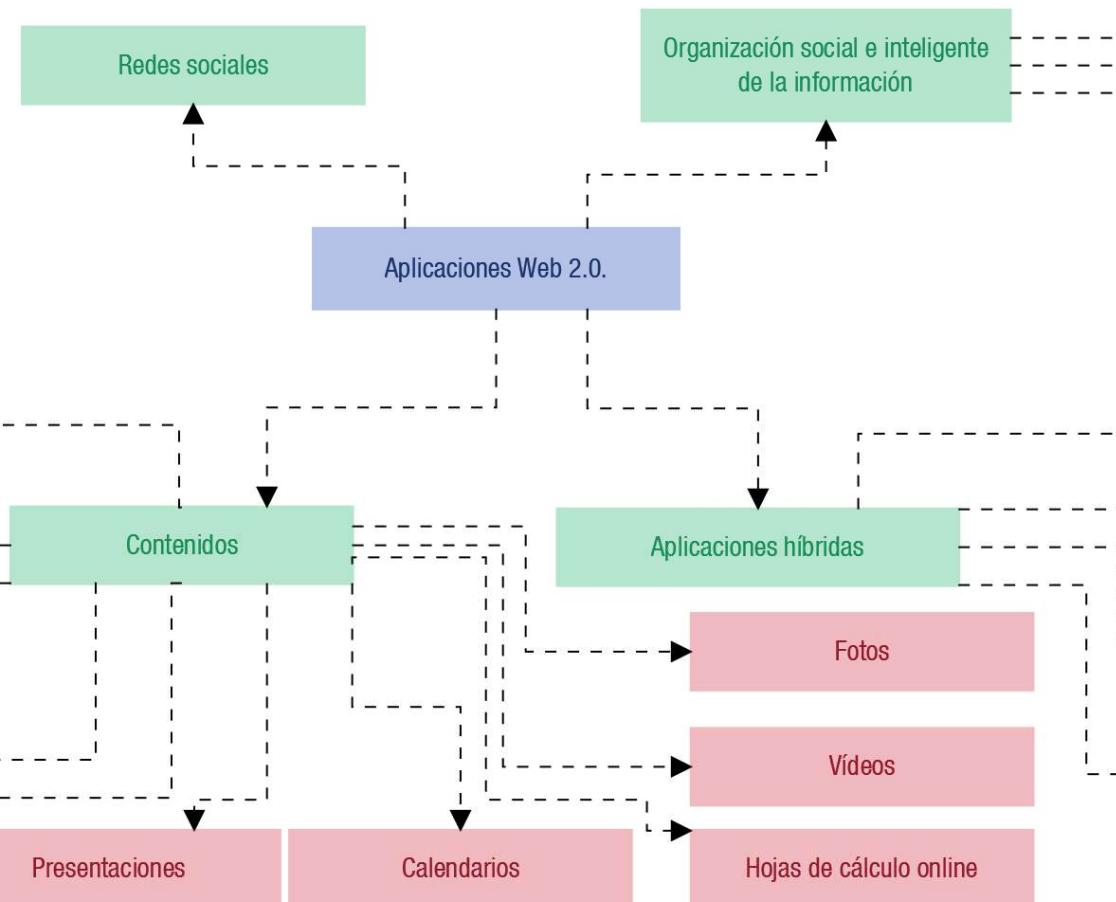


ESQUEMA DE LAS APLICACIONES WEB

Definición

Una aplicación web es una **plataforma orientada a automatizar los procesos de servicios que se quieran ofrecer a usuarios.**

Cualquier proyecto que se quiera desarrollar en Internet, conlleva el desarrollo de una aplicación web.



Clasificación según como se presenta la app web

Página web estática

Página web
dinámica

Página web animada

Portal

Tienda virtual
o
comercio electrónico

Página web con
"Gestor de
Contenidos

W Wikipedia, la enciclopedia libre × +

es.wikipedia.org/wiki/Wikipedia:Portada

Buscar en Wikipedia Buscar

Crear una cuenta Acceder ...

Wiki Loves Monuments: ¡Fotografía un monumento, ayuda a Wikipedia y gana!

Más información X

Portada Discusión Leer Ver código fuente Ver historial Herramientas ▾

1 894 525 artículos en español.

Página web Estática

Bienvenidos a Wikipedia,

la enciclopedia de contenido libre que todos pueden editar.

- Implementadas en HTML y pueden mostrar en alguna parte de la página objetos en movimiento tales como banners, GIF animados, vídeos, etc.

Artículo destacado

Ptyonoprogne rupestris

El avión roquero (*Ptyonoprogne rupestris*) es una especie de ave de la familia de los aviones y golondrinas (Hirundinidae). Mide 14 cm de largo y posee partes superiores color marrón ceniza y partes inferiores más claras, además de una cola corta y cuadrada que posee unas manchas blancas distintivas en la mayoría de sus plumas. Habita en las montañas del sur de Europa, el noreste de África y el sur de



Un dibujo de *P. rupestris* que data de 1905.

Actualidad

- Invasión rusa de Ucrania
- Enfrentamientos entre Azerbaiyán y Artsaj
- Crisis de Níger
- Tercera guerra civil sudanesa
- 25 de septiembre-1 de octubre: Abierto Británico de snooker
- 24 de septiembre: Maratón de Berlín
- 24 de septiembre: Gran Premio de la India de motociclismo
- 24 de septiembre: Gran Premio de Japón de Fórmula 1
- 23-24 de septiembre: Campeonato Mundial de Triatlón
- 22-30 de septiembre: Festival de Cine de San Sebastián

Página web Dinámica

- Existen muchos lenguajes de programación (Destacados PHP y ASP), que permiten una perfecta estructuración del contenido
- Gestionar el contenido a través de un panel de control

RED DE WEBS



Regístrate o INICIAR SESIÓN

Vaya GIF



ÚLTIMOS

TOP GIFS

GIFS ALEATORIOS

ENVIAR UN GIF

MODERAR GIFS

0

Espera aquí un momento, ¿por qué?

Publicado por michaelbuble el 24 sep 2023, 13:15

Página web Animada

- Se realizan con la tecnología FLASH; ésta permite que una página web presente el contenido con ciertos efectos animados continuados.
- Está descontinuada y desaconsejada ya que los navegadores la bloquean por su gran cantidad de vulnerabilidades.



Etiquetas: niñas, sótano, secuestro, mala gente

14 (16 votos)

Twitter

Compartir

Reportar por inadecuado o fuente incorrecta

0

Beverly Hills está cerrando todas sus tiendas, qué es lo que nos espera

Publicado por chuckbass el 24 sep 2023, 11:15

Búsqueda

BUSCAR

El mejor GIF de ayer



Carpeta Ciudadana

Portal

- Es un sitio web que en su página principal permite el acceso a múltiples secciones que, por lo general, son foros, chats, cuentas de correo, buscador, acceso registrado para obtener ciertas ventajas, las últimas noticias de actualidad, etc.



Tienda virtual o comercio electrónico

Solo para clientes Prime

- Sitio web que publica los productos de una tienda en Internet. Permite la compra on-line a través de tarjeta de crédito, domiciliación bancaria o transferencia bancaria en general.
- Ofrece al administrador un panel de gestión para poder subir los productos, actualizarlos, eliminarlos, etc.

Ofertas flash Ayuda a los demás Los más vendidos en Electrónica Identifícate para una mejor experiencia Inicia sesión de manera segura

The screenshot shows the top navigation bar of the Amazon.es website. It includes the logo, language selection (ES), sign-in options, and a shopping cart icon. Below the header, there's a main banner for the 'Fiesta de Ofertas Prime' (Prime Sale) with a blue background, gold confetti, and a large brown cardboard box. The text 'Fiesta de Ofertas Prime' is prominently displayed in white. Below the banner, there's a sub-section titled 'Tienda virtual o comercio electrónico' with a sub-note 'Solo para clientes Prime'. To the left of this section is a graphic of a red price tag with a percentage symbol. To the right are three smaller images: one showing two hands clasped, another showing various electronic devices like a tablet, smartphone, and headphones, and a third showing a yellow button labeled 'Inicia sesión de manera segura' (Sign in securely). The URL at the bottom is https://www.amazon.es/gp/bestsellers/electronics/?ie=UTF8&pd_rd_w=VtURO&content_id=amzn1.sym.16096117-fc35-4dec-96f6-4f7289176bff&pf_rd_p=16096117-fc35-4dec-96f6-4f7289176bff&pf_rd_r=YT5GN6FZ4H4PHJ8V5QBA&pd_rd_wg=vi8Me&pd_rd_r=383bc0e5-88ec-49...

Escritorio

Inicio

Actualizaciones

Entradas

Medios

Páginas

Comentarios

Apariencia

Plugins

Usuarios

Herramientas

Ajustes

Cerrar menú

¡Bienvenido a WordPress!

Hemos recopilado algunos enlaces para que puedas comenzar:

Comienza

Siguientes pasos

Escribe tu primera entrada en el b

Añade una página «Acerca de»

Establece tu página de inicio

Ver tu sitio

Página web con "Gestor de Contenidos".

- Se trata de un sitio web cuyo contenido se actualiza a través de un panel de gestión por parte del administrador del sitio.
- Este panel de gestión suele ser muy intuitivo y fácil de usar.
- En aquellas páginas web que requieran una actualización constante, se suele incorporar este panel de gestión para que la web pueda controlarse día a día por parte del cliente.

De un vistazo

1 entrada

1 página

1 comentario

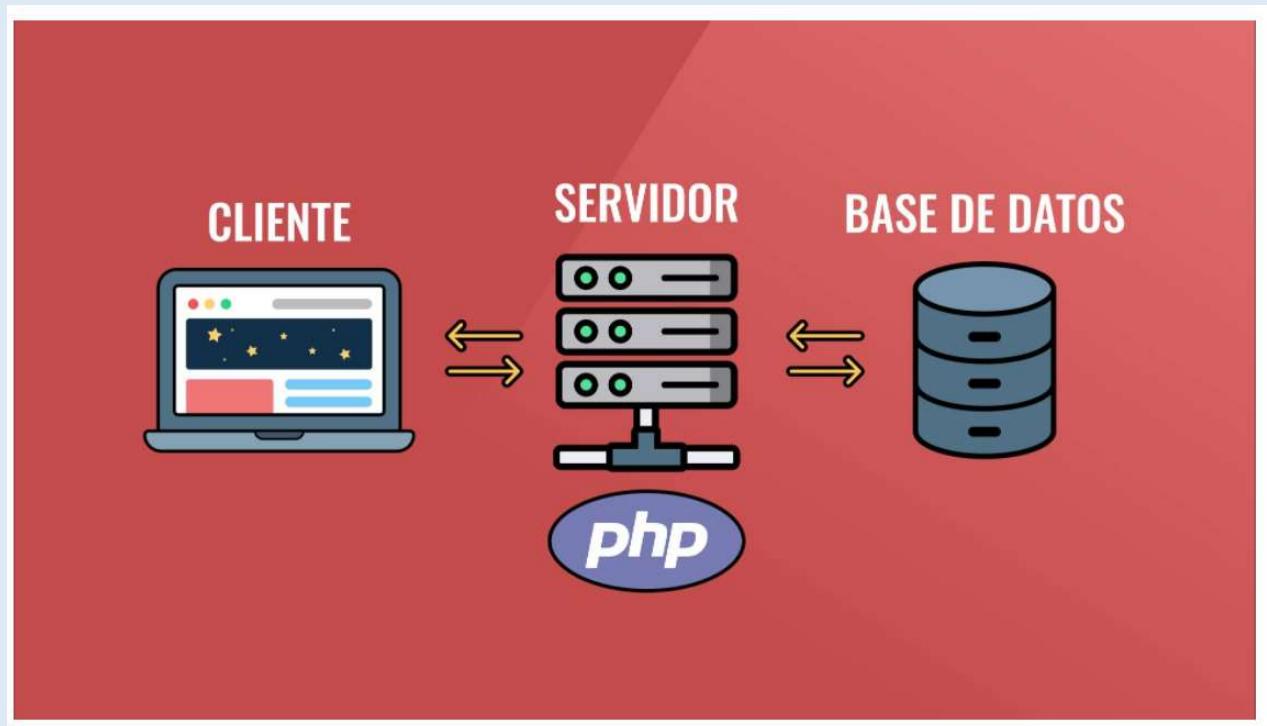
Borrador rápido

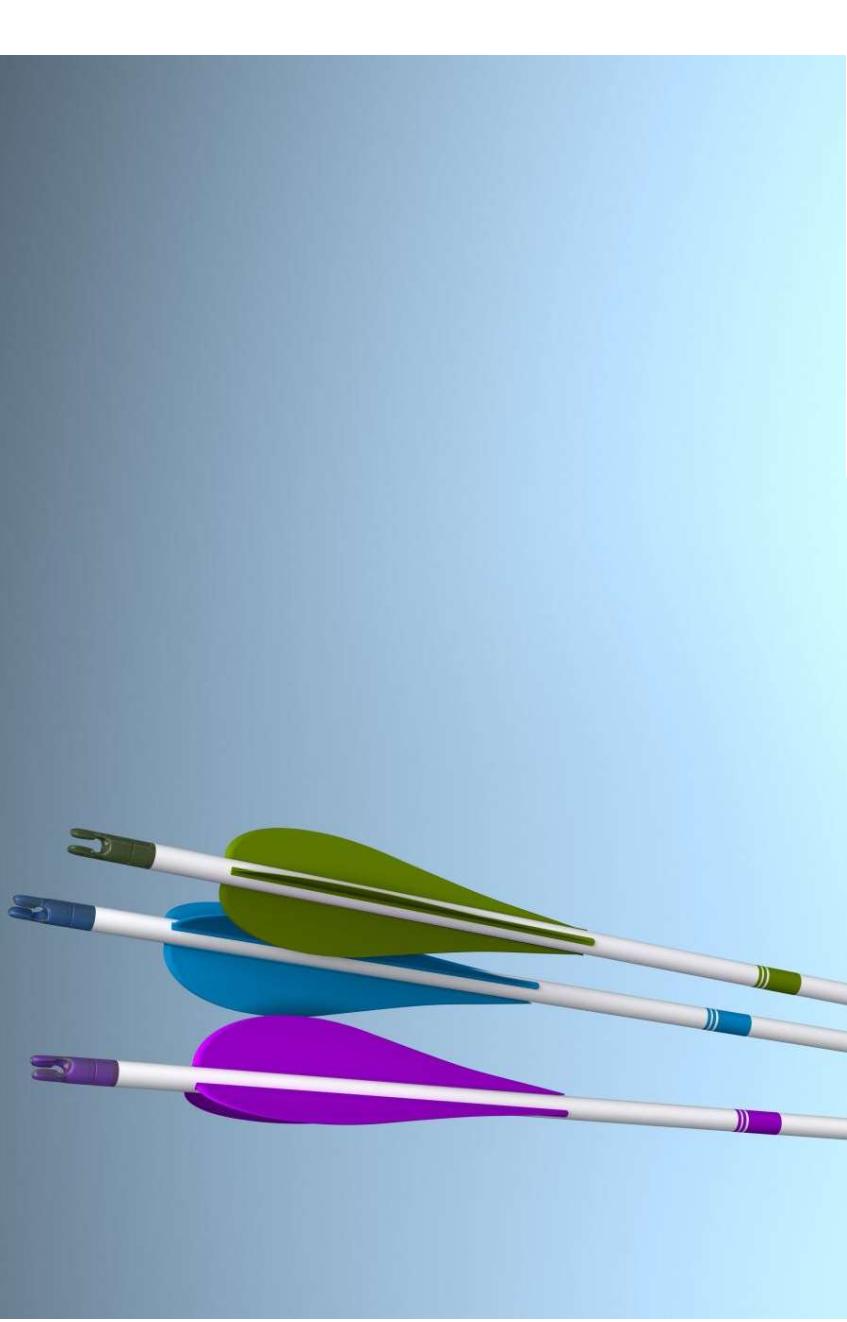
Título

Contenido

WordPress 5.3.2 está funcionando con el tema [Twenty Twenty](#).

1.4.- Arquitecturas web. Modelos

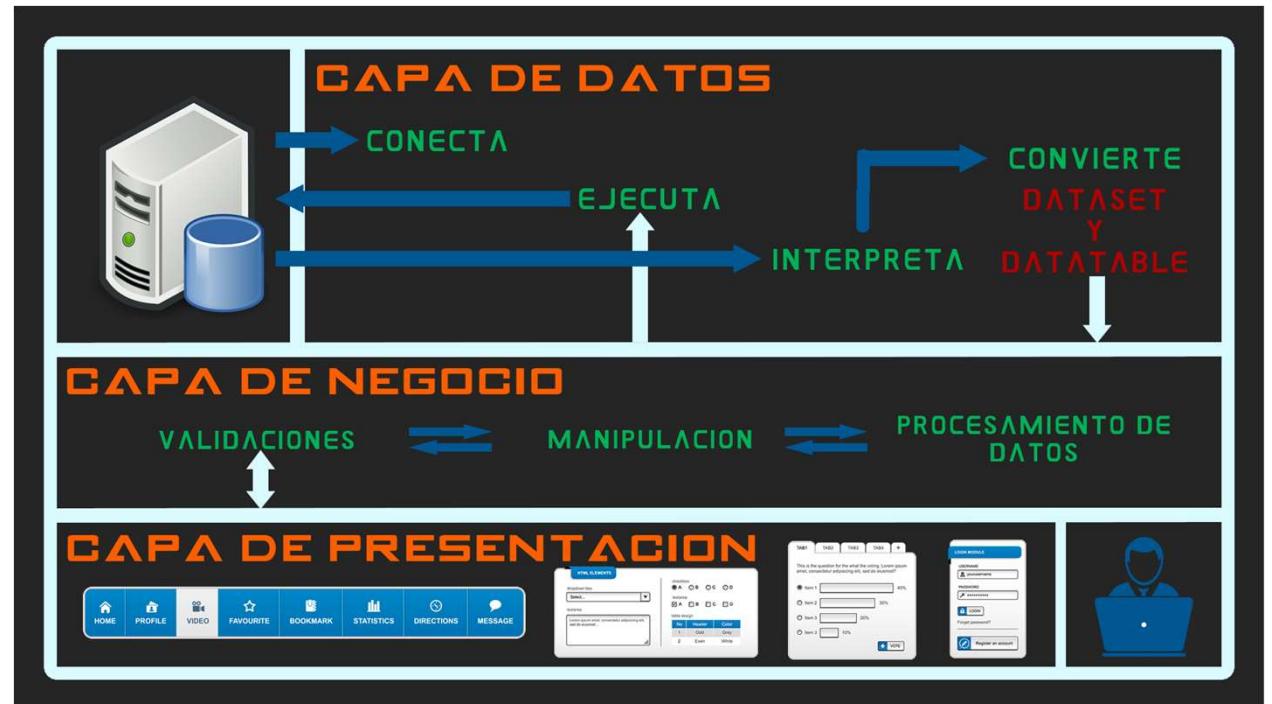




Objetivo

- Servir de ayuda a los usuarios a encontrar y manejar la información.

Modelos de arquitectura web



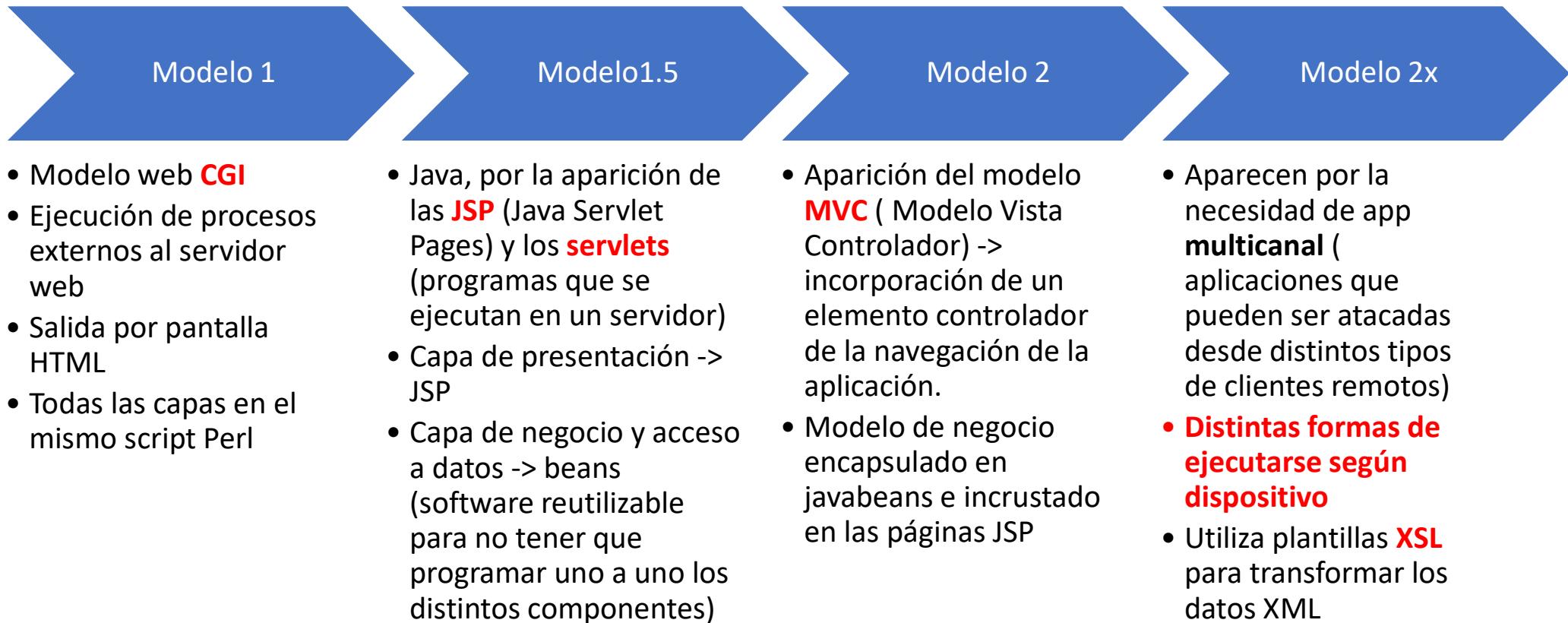


En función de cómo implementen cada una de estas capas vamos a poder hacer una división de los distintos modelos.



Con la evolución de las arquitecturas de las aplicaciones web, estos modelos han ido evolucionando.

Evolución de los modelos



1.5.- Plataformas web libres y propietarias



Definición de plataforma

- Entorno de desarrollo de software empleado para **diseñar y ejecutar un sitio web**

Componentes básicos



Sistema operativo



Servidor web



Gestor de bases de datos



Lenguaje de programación interpretado

Sistema operativo



Sobre el que trabaja el equipo donde se encuentran las páginas web



En ocasiones limita la elección de otros componentes

Servidor web

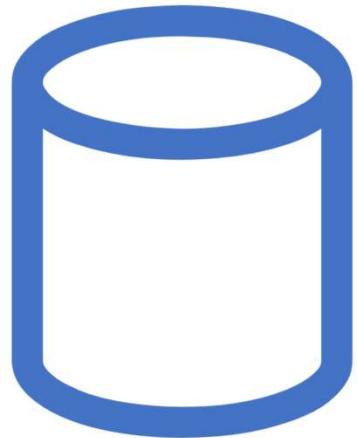
Software que maneja las peticiones desde otros equipos remotos a través de internet.

Páginas estáticas

- el servidor web simplemente provee el archivo solicitado, el cual se muestra en el navegador.

Páginas dinámicas

- el servidor web se encarga de pasar las solicitudes a otros programas que puedan gestionarlas adecuadamente.



+

.

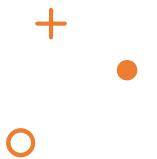
o

Gestor de bases de datos

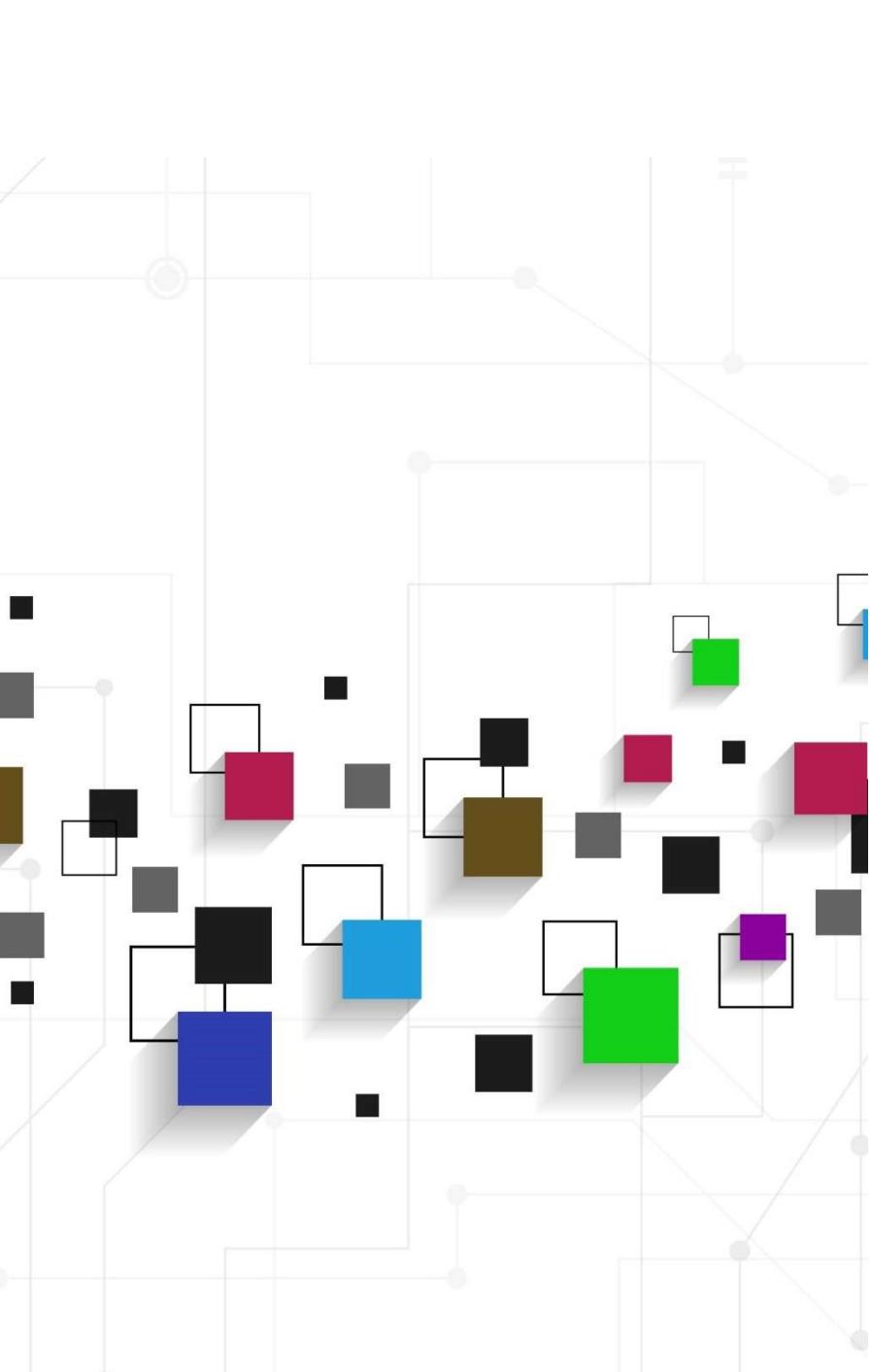
- Almacenar sistemáticamente un conjunto de registros de datos relacionados para ser usados posteriormente.



Un lenguaje de programación interpretado



- Controla las aplicaciones de software que corren en el sitio web.

- 
- Diferentes combinaciones de los cuatro componentes, basadas en las distintas opciones de software disponibles en el mercado, dan lugar a numerosas plataformas web.
 - Más importantes: LAMP y WISA.

LAMP

- **Software libre** y no está sujeta a restricciones propietarias
- Linux: Sistema operativo.
- Apache: Servidor web.
- MySQL: Gestor de bases de datos.
- PHP: Lenguaje interpretado PHP, aunque a veces Perl o Python.

WISA

- Desarrollado por Microsoft; **software propietario**
- Windows: Sistema operativo.
- Internet Information Services: servidor web.
- SQL Server: gestor de bases de datos.
- ASP o ASP.NET como lenguaje para scripting del lado del servidor.

Otras plataformas

Windows-Apache-MySQL-PHP (**WAMP**). Es bastante común pero sólo como **plataforma de desarrollo local**.

Servidor Windows con MySQL y PHP (**WIMP**).

Existen muchas otras plataformas.

1.4.- Escalabilidad



- En el entorno en que se ubican las aplicaciones web, uno de los principales **factores que puede afectar al rendimiento es el número de usuarios.**

- El éxito o el fracaso de un sitio web orientado al usuario común vendrá determinado, entre otros aspectos, por el dimensionamiento del sistema sobre el que se instala y soporta el software que sustenta dicho sitio.

Tipos



Verticalmente: de manera ascendente "upgrades" a cada nodo.

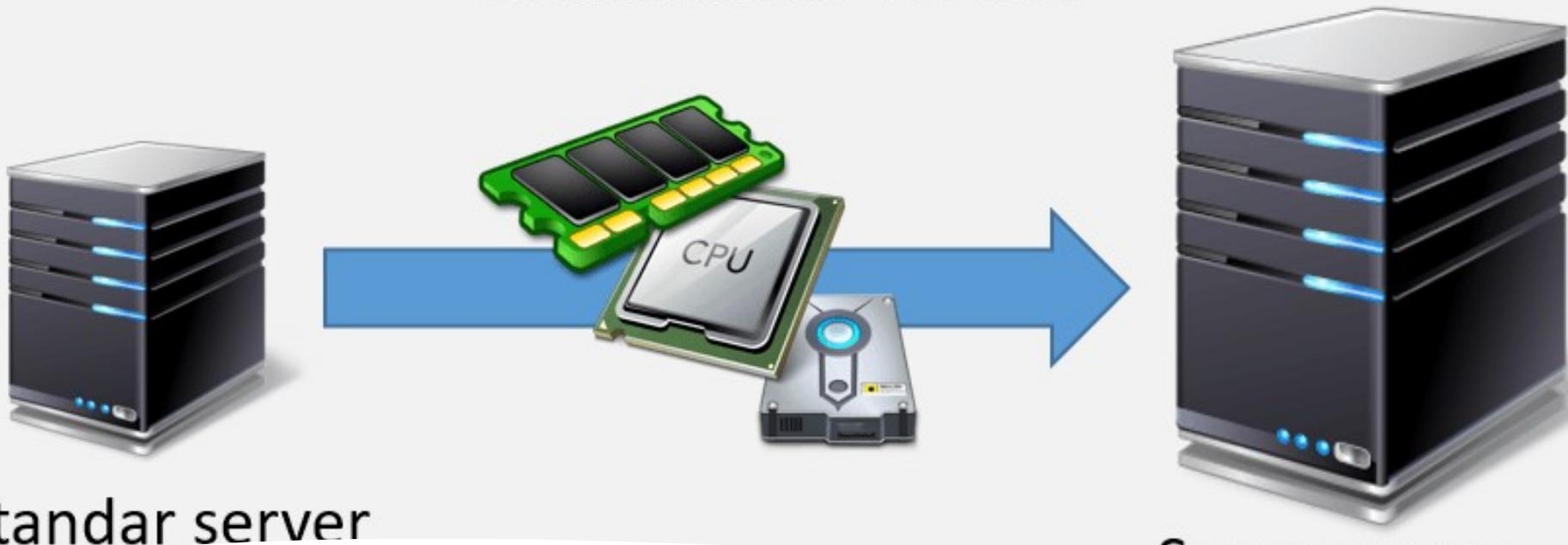


Horizontalmente: consiste en aumentar el número de nodos.



Cluster: consiste en crear agrupaciones de servidores.

Escalamiento Vertical

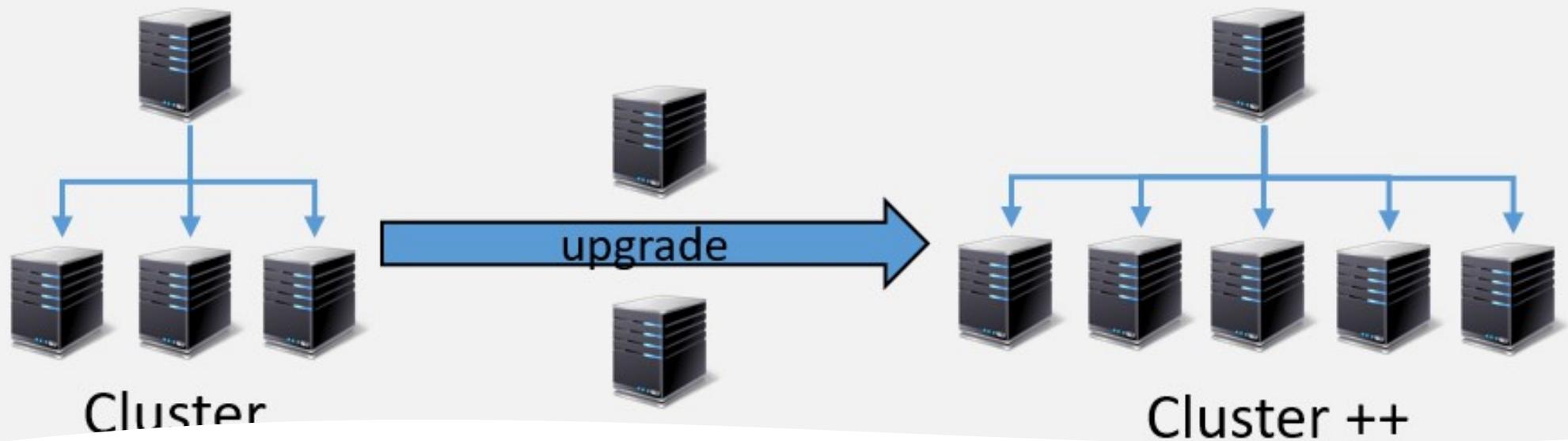


Estandar server

Super server

Escalabilidad vertical

- Habitualmente, **la separación lógica en capas** se implementa de tal forma que se permita una **separación física** de las mismas.
- Interponiendo elementos conectores que actúen de middlewares es posible distribuir la aplicación de forma vertical (una máquina por cada capa del sistema), e incluso si esto no fuera suficiente, distribuyendo los elementos de una misma capa entre distintas máquinas servidoras.



Escalabilidad horizontal.

- Se trata de **clonar el sistema en otra máquina de características similares y balancear la carga** de trabajo mediante un dispositivo externo.
- El balanceador de carga puede ser:
 - Software
 - Hardware
 - Hardware http

Balanceador Software:

Suele ser una **función del controlador** de distribución de aplicaciones (ADC) que se ejecuta en un servidor estándar o una máquina virtual.

Distribuye el tráfico entre un grupo de servidores de acuerdo con el **algoritmo elegido**.

Es la forma en que los administradores enrutan el tráfico de red a diferentes servidores.

Evalúan las **solicitudes** de los clientes **examinando** las características de las aplicaciones (la dirección IP, el encabezado HTTP y el contenido de la solicitud).

Balanceador hardware:

Se trata de **dispositivos** que, respondiendo únicamente **a algoritmos de reparto** de carga (Round Robin, LRU, etc.), redireccionan una petición http del usuario a la máquina.

Son **mucho más rápidos** que los anteriores, dado que se basan en conmutación de circuitos y **no examinan ni interpretan el paquete http**.

No garantiza el mantenimiento de la misma sesión de usuario en la misma máquina, condiciona seriamente el diseño, dado que fuerza a que la información relativa a la sesión del usuario sea almacenada por el implementador del mismo, bien en cookies o bien en base de datos.

Balanceador hardware http:

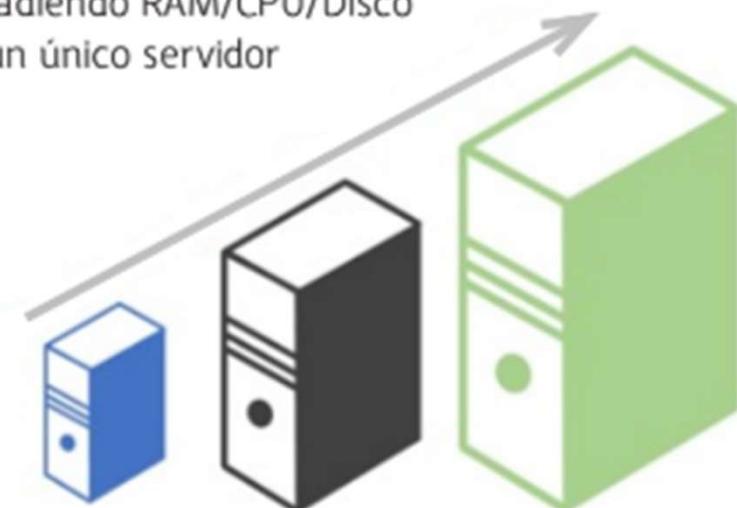
Se trata de **dispositivos** hardware pero que **examinan el paquete http y mantienen la relación usuario-máquina.**

Mucho más rápidos que los balanceadores software, pero algo menos que los hardware.

Una de las soluciones más aceptadas en el mercado

Escalado vertical

Aumentar la capacidad
añadiendo RAM/CPU/Disco
a un único servidor



Escalado horizontal

Aumentar la capacidad añadiendo servidores



Cluster

Dependiendo de cómo se aplique, podría clasificarse como **vertical u horizontal**.

Permite el despliegue de una aplicación web corriente, de forma que su carga de trabajo vaya a ser **distribuida entre la granja de servidores** que forman el cluster, de modo transparente al usuario y al administrador.

Mediante el mecanismo de **replicación de sesión**, garantiza que sea cual sea la máquina que sirva la petición http, tendrá acceso a la sesión del usuario (objeto **HttpSession** en java).

Este tipo de sistemas, debido precisamente a la replicación de sesión, suele presentar **problemas de rendimiento**.

¿En qué tipo de escalabilidad se emplean los balanceadores de carga?

Horizontal

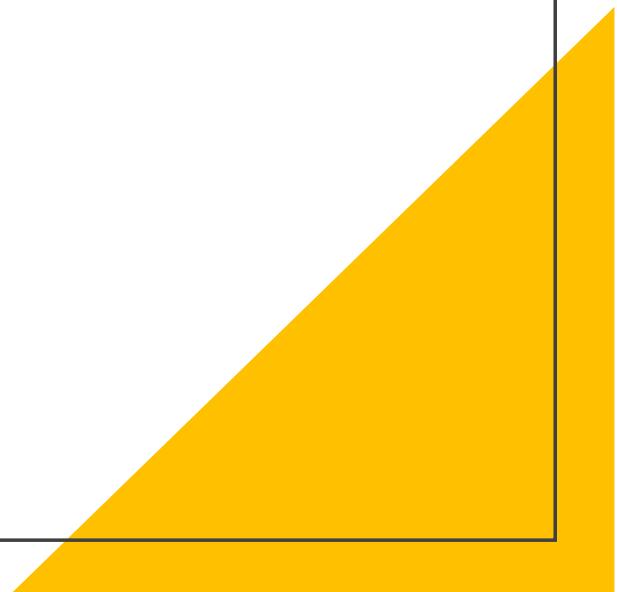
Vertical

- Correcto. Los balanceadores se encargan de repartir la carga entre las distintas máquinas; pudiendo existir tres tipos de balanceadores :
 - Balanceador software: Centran su trabajo basándose en las sesiones establecidas por los usuarios de la aplicación.
 - Balanceador hardware: Se basan en algoritmos de reparto de carga; redireccionando las peticiones http del usuario a la máquina indicada por el algoritmo.
 - Balanceador hardware-http: Situación intermedia entre los dos anteriores.



Revisa el concepto de escalabilidad vertical

Ejercicio 1



2. Servidor web

Apache

¿Qué es un servidor web?

- Programa que **se ejecuta de forma continua en un ordenador** (también se utiliza el término para referirse al ordenador que lo ejecuta), **se mantiene a la espera de peticiones** por parte de un cliente (un navegador de Internet) **y contesta** a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

¿Qué es Apache y para qué sirve?

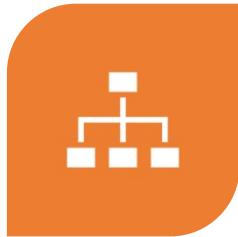
[https://www.youtube.com/watch](https://www.youtube.com/watch?v=eNvP7vwFCiQ)
?v=eNvP7vwFCiQ

Apache

- Uno de los servidores web más populares del mercado y el más utilizado
- **Código abierto y gratuito**
- Disponible para Windows y GNU/Linux, entre otros.
- Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.
- La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.



Arquitectura de Apache



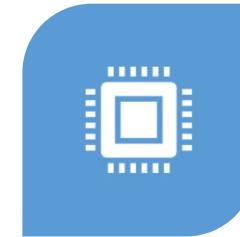
ESTRUCTURADO EN
MÓDULOS.



CADA MÓDULO
CONTIENE UN
CONJUNTO DE
FUNCIONES
RELATIVAS A UN
ASPECTO
CONCRETO DEL
SERVIDOR.



EL ARCHIVO
BINARIO HTTPD
CONTIENE UN
CONJUNTO DE
MÓDULOS QUE
HAN SIDO
COMPILADOS.



LA FUNCIONALIDAD
DE ESTOS
MÓDULOS PUEDE
SER ACTIVADA O
DEACTIVADA AL
ARRANCAR EL
SERVIDOR.

Módulos de Apache



Módulos base:

- Funciones básicas.

Módulos multiproceso:

- Unión de los puertos de la máquina, aceptando las peticiones y atendiéndolas.

Módulos adicionales:

- Añadir funcionalidad al servidor.



Para saber más

- Esta web sirve como manual de referencia, guía de usuario, tutoriales prácticos, etc., sobre el servidor web Apache. Se trata de la web oficial de Apache Software Foundation.
- [Documentación Servidor Apache en español](#)



2.1.- Instalación y configuración



Instalación

- Máquina con la última distribución de Ubuntu.
- Puede ser una máquina real, una máquina virtual o un contenedor.

Empezamos por identificarnos en la máquina con el usuario root y, a continuación, ejecutamos:

```
# apt-get update && apt-get install systemd apache2
```

```
# apt-get install php libapache2-mod-php php-mysql  
# apt-get install mysql-server
```

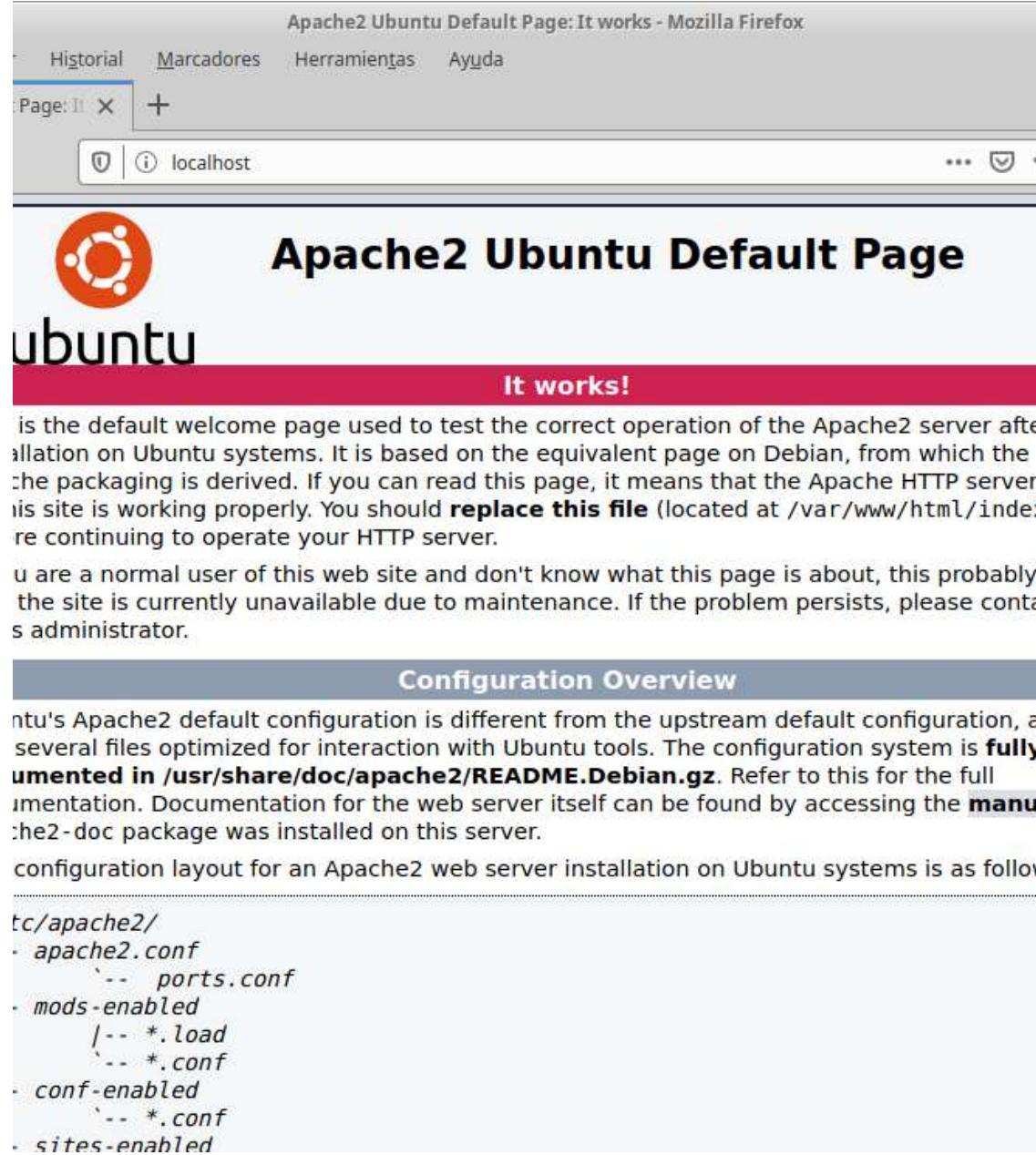
Debido a que pretendemos montar una plataforma LAMP, por sus ventajas derivadas de las características del software libre, instalaremos también los siguientes componentes: MySQL y PHP.

También es buena idea hacer que Apache arranque automáticamente:

```
# systemctl enable apache2
```

Verificar si funciona

- Desde un navegador, escribiendo en la barra de direcciones **http://localhost** o **http://127.0.0.1**,
- Desde otro equipo de la red a la dirección IP de esta máquina.
- Deberíamos obtener una página similar a la siguiente



Página por defecto de Apache

Por defecto, Apache sirve las páginas web que están en la carpeta
"`/var/www/html/`"

En esa carpeta, encontramos un archivo "**index.html**" que es el que contiene la página por defecto de Apache.

Podemos crear nuevas carpetas en donde ubicaremos nuevas páginas web que deseamos servir.

Todas las páginas web accesibles a través del **puerto 80**.

Donde guardar los archivos de las páginas web

- Si **una página web**, podemos integrar su contenido en la **carpeta**.
- **Más páginas web**, es más recomendable la utilización de los **Hosts Virtuales**.
 - Accedemos a la carpeta "**/etc/apache2/sites-enabled**"
 - Fichero llamado "**000-default**", que nos va a servir de ejemplo para la creación de hosts virtuales.
 - Nos permite servir varias webs desde una sola dirección IP utilizando para cada una un puerto distinto.

Configurar Apache

Colocando directivas en archivos de configuración de texto plano.

- El archivo principal de configuración se llama **apache2.conf**.

Se pueden añadir otros archivos de configuración mediante la **directiva "Include"**, y se pueden usar comodines para incluir muchos archivos de configuración.

- Todas las directivas deben colocarse en alguno de esos archivos de configuración.

Apache2 sólo reconocerá los cambios realizados en los archivos principales de configuración **cuando se inicie o se reinicie**.

Configuración de Apache2

/etc/apache2/apache2.conf

- Para configurar:
 - el número de puerto
 - la raíz de documentos
 - los módulos
 - los archivos de registros
 - los hosts virtuales
 - etc.

- Pasamos a ver alguna de las principales directivas:

ServerTokens:	ServerSignature:	Alias:	userDir:
<ul style="list-style-type: none">• configurar la cantidad de información que Apache aporta sobre sí mismo.	<ul style="list-style-type: none">• indicar datos sobre Apache en el pie de los mensajes de error.	<ul style="list-style-type: none">• permite direccionar a una carpeta que puede estar fuera del árbol de directorios especificado en DocumentRoot.	<ul style="list-style-type: none">• redireccionar al directorio personal del usuario si se recibe una solicitud de tipo ~usuario.

Para modificar el servidor virtual predeterminado

Editamos el archivo **/etc/apache2/sites-available/default**.

En el caso de querer configurar un **nuevo servidor o sitio virtual, copiaríamos ese archivo** dentro del mismo **directorio con el nombre** que se haya elegido, y **editaríamos** el nuevo archivo para configurar el nuevo sitio usando algunas de las directivas que se describen a continuación.

Directivas

ServerName:

- En el caso de no tener un dominio registrado emplearíamos **localhost**.

CustomLog:

- Define el archivo **.log** donde se guardan los logs de acceso.

ServerAdmin:

- Especifica la **dirección de correo del administrador del servidor**.
- El valor por omisión es **webmaster@localhost**.

Listen:

- Especifica el **puerto** (y, opcionalmente, la dirección IP) por el que escuchará Apache2.
- La directiva se puede encontrar y cambiar en **su propio archivo** de configuración, **/etc/apache2/ports.conf**.

DocumentRoot:

- Especifica **dónde Apache debe buscar los archivos** que forman el sitio.
- El valor predeterminado es **/var/www**.

RedirectMatchen:

- En **/etc/apache2/apache2.conf**, las peticiones se redirigirán a **/var/www/apache2-default**, que es donde reside el sitio predeterminado de Apache2.
- Cambiar este valor en el archivo de host virtual implica crear ese directorio si fuese necesario.



2.2.- Iniciar Apache



Probar su configuración por defecto

Si todo **está correcto** debería devolver un mensaje del tipo "**Syntax Ok**" y el **servidor** debería estar **arrancado**, con lo cual, si en un navegador introducimos la URL: **http://localhost** veríamos la **página de bienvenida de Apache**.

```
# apachectl configtest
```

- Si el **puerto** especificado en la **directiva Listen** del fichero de configuración es el que viene por defecto, es decir, el puerto 80 (o cualquier otro **puerto por debajo del 1024**), entonces es necesario tener **privilegios de usuario root** (superusuario) **para iniciar Apache**, de modo que pueda establecerse una conexión a través de esos puertos privilegiados.
- Una vez que el servidor Apache se ha iniciado y ha completado algunas tareas preliminares. El proceso principal, **httpd**, continúa ejecutándose como **root**, pero los **procesos hijo se ejecutan con menores privilegios** de usuario.

El demonio httpd

Se debería invocar empleando el **script de control apachectl**, que es el que se encarga de fijar variables de entorno y pasa al demonio (httpd) cualquier opción que se le pase como argumento por línea de comandos.

El script apachectl

Es capaz de interpretar los argumentos `start`, `restart`, `y stop` y traducirlos en las señales apropiadas para `httpd`, y es el comando genérico para Linux, sin embargo, para Debian y Ubuntu es más habitual usar los comandos que se comentan a continuación.

Ver el estado del servicio con el comando:

```
# systemctl status apache2
```

Y nos mostrará si el servidor está arrancado, desde cuándo, la versión, etc.

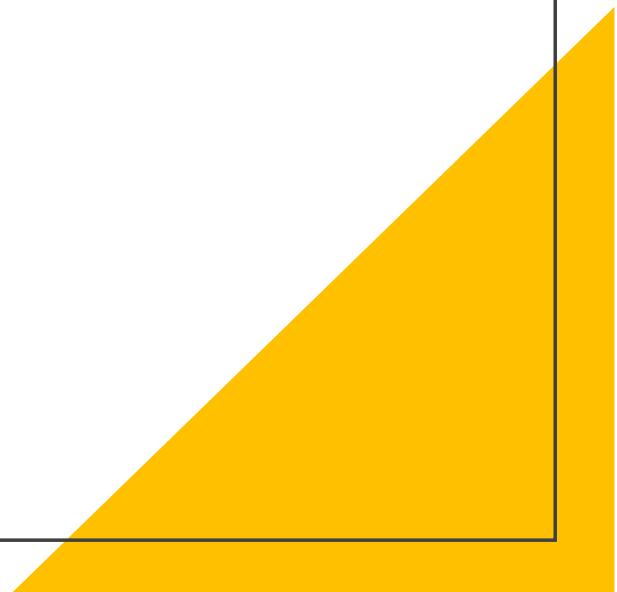
Parar, reiniciar o arrancar el servidor:

```
# /etc/init.d/apache2 stop  
# /etc/init.d/apache2 restart  
# /etc/init.d/apache2 start
```

Versiones más actuales Debian y Ubuntu:

```
# systemctl stop apache2  
# systemctl restart apache2  
# systemctl start apache2
```

Ejercicio 2



3. Aplicaciones web y servidores de aplicaciones

Aplicación web

Aplicación informática que se ejecuta en un entorno web, de forma que se trata de **una aplicación cliente-servidor** junto con un **protocolo** de comunicación previamente establecido:

- **Cliente:** navegador.
- **Servidor:** servidor web.
- **Comunicación:** protocolo HTTP.

Servidor de aplicaciones

Software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general, a través de Internet y utilizando el protocolo HTTP.

Se distinguen de los servidores web en el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos.

También es una máquina en una red de computadores que ejecuta determinadas aplicaciones, gestionando la mayor parte de las funciones de acceso a los datos de la aplicación.

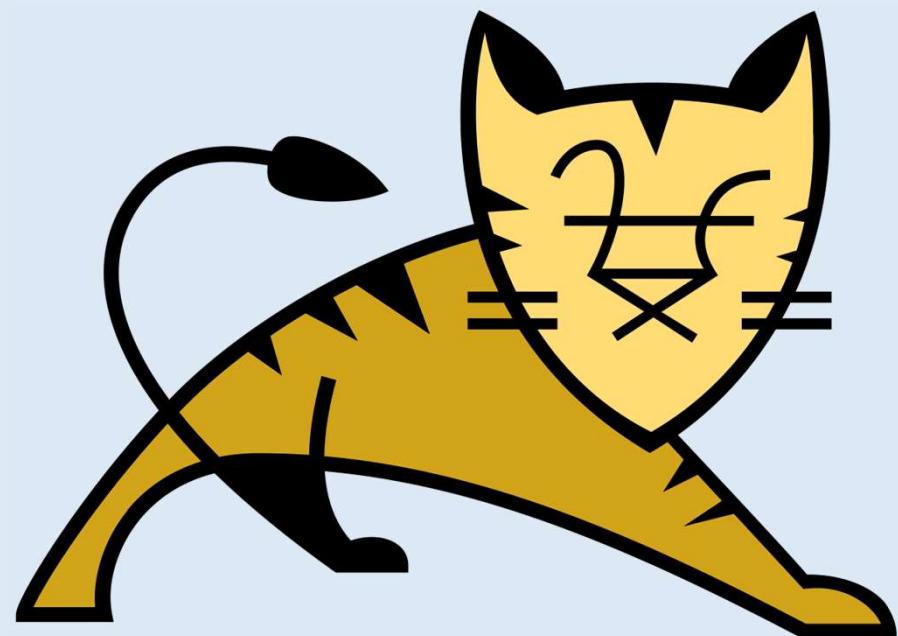
Ventajas de los servidores de aplicaciones

- **Centralización y disminución** de la complejidad en el desarrollo de las aplicaciones.
 - No necesitan ser programadas, sino que son ensambladas desde bloques provistos por el servidor de aplicación.
- **Integridad de datos y código**
 - Al estar centralizada en una o un pequeño número de máquinas servidoras, las actualizaciones están garantizadas para todos los usuarios.

Término servidor de aplicaciones

- **Se aplica a todas las plataformas.**
- Se utiliza para referirse a los servidores de aplicaciones basadas en web.
 - Ejemplo:
 - el control de las plataformas de comercio electrónico integrado
 - sistemas de gestión de contenido de sitios web
 - asistentes o constructores de sitios de Internet.

3.1.- El servidor de aplicaciones: Tomcat





Definición

Servidor web (incluye el servidor Apache) y de aplicaciones del proyecto Jakarta, con lo cual, gestiona las solicitudes y respuestas HTTP y, además, es servidor de aplicaciones o contenedor de Servlets y JSP.

Incluye el compilador Jasper, que compila JSP convirtiéndolas en servlets.

Es un contenedor de servlets con un entorno JSP. Un contenedor de servlets es un Shell de ejecución que maneja e invoca servlets por cuenta del usuario

Podemos dividir los contenedores de servlets en:

Contenedores
de servlets
**stand-
alone** (independientes):

Contenedores
de servlets
dentro-de-proceso:

Contenedores
de servlets
fuera-de-proceso:

Contenedores de servlets stand-alone (independientes):

Estos son una parte **integral del servidor web.**

Este es el caso en el que se usa un servidor web basado en Java,

- por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por iPlanet).

Por defecto Tomcat trabaja en este modo, sin embargo, la mayoría de los servidores no están basados en Java.

Contenedores de servlets

dentro-de-proceso:

El contenedor servlets es una **combinación de un plugin** para el servidor web y una implementación de **contenedor Java**.

El plugin del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él.

En el caso de que una petición debiera ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java.

Un contenedor de este tipo es adecuado **para servidores multi-thread de un sólo proceso** y proporciona un buen rendimiento pero está **limitado en escalabilidad**.

Contenedores de servlets fuera-de-proceso:

El contenedor servlets es una **combinación de un plugin** para el servidor web y una implementación de **contenedor Java** que se ejecuta en una JVM **fuera del servidor** web.

El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP).

Si una cierta petición tuviese que ejecutar un servlets, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs).

El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

||| Consideraciones

Tomcat puede utilizarse como un **contenedor solitario** (principalmente para desarrollo y depuración) o como **plugin para un servidor web existente** (actualmente soporta los servidores Apache, IIS).

Esto significa que siempre que despleguemos Tomcat tendremos que **decidir cómo usarlo y**, si seleccionamos las opciones 2 o 3, también necesitaremos instalar un adaptador de servidor web.

Las funciones del Servidor Apache y las funciones del servidor Tomcat, ¿son equivalentes?

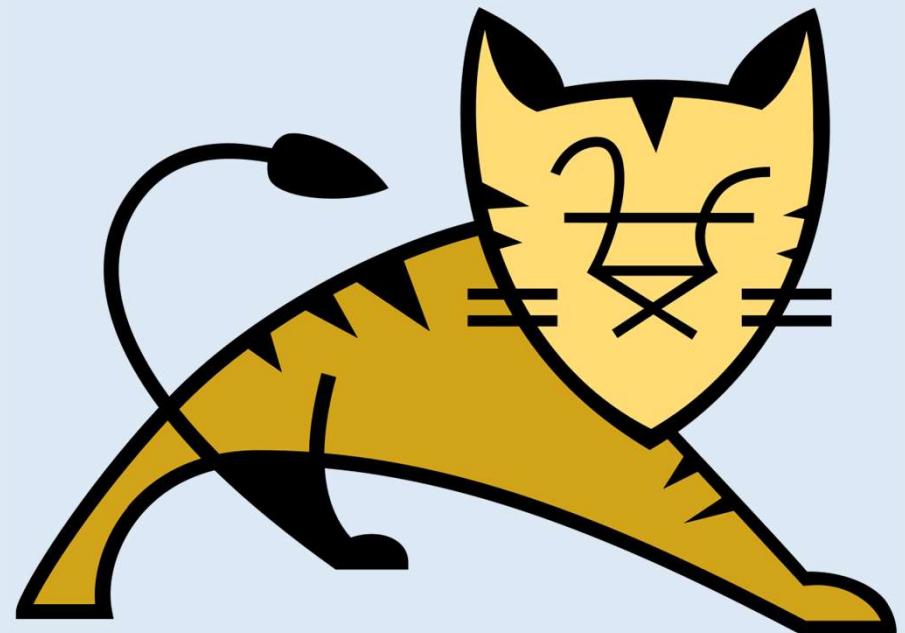
Si

No



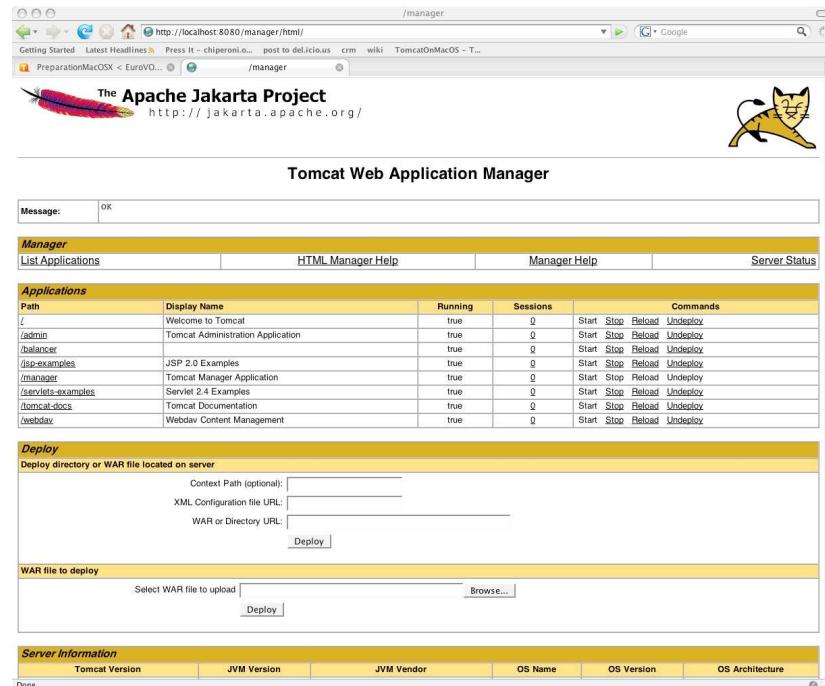
Básicamente, el servidor Apache es únicamente un servidor web, mientras que el servidor Tomcat es un servidor de aplicaciones.

3.1.1.- Instalación y configuración básica



En primer lugar

Para instalar cualquier versión de Tomcat es necesario tener **instalado JDK (Kit de desarrollo de Java)**, ya que el objetivo es que las peticiones a Apache se redirijan a Tomcat empleando un conector proporcionado por Java en este caso.



Empezamos actualizando el sistema:

```
$ sudo apt update
```

Instalamos el siguiente paquete por ser el que más se adapta a nuestras necesidades

```
$ sudo apt install default-jdk
```

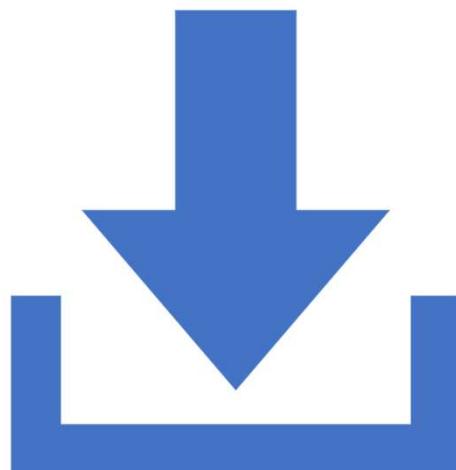


```
$ sudo groupadd tomcat  
$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```



Crearemos un grupo **tomcat**, y un usuario **tomcat** miembro del grupo anterior, con un directorio de inicio **/opt/tomcat** (donde instalaremos Tomcat) y un shell de **/bin/false** (de modo que nadie pueda iniciar sesión en la cuenta):

Ahora descargaremos e instalaremos Tomcat



Para descargar la última versión de Tomcat, iremos a su [página de descargas](#).

En este ejemplo vamos a usar la versión **9.0.35**, copiamos el enlace del archivo **.tar.gz**.

Nos movemos al directorio **/tmp** y descargamos el archivo:

(si no disponemos de curl, lo instalamos con sudo apte install curl)

```
$ cd /tmp  
$ curl -O https://apache.brunneis.com/tomcat/tomcat-9/v9.0.35/bin/apache-tomcat-9.0.35.tar.gz
```



```
$ sudo mkdir /opt/tomcat  
$ sudo tar xzvf apache-tomcat-*tar.gz -C /opt/tomcat --strip-components=1
```



Una vez que se complete la descarga,
extraeremos el archivo comprimido en el
directorio /opt/tomcat:

```
$ cd /opt/tomcat  
$ sudo chgrp -R tomcat /opt/tomcat
```

Nos cambiamos al directorio en el que desempaquetamos la instalación de Tomcat y le damos propiedad sobre todo el directorio de instalación al grupo tomcat:

```
$ sudo chmod -R g+r conf
$ sudo chmod g+x conf
$ sudo chown -R tomcat webapps/ work/ temp/ logs/
```

A continuación, le proporcionamos al grupo **tomcat** acceso de lectura al directorio **conf** y a todos sus contenidos, y acceso de ejecución al directorio, además nos aseguramos de que el usuario **tomcat** sea el propietario de los directorios **webapps**, **work**, **temp** y **logs**:

Ahora tenemos que crear el archivo tomcat.service en el directorio /etc/systemd/system:

```
$ sudo nano /etc/systemd/system/tomcat.service
```

Y escribiremos lo siguiente:

En la variable de entorno **JAVA_HOME** tendremos que poner la ruta que tengamos en nuestro equipo.

```
1 [Unit]
2 Description=Apache Tomcat Web Application Container
3 After=network.target
4
5 [Service]
6 Type=forking
7
8 Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
9 Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
10 Environment=CATALINA_HOME=/opt/tomcat
11 Environment=CATALINA_BASE=/opt/tomcat
12 Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
13 Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'
14
15 ExecStart=/opt/tomcat/bin/startup.sh
16 ExecStop=/opt/tomcat/bin/shutdown.sh
17
18 User=tomcat
19 Group=tomcat
20 UMask=0007
21 RestartSec=10
22 Restart=always
23
24 [Install]
25 WantedBy=multi-user.target
```

```
$ sudo systemctl daemon-reload  
$ sudo systemctl start tomcat
```

Por último, recargamos el demonio **systemd** para que reciba información sobre nuestro archivo de servicio e iniciamos el servicio Tomcat:

Para comprobar que el servicio está iniciado sin errores podemos escribir:

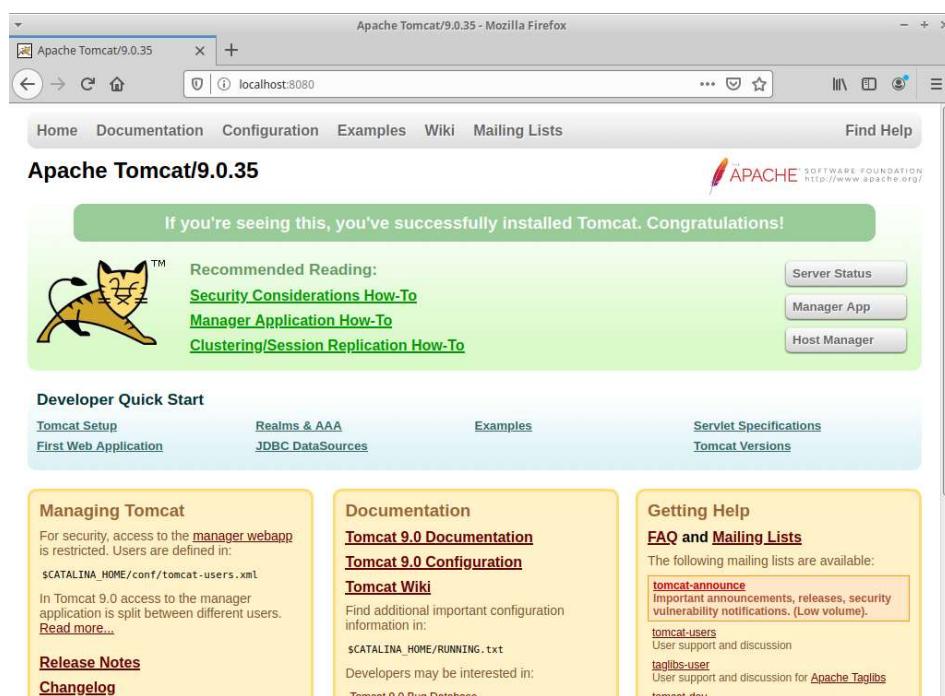
```
$ sudo systemctl status tomcat
```

Y nos mostrará si el servidor está arrancado, desde cuando, la versión, etc.

Gestión del servicio

Se realiza a través del script incluido llamado **catalina**, al que le podemos proporcionar los parámetros **start** y **stop**, con lo que arrancaríamos o pararíamos el servicio manualmente.

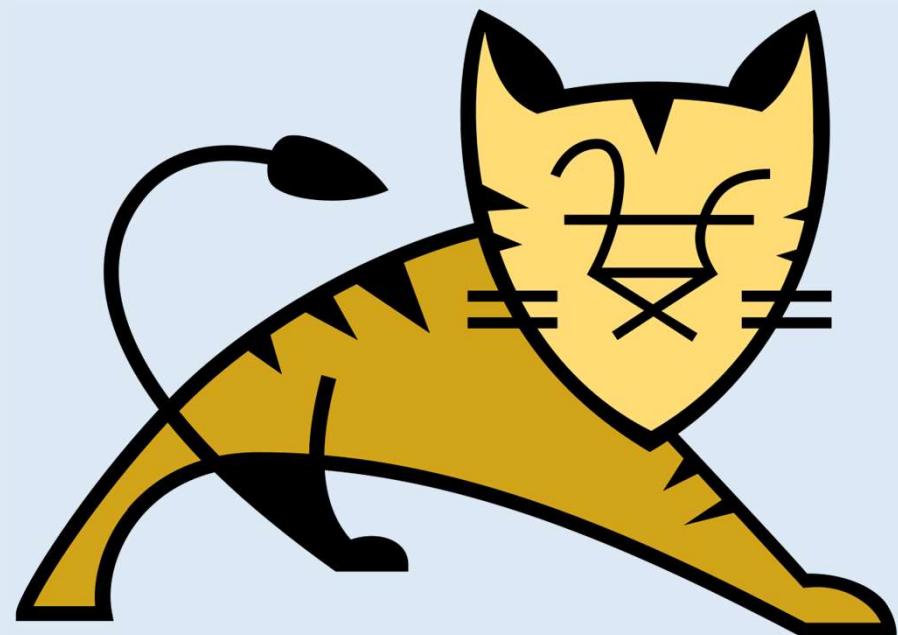
Comprobar que nuestro servidor está ya escuchando

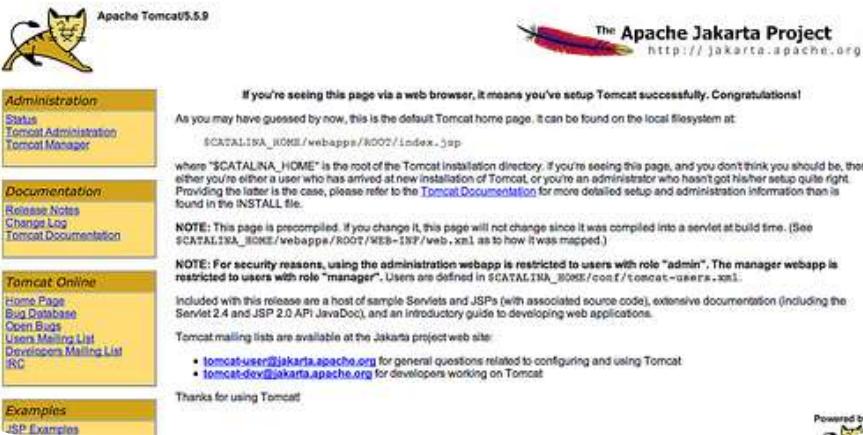


Para, introducimos en un navegador la URL: **http://localhost:8080** o **http://127.0.0.1:8080**, y éste debería mostrar la página de inicio de Tomcat similar a la siguiente



3.1.2.-Iniciar Tomcat





Tomcat va a estar escuchando en el puerto **8080** y va a tener su propio directorio de trabajo.

La misión de **apache2** va a ser interceptar todas las peticiones en el puerto **80** y derivar las que considere necesarias a Tomcat; de este modo observamos la ventaja de la escalabilidad, ya que apache, al funcionar como proxy, puede tener una batería de tomcats a los que balancear las conexiones, haciendo que, si nuestras necesidades crecen, nuestras máquinas puedan ampliarse en número siendo completamente transparente para los usuarios.

```
$ sudo nano /opt/tomcat/conf/server.xml
```

Apache por defecto busca los ficheros en **/var/www/html**, Tomcat trabaja sobre la carpeta **/opt/tomcat/webapps**.

La petición de una URL se puede gestionar, parte por apache y parte por Tomcat, por lo que vamos a cambiar la carpeta por defecto de trabajo para unificarlo.

Para ello editamos el fichero **/opt/tomcat/conf/server.xml**.

Encontraremos una línea con "**Host name=**" y
debería tener:

```
Host name="localhost" appBase="wwebapps"
```

```
$ sudo nano /opt/tomcat/conf/tomcat-users.xml
```

Para usar la aplicación de administración web que viene con Tomcat, debemos añadir un inicio de sesión a nuestro servidor de Tomcat.

Para ello editaremos el archivo **tomcat-users.xml**:

Añadir usuario

Que pueda acceder a **manager-gui** y **admin-gui** (aplicaciones web que vienen con Tomcat).

Podemos definir un usuario, por ejemplo como el que se muestra a continuación, entre las etiquetas **tomcat-users**, y poniendo nombres de usuarios y contraseñas seguras:

```
1 <tomcat-users . . .>
2   <user username="admin" password="password" roles="manager-gui,admin-gui"/>
3 </tomcat-users>
```

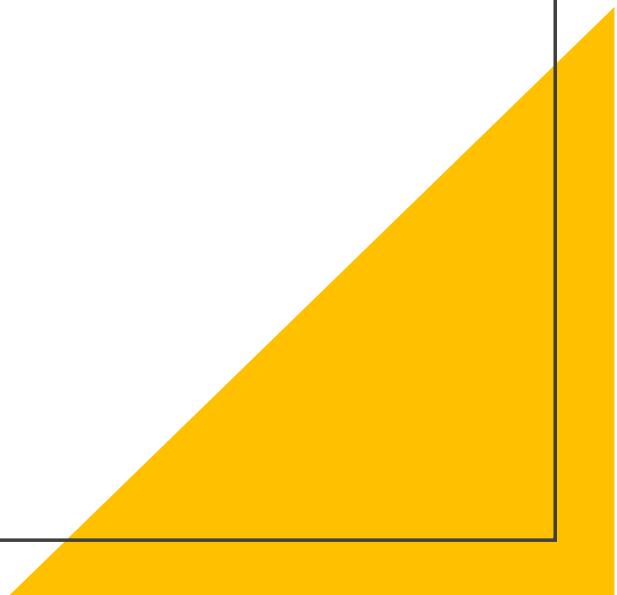
Si en cualquier momento deseásemos parar o arrancar el servidor, podríamos emplear los siguientes comandos respectivamente:

```
$ sudo systemctl stop tomcat  
$ sudo systemctl start tomcat
```

O bien:

```
$ sudo service tomcat stop  
$ sudo service tomcat start
```

Ejercicio 3



4. Estructura y despliegue de una aplicación web

Componentes de una app web

- 
- Una serie de servlets
 - páginas JSP
 - ficheros HTML
 - ficheros de imágenes
 - ficheros de sonidos
 - Texto
 - clases
 - etc.
-
- De forma que todos estos recursos se pueden empaquetar y ejecutar en varios contenedores distintos.

Servlet

- 
- Aplicación java encargada de realizar un servicio específico dentro de un servidor web.
 - Facilita la creación de contenido dinámico.
 - **A partir de Servlet 2.2:** Define la estructura que deben seguir las aplicaciones web, incluyendo cómo deben organizarse los archivos y directorios.

Estructura de directorios que define servlet

- El directorio **raíz** debería tener el **nombre de la aplicación** y define la raíz de documentos para la aplicación web.
- Todos los **ficheros debajo de esta raíz pueden servirse al cliente excepto** aquellos ficheros que están bajo los **directorios especiales**:
 - **META-INF**
 - **WEB-INF**

Directorios Especiales:

META-INF:

- Contiene **información sobre la aplicación**, como el descriptor de despliegue (web.xml).

WEB-INF:

- Aquí se almacenan los **archivos sensibles**, como los archivos .class de los servlets y otros recursos que no deben ser expuestos al usuario.

Durante la **etapa de desarrollo** de una aplicación web

- Se emplea la **estructura de directorios**
- El **código necesario para ejecutar** correctamente una aplicación web se encuentra distribuido en **una estructura de directorios, agrupándose** ficheros **según su funcionalidad.**

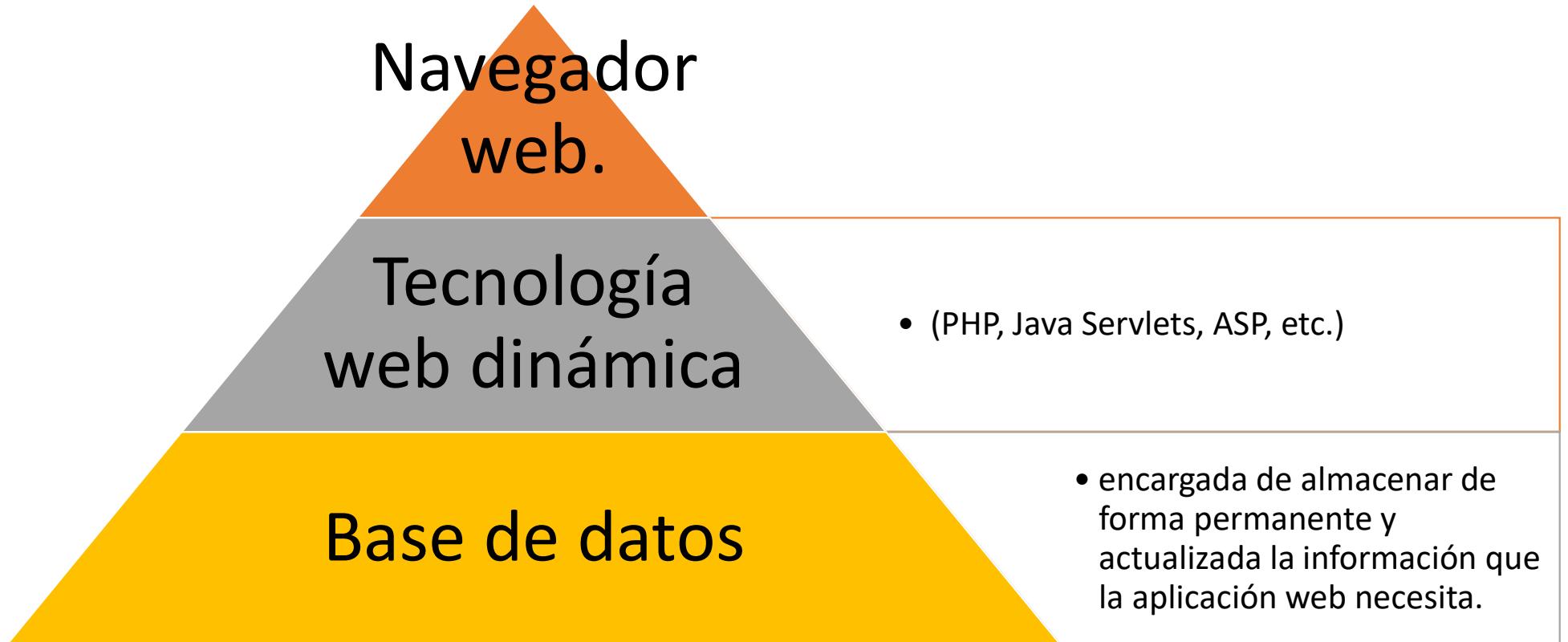
Etapa de producción,

- toda la estructura de la aplicación se empaqueta en un **archivo .war**.

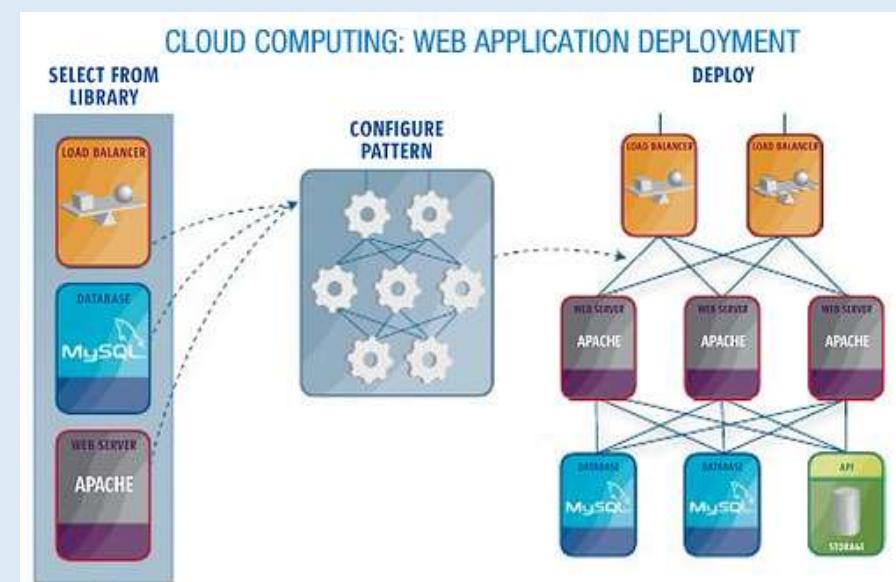
Un ejemplo de la estructura de carpetas de una aplicación web :

```
/index.jsp  
/WebContent/jsp/welcome.jsp  
/WebContent/css/estilo.css  
/WebContent/js/utils.js  
/WebContent/img/welcome.jpg  
/WEB-INF/web.xml  
/WEB-INF/struts-config.xml  
/WEB-INF/lib/struts.jar  
/WEB-INF/src/com/empresa/proyecto/action/welcomeAction.java  
/WEB-INF/classes/com/empresa/proyecto/action/welcomeAction.class
```

De forma genérica podríamos decir que una aplicación web se estructura en tres capas:



4.1.-Archivos WAR



Definición

- Su nombre procede de **Web Application Archive (Archivo de Aplicación Web)**.
- Permiten empaquetar en una sola unidad aplicaciones web de Java completas, es decir que su contenido sea:
 - Servlets y JSP.
 - Contenido estático: HTML, imágenes, etc.
 - Otros recursos web.

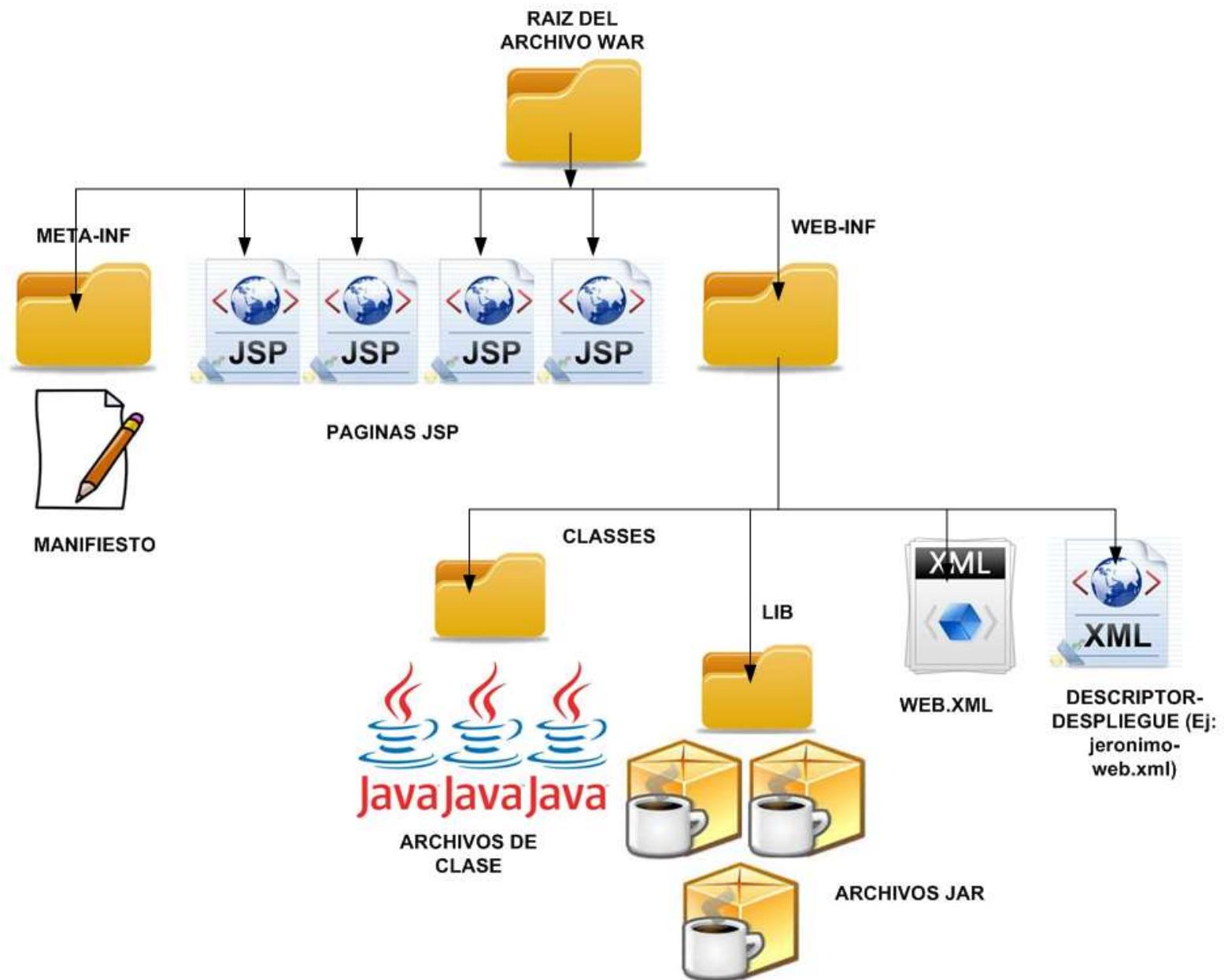
Ventaja



- La simplificación del despliegue de aplicaciones web:
 - Instalación es sencilla
 - Solamente es necesario un fichero para cada servidor en un cluster.
 - Incrementar la seguridad ya que no permite el acceso entre aplicaciones web distintas.

Su estructura:

/	<p>• En la carpeta raíz del proyecto se almacenan elementos empleados en los sitios web:</p> <ul style="list-style-type: none">• documentos html• hojas de estilo• los elementos JSP (*.html, *.jsp, *.css).	<p>/WEB-INF/</p> <ul style="list-style-type: none">• Aquí se encuentran los elementos de configuración del archivo .war como pueden ser:<ul style="list-style-type: none">• la página de inicio• la ubicación de los servlets• parámetros adicionales para otros componentes.• El más importante de éstos es el archivo web.xml.	<p>/WEB-INF/classes/</p> <ul style="list-style-type: none">• Contiene las clases Java empleadas en el archivo .war y, normalmente, en esta carpeta se encuentran los servlets.	<p>/WEB-INF/lib/</p> <ul style="list-style-type: none">• Contiene los archivos JAR utilizados por la aplicación y que normalmente son las clases empleadas para conectarse con la base de datos o las empleadas por librerías de JSP.
---	---	---	---	---



Generar archivos .war

- Se pueden emplear diversas herramientas desde entorno IDE.
- Por ejemplo:
 - NetBeans
 - Eclipse
 - Jbuilder de Borland
 - Jdeveloper de Oracle
 - Ant

Un archivo WAR

Es un archivo comprimido que se puede generar con cualquier tipo de compresor, por ejemplo winzip, winrar, tar, .

Es una aplicación web formada únicamente por archivos .

Es un archivo en el cual se empaqueta en una sola unidad, aplicaciones web completas.

Es un archivo que engloba el protocolo de comunicación de las aplicaciones web generadas con Java.

Un archivo WAR

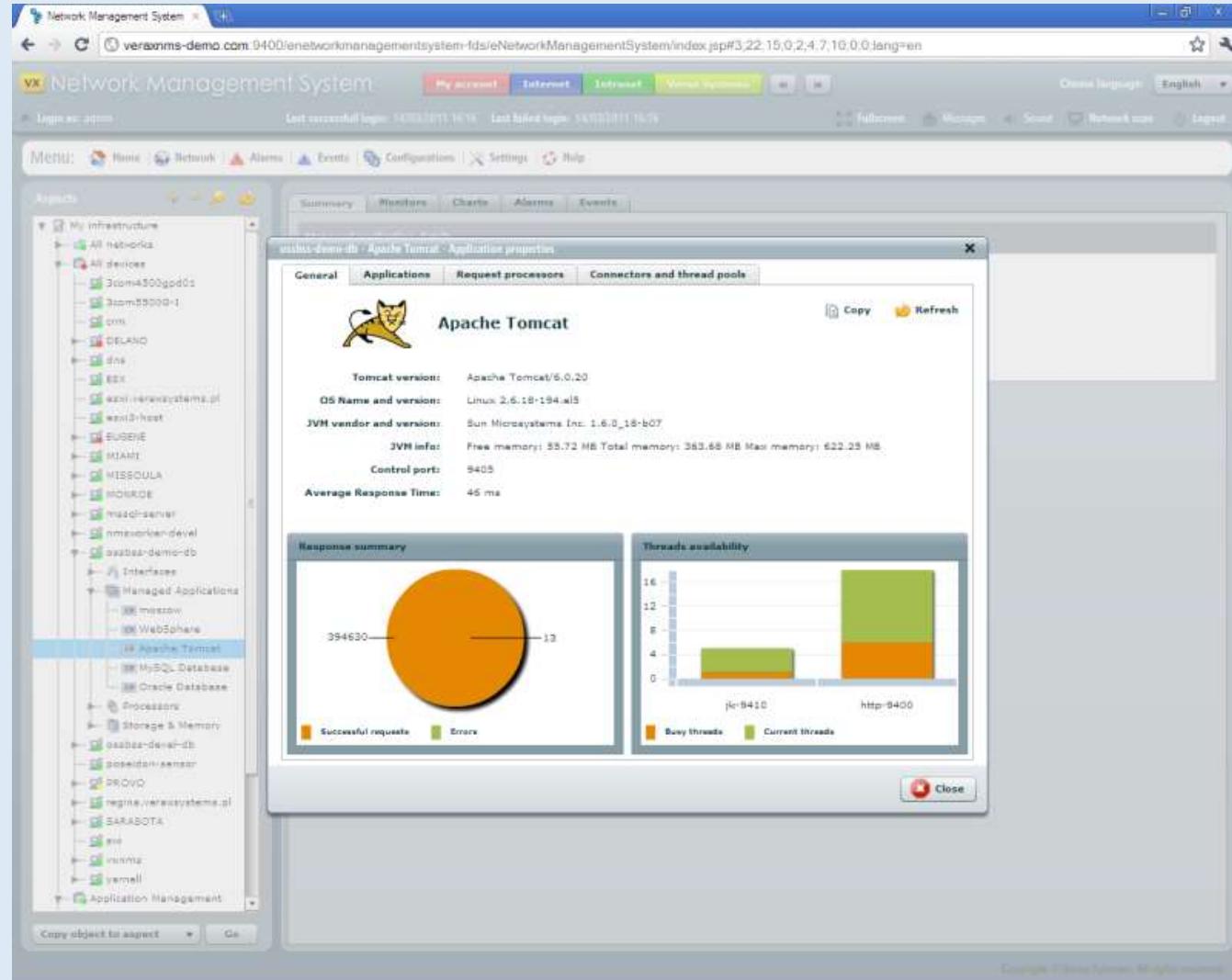
Es un archivo comprimido que se puede generar con cualquier tipo de compresor, por ejemplo winzip, winrar, tar, .

Es una aplicación web formada únicamente por archivos .

Es un archivo en el cual se empaqueta en una sola unidad, aplicaciones web completas.

Es un archivo que engloba el protocolo de comunicación de las aplicaciones web generadas con Java.

4.2.-Despliegue de aplicaciones con Tomcat



Métodos:

Por medio de archivos .war

- (Web Archive).

Editando los
archivos web.xml y server.xml.

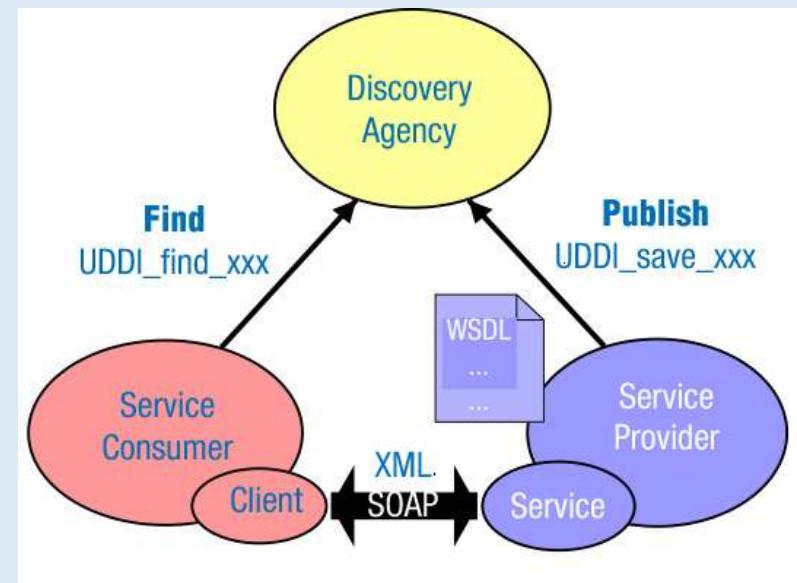
- Este método es el que se pasa a tratar a continuación.

Directorios de la aplicación web

www	Bin	src	tomcat	gwt-cache
<ul style="list-style-type: none">• Contiene una carpeta con el nombre y ruta del proyecto que contiene los ficheros que forman la interfaz (.html, .js, .css, etc.).	<ul style="list-style-type: none">• Contiene las clases de java de la aplicación.			<ul style="list-style-type: none">• en aplicaciones que utilizan Google Web Toolkit (GWT).



4.2.-Despliegue de aplicaciones con Tomcat



Descriptor de Despliegue

- Documento XML que describe las características de despliegue de una aplicación, un módulo o un componente.
- Por esto, la información del descriptor de despliegue es declarativa, y esta puede ser cambiada sin la necesidad de modificar el código fuente.

- Cualquier aplicación web tiene que aportar un descriptor de despliegue situado en WEB-INF/web.xml
- Tomcat el descriptor <TOMCAT_HOME>/conf/web.xml
 - descriptor por defecto
 - ejecuta siempre antes del descriptor de la aplicación
 - solamente, debería contener información general y no específica de la aplicación

Un ejemplo de descriptor de despliegue

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">

    <servlet>
        <servlet-name>MiServlet</servlet-name>
        <servlet-class>com.ejemplo.MiServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>MiServlet</servlet-name>
        <url-pattern>/miServlet</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>

</web-app>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
    <!-- Tus definiciones van aquí -->
</web-app>
```

Situadas entre las **etiquetas <web-app>** y **/<web-app>** estarían los **descriptores de despliegue** de servlets, los cuales deben contener las siguientes etiquetas en el siguiente orden:

```
<servlet>
    <servlet-name>nombre</servlet-name>
    <servlet-class>package.nombre.MiClass</servlet-class>
</servlet>
```

Para probar el servlet

- Una vez arrancado el servidor Tomcat
- Abrimos un navegador web
- Escribimos una URL con el siguiente formato:

```
http://{address}:{port}/{servletName}
```

Por ejemplo:

`http://localhost:8080/Servlet_de_prueba`

Escribe el nombre de tecnologías asociadas a aplicaciones siguiendo los enunciados siguientes:

ASP
PHP
CGI
CSS
PERL

- La Interface Común de Entrada es uno de los estándares más antiguos en Internet para trasladar la información desde una página web a un servidor web:

- Las Hojas de Estilo en Cascada se usan para formatear las páginas web:

- Las Páginas Activas se ejecutan del lado del servidor:

- Este lenguaje es, como ASP, usado en el lado del servidor, es similar a ASP y puede ser usado en circunstancias similares:

- Algo así como lenguaje práctico de extracción y de informes, nace con el objetivo principal de simplificar las tareas de administración de un sistema **UNIX**:

1. La Interface Común de Entrada es uno de los estándares más antiguos en Internet para trasladar la información desde una página web a un servidor web: **CGI**
2. Las Hojas de Estilo en Cascada se usan para formatear las páginas web: **CSS**
3. Las Páginas Activas se ejecutan del lado del servidor: **ASP**
4. Este lenguaje es, como ASP, usado en el lado del servidor, es similar a ASP y puede ser usado en circunstancias similares: **PHP**
5. Algo así como lenguaje práctico de extracción y de informes, nace con el objetivo principal de simplificar las tareas de administración de un sistema UNIX: **PERL**

