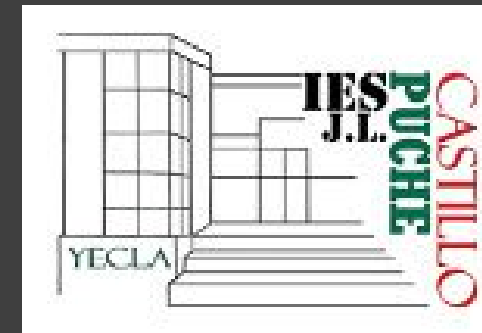




2º DAW

DESPLIEGUE DE APLICACIONES WEB



Unidad 5 – Servicios web

Profesora:

blanca.palao@murciaeduca.es

Contenidos

- Características generales de un servidor web.
- Automatización de máquinas virtuales.
- Configuración avanzada del servidor web.
- Módulos: instalación, configuración y uso.
- Hosts virtuales. Creación, configuración y utilización.
- Autenticación y control de acceso.
- El protocolo HTTPS.
- Certificados. Servidores de certificados.
- Despliegue de aplicaciones sobre servidores web.
- Virtualización en la nube.

Resultados de aprendizaje y criterios de evaluación

2. Gestiona servidores Web, evaluando y aplicando criterios de configuración para el acceso seguro a los servicios.

Criterios de evaluación:

- a) Se han reconocido los parámetros de administración más importantes del servidor Web.
- b) Se ha ampliado la funcionalidad del servidor mediante la activación y configuración de módulos.
- c) Se han creado y configurado sitios virtuales.
- d) Se han configurado los mecanismos de autenticación y control de acceso del servidor.
- e) Se han obtenido e instalado certificados digitales.
- f) Se han establecido mecanismos para asegurar las comunicaciones entre el cliente y el servidor.
- g) Se han realizado pruebas de funcionamiento y rendimiento del servidor Web.
- h) Se ha elaborado documentación relativa a la configuración, administración segura y recomendaciones de uso del servidor.
- i) Se han realizado los ajustes necesarios para la implantación de aplicaciones en el servidor Web.

Índice

1. Introducción
2. Arquitecturas web.
3. Evolución de la tecnología web.
4. Características generales de un servidor web.
5. Automatización de máquinas virtuales.
6. Recursos en Internet.
7. Cómo funciona la web.
8. WWW es un sistema Cliente/Servidor.
9. Tipos de alojamientos web y características.
10. El protocolo HTTP vs HTTPS.

Índice

11. Estructura y recursos que componen una aplicación web. Descriptor de despliegue. Archivos jar, war y ear. Maven.
12. Host virtuales. Creación, configuración y utilización.
13. Módulos: Instalación, configuración y uso.
14. Autenticación y control de acceso a directorios.
15. Certificados, servidores de certificados.
16. Pruebas de funcionamiento, monitorización y rendimiento del servidor web.
17. Prácticas:
 - Instalación y configuración del servicio Apache en SO Linux.
 - Instalación y configuración del servidor nginx en SO Linux.
 - Instalación y configuración del servidor IIS en Windows Server.

1. Introducción.

La **arquitectura web** es la planificación y el diseño de los componentes técnicos, funcionales y visuales de una aplicación web, antes de que sea desarrollado e implementado. Los diseñadores y desarrolladores la utilizan como un medio para ejecutar su trabajo.

Las aplicaciones web son una composición muy compleja de distintos sistemas integrados entre sí: Bases de datos, servidores (DNS, HTTP, etc.), redes, componentes de backup y seguridad, etc...).

El resultado final será una aplicación web que pueda resolver las necesidades de negocios: vender productos y servicios online y servir mejor a las necesidades de los clientes.

Ver ejemplos en Aula Virtual > **Recurso extra:**

<https://aulavirtual.murciaeduca.es/mod/page/view.php?id=3275080>

2. Arquitecturas web.

De forma genérica podríamos decir que la **arquitectura web** es un modelo compuesto de tres capas:

1. **Capa de Base de Datos**, donde estaría toda la información que se pretende administrar mediante el servicio web y emplearía un SGBD tipo MySQL, PostgreSQL, etc.
2. En una segunda capa estarían los **servidores de aplicaciones web** ejecutando aplicaciones de tipo Apache, Tomcat, Weblogic, Jboss, etc.
3. En una tercera capa estarían los **clientes** del servicio web al que accederían mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

Por último, estarían otros **sistemas terceros** si es que la aplicación web es completa y necesita integrarse con otros **sistemas externos**.

2. Arquitecturas web.

La **infraestructura** permite a los usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros. También permite a los desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

Los **aspectos generales** a destacar en una arquitectura web son los siguientes:

- Escalabilidad.
- Separación de responsabilidades.
- Portabilidad.
- Utilización de componentes en los servicios de infraestructura.
- Gestión de las sesiones del usuario.
- Aplicación de patrones de diseño.

3. Evolución de la tecnología web.

La evolución del uso de **servicios web** en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios. Entre los factores que han impulsado el uso de los servicios web se encuentran:

- El contenido se está volviendo más **dinámico**: Los sitios web actuales proporcionan contenidos "instantáneos". Un servicio web debe ser capaz de combinar contenido proveniente de fuentes muy diferentes.
- El **ancho de banda es menos costoso**: Actualmente un servicio web puede entregar tipos variables de contenidos como vídeo o audio. A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenidos.

3. Evolución de la tecnología web.

- El **almacenamiento en la nube** es cada vez más barato: Un servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.
- El éxito de **Internet de la cosas**: los ordenadores están dejando de ser el dispositivo más comunes en el uso de Internet por lo que los servicios web deben servir a todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.
- **En los orígenes** del mundo web nos situábamos ante un entorno estático, con páginas en formato HTML que raramente sufrían modificaciones o actualizaciones y en las que apenas había interacción con el usuario. Todo eso ha cambiado.

4. Características generales de un servidor web.

¿Qué es un servidor web y para qué sirve en Internet?

Un servidor web es un dispositivo que le brinda espacio y estructura a los sitios web para que almacenen sus datos y manejen sus páginas.

Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo **HTTP** (Hypertext Transfer Protocol).

4. Características generales de un servidor web.

¿Para qué sirve un servidor web en Internet?

El rol principal de un servidor web es almacenar y transmitir el contenido solicitado de un sitio web al navegador del usuario.

Este proceso, para los internautas no dura más que un segundo, sin embargo, a nivel del web server es una secuencia más complicada de lo que parece.

Para cumplir con sus funciones el servidor deberá tener la capacidad de estar siempre encendido para evitar interrumpir el servicio que le ofrece a sus clientes. Si dicho servidor falla o se apaga, los internautas tendrán problemas al ingresar al sitio web.

4. ¿Que es un servidor web?

Definición: “Un programa que implementa el protocolo HTTP (hypertext transfer protocol) Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language) textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos”.



4. ¿Que es un servidor web?

Arquitectura del servidor web

La arquitectura utilizada es **cliente/servidor** es decir, el equipo cliente hace una solicitud o petición al equipo servidor, y éste atiende dicha solicitud. En el equipo cliente se ejecuta una aplicación llamada **‘navegador o cliente web’** que:

- Sirve de interfaz con el usuario atiende sus peticiones, muestra los resultados de las consultas y proporciona al usuario un conjunto de herramientas que facilitan su comunicación con el servidor
- Se comunica con el servidor web transmite las peticiones de los usuarios.

En el equipo servidor la única tarea es:

- Atender las peticiones recibidas desde los navegadores o clientes web y hacerlo de forma eficiente y segura. Este es el caso de los servidores web seguros que solicitan un nombre de usuario y una contraseña para permitir el acceso sólo a usuarios registrados y por tanto, con permiso para visualizar la página/s.

4. ¿Que es un servidor web?

Objetivo del servidor web

El objetivo de un servidor web es servir o suministrar páginas web a los clientes web o navegadores que las solicitan. Ejemplo:

- Un usuario desde el navegador Firefox, al hacer 'click' sobre un enlace a una página web está enviando una solicitud al servidor web que aloja dicha página para que se la muestre (eso se llama servir la página).
- El servidor web si la encuentra la envía y si no la encuentra devuelve al cliente un mensaje de error.
- El cliente cuando recibe la página web interpreta el código HTML mostrando las fuentes, colores, imágenes que la componen, etc. de forma correcta.

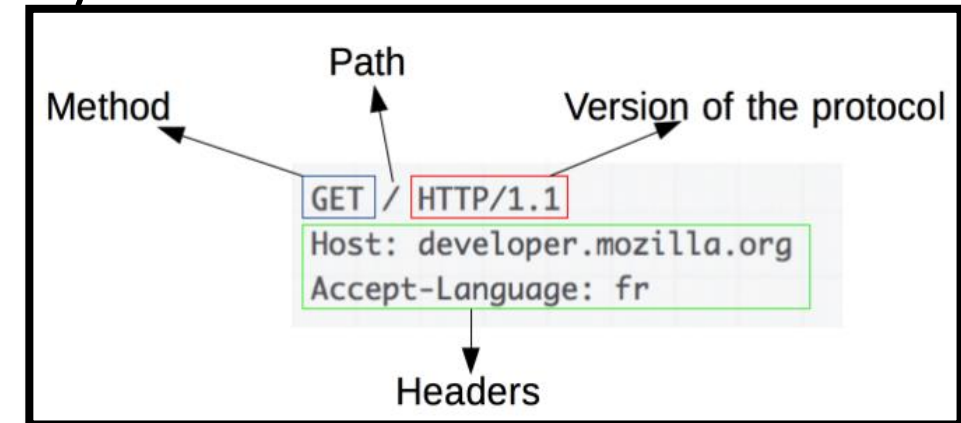
Pero el servidor web, además de servir páginas web HTML estáticas, también permite la ejecución de una serie scripts en diferentes lenguajes de programación, que proporcionan dinamismo a las páginas web. Estos lenguajes son PHP, Java, etc.

4. ¿Que es un servidor web?

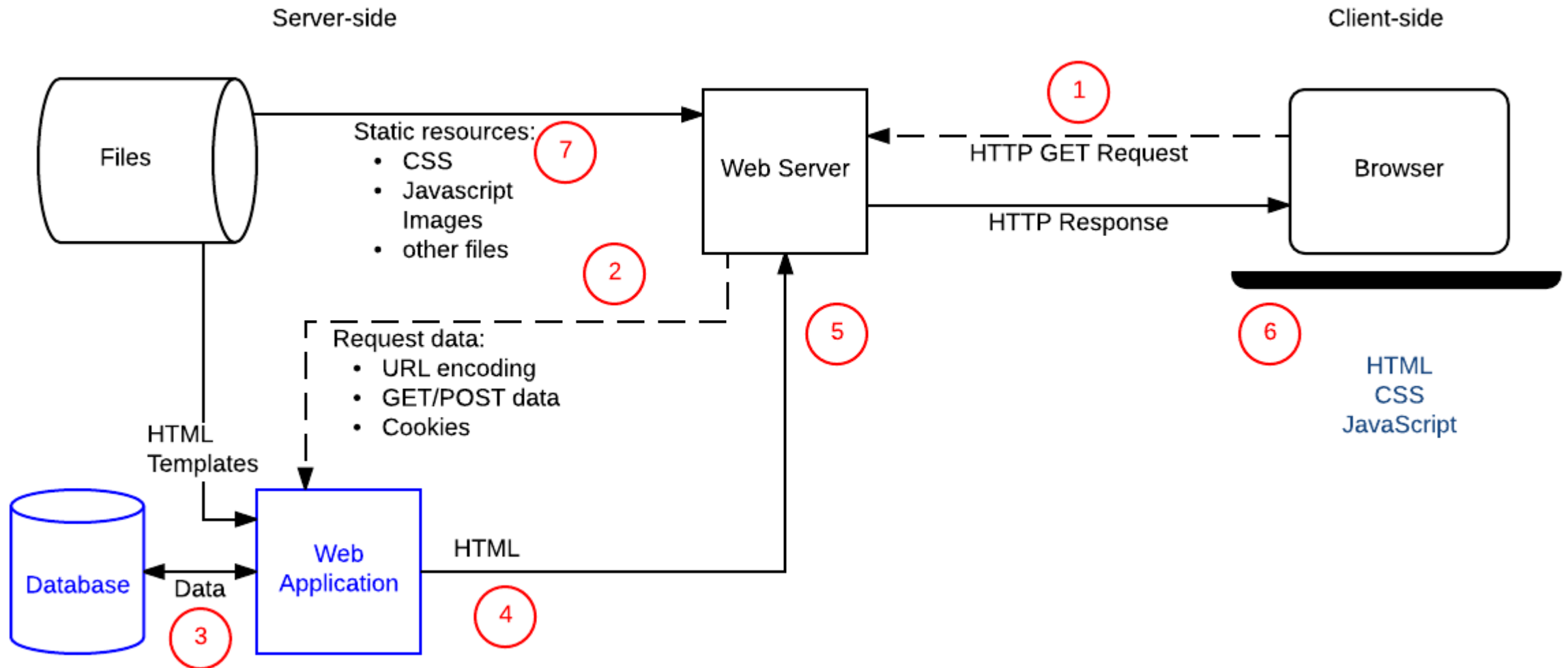
Esquema

La imagen que aparece en la siguiente página, se muestra el esquema de funcionamiento de un servidor web. En ella observamos los elementos básicos que componen las peticiones web dinámicas:

- El cliente que hace la petición (HTTP GET).
- El servidor que atiende la petición y se accede a la aplicación web.
- La aplicación web accede a la base de datos para recuperar información almacenada, imágenes, contenidos, etc.
- La aplicación web devuelve a la información al servidor y los datos solicitados mediante la petición HTTP response son mostrados en
- el navegador web del cliente.
- La parte de ***user experience*** (HTML, CSS y JavaScript)
- consumen recursos del cliente.



4. ¿Que es un servidor web?



5. Automatización de máquinas virtuales.

La necesidad de tener un 100% de disponibilidad en las máquinas que alojan los servidores web requiere de los mecanismos automatizados para que el servidor se balancee o escale de forma que esté algún servidor siempre disponible y que puedan estos crecer según las necesidades cambiantes.

Antiguamente, y puede que lo sigáis viendo a día de hoy, se requería configurar una serie de scripts que controlaran el crecimiento de la máquina. La idea era que si se llegaba a un límite de porcentaje alto de rendimiento en la CPU en unas máquinas concretas, se arrancaban automáticamente las máquinas de reserva. La solución era la ideal para plataformas web y soluciones on-premise (servidores ubicados físicamente en las dependencias de la empresa).

5. Automatización de máquinas virtuales.

Actualmente con la infraestructura en cloud (IaaS), este tipo de necesidades son un quebradero de cabeza menos para la empresa pues pueden pagar por servicios de balanceo de carga para que los servidores repartan la carga de peticiones de forma equilibrada así como servicios de escalado pensados precisamente para levantar máquinas automáticamente y aumentar los recursos según las necesidades del servidor.

Los servicios de escalado de máquinas virtuales de los diferentes proveedores de IaaS permiten crear y administrar un grupo de máquinas virtuales con equilibrio de carga. El número de instancias de máquina virtual puede aumentar o disminuir automáticamente según la demanda, o de acuerdo a una programación definida. Por ejemplo, este servicio en **AZURE**:

<https://learn.microsoft.com/es-es/azure/virtual-machine-scale-sets/overview>

5. Automatización de máquinas virtuales.

El servicio de **AWS Auto Scaling** monitoriza las aplicaciones y ajusta automáticamente la capacidad para mantener un desempeño predecible y estable al menor costo posible. Con AWS Auto Scaling, resulta sencillo configurar el escalado de aplicaciones para distintos recursos en varios servicios en cuestión de minutos.

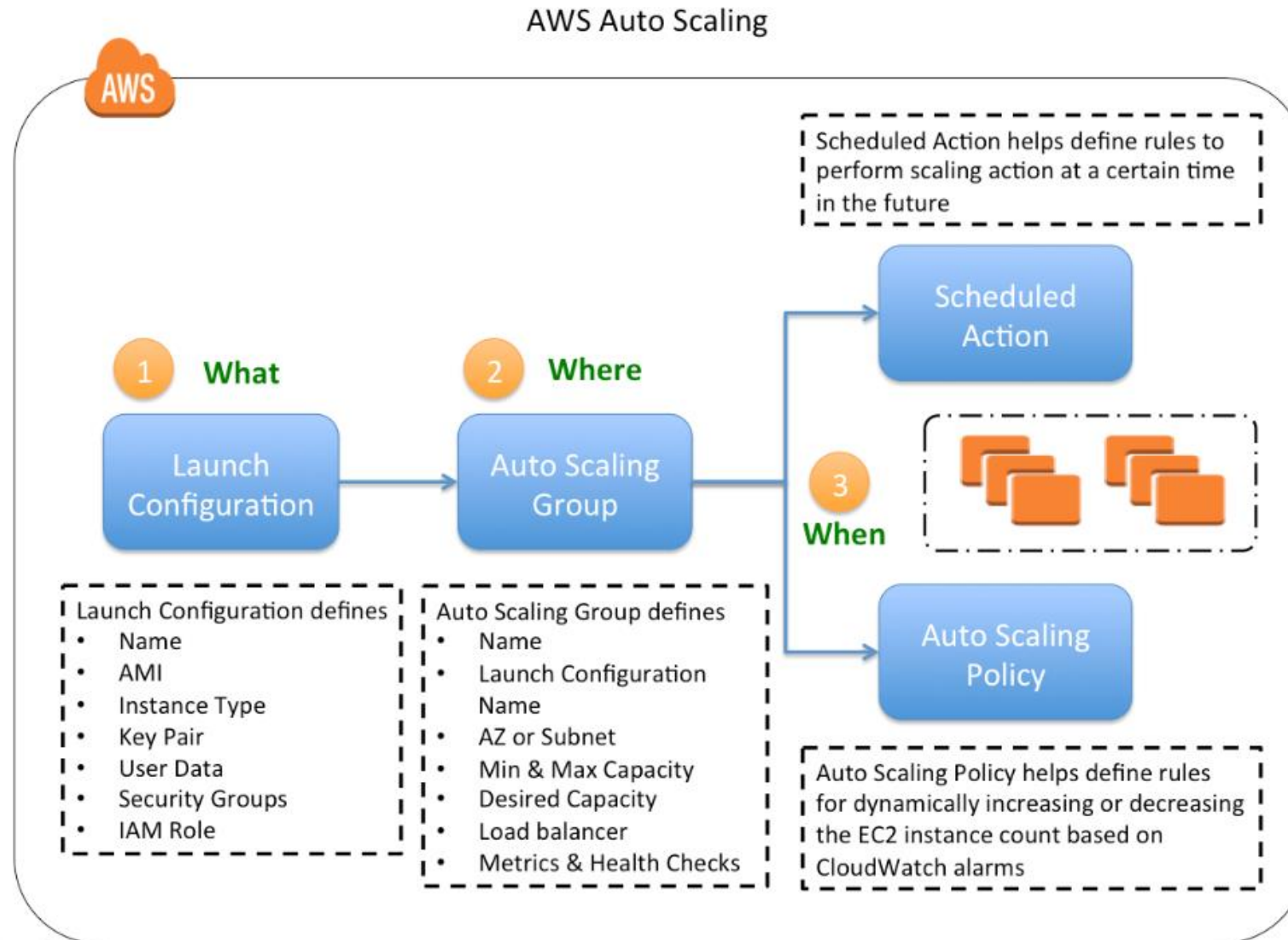
Desde la consola de AWS con la que hemos trabajado, podemos ver lo fácil que permite crear planes de escalado para recursos, incluidas las instancias de Amazon EC2 y otros recursos como SGBD.

AWS Auto Scaling facilita el proceso de escalado con recomendaciones sencillas que nos permiten optimizar el desempeño, los costos, o bien lograr un equilibrio entre ambos. Con AWS Auto Scaling, las aplicaciones siempre tienen los recursos adecuados en el momento correcto.

<https://aws.amazon.com/es/autoscaling/>



5. Automatización de máquinas virtuales.



6. Recursos en Internet

- Los recursos accesibles en Internet se identifican por un **URI** (identificador de recurso uniforme):
 - **URI**: aulavirtual.murciaeduca.es
 - **URL**: <https://aulavirtual.murciaeduca.es/my/index.php> :
 - ✓ **El protocolo** con que acceder al recurso (**https**).
 - ✓ **El nombre** (o dirección) del **host**, nombre del dominio:
aulavirtual.murciaeduca.es
 - ✓ **El nombre del recurso** (/my/index php).

7. Como funciona la web

- Para poder visualizar una página web es preciso:
 1. Tener un ordenador conectado a Internet
 - Establecer una conexión PPP con la máquina del **ISP** o estar conectado a una puerta de enlace que tenga acceso a internet.
 - El ordenador debe tener una dirección IP y una dirección de servidor DNS.
 2. Iniciar sesión en un navegador Web (Internet Explorer, Firefox, Safari, Chrome,...)
 3. Indicar al navegador el URL de la página que se desea.
 4. El navegador pide al DNS la dirección IP correspondiente al servidor que contiene el documento cuyo URL se ha indicado.
 5. El navegador abre una sesión TCP con la máquina cuya dirección IP se ha obtenido.
 6. El navegador solicita al servidor que le transmita el documento (orden GET)
 7. El servidor web envía el documento.
 8. La conexión TCP finaliza.
 9. El navegador muestra el documento.

7. Como funciona la web

Abrir dirección "http://www.uv.es/cerveron/index.html"



7. Como funciona la web

URL = Uniform Resource Locator

<http://www.uv.es/cerveron/index.html>

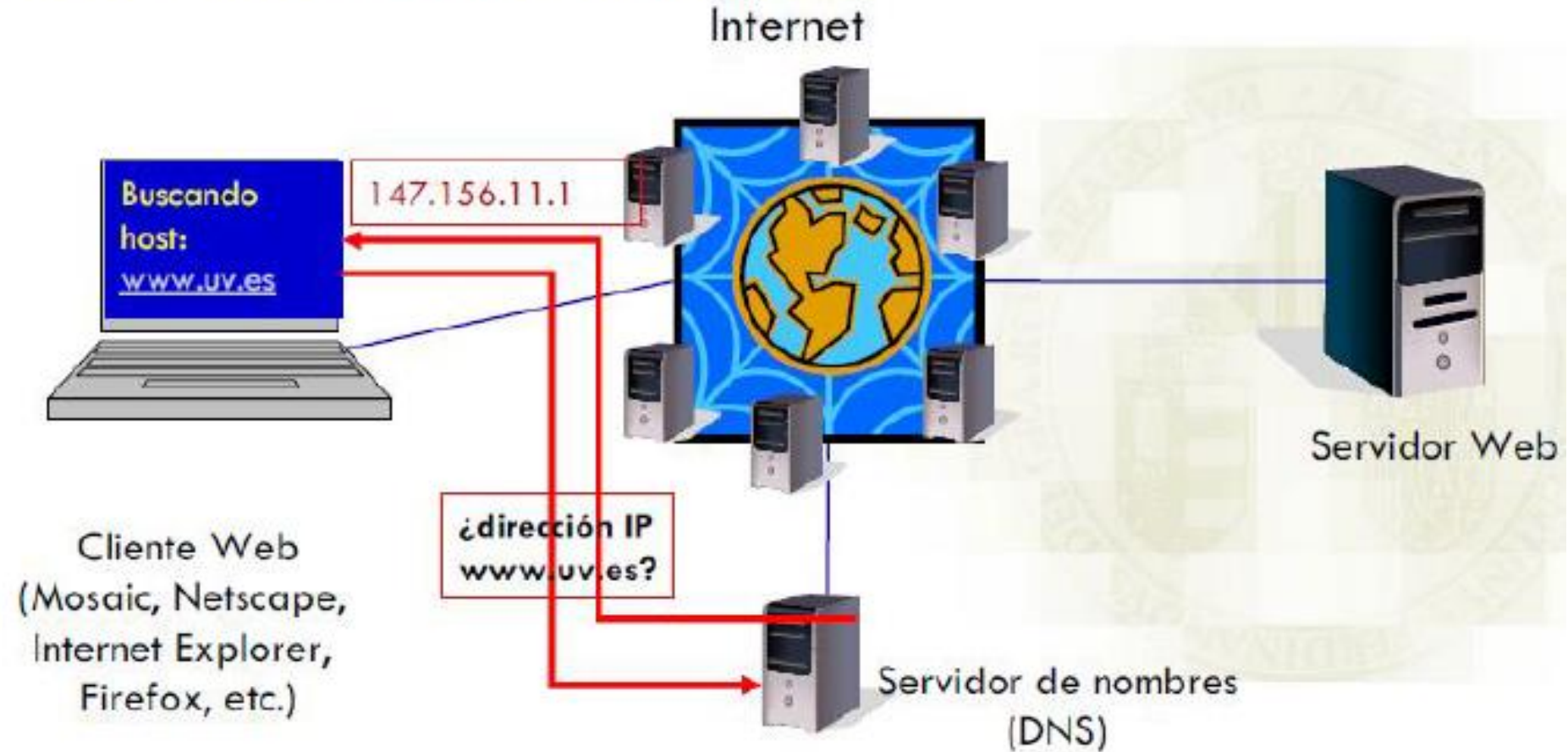
Nombre del protocolo de comunicación
con el servidor
(http es el estándar para web)

Localización del documento dentro del
sistema de archivos del servidor web

Nombre del dominio del servidor web
donde se almacena el documento

7. Como funciona la web

Abrir dirección "http://www.uv.es/cerveron/index.html"



7. Como funciona la web

Abrir dirección "http://www.uv.es/cerveron/index.html"



8. WWW es un sistema Cliente/Servidor

Cientes Web:

- Utilizan el protocolo HTTP para conectar con los servidores.
- Solicitan y muestran las páginas web almacenadas en los servidores.
- Clientes típicos: navegadores web (Chrome, Firefox, ...).

Servidores Web:

- "Escuchan" conexiones entrantes desde clientes.
- Utilizan el protocolo HTTP para conversar con los clientes.
- Almacenan y transmiten páginas web a los clientes.
- Actualmente interactúan con el usuario y generan dinámicamente páginas web.

9. Tipos de alojamientos web y características

Elegir un alojamiento web es un paso muy importante a la hora de publicar nuestro sitio web en Internet. Si buscamos en Google por alojamiento web (o web hosting) obtendremos millones de resultados, y el saber buscar el hosting adecuado no es una tarea fácil.

¿Qué es el alojamiento web?

Según la Wikipedia, “el alojamiento web llamado en inglés web hosting es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web”.



9. Tipos de alojamientos web y características

¿Qué es un proveedor de alojamiento web?

Es una empresa que alquila espacio web y ancho de banda para la publicación de sitios web.

Normalmente se trata de una cuenta en un sistema Linux o UNIX que está permanentemente encendido donde alojamos los archivos de nuestro sitio web a través de la herramienta FTP (o SSH), nuestra web es servida mediante un servidor web (por ejemplo: Apache o IIS).

Top In Web Server Usage Distribution on the Entire Internet

| Technology | Websites | % |
|---|------------|-------|
|  Apache | 60,851,533 | 41.16 |
|  nginx | 43,644,581 | 29.52 |
|  IIS | 30,479,537 | 100 |
|  Varnish | 4,774,695 | 3.23 |
|  LiteSpeed | 3,024,151 | 2.05 |
|  Citrix NetScaler | 2,024,363 | 1.37 |
|  Phusion Passenger | 527,725 | 0.36 |
|  Apache Traffic Server | 489,201 | 0.33 |
|  Tengine | 464,342 | 0.31 |

9. Tipos de alojamientos web y características

Tipos de alojamiento web

Los requerimientos de una página personal no son los mismos que los de una gran empresa. En este último caso, el volumen de información y la cantidad de accesos al servidor será muchísimo mayor y, por tanto, el servidor que maneje dicha información deberá disponer de más recursos.

En una web con miles de visitas al día, la no disponibilidad del servicio durante unas horas supone diversas pérdidas que no deben ser admisibles, por lo cual el hosting a contratar deberá ofrecer más calidad que, supongamos, en el caso de una web personal de un alumno que cuelga sus apuntes.

9. Tipos de alojamientos web y características

Tipos de alojamiento web -> Alojamientos de pago

- Compartido
- Servidor virtual privado
- Servidor dedicado
- Housing
- En la nube



9. Tipos de alojamientos web y características

1. Alojamiento gratuito

- Normalmente dispone de poca funcionalidad y estabilidad.
- No tienes derechos sobre el tiempo de gratuidad del servicio y cedes derechos a la empresa del alojamiento.
- Son útiles para:
 - Publicar páginas personales.
 - Hacer las primeras pruebas cuando se está aprendiendo a hacer páginas web.
 - Cuando no queremos gastar dinero.

Una empresa seria no debería emplear este sistema, ya que estos sistemas son limitados, ofrecen malas prestaciones (espacio y tráfico limitado) y suelen añadir publicidad a la página web.

9. Tipos de alojamientos web y características

2.1 Alojamiento de pago: Compartido

- El alojamiento compartido es la solución más común.
- Ofrece mejores prestaciones que la solución gratuita, pero a un precio reducido ya que se comparten los recursos software y hardware del servidor.
- Se dice que es compartido porque se alojan las aplicaciones de varios clientes en un mismo servidor. Se asigna a cada empresa un espacio privado y exclusivo y se asigna un porcentaje de recursos a utilizar dependiendo de las necesidades.
- Los recursos compartidos son: SO, IP del servidor, memoria RAM, espacio de disco, ancho de banda, etc.
- Es una buena solución para pequeños y medianos clientes, ya que ofrece una buena relación calidad/precio ya que al compartirse el servidor, el costo se reduce considerablemente.

9. Tipos de alojamientos web y características

2.2 Alojamiento de pago: Servidor virtual privado (VPS)

- Mediante un software especial que crea diferentes máquinas virtuales en un mismo servidor, la empresa de alojamiento ofrece al cliente el control de un ordenador aparentemente no compartido.
- Es una buena solución para clientes avanzados que necesitan un mayor control sobre su servicio web.
- Permite aislar nuestro sitio web de posibles errores que produzcan otros sitios web de otros clientes alojados en el mismo servidor.
- Este tipo de alojamiento web generalmente es elegido por los propietarios de sitios web que tienen un tráfico de nivel medio que excede los límites de los planes de hosting compartido pero que aún no necesitan los recursos de un servidor dedicado.

9. Tipos de alojamientos web y características

2.3 Alojamiento de pago: Servidor dedicado

- Es la solución más adecuada para un cliente que desea las mejores prestaciones, pero no quiere preocuparse de algunas tareas de administración y mantenimiento.
- El cliente alquila o compra un servidor completo y tiene el control completo sobre él.
- Hay una completa exclusividad para el cliente que lo alquila/compra y dispone totalmente de los recursos de: Memoria, almacenamiento, ancho de banda, CPU, disco duro, entre otras cosas; estarán disponibles solamente para un cliente.
- Un servidor dedicado da mayor seguridad y velocidad. Recomendado para sitios web con un tráfico muy elevado y grandes empresas.

9. Tipos de alojamientos web y características

2.4 Alojamiento de pago: Housing

- En el housing (también llamado *co-location*) la empresa de alojamiento dispone de un datacenter y sólo se encarga del cuidado físico del ordenador (básicamente, proporciona sistemas de alimentación ininterrumpida y un control de la temperatura ambiental) y además proporciona la conectividad a Internet.
- Los servicios de housing son ideales para clientes que tienen su propia plataforma tecnológica y que desean alojarla en unas instalaciones adecuadas con una conectividad de calidad.
- Básicamente, se alquila el espacio físico dónde se aloja el servidor y es el cliente el responsable de la mantención interna del servidor dedicado; es su responsabilidad asegurar que todo esté funcionando correctamente desde el hardware hasta el software.

9. Tipos de alojamientos web y características

2.5 Alojamiento de pago: Nube

- Es el sistema más innovador, que permite a un gran número de servidores actuar como un único servidor.
- El alojamiento en la nube utiliza múltiples servidores y recursos computacionales distribuidos en la red que llamamos nube o Cloud.
- Es más seguro porque el sitio web está alojado en varios servidores en vez de uno solo.
- Y es un sistema muy flexible porque permite aumentar las prestaciones en función de la demanda; la escalabilidad es transparente.
- Estas características hacen que el **alojamiento web en la nube sea ideal** para sitios **web con altos volúmenes de tráfico**, o portales en los que es **crítico garantizar un tiempo de funcionamiento cercano al 100%**.

9. Tipos de alojamientos web y características

3. On-premise

- La empresa instala la solución dentro de sus servidores e infraestructuras.
- El término on-premise o en local se refiere al tipo de instalación de una solución de software. Esta instalación se lleva a cabo dentro del servidor y la infraestructura (TIC) de la empresa. Es el modelo tradicional de aplicaciones empresariales.
- Con el modelo on-premise, la empresa es la responsable de la seguridad, disponibilidad y gestión del software. Por lo que la empresa debe tener un departamento de sistemas que dedique parte de sus recursos a la gestión de la infraestructura in situ. Sin embargo, el proveedor también suele proporcionar servicios de integración y soporte post-venta. Se aloja la gestión documental, sistemas ERP o gestión de relaciones con el cliente (CRM), copias de seguridad, servicios en red.

9. Tipos de alojamientos web y características

Consideraciones generales

- Para una cliente pequeño o mediano, lo mejor es empezar con la opción más básica, con el servidor compartido, y migrar a soluciones mejores y más costosas conforme aumenten las necesidades.

Ejercicio en el aula

Realizar una investigación en Internet sobre alguno de los tipos de alojamientos que hemos visto en el tema. Leed varios artículos sobre el tipo de alojamiento seleccionado y cada alumn@ realizará un pequeño resumen sobre ese tipo de alojamiento en concreto realizando una exposición en clase.

Foro en Aula Virtual:

<https://aulavirtual.murciaeduca.es/mod/forum/view.php?id=3199110>

9. Tipos de alojamientos web y características

Algunas propuestas de alojamientos gratuitos:

- <https://wordpress.com/> el sitio de alojamiento de webs más utilizado con un plan de alojamiento asequible para todos los bolsillos.
- <https://www.webs.com/> ofrece numerosas herramientas para crear sitios web de una forma sencilla. Posee un producto gratuito, pero las prestaciones que ofrece son bastante limitadas. También ofrece varios productos de pago.
- <https://es.wix.com/> Editor intuitivo de arrastrar y soltar elementos. Magníficas plantillas para crear página web de forma más fácil.
- **Google Sites** que permite crear sitios web públicos o privados para compartirlos con los usuarios que nosotros queramos. La principal ventaja de Google Sites es que permite integrar fácilmente otros servicios de Google, como mapas de Google Maps, documentos de Google Docs.

9. Tipos de alojamientos web y características

Recomendaciones para elegir alojamiento:

1. La calidad que la empresa cumpla lo que hemos contratado y que resuelva los problemas que tengamos rápidamente.
2. El sistema operativo que emplea el servidor web.
3. El espacio de almacenamiento que proporciona.
4. Los límites mensuales para la transferencia de datos.
5. La velocidad de transferencia máxima que podemos tener.
6. Los lenguajes de programación web que permite emplear.
7. Si permite crear bases de datos y cuántas.
8. Si admite múltiples dominios y subdominios.
9. Otros como el número de cuentas de correo que se pueden crear o si proporciona plantillas para crear el sitio web.
10. Y por último, claro está, el precio.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 1. La calidad

- La calidad se puede resumir en que la empresa cumpla lo que hemos contratado, que resuelva los problemas que tengamos y que nos ofrezca la máxima disponibilidad y fiabilidad, es decir, que el servidor web donde esté alojado nuestro sitio web no esté apagado o tenga problemas de conectividad frecuentemente.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 1. La calidad

- Algunos indicadores que nos pueden ayudar a averiguar la calidad de una empresa de alojamiento son:
 - ✓ La calidad de la página web de la empresa de alojamiento, mirando la profesionalidad de la misma.
 - ✓ Las opiniones de clientes antiguos y actuales que encontremos en Internet.
 - ✓ El tiempo que lleva operando la empresa.
 - ✓ El nivel de soporte que ofrece la empresa, por ejemplo, si ofrece una sección de documentación y manuales, o si ofrece un apartado de “preguntas más frecuentes”.
 - ✓ Y por último, si la empresa ofrece un **servicio de atención al cliente y resolución de problemas**. Además, hay que comprobar los mecanismos de contacto que ofrece la empresa teléfono, correo electrónico, chat, etc. Un parámetro importante es el tiempo de respuesta, pero esto también es difícil de conocer a priori, a no ser que nos lo garanticen por escrito en el contrato del servicio.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 2.El sistema operativo

Los dos sistemas operativos que normalmente vamos a encontrar en los planes de alojamiento son, por un lado Linux y por otro, Windows.

Normalmente, aunque no es obligatorio que sea así, cuando elegimos Linux tendremos como servidor web **Apache** y cuando elijamos Windows tendremos como servidor web **Internet Information Server (IIS)**.

La decisión de qué sistema operativo elegir depende principalmente del lenguaje de programación y de los servicios adicionales que queramos utilizar.

-> Elegiremos Linux cuando queramos utilizar los lenguajes de programación PHP, Python, Perl o JSP y la base de datos MySQL o PostgreSQL.

-> Elegiremos Windows cuando queramos utilizar los lenguajes de programación ASP o ASP NET y la base de datos Access o SQL Server.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 2.El sistema operativo

Normalmente, un alojamiento en Linux es más barato que en Windows, ya que la empresa de alojamiento se ahorra la licencia del sistema operativo. También suele ser más barato porque hay mejores herramientas para la administración de servidores Linux, lo cual abarata el coste de administración.

Normalmente los planes basados en Linux suponen un 30% de ahorro más que los planes basados en Windows.

<https://blog.infranetworking.com/linux-vs-windows-servidores/>

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 3. Almacenamiento

El almacenamiento, también llamado espacio web o espacio en disco es el espacio del sistema de archivos del servidor que nos ofrece la empresa de alojamiento para almacenar nuestro sitio web.

-> En inglés se suele llamar **disk space, disk storage o hosting space**.

El espacio web suele ser un cebo para atraer clientes. A veces se ofrecen tamaños muy grandes, para que el cliente piense que cuantos más megabytes de espacio, será mucho mejor. Sin embargo, para la empresa de alojamiento el almacenamiento en el servidor es lo más irrelevante y económico, por lo que un espacio web muy grande no implica obligatoriamente que la empresa de alojamiento sea buena.

En algunos planes de alojamiento nos pueden penalizar si superamos el límite de almacenamiento fijado. Por ejemplo, un plan de alojamiento nos indican que podemos tener un número ilimitado de bases de datos, pero las bases de datos consumen el espacio de almacenamiento disponible.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 4. Transferencia

La transferencia, también llamada volumen de transferencia o tráfico contabiliza el total de datos enviados en un periodo de tiempo.

-> En inglés se llama **file o site transfer**.

Esta característica sí que es importante que la contratemos adecuadamente, porque a veces en la “letra pequeña” del contrato de alojamiento puede poner que nos penalizan si la superamos. Por ejemplo, un plan de alojamiento nos ofrece una transferencia mensual de 2 GB. Sin embargo, si nos pasamos, nos cobrarán 15 €/mes por cada GB de exceso.

El volumen de transferencia también es difícil de calcular, más difícil aún que el espacio de almacenamiento, porque su valor depende del impacto que tenga nuestro sitio web, de cuántas visitas vayamos a recibir, etc. Y eso es algo que no depende de nosotros. Hay que calcular bien el consumo previsto y saber cuándo lo superaríamos y cuánto habría que pagar por él una vez superado el límite contratado. Un sitio web consume más tráfico que el correo electrónico y las transferencias de FTP, por lo que es suficiente con evaluar el tráfico del sitio web.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 5. Velocidad de transferencia

La velocidad de transferencia o ancho de banda (en inglés, **bandwidth**), es la cantidad de datos que se pueden enviar a través de un canal de comunicación por unidad de tiempo. Para el caso de los servidores web, las medidas se hacen en Kilobits por segundo o Megabits por segundo (Kbps/Mbps).

Al igual que con la transferencia, un ancho de banda ilimitado es totalmente imposible, siempre existirá un límite, aunque haya algunas empresas que lo prometan. El ancho de banda indica el límite de transferencia de datos que podemos alcanzar en un momento determinado e influye en el tiempo de respuesta del servidor web y en el tiempo que necesita un visitante para acceder a una página web. Por tanto, es un parámetro que afecta a la percepción que tienen los visitantes de nuestro sitio web. El cálculo del ancho de banda que necesitamos ha de realizarse teniendo presente los momentos de mayor actividad, lo que hace más complejo el cálculo, ya que el número medio de visitantes no es un valor apropiado.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 6. Lenguaje de programación.

Se emplea para dar las instrucciones de cálculo al ordenador y permite crear aplicaciones web.

Si nuestro sitio web es estático no tiene programación, esta característica no nos importará mucho porque no la utilizaremos. Sin embargo, si pensamos publicar una aplicación web, es muy importante comprobar qué lenguajes de programación admite el plan de alojamiento que vamos a contratar.

Existen múltiples lenguajes de programación. La decisión de emplear uno u otro depende más de los gustos y conocimientos que posea la persona que va a realizar la programación, ya que la mayoría de los lenguajes de programación modernos ofrecen la misma potencia y permiten realizar las mismas tareas con un costo similar.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 7. Bases de datos.

El sistema gestor de bases de datos es el software que permite almacenar y posteriormente recuperar los datos que necesita una aplicación de forma rápida, segura y estructurada.

Si nuestro sitio web es estático y no tiene programación, la base de datos es otra característica por la que, en principio, no nos tenemos que preocupar mucho.

Respecto los sistemas gestores de bases de datos que podemos encontrar en los planes de alojamiento, si el sistema operativo es Linux, se suelen ofrecer MySQL, MariaDB y PostgreSQL, mientras que si el sistema operativo es Windows, se suelen ofrecer Microsoft SQL Server.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 8. Dominios.

El dominio o nombre de dominio en inglés **domain o domain name**, es el nombre mediante el cual los visitantes van a encontrar nuestro sitio web. A un mismo sitio web se puede llegar a través de diferentes nombres de dominio.

Recordad que los subdominios permiten crear subdivisiones del dominio principal. Normalmente se emplean para crear sitios web adicionales bajo el paraguas del sitio web principal. Por ejemplo, los subdominios se pueden emplear para crear sitios web adicionales en función de la situación geográfica.

Algunas empresas de alojamiento ofrecen un número ilimitado de dominios y subdominios. Necesitaremos disponer de múltiples dominios y subdominios si queremos organizar el sitio web en diferentes secciones, como puede ser por la situación geográfica como hemos visto antes, o por el contenido.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 9. Otras características.

- El número de cuentas de **correo electrónico**, de **FTP** y **SSH** que se ofrecen.
- Si se facilitan scripts y librerías preinstalados.
- Las aplicaciones preinstaladas que existen, como de **foro**, de **blogs**, **gestores de contenido**, de **comercio electrónico**.
- El panel de control que ofrece, que es la **herramienta que permite gestionar** el plan de alojamiento.
- Si ofrece **análisis estadístico** de las visitas al sitio web y acceso directo a los ficheros de registro (que es donde se almacenan los datos estadísticos de las visitas al sitio web
- Si ofrece un servicio de **copias de seguridad**.
- Si ofrece el servicio **cron**, que permite programar procesos en segundo plano para que se ejecuten de forma periódica.
- Si se pueden realizar **conexiones seguras** al servidor con SSL.
- Y si tiene capacidades de **streaming** de vídeo y audio.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 10. El precio.

- Hay que encontrar un equilibrio entre lo que queremos obtener, qué calidad queremos, y el precio que estamos dispuestos a pagar.
- Al comparar las ofertas de diferentes empresas de alojamiento podemos encontrar lo más acorde a nuestras necesidades.
- Y un factor que influye en el precio es el período de contratación.
- Muchas empresas de alojamiento ofrecen grandes descuentos si se realizan contratos por períodos largos de tiempo. Por ejemplo, si se contrata el alojamiento por uno o dos años, se pagará mucho menos que si se realiza el contrato por trimestre.
- Además, normalmente un alojamiento en Linux es más barato que en Windows, ya que la empresa de alojamiento se ahorra la licencia del sistema operativo. También suele ser más barato porque hay mejores herramientas para la administración de servidores Linux, lo cual abarata el coste de administración.

9. Tipos de alojamientos web y características

Criterios a tener en cuenta en los alojamientos de pago: 10. El precio.

- Los agentes registradores, las entidades autorizadas para vender los nombres de dominio, también suelen ofrecer planes de alojamiento. Normalmente suelen ofrecer un paquete completo, que incluye el registro del nombre de dominio y el alojamiento.
- Pero no hay ninguna obligación en contratar ambos productos a la vez el nombre de dominio y el plan de alojamiento se pueden contratar a diferentes empresas.
- Como existen múltiples empresas, tanto españolas como de otros países, que ofrecen planes de alojamiento, la decisión de qué plan contratar puede ser un poco complicada.
- Para ayudarnos, existen algunas páginas web donde se realizan comparativas de alojamientos.
- Pero hay que tener cuidado y asegurarse de que la comparativa es reciente y está actualizada con las últimas prestaciones y precios.

Trabajo en aula

Buscad comparativa entre Nginx vs Apache vs IIS.

Comentaremos en el aula las diferencias entre un servidor y otro.



VS



Realizar prácticas

Prácticas en Aula Virtual



VS



10. El protocolo HTTP.

El protocolo de **transferencia de hipertexto** (**HTTP**, hyperText transfer protocol) es el motor que da vida a Internet, la base de la navegación web (www, world wideweb).

Es en los inicios del protocolo HTTP, a mediados del año 1990, cuando encontramos la versión 0.9. Esta versión tenía como único fin transferir datos por Internet en forma de páginas web escritas en lenguaje de marcado de hipertexto (HTML, Hyper Text Markup Language). A partir de la versión 1.0 del protocolo surgió la posibilidad de transferir mensajes con encabezamientos que describían el contenido de los mensajes. El protocolo de transferencia de hipertexto (HTTP) es un protocolo cliente-servidor bastante sencillo que articula los intercambios de información entre los clientes web y los servidores HTTP. HTTP fue desarrollado por el consorcio W3C y la IETF. Esta colaboración culminó en 1999 con la publicación de una serie de RFC, el más importante de los que fue el RFC 2616, que especificaba la versión 1.1. Desde el punto de vista de las comunicaciones, está soportado en los servicios de conexión TCP/IP y funciona de la misma forma que el resto de servicios en red.

10. El protocolo HTTP.

Técnicamente, un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el **80**) y espera las solicitudes de conexión de los clientes web.

Una vez establecida la conexión, el **protocolo TCP** se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores. El protocolo de transferencia de hipertexto se basa en operaciones sencillas de solicitud/respuesta.

Cuando un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud, el servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto web (documento **HTML**, archivo multimedia o aplicación CGI) es conocido por su uniforme de recursos (**URL**, uniform resource locator). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta en una base de datos, la traducción automática de un documento, etc.

10. El protocolo HTTP.

El estándar HTTP/1.0 recoge únicamente tres órdenes que representan las operaciones de recepción, envío de la información y revisión del estado.

- 1. GET.** Se utiliza para recoger cualquier tipo de información del servidor. Siempre que pulsamos sobre un enlace o escribimos una dirección de Internet en la barra de navegación de un navegador web, estamos utilizando este pedido. Como resultado, el servidor HTTP envía el documento correspondiente a la dirección de Internet seleccionada o activa un módulo, a su vez, generará la información de vuelta.
- 2. HEAD.** Solicita información sobre un objeto (archivo) como, por ejemplo, su tamaño, tipo, fecha de modificación, etc. Suele ser utilizado por los gestores de caché de páginas o por los servidores intermediarios (proxy server) para conocer cuándo es necesario actualizar la copia que se mantiene de un archivo determinado.
- 3. POST.** Se utiliza para enviar información al servidor como, por ejemplo, los datos contenidos en un formulario. El servidor pasará esta información a un proceso encargado de su tratamiento. La operación que se llevará a cabo con la información proporcionada dependerá de la dirección de Internet (URL) utilizada.

11. Estructura y recursos que componen una aplicación web.

- Descriptor de despliegue.
- Archivos jar, war y ear.
- Maven.

Ver pdf independiente

12. Host virtuales. Creación, configuración y utilización.

Anteriormente hemos visto como poder alojar múltiples páginas web en el servidor web Apache, pero todas pertenecientes al mismo sitio/dominio, es decir, todas pertenecientes a blanca.com, entonces, ¿no se puede alojar páginas de distintos dominios en el mismo servidor web? La respuesta es que si, si se puede, ¿cómo?, mediante la configuración de hosts virtuales o virtualhosts.

Éstos básicamente lo que hacen es permitir que un mismo servidor web pueda alojar múltiples dominios, así configurando hosts virtuales podemos alojar: paco.com, elvira.com,... en el mismo servidor web. Cada empresa tendrá su virtualhost único e independiente de las demás.

12. Host virtuales. Creación, configuración y utilización.

Aunque cada virtualhost es único e independiente de los demás, todo aquello que no esté incluido en la definición de cada virtualhost se heredarán de la configuración principal, por ejemplo, en apache: `/etc/apache2/apache2.conf`.

O sea, si quieres definir una directiva común en todos los virtualhost no debes modificar cada uno de los virtualhost introduciendo esa directiva sino que debes definir esa directiva en la configuración principal del servidor web Apache, de tal forma que todos los virtualhost heredarán esa directiva, por ejemplo en `apache2.conf` puedes encontrar la directiva `Timeout 300`, que indica el número de segundos antes de que se cancele una conexión por falta de respuesta.

-> **Realizad práctica asociada.**

13. Módulos: Instalación, configuración y uso.

Caso práctico real: Debes configurar un nuevo servidor web en Apache en tu empresa y necesitas montar:

- **Varias aplicaciones web** atiendan en el mismo dominio, tal que:
sucursal-zonaX.empresa-proyecto.com, www.sucursal-zonaX.empresa-proyecto.com, etc.
- Un único **panel de control** de usuarios, en la URL www.empresa-proyecto.panel-de-control.com.
- También soporte **SSL** para cifrado.
- Soporte para páginas dinámicas mediante PHP.
- Y soporte para control de usuarios **LDAP**.

Por lo tanto, **tienes que montar Apache con varios módulos.**

13. Módulos: Instalación, configuración y uso.

La importancia de un servidor web radica en su: **estabilidad, disponibilidad y escalabilidad**. Es muy importante poder dotar al servidor web de nuevas funcionalidades de forma sencilla, así como del mismo modo quitárselas. Si que necesitas soporte SSL puedes configurar el módulo SSL, si necesitas soporte PHP el módulo PHP, hay soporte para LDAP, etc.

En Ubuntu, existen dos comandos fundamentales para el funcionamiento de los módulos en el servidor web Apache: **a2enmod** y **a2dismod**.

13. Módulos: Instalación, configuración y uso.

- **a2enmod**: Utilizado para habilitar un módulo de apache. Sin ningún parámetro preguntará que módulo se desea habilitar. Los ficheros de configuración de los módulos disponibles están en `/etc/apache2/mods-available/` y al habilitarlos se crea un enlace simbólico desde `/etc/apache2/mods-enabled/`.
- **a2dismod**: Utilizado para deshabilitar un módulo de Apache. Sin ningún parámetro preguntará que módulo se desea deshabilitar. Los ficheros de configuración de los módulos disponibles están en `/etc/apache2/mods-available/` y al deshabilitarlos se elimina el enlace simbólico desde `/etc/apache2/mods-enabled/`.

Si no dispones de esos comandos para poder habilitar y deshabilitar módulos Apache simplemente haces lo que ellos: crear los enlaces simbólicos correspondientes desde `/etc/apache2/mods-enabled/` hasta `/etc/apache2/mods-available/`.

14. Autenticación y control de acceso a directorios.

En la mayoría de aplicaciones web interesa controlar el acceso hacia un servicio a través de la web. Para este tipo de casos tenemos que pensar en la autenticación y el control de acceso.

Un **sistema de autenticación** verifica la identidad de un usuario para dejarle acceder o no a un servicio, en este caso al servicio web. Si se establece la autenticación en un sitio web, al acceder al sitio web desde un navegador, se pide un nombre de usuario y una contraseña. Si son válidos, se envía al navegador la página web solicitada y si no es así, se envía una respuesta HTTP de error por fallo en la autenticación. Otros sistemas de autenticación más avanzados usan los **certificados** en lugar de nombres de usuario y contraseñas.

14. Autenticación y control de acceso a directorios.

Un **sistema de control de acceso** comprueba la identidad del ordenador cliente que solicita un recurso, y establece limitaciones sobre el acceso al recurso en función de la identificación del ordenador. Sobre un mismo recurso, se puede establecer un sistema de control de acceso y un sistema de autenticación. El control de acceso controla la máquina cliente, mientras que la autenticación controla al usuario cliente. Para el acceso a un recurso, el servidor Apache realiza primero el control de acceso y después la autenticación.

Directorio privado

Prueba de autentificación en Apache con diversos metodos.

Si estas viendo esto es que tienes acceso. Felicidades.



The screenshot shows a standard Windows-style dialog box titled "Ventana de autorización". It contains a key icon and the text: "Debe suministrar un nombre de usuario y una contraseña para acceder a este sitio." Below this, there are two input fields: "Nombre de usuario:" with the text "josemaria" and "Contraseña:" with a masked password of ten dots. To the right of the password field is a checkbox labeled "Recordar la contraseña". At the bottom right, there are two buttons: "Accesar" (with a key icon) and "Cancelar" (with a red X icon).

14. Autenticación y control de acceso a directorios.

Hay tres sistemas de autenticación:

- **Autenticación básica con ficheros:** el mecanismo más simple para implementar el control de acceso a recursos de una sede web es utilizar archivos de usuarios y grupos propios del servidor. Apache proporciona herramientas para crearlos. La ventaja principal de este método es la facilidad de administración. El inconveniente es que conlleva una gestión diferenciada de los usuarios del servicio web y los del sistema. De hecho, esto puede ser un inconveniente o una ventaja, si lo que interesa es tenerlos segregados.
- **Autenticación mediante PAM:** en los sistemas GNU / Linux actuales la autenticación de los usuarios se hace vía PAM (Pluggable Authentication Module). El PAM comprueba el directorio / etc / passwd, el LDAP, el Kerberos, las huellas dactilares o lo que sea necesario. Usar el vínculo con el módulo del PAM es un buen mecanismo para validar los usuarios del servicio web al igual que se validan los usuarios del sistema.

14. Autenticación y control de acceso a directorios.

- **Autenticación mediante LDAP** : uno de los mecanismos más populares actualmente para la autenticación es el LDAP. El módulo del LDAP permite pasar la validación de los usuarios al encargado de gestionar la autenticación LDAP de los usuarios del sistema. También se puede tener en funcionamiento un servicio LDAP específico para las validaciones del servicio web.

HTTP proporciona un método de autenticación básico de usuarios: **basic**. Este método ante una petición del cliente (navegador web) al servidor cuando se solicita una URL mostrará un diálogo pidiendo usuario y contraseña. Una vez autenticado el usuario, el cliente volverá a hacer la petición al servidor pero ahora enviando el usuario y contraseña, en texto claro (sin cifrar) proporcionados en el diálogo. Es recomendable entonces si empleas este método que lo hagas combinado con conexión SSL (HTTPS).

15. Certificados, Servidores de certificados. El protocolo HTTP -> HTTPS.

Certificados. Servidores de certificados

HTTPS es la encriptación que obtienes cuando tu web utiliza un certificado como el SSL. **SSL** son las siglas de Secure Sockets Layer (capas de puerto seguras). Características:

- 1) Encripta los datos de manera que si algo estuviese en medio de ese flujo de datos e intentase leerlos, no podría hacerlo. Podría ver el paquete de datos, eso sí, pero no entenderlos puesto que están encriptados.
- 2) Garantiza la integridad de los datos mientras se transfieren entre el servidor y el cliente ya que no pueden modificarse ni intencionalmente ni por error.
- 3) Provee autenticación comunicando a los usuarios que tu aplicación web es realmente la que dices que es y así el usuario puede estar seguro de que esa web es segura y no una copia creada por alguien para phishing, robos de identidad, etc.

15. Certificados, Servidores de certificados. El protocolo HTTP -> HTTPS.

Certificados. Servidores de certificados

¿Cómo conseguimos el certificado para nuestro servidor web? ¡Lo practicaremos!

Necesitas activar el certificado SSL que te ofrezca tu hosting. Para ello, de normal basta con acceder al panel de control del alojamiento de la web contratado y buscad el apartado de “Seguridad”. Allí deberías encontrar un sub apartado llamado “Certificados SSL” (o algo parecido). Accede a ese apartado y activa el certificado. Suele ser un servicio gratuito y se hace en un solo clic. De normal con estas acciones, se instala y al navegar hacia tu servidor web, ya verás que el http cambia a https directamente.

Si tu hosting no ofrece este servicio, puedes encontrarlo soluciones en:

- <https://letsencrypt.org/>
- <https://www.sslforfree.com/>

16. Pruebas de funcionamiento, monitorización y rendimiento del servidor web.

Tan importante como es configurar un servidor web lo es **mantener** y **comprobar** su **correcto funcionamiento**, y para ello debes ayudarte de los logs o archivos de registro que te permiten revisar y estudiar su funcionamiento. El administrador de un servidor web debe conocer donde registra un servidor los errores que se produzcan, y donde monitoriza la actividad en el servidor y los accesos de los clientes. Esta información se almacena en archivos (aunque normalmente se los llama registros). Además un administrador debe ser capaz de analizar el contenido de esos archivos y sacar conclusiones que le permitan mejorar las prestaciones del servidor.

16. Pruebas de funcionamiento, monitorización y rendimiento del servidor web.

En el servidor web Apache se tienen principalmente los siguientes archivos de registro:

- 1. Registro de errores:** por defecto, en Debian/Ubuntu, es el archivo `/var/log/apache2/error.log`. De cada error o fallo registrado, se indica la fecha y hora, la gravedad del error, la dirección IP del cliente que provocó el error (si es el caso), el mensaje de error y el nombre del recurso solicitado por el cliente (si es el caso). En este archivo hay varios avisos que no indican errores graves pero que alertan de anomalías que deberían solucionarse.
- 2. Registro de accesos:** por defecto, en Debian/Ubuntu, es el archivo `/var/log/apache2/access.log`. Monitoriza o registra información sobre todas las peticiones de clientes procesadas por el servidor. De cada petición registra la IP del cliente, la fecha y hora de la petición y el mensaje de petición del recurso. Para cada servidor virtual hay un registro de acceso. Estos registros de acceso tienen nombres `other_vhosts_Access`.

16. Formato de los archivos de registro (logs).

Apache permite mediante diversas directivas crear archivos de registro que guardarán la información correspondiente a las conexiones con el servidor. Esta información es guardada en formato **CLF** (Common Logon Format) por defecto. Ésta es una especificación utilizada por los servidores web para hacer que el análisis de registro entre servidores sea mucho más sencillo, de tal forma que independientemente del servidor web utilizado podamos emplear el mismo método de análisis de registro, ya sea mediante lectura, mediante programas ejecutables (scripts) o mediante programas propios de análisis de registro.

En un archivo de registro en formato CLF cada línea identifica una solicitud al servidor web. Esta línea contiene varios campos separados con espacios. Cada campo sin valor es identificado con un guión (-).

16. Formato de los archivos de registro (logs).

| Campos (especificadores) | Definición | Ejemplo |
|-----------------------------|--|------------------------------|
| host (%h) | Identifica el equipo cliente que solicita la información en el navegador. | 192.168.200.100 |
| ident (%l) | Información del cliente cuando la máquina de éste ejecuta identd y la directiva IdentityCheck está activada. | |
| authuser (%u) | Nombre de usuario en caso que la URL solicitada requiera autenticación HTTP. | |
| date (%t) | Fecha y hora en el que se produce la solicitud al servidor. Va encerrado entre corchetes. Este campo tiene su propio formato: [dia/mes/año:hora:minuto:segundo zona] | [05/May/2011:17:19:18 +0200] |
| request (%r) | Petición del cliente, esto es, la página web que está solicitando. En el ejemplo: /index.html, esto es, dentro de la raíz del dominio que se visite la página | /index.html |
| status (%s ó %>s) | Identifica el código de estado HTTP de tres dígitos que se devuelve al cliente. | 200 |
| Bytes (%b) | Sin tener en cuenta las cabeceras HTTP el número de bytes devueltos al cliente. | 20 |

16. Directivas para archivos de registro.

El contexto de aplicación de todas las directivas que se indican a continuación en la siguiente tabla puede ser el de la configuración principal del servidor así como el de la configuración de los host virtuales.

Directivas para archivos de registro.

| Directivas | Definicion |
|-------------|---|
| TransferLog | Directiva que define el nombre del archivo de registro o al programa al que se envía la información de registro. Emplea los especificadores asignados por la directiva LogFormat. |
| LogFormat | Directiva que define el formato del archivo de registro asignado con la directiva TransferLog |
| ErrorLog | Directiva que permite registrar todos los errores que encuentre Apache. Permite guardar la información en un archivo de registro o bien en syslog |
| CustomLog | Directiva similar a la directiva TransferLog, pero con la particularidad que permite personalizar el formato de registro empleando los especificadores anteriormente vistos. |
| CookieLog | Directiva que define el nombre del archivo de registro donde registrar información sobre cookies |

16. Rotación de los archivos de registro.

Como los archivos de registro a medida que pasa el tiempo van incrementando su tamaño, debe existir una política de mantenimiento de registros para que éstos no consuman demasiados recursos en el servidor, así es conveniente rotar los archivos de registro, esto es, hay que depurarlos, comprimirlos y guardarlos. Básicamente tienes dos opciones para rotar tus registros: **rotatelogs** un programa proporcionado por Apache, o **logrotate**, una utilidad presente en la mayoría de los sistemas GNU/Linux.

No debes olvidar que la información recopilada en los **ficheros log** se debe conservar al menos durante 1 año por eventuales necesidades legales, de este modo, además de rotarlos se opta habitualmente por **comprimir logs**.



17 - Prácticas

Ver guías y prácticas en Aula Virtual.