

Instalación y administración de servidores de ficheros de alta disponibilidad y alto rendimiento.



Caso práctico

La empresa BK Programación se ha dedicado últimamente a la creación de entornos web en servidores dedicados y compartidos. Siempre se ha decantado para la instalación, administración y configuración de servidores de software libre. Y es por ello que varias empresas se han puesto en contacto con BK Programación para contratarles sus servicios. Una de estas empresas con las que trabaja BK Programación tuvo un problema subiendo un archivo a su dominio, por lo cual se ha recibido una llamada dirigida a soporte técnico en BK Programación. En ésta se mantuvo la siguiente conversación:



- Hola, buenos días. BK Programación, ¿en qué puedo ayudarle?
- Hola, buenos días. Tengo un problema en el servidor de nuestra página, no puedo subir un archivo de vídeo.
- ¿No le aparece la opción de subida o, en medio del proceso, se le corta la conexión?
- Sí, soy capaz de empezar a subir el archivo pero, pasado un tiempo, el proceso se corta.
- ¿Cuánto pesa el archivo? Es decir, ¿qué tamaño posee?
- No sé, espere un momento... —pasado un tiempo—, sí..., mire, el archivo ocupa 300 MB.
- Vale, parece claro, tal como está procediendo hasta ahora no podrá subir el archivo, debido a la limitación establecida en el servidor web para la subida de archivos, por lo tanto debe utilizar su cuenta ftp para la transferencia de archivos.
- Y eso, ¿cómo procedo? ¿está estipulado en el contrato?
- Sí, no se preocupe. El contrato estándar ya establece una cuenta ftp por dominio, pero eso sí, dependiendo del contrato poseerá una cuota de disco u otra. En cuanto al método para proceder, aparece explicado en nuestra página web, paso por paso, en la documentación que podrá encontrar en la pestaña descargas.
- Pues, poseo el contrato estándar.

- Bien, entonces posee una cuota de 2 GB.
- Vale, gracias, entonces ¿cuándo podré contar con la cuenta ftp para subir el archivo?
- Ya la tiene operativa, solamente debe seguir los pasos del documento que le he comentado. Si tiene cualquier problema no dude en contactar de nuevo con nosotros.
- De acuerdo, muchas gracias. Hasta luego.
- Hasta luego.

Este tipo de incidencias son típicas en BK Programación, por lo cual Ada, la directora de BK Programación ha establecido un protocolo de actuación según el tipo de incidencia. Las incidencias primero serán atendidas por una unidad de servicio telefónico, en caso de que no se encuentre la solución dentro de las posibles será escalada a su correspondiente área técnica, así las incidencias de servidores serán escaladas a María, las de programación a Juan y éstos derivarán la incidencia a un técnico de la empresa. En el caso de la incidencia por llamada telefónica anterior la solución ya existía dentro de las posibles con lo cual la incidencia no fue escalada.

Las posibles soluciones fueron proporcionadas por el personal responsable para cada área. Así como la incidencia que hablamos era sobre servidores fue propuesta la solución por María. Además, en este caso, María había estudiado varios servidores ftp y se decantó por la versatilidad, funcionalidad y seguridad del servidor ftp ProFTPD.

Pero antes de instalar un servidor tan importante en la estructura de red de la empresa, Ada le pidió a María que previamente hiciera un estudio básico de las soluciones de alta disponibilidad software que se podrían implementar en BK Programación para asegurar la calidad del servicio ante problemas.



Materiales actualizados por el profesorado de la Junta de Andalucía

Jesús Manuel Marín Navarro



[Aviso legal](#)



**Materiales desarrollados inicialmente por el Ministerio de Educación,
Cultura y Deporte**

[Aviso Legal](#)

1.- Introducción a la Alta disponibilidad y alto rendimiento.



Caso práctico

María, que ya sabía los planes de la empresa para implementar un servicio de Voz sobre IP, decidió no solo buscar las soluciones de alta redundancia software para su servicio de FTP, sino también de paso que sirvieran para futuros servicios dónde la alta disponibilidad sería muy importante como el de Voz sobre IP o Telefonía IP.



En una primera investigación sobre la alta disponibilidad se informó sobre los costes de las soluciones de alta disponibilidad a nivel de hardware, así como las soluciones software propietarias de empresas consagradas. En ambos casos el costo (e incluso la complejidad) era demasiado grande como para que mereciera la pena. Así que comenzó su investigación sobre las soluciones de alta disponibilidad software basadas en software libre. Que de paso se complementarían perfectamente con el resto de infraestructura ya montada en BK Programación y además le serviría como modelo de desarrollo de nuevos servicios para posibles futuros clientes que demandaran alta disponibilidad en sus servicios locales.

De igual forma, para el caso de que hubiera ancho de banda suficiente, también se ha propuesto investigar las soluciones de alto rendimiento software ya que en muchos casos, ambas soluciones se complementan o se desarrollan de igual forma.

Los usuarios habituados a la red de telefonía tradicional son conscientes de que la probabilidad de fallo en la línea telefónica a la hora de realizar una llamada es muy escasa, ya que la fiabilidad de la red telefónica tradicional puede llegar a ser superior al 99'99%.

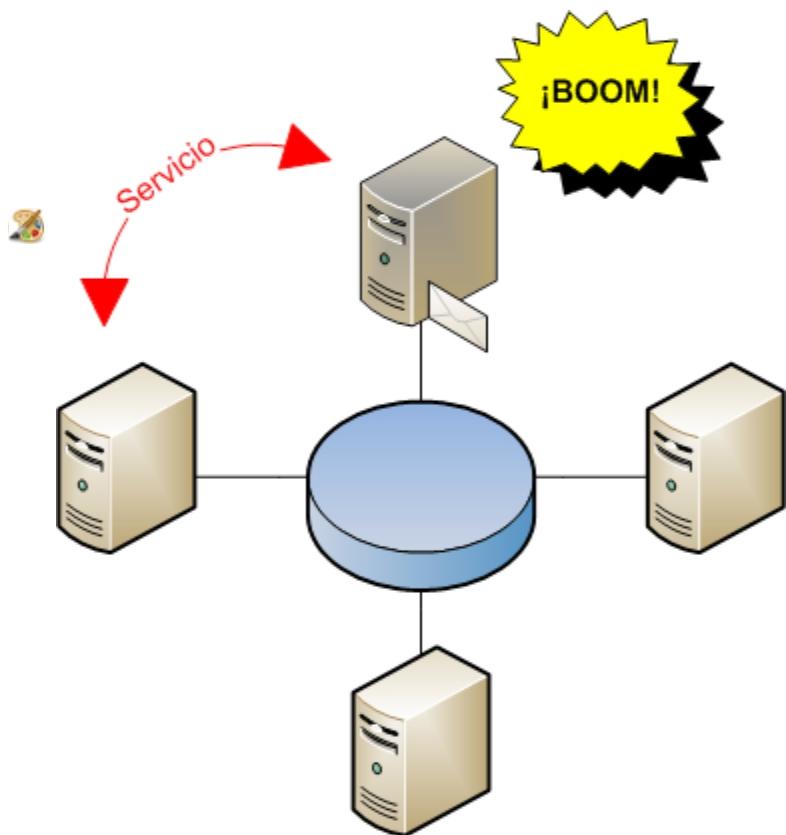
Dada la importancia de asegurar las comunicaciones en una empresa, es muy importante asegurar que el servicio VoIP funcione de forma ininterrumpida. En definitiva, por un motivo u otro siempre necesitamos asegurar que el servicio de telefonía se encuentra disponible y además, que funcione de manera rápida y eficiente.

Para conseguir este objetivo (24 horas al día y 7 días a la semana) en GNU/Linux, es necesario emplear diferentes técnicas y herramientas que permitan alta disponibilidad y alto rendimiento.

La alta disponibilidad permite que un servicio funcione correctamente ante un fallo software o hardware. De esta forma, aunque se produzca un fallo en el sistema, el servicio seguirá funcionando de forma transparente para los usuarios como si el fallo no hubiera ocurrido. El fallo de un servicio que no dispone de alta disponibilidad provocará una parada de estos, teniendo como resultado unas consecuencias muy negativas en la empresa.

La alta disponibilidad se puede implementar mediante una configuración basada en hardware o software. Una solución hardware trata de asegurar que el servidor funcione de forma ininterrumpida y para ello suele utilizar sistemas redundantes de alimentación, discos duros (RAID), de red, etc. De esta forma, si falla cualquiera de esos elementos hardware, el sistema funcionará correctamente y lo único que tendremos que hacer es reemplazar el dispositivo defectuoso en "caliente".

Una configuración basada en software consiste en una serie de servidores, denominados nodos conectados entre sí de tal manera que, ante un fallo hardware/software, el servicio o servicios ofrecidos por el nodo fallido son retomados por otro nodo del clúster. De esta manera el servicio que se ofrece sigue en funcionamiento de manera casi ininterrumpida. En la siguiente figura se muestra un clúster de alta disponibilidad basado en software.



El alto rendimiento permite distribuir la carga de un servicio entre varios equipos consiguiendo así un mayor rendimiento e impidiendo que se sature un servidor. Para permitir el alto rendimiento se crea un cluster que está compuesto por nodos. Cada nodo se encarga de realizar una serie de tareas que se van distribuyendo dependiendo de diversos factores: equipos activos, rendimiento etc..

Esta solución ofrece dos características muy importantes. La primera de ellas es que evita la saturación de una máquina. En un entorno empresarial, por lo general, la máquina que ofrece, por ejemplo, un servicio web tiende a saturarse durante determinados momentos del día debido al acceso masivo de los visitantes. Si la saturación alcanza picos muy elevados, la máquina podría fallar y el servicio pararse de manera inesperada. Una solución de alto rendimiento permitiría la posibilidad de repartir la carga entre varios servidores encargados de ofrecer el servicio web. Esta solución también se le conoce como balanceo de carga.

La segunda característica es que los recursos son gestionados de manera inteligente, ya que las tareas se van distribuyendo entre los nodos para impedir que se sobrecarguen.

En definitiva, las soluciones de alto rendimiento ofrecen: mayor capacidad de gestión, escalabilidad y ampliación así como mayor disponibilidad para los servicios de entorno empresarial.

Por otro lado, las soluciones de alta disponibilidad ofrecen: tolerancia a fallos, flexibilidad y tranquilidad a la empresa.



Reflexiona

Si queremos alcanzar una alta disponibilidad mediante la redundancia de componentes como fuentes de alimentación, discos duros, ventilación, tarjetas de red, etc. así como la inversión de elementos hardware de calidad y más fiables, ¿cómo crees que afecta al coste económico y a la complejidad técnica?

[Mostrar retroalimentación](#)

1.1.- Alta disponibilidad en GNU/Linux.

En los sistemas GNU/Linux existen muchas herramientas que nos permiten ofrecer alta disponibilidad de servicios. Dichas herramientas las podemos clasificar en las categorías de software y hardware. Las soluciones de alta disponibilidad basadas en software están destinadas a ofrecer alta disponibilidad en servicios y alta disponibilidad en datos.

- ✓ **Servicios.** Las soluciones de alta disponibilidad enfocadas a servicios son especialmente utilizadas en servicios críticos. Normalmente, estos servicios suelen ir acompañados de ciertos complementos requeridos para un correcto funcionamiento como: direccionamiento IP, reglas de firewall, copias de seguridad, etc. Por ejemplo, la mayoría de las empresas requieren de la utilización del servicio de "directorio activo", por lo que éste se convierte en un servicio crítico. Una configuración de alta disponibilidad podría consistir, por ejemplo, en dos nodos donde sólo uno de ellos dispone de una dirección IP Virtual, mediante la cual las aplicaciones acceden al directorio activo. En caso de fallo del nodo que ofrece el servicio, la dirección IP Virtual se establece en el otro nodo (nodo secundario o backup) lo que permite a los clientes y aplicaciones seguir accediendo al directorio activo como si no hubiera ocurrido nada. Algunas de las herramientas GNU/Linux enfocadas a la alta disponibilidad en servicios son: "KeepAlive" [KeepAlive], "Heartbeat" [Heartbeat], "Veritas Cluster" [VeritasCluster], "HP Service Guard" [HPServiceGuard], etc.
- ✓ **Datos.** Las soluciones de alta disponibilidad enfocadas a datos son fundamentales y por lo general es necesaria su integración con los sistemas de alta disponibilidad enfocados a servicios. El objetivo que se persigue con la alta disponibilidad enfocada a datos es mantener en varias localizaciones, una serie de datos que no presentan problemas de consistencia ni integridad entre ellos.

Autoevaluación

Si tenemos un servidor NAS con 2 discos duros replicados mediante RAID 1, ¿tendremos alta disponibilidad del servicio?

- Verdadero Falso

1.1.1.- Replicación de servicios.

En GNU/Linux existe una gran variedad de herramientas para ofrecer alta disponibilidad en los servicios. A continuación se relacionan algunas de las herramientas más importantes:

- ✓ **HA-OSCAR.** Es un proyecto Open Source basado en un sistema de clustering Beowulf enfocándose, por tanto, a aplicaciones que requieren un grado elevado de disponibilidad y alto rendimiento. HA-OSCAR tiene en cuenta los puntos débiles, por lo que adopta redundancia de componentes. HA-OSCAR incorpora detección de fallos, mecanismos de recuperación, etc.
- ✓ **The High Availability Linux Project.** Es la solución más extendida y proporciona una solución de alta disponibilidad ofreciendo fiabilidad y disponibilidad. Se integra perfectamente con multitud de servicios y aún sigue mejorando dicha integración gracias a la aportación de su activa comunidad.
- ✓ **Kimberlite.** Es considerado un proyecto único, ya que es una infraestructura completa de alta disponibilidad y de código abierto que incluye incluso garantía sobre la integridad de los datos. Actualmente es un proyecto abandonado o de muy baja actividad.
- ✓ **LifeKeeper.** Es una alternativa moderna y flexible comparada con los productos que ofrecen soluciones de alta disponibilidad. Dispone de mecanismos para mantener la integridad de los datos. Una numerosa colección de kits de recuperación para los distintos servicios y/o aplicaciones permiten instalar LifeKeeper en cuestión de horas. Es un producto de la empresa SteelEye.
- ✓ **HP Serviceguard.** Es un software especializado para ofrecer alta disponibilidad a las aplicaciones más críticas ante fallos tanto hardware como software. HP Serviceguard monitoriza el estado de cada nodo y responde rápidamente en caso de fallo permitiendo minimizar el tiempo durante el cual no se está ofreciendo el servicio. Está disponible tanto para Linux como para HP-UX.
- ✓ **Red Hat Cluster Suite.** Ha sido diseñado específicamente para Red Hat Enterprise Linux y permite alta disponibilidad y balanceo de carga. Dispone de soluciones listas para implantar para la mayoría de los servicios, pudiendo además proporcionar soporte cuando sea necesario.
- ✓ **Veritas Cluster Server.** Es un producto de la empresa Symantec, una solución clustering de alta disponibilidad para la industria.
- ✓ **KeepAlive.** El objetivo principal del proyecto es el de proporcionar alta disponibilidad a proyecto Linux Virtual Server. KeepAlive es un servicio que monitoriza el estado del clúster LVS para que en caso de fallo el cluster siga en funcionamiento.

1.1.2.- Replicación de datos.

En **GNU/Linux** existen múltiples herramientas que ofrecen alta disponibilidad de datos. Esta herramientas se pueden clasificar en varias categorías dependiendo de los datos que replica bloques de datos, bases de datos en MySQL o servidores NAS .

La **replicación de bloque de datos** consiste en replicar la información bloque a bloque de sistema de ficheros. Una de las herramientas que más se utilizan para la replicación de datos es **DRBD** (*Distributed Replicated Block Device*). DRBD permite una replicación en tiempo real de manera continua y transparente. La principal limitación que presenta DRBD se produce cuando los cambios se están realizando en más de una localización a la vez, ya que la “unión” de los datos de más de una localización no es posible pues sólo se permite tomar una de las localizaciones como modelo de copia (master), siendo el resto esclavos (slaves). Otra herramienta de este tipo es rsync una aplicación para sistemas de tipo Unix que ofrece transmisión eficiente de datos incrementales comprimidos y cifrados. Finalmente, **slony-I** es un sistema de replicación de “maestro a múltiple esclavos”, el cuál además soporta el concepto de “cascada” (un nodo puede proveer a otro nodo el cual puede proveer a otro) permitiendo así la recuperación ante errores. Incluye las principales características para replicar bases de datos de gran información a un número límite razonable de esclavos. Es por tanto, un sistema diseñado para centros de datos y sitios de backup donde el funcionamiento normal requiere que todos los nodos estén disponibles en un momento determinado.

La **replicación de bases de datos (MySQL)** permite replicar una base de datos entre varios servidores MySQL. A diferencia de DRBD, la replicación se lleva a cabo sentencia a sentencia. Gracias a la flexibilidad de diseño que permiten las bases de datos, actualmente es posible realizar modificaciones en las mismas para más de una localización. Flexibilidad presente en las funcionalidades como la Replication de MySQL y sus atributos como auto_increment_incremer y auto_increment_offset permiten superar la limitación que tienen las soluciones basadas en el bloque de datos. En GNU/Linux se dispone también de otras muchas herramientas de replicación de base de datos como Slony-I, además de la replicación que ofrece Oracle, entre muchos otros.

La utilización de un **servidor NAS** (Network-Attached Storage) permite un nivel de replicación de datos superior a las otras soluciones ya que permite crear una unidad de red en la que se guardan todos los datos. Estas unidades suelen tener varios discos duros formando un RAID1, RAID5, etc. Algunas de las herramientas que permiten montar un servidor NAS son:

- ✓ **FreeNAS.** FreeNAS es basado en FreeBSD. Entre sus características más importantes están el soporte de una gran variedad de sistemas de ficheros, sistemas RAID (0, 1 y 5), autenticación de usuarios, interfaz web de configuración, etc.
- ✓ **Openfiler.** Es un sistema operativo que permite crear un sistema NAS. Provee de buena integración con iSCSI (Internet SCSI) y “fibre channel”, muy útiles en entornos empresariales y en virtualización tal como Xen y VMware. Cuenta con una interfaz Web para su administración y soporta clientes de todos los sistemas operativos.
- ✓ **NAS Lite-2.** Es un sistema operativo que permite crear una unidad NAS. Su principal característica es que está optimizado para ofrecer el máximo rendimiento, soporta múltiples sistemas de ficheros y lo más sorprendente es que se ejecuta directamente sobre la RAM necesitando únicamente 8MB de disco.

En la tabla del apartado 1.2.1 se muestra un resumen de las herramientas que permiten alta disponibilidad.



Para saber más

Openfiler con iSCSI, DRBD y Heartbeat

[Descripción del video](#)

1.2.- Alto rendimiento en GNU/Linux.

En los sistemas *GNU/Linux* existen muchas herramientas que permiten ofrecer alto rendimiento para servicios que tienen que soportar una gran carga de trabajo. Dichas herramientas se pueden utilizar de forma general para cualquier servicio (p.e. Web, correo) o de forma individual, como por ejemplo, para telefonía IP.

1.2.1.- Soluciones Generales.

Al hablar de una solución de alto rendimiento general nos referimos a una solución aplicable a servicios, que por lo general, son propensos a gestionar un gran volumen de peticiones, lo que obliga a dividir la carga. Una de las aplicaciones GNU/Linux más utilizada es la solución de Linux Virtual Server o LVS, una avanzada solución de balanceo de carga con la finalidad de dotar a los servicios como la Web, Cache, Mail, Ftp, etc, de alto rendimiento y escalabilidad.

Algunos de los proyectos destinados a ofrecer alto rendimiento en servicios son:

- ✓ **Linux Virtual Server.** Con él se puede construir un cluster de alto rendimiento, mediante un servidor dedicado a balancear la carga. La arquitectura del cluster es totalmente transparente para los usuarios, interactuando estos como si de un único servidor se tratase. Es uno de los proyectos más utilizados.
- ✓ **Linux Virtual Server Manager.** Es un paquete que simplifica la creación y mantenimiento de cluster basados en *LVS*. Permite administrar el cluster vía *Web* y modificar la configuración del cluster en tiempo real generando además diversas estadísticas sobre el funcionamiento del cluster.
- ✓ **Red Hat Cluster Suite.** Ha sido diseñado específicamente para *Red Hat Enterprise Linux*. Proporciona además de alta disponibilidad, balanceo de carga. Dispone de soluciones lista para ser implantadas por la mayoría de los servicios.
- ✓ **UltraMonkey.** Es un proyecto que dota a los servicios tanto de alta disponibilidad como de alto rendimiento, ya que integra los proyectos *Heartbeat* y *LVS*. *UltraMonkey* hace uso de *GNU/Linux* para proveer una solución fácilmente adaptable a un gran número de necesidades.
- ✓ **Bind9.** *Berkeley Internet Name Domain* es el servidor de *DNS* más comúnmente usado en Internet, especialmente en sistemas *Unix*, en los cuales es un estándar de facto.

1.2.2.- Soluciones para VoIP.

El protocolo *SIP*, definido en el *RFC3261*, es actualmente uno de los protocolos de sesión con mayor auge en VoIP. En *GNU/Linux* existen varias herramientas cuya funcionalidad es la de implementar un *proxy SIP* de alto rendimiento y muy configurable. Unas de estas primeras aplicaciones fue *SER (SIP Express Router)*, responsabilidad de *iptel.org*. Posteriormente y debido a problemas internos en la organización, se creó *OpenSER*. Posteriormente *OpenSER* pasó a denominarse *KAMAILIO*, manteniendo incluso la misma comunidad, y al mismo tiempo se creó un nuevo proyecto a partir de *OpenSER*, denominado *OpenSIPS*.

- ✓ **SER.** Es un proyecto libre, configurable y de muy alto rendimiento bajo la licencia *GNU*. *SER* soporta e implementa todas las funcionalidades descritas en el *RFC3261* (soporta el uso de diversas bases de datos, *NAT*, multidominio, etc).
- ✓ **Kamailio.** Es el proyecto *OpenSER*
- ✓ **OpenSIPs.** Es un *fork* de *OpenSER*
- ✓ **SIP-Router.** *Kamailio + SER*. Unificación de los proyectos *KAMAILIO* y *SER*.

En la tabla 1 se muestra un resumen de las herramientas comentadas.

Tabla 1. Herramientas de alta disponibilidad y alto rendimiento

Nombre	Alta disponibilidad	Alto rendimiento	Características	Licencia	
DRBD	X		De datos	GPLv2	
FreeNAS	X	X	De NAS	BSD	
Openfiler	X	X	De SAN-NAS	GPLv2	
NAS Lite-2	X	X	De NAS	Propietario	www.sagivsoft.com
Mdadm	X	X	De RAID	GPLv2	http://mdadm.org
MySQL Replication	X		De base de datos	GPLv2	http://dev.mysql.com/doc/refman/5.1/en/replication.html
Slony-I	X		De base de datos	BSD	
HA-OSCAR	X		De servicio	GPL	http://www.oscar-project.org
Linux-HA	X		De servicio	GPL	
Kimberlite	X		De servicio	GPL	www.mirantis.com
LifeKeeper	X		De servicio	Propietario	

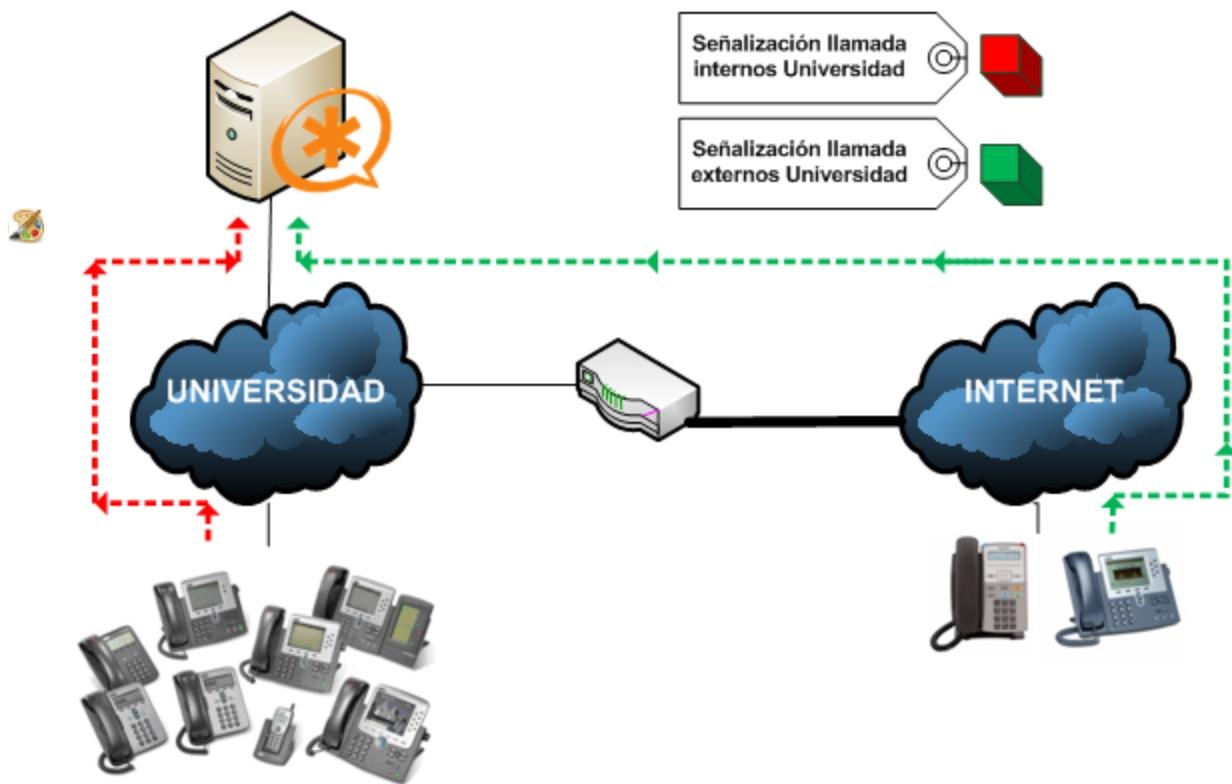
HP Serviceguard	X		De servicio	Propietario	https://do
RedHat Cluster Suite	X	X	De servicio	Propietario	http://www.redhat.com
Veritas Cluster Server	X		De servicio	Propietario	www.symantec.com
KeepAlive	X		De servicio	GPL	
Linux Virtual Server		X	De servicio	GPL	
Linux Virtual Server Manager		X	De servicio	GPL	
UltraMonkey	X	X	De servicio	GPL	
Bind9		X	De servicio	GPL	
SER		X	Específico VoIP	GPL	
Kamailio		X	Específico VoIP	GPL	
OpenSIPS		X	Específico VoIP	GPL	
SIP-Router		X	Específico VoIP	BSD	

1.3.- Esquemas de sistemas de VoIP de alta disponibilidad y alto rendimiento.

El objetivo del artículo es mostrar la posibilidad de dotar a los servidores de telefonía IP de alta disponibilidad y alto rendimiento. Para ello se va a llevar a cabo el diseño de varios escenarios donde se analizarán y estudiarán las ventajas e inconvenientes de cada uno de ellos desde el punto de vista de la disponibilidad del servicio, del rendimiento, así como del coste de implantación que dicha configuración supondría a una empresa.

1.3.1.- Esquema básico.

Supongamos como escenario básico una Universidad. Para simplificar, supongamos que la red de cada uno de los edificios de la Universidad se conecta a un *switch* central, el cual conecta directamente con la *troncal* que ofrece acceso a Internet a toda la Universidad. En la figura siguiente se muestra esta arquitectura.



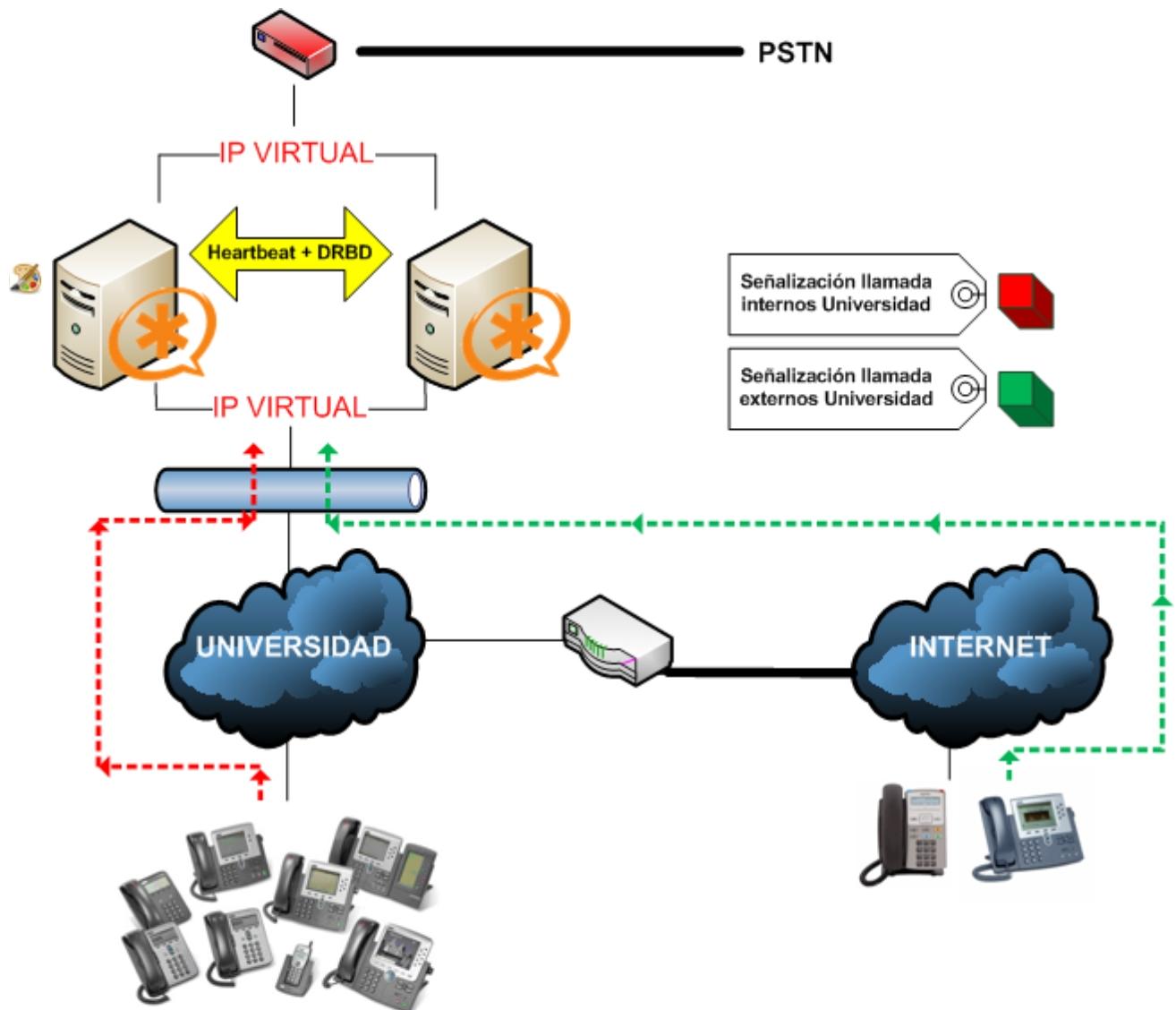
El sistema se compone de una centralita *Asterisk* sobre la cual se registran todos los teléfonos de la Universidad. *Asterisk* puede tener varios tipos de proveedores, entre ellos proveedores de VoIP y proveedores de la red telefónica tradicional.

Además el sistema permite que usuarios externos se puedan conectar a través de Internet para realizar llamadas desde cualquier ubicación.

El sistema cuenta con dos “puntos únicos de fallo”. El punto de fallo más importante es el servicio *Asterisk*, ya que en caso de fallo, toda la Universidad se queda sin poder realizar o recibir llamadas. El otro punto de fallo, y ciertamente menos crítico, es la troncal de Internet de la propia Universidad. Si ésta no se encuentra disponible durante un tiempo, los usuarios externos no podrán hacer uso del sistema telefónico, por encontrarse inaccesible en el interior de la Universidad. Del mismo modo y en sentido inverso, no podrán realizar llamadas a través de un proveedor de VoIP por lo que se tendrán que realizar las llamadas a través de la red tradicional.

Planteado el escenario inicial, se presentan a continuación diversos esquemas en los cuales se irán eliminando los “puntos únicos de fallo”, analizándose, además, para cada uno de ellos las ventajas e inconvenientes que presentan.

El primer esquema consiste en una arquitectura VoIP de alta disponibilidad. El servicio de telefonía es ofrecido por Asterisk y la alta disponibilidad se consigue mediante *Heartbeat*. La arquitectura por tanto se compone de un cluster de alta disponibilidad formado por dos nodos Asterisk + Heartbeat. En la figura 3 se muestra la arquitectura.



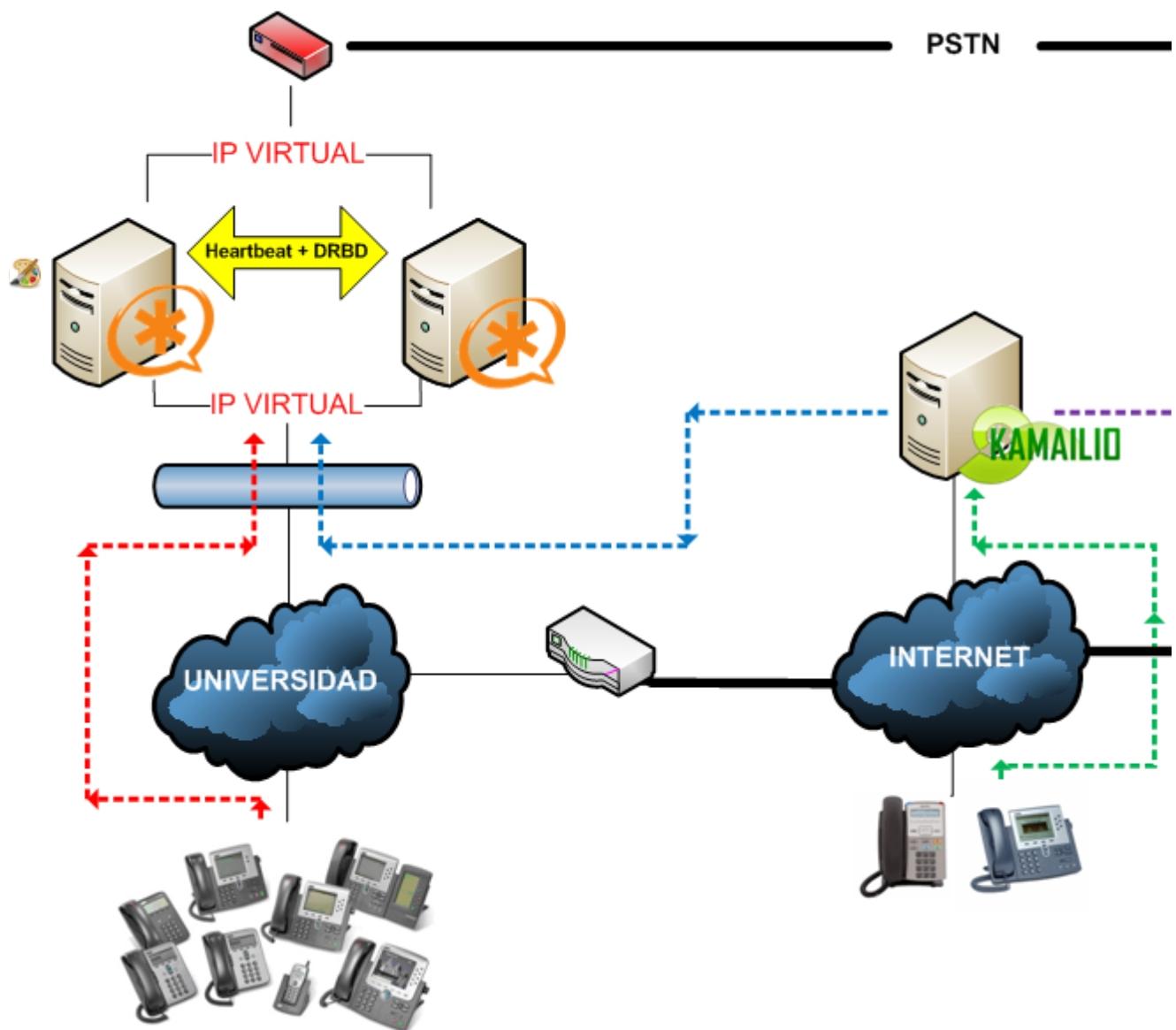
En la configuración de Asterisk se definen cuáles son los posibles destinos de cada terminal, los parámetros de autenticación, buzones de voz, etc. Esta configuración ha de mantenerse igual en todo momento en los dos nodos Asterisk. Por tanto, para hacer que el sistema sea más fácil de mantener es necesario replicar los datos utilizando DRDB o NAS.

En este primer esquema, eliminamos el punto de fallo de disponer de una única máquina Asterisk ofreciendo el servicio. Además cada nodo Asterisk se puede localizar en edificios distintos. De esta manera además de solucionar el problema de fallo hardware/software de uno de los dos nodos también es posible evitar la caída del servicio ante un fallo eléctrico en el edificio que alberga los sistemas. Los teléfonos se registrarán y autenticarán en una IP Virtual asociada al servidor Asterisk que se encuentra activo.

Además, para que cualquier servidor Asterisk pueda hacer uso de la red tradicional de telefonía, el sistema se puede complementar con un balanceador de primarios (*Redfone Communications* - www.red-fone.com). Este se integra con HeartBeat de tal manera que las llamadas recibidas a través de la red tradicional son enviadas al Asterisk activo en cada momento.

1.3.2.- Arquitectura de VoIP con alta disponibilidad y balanceo.

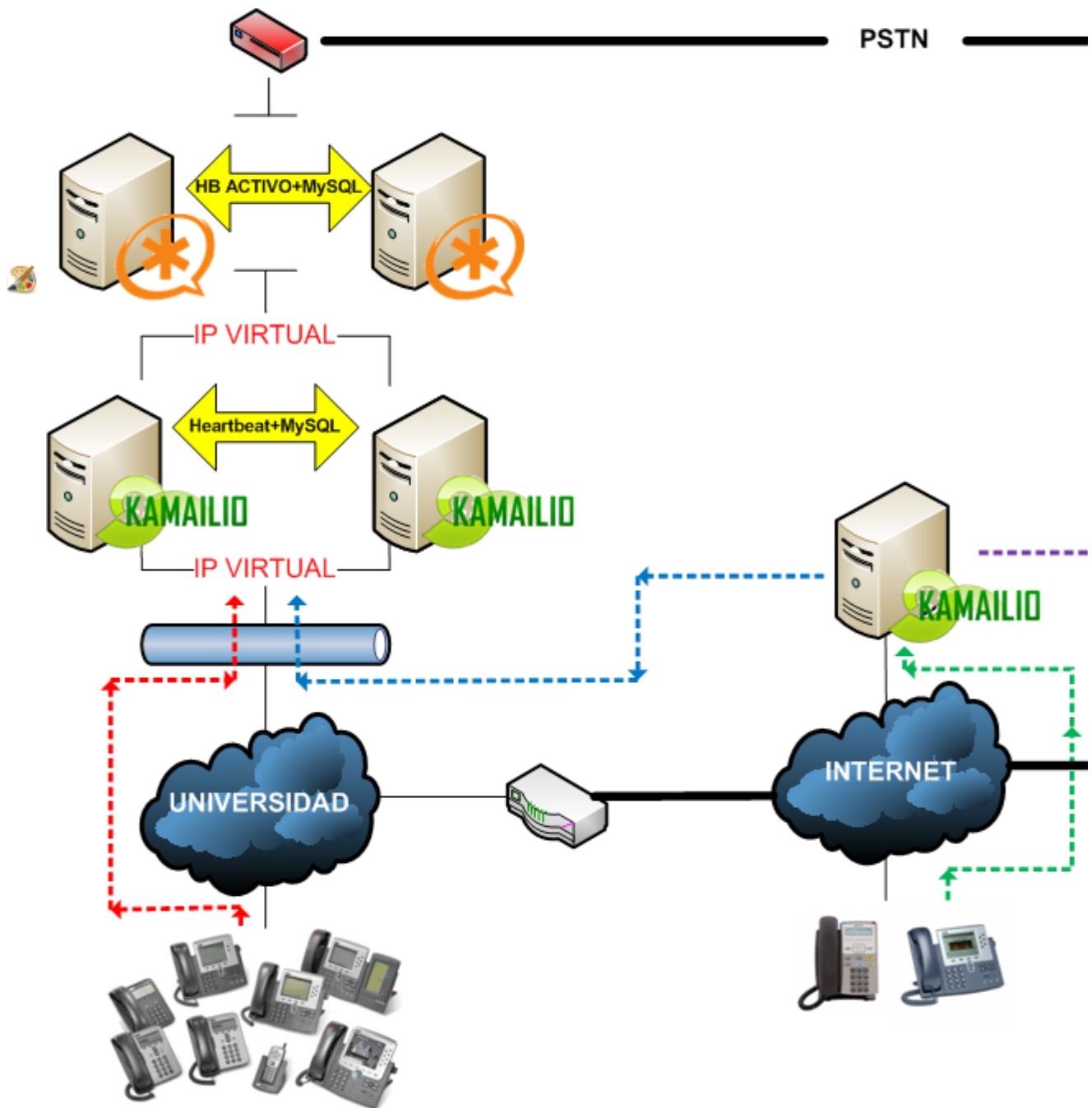
El segundo esquema consiste en una arquitectura VoIP de alta disponibilidad con balanceo. Se compone de un clúster de alta disponibilidad formado por dos nodos Asterisk con Heartbeat en la Universidad, un clúster de alta disponibilidad de dos nodos Asterisk con Heartbeat en otra ubicación remota y un servidor KAMAILIO situado en un lugar de muy alta disponibilidad como lo mantienen por *RedIris*. En la siguiente figura se muestra la arquitectura.



En este esquema también es eliminado el punto de fallo hardware/software de disponer de una única máquina Asterisk ofreciendo el servicio, por lo que para los funcionarios de la propia Universidad es muy poco probable que no se puedan comunicar entre sí.

1.3.3.- Esquema de alta disponibilidad y alto rendimiento.

El tercer esquema consiste en una arquitectura VoIP de alta disponibilidad y alto rendimiento. Esta nueva arquitectura es necesaria cuando el volumen de llamadas es muy alto como para ser soportado por un único y “saturado” nodo Asterisk. La arquitectura se compone de un clúster de alta disponibilidad y alto rendimiento formado por dos nodos Asterisk + Heartbeat y dos nodos KAMAILIO + Heartbeat en la Universidad, un clúster de alta disponibilidad de dos nodos Asterisk + Heartbeat en la ubicación remota y un servidor KAMAILIO situado en un lugar de muy alta disponibilidad como los mantenidos por la empresa RedIris. En la siguiente figura se muestra la arquitectura.



Este tercer esquema añade al segundo esquema alto rendimiento en el interior de la propia Universidad. Es necesario por tanto modificar la configuración de Heartbeat utilizada hasta el momento. Anteriormente la configuración del cluster de alta disponibilidad era un configuración Activo/Pasivo, es decir, en cada instante sólo hay un nodo activo y funcionando, mientras que el otro nodo queda configurado en modo pasivo, es decir, está sin funcionar a la espera de que el nodo activo falle. Ahora, con la nueva configuración de Heartbeat, se va a establecer un cluster de alta disponibilidad Activo/Activo, es decir, ambos nodos funcionan simultáneamente, lo que permite tratar con el doble de llamadas que con los esquemas anteriores.

En esta nueva arquitectura, los usuarios internos se registran en el servidor KAMAILIO interno de la Universidad, mediante una IP Virtual asociada al servidor KAMAILIO activo. De esta manera el KAMAILIO interno balanceará la carga entre los dos Asterisk disponibles haciendo uso, por ejemplo, del algoritmo Round Robin.

No se ha considerado añadir también alto rendimiento en el clúster de alta disponibilidad remoto ya que el volumen de llamadas que puede ser generado por los usuarios externos es soportado por una única máquina Asterisk en funcionamiento.

1.4.- Introducción a la detección de vulnerabilidades en servidores Web.



Reflexiona

Está muy bien tener varios equipos redundantes que nos ofrezcan más rendimiento y disponibilidad, pero... ¿de qué valdría todo esto si los sistemas tuvieran fallos de seguridad en el sistema operativo o en los programas instalados encima que permitan a cualquier atacante realizar un DoS o un DDoS?

Ver la definición de [Denegación de Servicio](#) en Wikipedia para más información y detalles técnicos, aunque como tantas veces, la entrada en [inglés](#) está mucho más completa.

En esta sección haremos una breve introducción a algunas herramientas típicas que pueden poner en compromiso a muchos de los servidores Web actualmente en producción.

Para saber cómo defenderse hay que conocer cómo se ataca, por eso vamos a ver las herramientas de ataque a servidores web.

Antes de nada, hay que advertir que jugar con este tipo de herramientas en un ambiente local seguro puede estar bien, pero si se sale a Internet y se provoca un DoS puede ser considerado un delito en muchos países. Los proveedores de servicio de Internet (ISP's) o los servidores web de destino pueden detectar nuestras pruebas (mediante IDS's) como intentos reales de ataques y tomar medidas contra nuestra conexión o incluso contra el propietario de la conexión.

Hay formas de camuflar el origen de nuestros paquetes de red en Internet como la red [TOR](#). Pero de nuevo, se debe usar con precaución.

Otra advertencia es que siempre que se juegue con herramientas de cracking en general es conveniente usar máquinas virtuales o en las que no se trabaje personalmente para no poner nuestros datos e identidad en riesgo. Cualquier programa o script que se use proveniente desde el mundo del cracking puede contener instrucciones perniciosas para nuestro sistema.

hping3

hping es una herramienta en linea de comandos que nos permite crear y analizar paquetes TCP/IF y como tal tiene muchas utilidades: hacer testing de firewalls, escaneo de puertos, redes y como no... también tiene la capacidad de provocar un SYN Flood Attack mediante denegación de servicio (DoS).

En Ubuntu y derivados se instalaría simplemente con el comando:

```
sudo apt-get install hping3
```

y el comando típico para usarlo sería:

```
sudo hping3 <IP-destino> --flood
```

Esto mandaría una inundación de paquetes HTTP en la IP de destino ocupando todo el ancho de banda posible. El comando tiene mas variaciones como por ejemplo para falsear tu IP de origen.

Puedes ver más detalles de cómo usarlo [aquí](#).

Slowloris

Slowloris es un ataque de denegación de servicios lenta contra un servicio particular, en lugar de inundar las redes, es un concepto emergente que podría permitir a una sola máquina hacer caer un servidor web de otra máquina con un mínimo uso de ancho de banda y sin efectos sobre servicios y puertos no relacionados.

La situación ideal para varios ataques de negación de servicio es cuando todos los servicios permanecen intactos pero el servidor web por si mismo es inaccesible completamente. Slowloris nace de este concepto, y es por lo tanto relativamente oculto comparado con la mayoría de las herramientas de inundación.

Más información y formas de protegerse con comandos específicos se pueden encontrar [aquí](#).

Para instalar Slowloris estos son los comandos:

```
wget http://ha.ckers.org/slowloris/slowloris.pl  
chmod 777 slowloris.pl  
perl slowloris.pl
```

Una vez instalado, un ejemplo de aplicación podría ser:

```
sudo perl slowloris.pl -dns example.com -port 80 -timeout 240 -num 500 -tcpto 5
```

Donde 240 es el tiempo que el script espera para mandar el siguiente bloque de peticiones. Mínimo 5 para que no aparezcan errores entre los bloques.

El 500 es el numero de bloques que manda cada vez. Máximo recomendado serían 1000, si supone mas aparecen errores.

2.- Servicio de transferencia de ficheros. FTP.



Caso práctico

María, sabía que, llegado el momento, las empresas a las que darían soporte web necesitarían subir archivos a sus dominios, por lo cual necesitarían una alternativa a la aplicación web destinada para tal fin: un servicio ftp. Así realizó un estudio sobre servidores ftp y se decantó por la versatilidad, funcionalidad y seguridad del servidor ftp **ProFTPD**. En ese estudio quería llegar a saber del servidor ftp lo siguiente:

1. ¿Cómo funciona?
2. Posibilidades de autenticación y control de acceso.
3. Seguridad. ¿Es posible cifrar la transferencia de archivos?
4. ¿Permite cuotas de disco?
5. ¿Permite cuotas de subida y bajada de archivos?
6. ¿Qué clientes ftp soporta?



Pero antes de ponerlo en producción necesitaba probarlo, es por eso que construyó el siguiente escenario de pruebas, similar al escenario de producción para una empresa que ofrezca sus servicios web por medio de la infraestructura proporcionada por BK Programación y totalmente transparente al cliente final:

- ✓ Sistema Operativo: Debian GNU/Linux 6.0
- ✓ Servidor FTP: ProFTPD
- ✓ Configuración de Red:
 - ➔ Servidor FTP: 192.168.200.250
 - ➔ Cliente de pruebas, desde donde se lanza el cliente ftp: 192.168.200.100



Reflexiona

Hoy en día encontramos muchísima información en Internet, es más, en muchas ocasiones cuando buscamos una determinada información es muy probable que tengamos que filtrarla, ya que encontramos demasiada. Pero, una vez encontrada ¿la podemos guardar? ¿y descargar? Y si es así ¿cómo fue subida?

Normalmente para subir archivos en Internet, ya sean de texto, imágenes, vídeo... hubo que emplear algún método de transferencia de archivos para ubicarlos.

Uno de los métodos más empleados como servicio de transferencia de archivos se realiza mediante el servicio ftp. Éste utiliza el protocolo FTP empleando la arquitectura cliente-servidor. Así el servidor ftp esperará peticiones para transferir los archivos y el cliente ftp, ya sea por terminal o de modo gráfico, realizará esas peticiones.

Uno de los principales problemas, a pesar de ser uno de los métodos más utilizados del protocolo FTP es la no seguridad de la información, esto es, la transferencia tiene lugar sin cifrar la información transferida. Este no sólo es un problema del protocolo FTP sino de muchos de los protocolos utilizados en Internet, puesto que en el comienzo de Internet no se preveía su expansión actual y no se pensaba en asegurar la información mediante cifrado, sino simplemente asegurar el buen funcionamiento. Hoy en día existen extensiones sobre el protocolo FTP que aseguran el cifrado en la transferencia, como FTPS, empleando el cifrado SSL/TLS.

No confundir FTPS con SFTP, ya que este último es implementado con otro servicio, el servicio SSH, y es utilizado para conexiones remotas seguras a través de un terminal de comandos.



Para saber más

En el siguiente enlace, página web del RFC 959 sobre FTP, puedes encontrar traducido el estándar RFC sobre FTP.

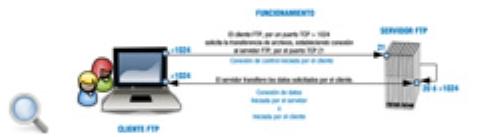
[Página web del RFC 959 sobre FTP.](#)

En el siguiente enlace, página web del RFC 4251 sobre SSH, puedes encontrar el estándar RFC sobre SSH.

[Página web del RFC 4251 sobre SSH.](#)

2.1.- ¿Cómo funciona?

El protocolo FTP emplea una arquitectura cliente/servidor, siendo el cliente ftp quien solicita la transferencia de archivos y el servidor ftp quien ofrece los archivos. Pertenece a la familia de protocolos de red TCP y por lo tanto es un protocolo orientado a conexión, esto es, el cliente ftp necesita establecer una conexión con el servidor para empezar la transferencia de ficheros. Si no se establece la conexión ésta no tiene lugar.



Puesto que FTP es un protocolo que no utiliza una autenticación de usuarios y contraseña cifrada se considera un protocolo inseguro y no se debería utilizar a menos que sea absolutamente necesario. Verás que existen otras alternativas al FTP, como por ejemplo el protocolo FTPS, para mantener comunicaciones cifradas. Aún así, el protocolo FTP está muy extendido en Internet ya que a menudo los usuarios necesitan transferir archivos entre máquinas sin importar la seguridad.

El protocolo FTP requiere de dos puertos TCP en el servidor para su funcionamiento, a diferencia de la mayoría de los protocolos utilizados en Internet que solamente requieren un puerto en el servidor. Un puerto es necesario para establecer el control de la conexión y otro se utiliza para el control de la transmisión, es decir, un puerto se utiliza para establecer la conexión entre el cliente y el servidor y otro para la transferencia de datos.

Los puertos TCP del servidor en cuestión, suelen ser el 21 para el control de la conexión y otro para determinar según el modo de conexión: podría ser el 20 o incluso uno mayor de 1024. Hay que tener en cuenta que estos puertos pueden ser modificados en la configuración del servidor, así que no es obligatorio que los puertos 21 y 20 sean los asignados al servidor FTP, pero sí son los que éste maneja por defecto. El puerto 21 también es conocido como puerto de comandos y el puerto 20 como puerto de datos.

La ventaja que supone utilizar el protocolo FTP se basa en su alto rendimiento y sencillez, que lo hacen una opción conveniente para la transferencia de archivos a través de Internet.

Autoevaluación

A través de un cliente ftp descargas un archivo a tu equipo desde el servidor ftp: ¿cuáles de las siguientes afirmaciones son correctas teniendo en cuenta que el archivo puede descargarse sin problemas?

- El servidor ftp posee dos puertos TCP: uno para el control de la transmisión y otro para la transferencia de datos.
- El servidor ftp posee siempre los puertos TCP 21 y 20: el 21 para el control de la transmisión y el 20 para la transferencia de datos.
- El servidor ftp posee los puertos TCP 21 y 20: el 21 para el control de la transmisión y el 20 para la transferencia de datos.
- El servidor ftp, configurado por defecto, posee el puerto TCP 21 válido para el control de la transmisión y para la transferencia de datos.

[Mostrar retroalimentación](#)

2.2.- Cliente FTP.



Citas para pensar

Lo importante es no dejar de hacerse preguntas.

Albert Einstein

Para poder establecer una conexión con el protocolo FTP son necesarias dos partes: un servidor y un cliente.

ftp://ftp.

Existen múltiples tipos de clientes ftp, desde clientes en terminal de comandos, como ftp o lftp, clientes gráficos como gftp o FileZilla, hasta un cliente ftp en los navegadores mediante ftp://

¿Cuál elegir? Pues, como todo, depende:

- ✓ ¿Conoces la consola ftp? Si te manejas con soltura en la consola ftp puedes pensar en un cliente ftp de comandos que permita utilizar la tecla "tabulado" después de escribir uno o más caracteres para complementar los nombres de archivos.
- ✓ ¿Cuál es el uso que necesitas? ¿Para qué lo vas a utilizar? A lo mejor solamente quiere visitar un servidor ftp y descargar un archivo sin tener que andar instalando nuevos programas. En este caso puedes utilizar el cliente ftp del navegador, ftp://
- ✓ ¿Quieres reanudar la conexión en caso de corte en la misma? En este caso mejor un cliente tipo gráfico.
- ✓ ¿Deseas facilidad de manejo? Un cliente terminal de comandos suele ser menos interactivo que uno gráfico, debes saber manejarlo con comandos en la consola ftp, mientras que en un cliente gráfico puedes manejarlo a través de clics del ratón. Los clientes gráficos suelen ser más amigables y por lo tanto más utilizados.
- ✓ ¿Qué tipo de conexión quieras establecer? ¿Cifrada? ¿No cifrada? Dependiendo del tipo de conexión debes emplear un cliente u otro, ya que no todos los clientes ftp permiten conexiones cifradas.
- ✓ ¿Deseas recordar conexiones? Pues lo mismo, no todos los clientes ftp lo permiten.

Un cliente ftp muy recomendable es el cliente gráfico ftp FileZilla, ya que posee las siguientes características:

- ✓ Fácil de usar.
- ✓ Soporta FTP, FTP sobre SSL / TLS (FTPS) y SFTP.
- ✓ Compatibilidad con múltiples plataformas: se ejecuta en Windows, Linux, BSD, Mac OS X y más.
- ✓ Soporte IPv6.
- ✓ Disponible en varios idiomas.
- ✓ Soporta y reanuda la transferencia de archivos de gran tamaño (mayores de 4 GB).
- ✓ Interfaz de usuario con pestanas.
- ✓ Potente administrador de sitios y cola de transferencia.
- ✓ Marcadores.
- ✓ Arrastrar y soltar.
- ✓ Permite configurar límites de velocidad de transferencia.
- ✓ Nombre de filtros.
- ✓ Directorio de comparación.
- ✓ Asistente de configuración de la red.
- ✓ Edición de archivos remoto.
- ✓ Automantenimiento de la conexión.
- ✓ HTTP/1.1, SOCKS5 y soporte de FTP-Proxy.
- ✓ Fichero de registro.
- ✓ Sincronización de directorios de navegación.
- ✓ Búsqueda de archivos remoto.



Para saber más

En el siguiente enlace puedes acceder a la página web oficial de FileZilla donde puedes descargarlo y encontrar documentación sobre el mismo.

[Página web oficial de FileZilla.](#)

2.3.- Tipos de usuarios.

¿Qué usuarios se pueden conectar al servidor ftp?
¿cualquiera? ¿sólo los usuarios del sistema?

Bien, típicamente existen dos tipos de usuarios:

- ✓ Usuarios anónimos: usuarios que tienen acceso y permisos limitados por el sistema de archivos. Al conectarse al servidor FTP sólo deben introducir una contraseña simbólica, normalmente cualquier dirección de correo -real o ficticia-, por ejemplo: a@ .
- ✓ Usuarios del sistema: aquellos que disponen de una cuenta en la máquina que ofrece el servicio FTP. Al conectarse al servidor FTP deben introducir su contraseña de sistema.



Pero en ciertos servidores, como el servidor ProFTPD, existe una tercera posibilidad muy interesante: usuarios virtuales. Los usuarios virtuales poseen acceso y permisos al servidor FTP sin necesidad de ser usuarios del sistema, por lo tanto si un usuario virtual quisiera acceder al sistema operativo como si fuese un usuario del sistema, ya sea de forma local o remota no podría, pues su cuenta de usuario no existe en el sistema. Los usuarios virtuales tienen definida una contraseña propia y pueden estar definidos en ficheros de autenticación (de texto) con el mismo formato que los del sistema operativo GNU/Linux /etc/passwd, directorios [LDAP](#), bases de datos [SQL](#) o servidores [RADIUS](#).

Dependiendo del servidor ftp, podrás tener unos métodos de autenticación de usuarios u otros por ejemplo en el servidor ftp ProFTPD se permite los siguientes métodos:

- ✓ Ficheros de autenticación del sistema operativo: /etc/passwd y /etc/shadow: Para ello usa las directivas AuthUserFile y AuthGroupFile.

[Howto AuthFiles.](#)

- ✓ Usuarios virtuales definidos mediante ficheros de autenticación (de texto) propios, distintos de los del sistema operativo: para ello también usa las directivas AuthUserFile y AuthGroupFile.
- ✓ Autenticación [PAM](#): Es necesario establecer la directiva AuthPAMAuthoritative a 'on'.

[Directiva AuthPAMAuthoritative.](#)

- ✓ Bases de datos SQL, tales como MySQL o Postgres. Para ello emplea el módulo mod_sql más información sobre el uso de mod_sql lo puedes encontrar en el HowTo SQL

[Howto SQL.](#)

- ✓ LDAP: Para ello emplea el módulo mod_ldap.
- ✓ RADIUS: Para ello emplea el módulo mod_radius.

Mediante la [directiva UserPassword](#) se puede crear una contraseña para un usuario particular que sobreescribe la contraseña del usuario en /etc/passwd (o /etc/shadow), esta contraseña es solamente efectiva dentro del contexto en el cual la directiva es aplicada, esto es, no se modifica el fichero /etc/passwd (o /etc/shadow) sino que se da la posibilidad de que el usuario emplee otra contraseña distinta de la definida en los ficheros del sistema operativo.



Para saber más

En el siguiente archivo encontrarás más información sobre PAM.

 [Información sobre PAM](#) (0.17 MB)

2.4.- Modos de conexión del cliente.



Reflexiona

Si en una transferencia de archivos mediante el protocolo FTP el cliente posee un cortafuegos configurado para impedir acceso local a puertos TCP menores de 1024, ¿es posible la transferencia?

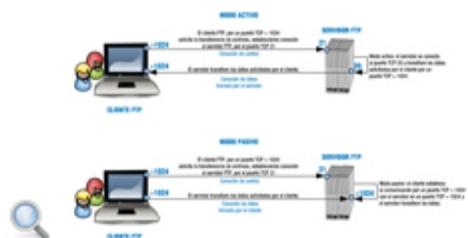
Ya se ha comentado que el servidor FTP a diferencia de otros servidores necesita dos puertos TCP para hacer posible la transferencia de archivos. Ahora bien, ¿son estos puertos siempre los mismos o no? ¿son independientes del tipo de cliente y servidor o no? Pues, básicamente depende de dos factores: del modo de conexión del cliente ftp y de la configuración del servidor ftp.

A priori, si no modificamos la configuración del servidor ftp éste otorgará siempre el puerto TCP 21 para el canal de conexión de control. Es el puerto del canal de transmisión de datos, el que varía ¿cómo?, pues según el modo de conexión del cliente ftp, que puede ser activo o pasivo.

Cuando una aplicación cliente FTP inicia una conexión a un servidor FTP, abre el puerto 21 en el servidor. Se utiliza este puerto para arrojar todos los comandos al servidor. Cualquier petición de datos desde el servidor se devuelve al cliente a través de otro puerto TCP del servidor dependiendo del modo de conexión del cliente. Así:

- ✓ El modo activo es el método original utilizado por el protocolo FTP para la transferencia de datos a la aplicación cliente. Cuando el cliente FTP inicia una transferencia de datos, el servidor abre una conexión desde el puerto 20 en el servidor para la dirección IP y un puerto aleatorio sin privilegios (mayor que 1024) especificado por el cliente. Este arreglo implica que la máquina cliente debe poder aceptar conexiones en cualquier puerto superior al 1024. Con el crecimiento de las redes inseguras, tales como Internet, es muy común el uso de cortafuegos para proteger las máquinas cliente. Debido a que estos cortafuegos en el lado del cliente normalmente rechazan las conexiones entrantes desde servidores FTP en modo activo, se creó el modo pasivo.
- ✓ La aplicación FTP cliente es la que inicia el modo pasivo, de la misma forma que el modo activo. El cliente FTP indica que desea acceder a los datos en modo pasivo y el servidor proporciona la dirección IP y el puerto aleatorio, sin privilegios (mayor que 1024) en el servidor. Luego, el cliente se conecta al puerto en el servidor y descarga la información requerida.

A continuación puedes ver una imagen que muestra el funcionamiento de los dos modos: el activo y el pasivo.



En sistemas GNU/Linux es típico encontrar el archivo **/etc/services** que contiene una lista de puertos TCP/UDP relacionado con los servicios estándar que trabajan en los mismos. Ejecuta el comando `cat /etc/services | grep ftp` y encontrarás todos los puertos y servidores relacionados con la cadena **ftp**.

2.5.- Tipos de transferencia de archivos.



Reflexiona

¿Es lo mismo descargar/subir mediante FTP un archivo de vídeo, que uno de texto o uno ejecutable?

i

Desde el punto de vista de FTP, los archivos se agrupan en dos tipos:

- ✓ Archivos ASCII: son archivos de texto plano (.txt, .ps, .html...)
- ✓ Archivos binarios: todo lo que no son archivos de texto: ejecutables (.exe), imágenes (.jpg, .png ...), archivos de audio (.mp3, .wav ...), vídeo (.avi, .mov ...) , etcétera.



Es muy importante saber con qué tipo de archivos estás trabajando en la transferencia ya que si no utilizas las opciones adecuadas puedes destruir la información del archivo. El servidor ftp permite configurar la transferencia de archivos según el tipo del mismo, es por eso que al ejecutar el cliente FTP, antes de transferir un archivo, debes utilizar uno de los siguientes comandos e poner la correspondiente opción en un programa con interfaz gráfica:

- ✓ **ascii** para tipos de archivos ascii.
- ✓ **binary** para tipos de archivos binarios.

Autoevaluación

Relaciona cada extensión de archivo con el tipo de transferencia ftp correspondiente, escribiendo el número del tipo de transferencia en el cuadro correspondiente:

Ejercicio de relacionar

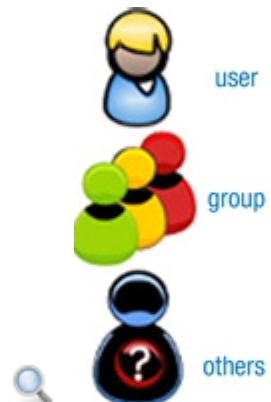
Extensión de archivo	Relación	Tipo de Transferencia
txt (texto).	0	1. ascii.
html (página web).	0	2. binario.
doc (documento).	0	
zip (comprimido).	0	
ps (postscript).	0	
mp3 (audio).	0	
tar (comprimido).	0	
tgz (comprimido).	0	
bz2 (comprimido).	0	

Enviar

2.6.- Establecer permisos en ftp.

El protocolo FTP sigue los permisos establecidos en entornos de tipo UNIX y sus similares GNU/Linux, con lo cual existen tres grupos de permisos en el siguiente orden: propietario, grupo y otros:

- ✓ **Propietario(user=u):** El creador o el que ha subido el archivo al servidor FTP.
- ✓ **Grupo(group=g):** Se refiere a un grupo de usuarios que posee la propiedad del archivo, al que probablemente pertenece el propietario.
- ✓ **Otros(others=o):** Son el resto de usuarios no propietarios o que no pertenecen al grupo indicado. Son el resto del mundo.



Cada grupo a su vez puede tener tres permisos en el siguiente orden: lectura, escritura y ejecución identificados respectivamente por una 'r', una 'w' y una 'x'. La ausencia de permiso es identificada con el carácter '-'. Cada permiso tiene un equivalente numérico, así: r=4, w=2, x=1 y -=0. Por ejemplo: rw- identifica permiso de lectura y escritura o lo que es lo mismo $4+2+0=6$

En un sistema operativo tipo GNU/Linux mediante el comando 'ls -l' puedes ver los permisos asignados a ficheros y directorios, por ejemplo si la salida del anterior comando es:

```
-rw-r--r-- 1 alumno clase 0 jun 20 01:15 prueba1.txt
```

significa que,

- ✓ **prueba1.txt** es un fichero ya que **-rw-r--r--** comienza con '-', si fuese un directorio aparecería un 'd'
- ✓ **rw-r--r--** identifica los permisos del fichero prueba1.txt, que divididos 3 a 3 representan de izquierda a derecha: propietario, grupo, otros.
- ✓ **rw-** identifican los permisos del usuario propietario, en este caso **alumno**. Por lo tanto alumno posee los permisos de **lectura** y **escritura** sobre el fichero prueba1.txt o lo que es lo mismo $4+2+0=6$
- ✓ **r--** identifican los permisos del grupo propietario, en este caso **clase**. Por lo tanto clase posee solamente el permiso de **lectura** o lo que es lo mismo **4+0+0=4**
- ✓ **r--** identifican los permisos de los **otros** (resto del mundo). Por lo tanto todos los usuarios que no son alumno y aquellos que no pertenecen al grupo clase poseen solamente el permiso de **lectura** o lo que es lo mismo **4+0+0=4**

Por lo tanto los permisos **rw-r- -r- -** equivalen a **644**.



Para saber más

Es conveniente que le des un vistazo al manual de chmod y umask: `man chmod` y `man umask`.

Por otro lado en un sistema GNU/Linux, en principio, no todos los usuarios del sistema tienen acceso por ftp, así existe un fichero `/etc/ftpusers` que contiene una lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: root, bin, uucp, news. Ten en cuenta que las líneas en blanco y las líneas que comiencen por el carácter '#' serán ignoradas.

Autoevaluación

Ejecutas en una consola de comandos en la ruta /home/alumno el comando ls -l obteniendo la siguiente salida:

drwxr-x--- 1 alumno clase 0 jun 20 01:16 Documentos

Entonces, con esa información puedes deducir que:

- Documentos es un directorio con permisos 750.
- Documentos es un fichero con permisos 750.
- Documentos pertenece al usuario propietario alumno y al grupo propietario clase. Además el usuario alumno posee permisos de modificación, mientras que el grupo clase y los demás no poseen ese permiso.
- Documentos pertenece al grupo propietario alumno y al usuario propietario clase. Además el usuario alumno posee permisos de modificación, mientras que el grupo clase y los demás no poseen ese permiso.

[Mostrar retroalimentación](#)

2.7.- Servicio de transferencia de archivos en modo texto.

Como se comentó anteriormente, existen varios tipos de clientes ftp, entre los cuales los cliente en modo texto desde siempre estuvieron incorporados en las distribuciones GNU/Linux.

De entre los clientes tipo texto cabe destacar dos: el cliente en modo texto **ftp** y el cliente en modo texto **Iftp**. En GNU/Linux Debian 6 se dispone del cliente modo ftp en una instalación básica. Para poder utilizarlo en el sistema simplemente hay que ejecutarlo como comando: el comando **ftp**.

Vamos a ver, a continuación, el comportamiento del cliente en modo texto ftp en la conexión a servidor ftp ftp.rediris.es:

1. Básicamente la sintaxis es la siguiente:

ftp [-pinegvd] [host [port]]

donde

- ✓ host identifica el servidor ftp
- ✓ port identifica el puerto, por defecto 21, por lo cual si conectas a un servidor ftp configurado en ese puerto no es necesario escribirlo, ya se considera.

Puedes ver la ayuda del comando ftp mediante: **man ftp** ó **info ftp** .



Para saber más

En el siguiente archivo puedes ver el mismo ejemplo de conexión al servidor ftp: [ftp.rediris.es](ftp://ftp.rediris.es) utilizando el cliente en modo texto lftp:



2.7.1.- Comandos ftp.



En la consola ftp pueden estar disponibles múltiples comandos, algunos de los más empleado son los recogidos en la siguiente tabla:

Comandos FTP

ABRIR/CERRAR CONEXIÓN	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
open servidor	Inicia conexión remota con un servidor ftp.
close / disconnect	Finalizan la sesión ftp sin cerrar la consola ftp.
bye / quit / exit	Terminan la sesión ftp y salen de la consola ftp.
!	Sale a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver, teclea exit en la línea de comandos.
AYUDA	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
? / help	Muestra una lista de los comandos disponibles.
? comando / help comando	Muestra la información relativa al comando.
TRABAJAR CON DIRECTORIOS	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
cd directorio	Cambia de directorio en el servidor remoto.
lcd directorio	Cambiarse de directorio en el equipo local (cliente ftp).

dir directorio / ls directorio	Listan el contenido del directorio remoto actual.
pwd	Muestra el directorio activo en el servidor.
lpwd	Muestra el directorio activo en el equipo local (cliente ftp).
rmdir directorio	Elimina un directorio vacío en el servidor.
mkdir directorio	Crea un directorio en el servidor.Crea un directorio en el servidor.

TRABAJAR CON FICHEROS

COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
delete archivo	Borrar un archivo en el servidor remoto.
mdelete patrón	Borrar varios archivos según un patrón.
get archivo	Obtiene archivo en el equipo cliente desde el servidor remoto.
mget archivos	Obtiene varios archivos desde el servidor remoto.
put archivo	Envía un archivo al servidor remoto.
mput archivos	Envía varios archivos al servidor remoto.
rename archivo	Cambia el nombre a un archivo en el servidor.
ascii	Para configurar y transferir archivos tipo ascii.
binary	Para configurar y transferir archivos tipo binario.
less archivo	Leer contenido de archivo mediante el comando less.

TRABAJAR CON PERMISOS

COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
chmod	Cambio de permisos en el servidor remoto.
umask	Configura el sistema de permisos en el lado remoto.



Para saber más

Puedes descargar en el siguiente documento una tabla con más comandos ftp.

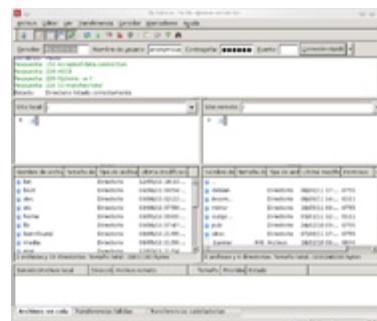
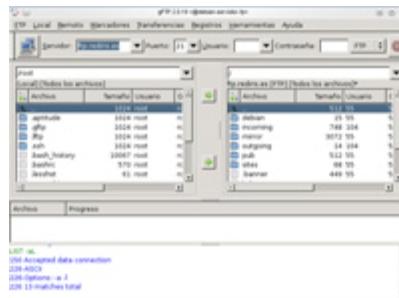
 [Tabla con más comandos ftp.](#) (0.05 MB)

2.8.- Servicio de transferencia de archivos en modo gráfico.

El servicio de transferencia de ficheros está bien, pero obliga a entender el funcionamiento de un servidor ftp mediante el uso de sus comandos. La verdad, es que no es muy interactivo. ¿Entonces..., no existe la posibilidad de trabajar de otro modo más interactivo? Pues si, mediante clientes ftp de modo gráfico o mediante el navegador, ya que éste incorpora su propio cliente ftp.

Típicamente los clientes gráficos se comportan todos igual, esto es, tienen una interfaz parecida básicamente presentan una ventana partida en dos secciones: la de la izquierda suele representar el equipo cliente ftp -desde donde se intenta establecer la conexión- y la de la derecha suele representar el equipo servidor ftp -quién recibe la conexión-. Luego suelen existir, en alguna zona determinada de la ventana: en el centro entre las dos secciones, arriba de las dos secciones, etc., una serie de botones, usualmente representados como flechas que indican la posibilidad de subir o descargar archivos. Incluso dependiendo del cliente en modo gráfico es posible guardar los datos de las conexiones como plantillas, de tal forma que la próxima vez que intentes establecer la conexión con un mismo servidor ftp en vez de tener que rellenar los campos referentes a la conexión puedes hacerlo a través de la plantilla que ya posee el valor de esos campos.

Dentro de los clientes ftp en modo gráfico cabe destacar dos: **gftp** y **filezilla**. A continuación puedes ver un ejemplo de como utilizarlos para establecer una conexión con un servidor ftp, en el servidor ftp.rediris.es :





Para saber más

En el siguiente archivo puedes ver información de la instalación del cliente en modo gráfico gftp.

[Instalación del cliente en modo gráfico gftp.](#) (0.01 MB)

Te proponemos el siguiente enlace de un vídeo práctico sobre la instalación y uso de FileZilla en una distribución GNU/Linux basada en Debian.

[Resumen textual alternativo](#)

2.9.- Servicio de transferencia de archivos desde el navegador.

El navegador web también puede ejercer de cliente ftp y, puesto que la mayoría de los sistemas operativos cuentan con un navegador en su instalación, es una de las herramientas más usadas para transferencia de archivos.

Para poder usar el navegador como cliente ftp solamente debes escribir en la barra de dirección una  dirección URL tipo, como la siguiente:

ftp://nombre_servidor_ftp:puerto

donde,

- ✓ **ftp://** indica que el protocolo que deseas que interprete el navegador sea el ftp.
- ✓ **nombre_servidor_ftp** representa el nombre o la IP del servidor ftp.
- ✓ **puerto** indica el puerto TCP, por defecto 21. Puedes omitirlo siempre y cuando sea el 21.

Si el servidor ftp permite la conexión a un usuario anónimo, al ejecutar **ftp://nombre_servidor_ftp:puerto** entrarás directamente al servidor ftp, esto es, el navegador no preguntará qué usuario y contraseña necesitas para establecer la conexión.

En la siguiente imagen puedes ver como puedes acceder al servidor ftp de rediris utilizando el navegador:



Así, lo único que tienes que hacer es escribir en la dirección URL: **ftp://ftp.rediris.es** y pulsa **Enter**, con lo cual, automáticamente, conectas con el servidor ftp, pudiendo visitar las carpetas y ver los ficheros como si de un explorador de archivos se tratara.

Para descargar las carpetas o archivos simplemente debes pulsar con el botón derecho del ratón sobre ellos y elegir la opción **Guardar enlace como...** -que aparece en Firefox y es similar en otros navegadores-.

Pero no todo van a ser ventajas al utilizar el navegador como cliente ftp, puesto que otros clientes tienen la posibilidad de continuar las descargas cuando estás sufrieron algún tipo de interrupción cosa que no pasa con el cliente ftp del navegador, como por ejemplo el cliente gráfico FileZilla que soporta y reanuda la transferencia de archivos de gran tamaño(> 4 GB).

2.10.- Asegurando el servicio de transferencia de archivos.

Bien, pero ¿qué pasa con los datos en la transferencia? ¿viajan cifrados? ¿no? Pues empleando el protocolo ftp cualquiera que tenga acceso al canal de transmisión podrá ver en texto claro todo lo que se transmite, esto es, los datos no se cifran. Esto puede carecer de importancia, o no, segú el contexto de la transmisión. Así, puede que a un organismo público no le importe compartir información a través de ftp y que los datos en la transferencia viajen sin cifrar y, sin embargo, a una empresa si le interese que los datos viajen cifrados.

Entonces, cuando interese asegurar el servicio de transferencia de archivos debes descartar el protocolo ftp y empezar a pensar en otras alternativas, como: **ftps** o **sftp**.

FTPS es una extensión del protocolo FTP que asegura el cifrado en la transferencia mediante los protocolos SSL/TLS. Permite tres tipos de funcionamiento:

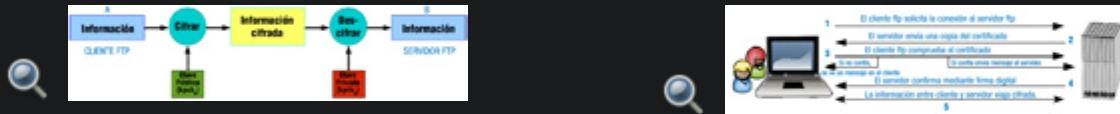
- ✓ SSL Implícito:
 - ↳ Como conexiones  [HTTPS](#).
 - ↳ Usa los puertos 990 y 989.
- ✓ SSL Explícito
 - ↳ El cliente usa los mismos puertos estándar FTP: 20 y 21 pero se efectúa el cifrado en ellos.
 - ↳ Usa [AUTH SSL](#).
- ✓ TLS Explícito:
 - ↳ Similar a SSL Explícito pero usa [AUTH TLS](#).

El cifrado al que nos referimos es el [cifrado de clave pública o asimétrico](#): **clave pública(kpub)** y **clave privada(kpriv)**. La **kpub** interesa publicarla para que llegue a ser conocida por cualquiera la **kpriv** no interesa que nadie la posea, solo el propietario de la misma. Ambas son necesaria para que la comunicación sea posible, una sin la otra no tienen sentido, así una información cifrada mediante la **kpub** solamente puede ser descifrada mediante la **kpriv** y una información cifrada mediante la **kpriv** sólo puede ser descifrada mediante la **kpub**.

En el cifrado asimétrico podemos estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el **cliente ftp(A)** y el **servidor ftp(B)**. Ver la siguiente tabla como ejemplo de funcionamiento del cifrado asimétrico:

Funcionamiento del cifrado asimétrico.

Funcionamiento del cifrado asimétrico



$A(\text{inf}) \rightarrow \text{inf cifrada} \rightarrow B \text{ [descifrar inf]} \rightarrow B(\text{inf}) = A(\text{inf})$
 $A(\text{inf}) \rightarrow \text{inf cifrada} = [(\text{inf})]k\text{pubB} \rightarrow B \text{ [\text{inf. cifrada}]k\text{privB}} \rightarrow B(\text{inf}) = A(\text{inf})$



Identificación

A	Cliente ftp.
inf cifrada = [(inf)]kpubB	Información cifrada mediante la clave pública de B obtenida a través de un certificado digital.
[inf. cifrada]kprivB	Información descifrada mediante la clave privada de B.
B	Servidor ftp.



Para saber más

En el siguiente enlace encontrarás más información sobre asegurar FTP con TLS.

[Asegurar FTP con TLS.](#)

En este otro enlace encontrarás más información sobre el protocolo TLS.

[Protocolo TLS.](#)

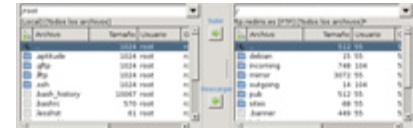
2.11.- El servicio de transferencia de archivos en el proceso de despliegue de una aplicación Web.



Reflexiona

¿Cómo actualizas una página web de forma remota? ¿Qué servicios suelen ofrecer las empresas de alojamiento web para que puedas subir archivos a tus aplicaciones? ¿Cuánto tiempo se mantiene la conexión subiendo un archivo?

Suele ser típico que cualquier aplicación web en Internet disponga de la posibilidad de subir archivos mediante una configuración del código fuente de la misma, una aplicación propia o una aplicación de terceros, como los paneles de administración web.



Si empleas una aplicación web para subir archivos debes tener en cuenta cuánto tiempo puede mantener la conexión abierta con el servicio web y cuál es el tamaño máximo de subida de un archivo. Estas cuestiones suelen ser típicas de la configuración del servidor web. Por la contra, si empleas un servidor ftp dependerá de éste las cuestiones anteriores.

Se suele configurar el servidor web con unos parámetros: tiempo de conexión y tamaño máximo de subidas de archivos diferentes del servidor ftp, de tal forma que para archivos de tamaño no muy grandes se puedan emplear aplicaciones web y no se sufra un corte en la subida de archivo, y, para archivos grandes, se emplee el servidor ftp.

Normalmente las empresas de alojamiento web (hosting) permiten la subida de archivos mediante un servidor ftp y poseen documentos sobre cómo operar con éste, esto es, documentación que explica cómo conectarse a sus servidores ftp a través de algún cliente ftp, como por ejemplo FileZilla, Cute FTP, Fetch o Transmit. También suelen permitir usar SCP o SFTP para transferir ficheros de forma segura mediante un canal cifrado. Por ejemplo en Filezilla se puede establecer la conexión de forma cifrada directamente, sólo con indicar como puerto TCP el número del servicio SSH, por defecto, 22.

También debes saber que muchos editores web permiten subir tu aplicación web al servidor con el protocolo FTP, esto te puede resultar más sencillo que el uso de una aplicación de FTP independiente.

Eso sí, sea cual sea el método ftp que utilices para subir archivos y actualizar tu web, si desaconseja el uso de aplicaciones no actualizadas que podrían comprometer la seguridad de tu web.

A continuación puedes ver errores típicos, junto con sus soluciones, que puedes encontrar al subir tu aplicación mediante un servidor FTP:

- ✓ Tu cliente FTP muestra el error **access denied**, o similar, cuando subes o borras ficheros o carpetas: Comprueba que tu usuario FTP tenga permisos suficientes sobre la carpeta o fichero en la que deseas subir o que deseas borrar.
- ✓ Tus páginas no son reconocidas de forma automática al acceder a tu dominio: Los servidores GNU/Linux son sensibles a mayúsculas y minúsculas por lo que verifica el nombre de tus archivos.
- ✓ El cliente de FTP te muestra el mensaje **too many connections from your IP address**: Esto quiere decir que existen más conexiones abiertas con el servidor FTP desde la misma dirección IP de las permitidas. En ese caso, asegúrate que no exista ninguna aplicación como un cortafuegos, que pueda estar bloqueando las conexiones abiertas, y provocando de esta forma, que se establezcan más intentos de conexión de los necesarios.

3.- Instalación del servidor proftpd.



Caso práctico

Bien —pensó María— ya es la hora, una vez configurado el servidor web, alojada la página y comprobada su visibilidad a través de Internet, toca permitir la subida de archivos de gran tamaño a la carpeta upload preparada para tal fin. Así que, déjame pensar, María, ¿qué debo hacer?:



1. Esta empresa maneja archivos de gran tamaño, por lo que habrá que configurar un servidor ftp para que en la subida de archivos la conexión no se corte, que es lo más probable que ocurra a través de la subida de archivos mediante la propia aplicación web.
2. Mirar qué servicio han contratado, puesto que el nivel de cuota en disco puede variar según el servicio.
3. Crear un usuario virtual para que pueda subir archivos, —¿cómo lo haré? ¿mediante base de datos SQL? —Uhm..., creo que lo mejor será crear un usuario virtual y, como solamente se trata de un usuario, pues lo crearé mediante un archivo de autenticación.
4. Si se necesita crear otro usuario, pues otro en el archivo de autenticación y listo. ¿Y si necesito crear grupos? Pues lo mismo, en un archivo de autenticación de grupos.
5. Se necesita que la comunicación sea cifrada, con lo cual se debe emplear el protocolo SSL. —¿Uhm...? Mejor el protocolo TLS, que es el sucesor de SSL. Pero, claro, si empleo cifrado voy a tener que utilizar otros puertos TCP distintos del servidor ftp que maneja por defecto: el 21 para la conexión de control y el 20 para la conexión de datos.
6. Ya está, mejor empleo TLS Explícito, de tal forma que puedo seguir utilizando los mismos puertos 20 y 21 para el cifrado.
7. Y todo esto no debe modificar la configuración ya realizada para otras empresas.

Pues clarísimo, lo que tengo que hacer es utilizar el servidor **ProFTPD**.

Entonces:

- ✓ Primero, comprobar si en este servidor dedicado está instalado y, si no lo está, instalarlo.
- ✓ Segundo, configurar el servidor proftpd de la siguiente forma:
 - ⇒ Configuración independiente para esta empresa.
 - ⇒ Usuario virtual en un archivo de autenticación.
 - ⇒ Cifrado TLS Explícito para asegurar el cifrado.
 - ⇒ Cuota de disco.
 - ⇒ Permisos de subida en la carpeta upload correspondiente.

Pues, manos a la obra María, que se va haciendo tarde.

¿Por qué ProFTPd? Pues porque es un servidor FTP bajo licencia GPL altamente configurable, así como:

- ✓ Usuarios virtuales con:
 - ◆ LDAP
 - ◆ BBDD: MySQL, PostgreSQL...
 - ◆ Ficheros de autenticación (ficheros de texto).
- ✓ Personalizar opciones según usuario/grupo.
- ✓ Seguridad mediante cifrado SSL/TLS.
- ✓ Configuraciones independientes mediante virtualhosts.

Para instalar el servidor proftpd en un sistema Operativo Debian 6.0 (squeeze) ejecutar el comando apt-get install proftpd. En la instalación deberás elegir si ProFTPD va a ejecutarse como un servicio desde **inetd** o como un **servidor independiente**. Ambas opciones tienen sus ventajas. Si sólo recibes unas pocas conexiones FTP diarias, probablemente sea mejor ejecutar ProFTPD desde inetd para ahorrar recursos. Por otro lado, con más tráfico, ProFTPD debería ejecutarse como un servidor independiente para evitar crear un proceso nuevo por cada conexión entrante.

En la instalación se crearán los usuarios **proftpd** y **ftp** con grupo **nogroup** y sin posibilidad de acceso a una consola del sistema. Se puede comprobar en el fichero **/etc/passwd** donde encontrarás nuevas líneas similares a las siguientes:

```
proftpd:x:106:65534::/var/run/proftpd:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
```



Para saber más

Te proponemos el siguiente enlace a un vídeo práctico sobre la instalación y uso de proftpd en una distribución GNU/Linux basada en Debian.

[Resumen textual alternativo](#)

3.1.- Configuración de proftpd.

Su configuración es similar a la configuración del Servidor Apache, con lo cual si posees conocimientos sobre Apache tendrás mucho ganado, así tiene:

- ✓ Un fichero de configuración principal [/etc/proftpd/proftpd.conf](#)
- ✓ La posibilidad de configurar  [hosts virtuales](#) o  [virtualhosts](#), de tal forma que un mismo servidor ftp puede alojar múltiples dominios con sus configuraciones correspondientes, y todo lo que no esté incluido en la definición de cada virtualhost se heredará de la configuración principal.
- ✓ Configuración a través de directivas.
- ✓ Contextos de configuración: global, directorio, virtualhost, anonymous.
- ✓ Modularización. Al igual que Apache se pueden activar/desactivar funcionalidades a través de módulos.



Debes conocer

En el siguiente enlace encontrarás la página web oficial de ProFTPD.

[Página web oficial de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre las directivas de ProFTPD.

[Directivas de ProFTPD.](#)

Una vez instalado ProFTPD existirán dos ficheros de especial interés:

- ✓ El fichero [/etc/ftpusers](#) (0.01 MB) , ya comentado, que contiene una lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: root, bin, uucp, news. Ten en cuenta que las líneas en blanco y las líneas que comienzan por el carácter '#' serán ignoradas.
- ✓ El fichero de configuración principal ([/etc/proftpd/proftpd.conf](#)) similar al del siguiente enlace [proftpd.conf](#) (0.01 MB)

En el fichero protfd.conf:

- ✓ Las líneas en blanco y las líneas que comienzan por el carácter '#' serán ignoradas.
- ✓ Las líneas que comienzan por **Include** recogerán la configuración de los ficheros que lo acompañan.
- ✓ User proftpd y **Group nogroup** identifican al usuario y grupo con el que se ejecuta proftpd.
- ✓ El soporte LDAP, SQL, TLS, virtualhosts y cuotas están desactivados, ver líneas: `##include /etc/proftpd/ldap.conf, #Include /etc/proftpd/sql.conf, #Include /etc/proftpd/tls.conf` `#Include /etc/proftpd/virtuals.conf` y QuotaEngine off.
- ✓ El mensaje de bienvenida se encuentra en el fichero `welcome.msg`
- ✓ Está configurado por defecto el modo de conexión ftp activo en el puerto TCP 21.
- ✓ Los usuarios que puedan conectarse por ftp:
 - ➡ Necesitan una consola de comandos activa, esto es, debe poseer una consola presente dentro del fichero **/etc/shells**
 - ➡ Pueden moverse por todo el sistema de ficheros, esto es, no están encerrados (jaula chroot) en sus directorios / home, puesto que la directiva DefaultRoot ~ está comentada. Por seguridad sería conveniente descomentar la línea y recargar la configuración del servidor.
- ✓ Para evitar ataques de denegación de servicio solamente se permiten 30 conexiones simultáneas: MaxInstances 30
- ✓ Los permisos para los ficheros y directorios creados en la conexión ftp son: 644 y 755 respectivamente, ya que, **umask**, donde el primer grupo de tres números identifican los permisos de los ficheros y el segundo grupo identifica los permisos de los directorios.
- ✓ Encontrarás al final del mismo un ejemplo de configuración para usuarios anónimos.

Una vez retocada la configuración del servidor proftpd sólo reconocerá estos cambios cuando recargas su configuración, con lo cual debes ejecutar el comando:

```
/etc/init.d/proftpd restart
```

Si la configuración es correcta, y no quieras reiniciar proftpd, puedes recargar la configuración mediante el comando:

```
/etc/init.d/proftpd reload
```

A continuación veremos distintas configuraciones del servidor proftpd.

3.2.- Configurar el servidor como ftp privado.

Una vez instalado el servidor proftpd, en Debian 6 (squeeze), disponemos de un archivo de configuración [/etc/proftpd/proftpd.conf](#) (0.01 MB)

Como has podido comprobar en la sección anterior, posee una configuración tipo por defecto. Ésta ya permite la conexión a tu servidor. ¿Con qué usuarios? Con cualquier usuario del sistema que posea una consola de comandos activa definida en /etc/shells.



¿Cómo? Pues simplemente ejecutando cualquier cliente ftp que establezca una conexión con el puerto TCP 21 a tu servidor ftp. Por ejemplo utilizando el cliente de comandos ftp sería:

```
ftp usuario_del_sistema@servidor_ftp
```

donde,

servidor_ftp: puede ser el nombre de tu servidor ftp en /etc/hosts o resuelto por DNS, o la IP de tu servidor ftp

Si no eres capaz de conectar revisa la configuración de tu cortafuegos y la sección 1.6 donde se exponen soluciones a problemas típicos en conexiones ftp.



Para saber más

En el siguiente enlace encontrarás ejemplos de configuraciones del servidor ProFTPD.

[Ejemplos de configuraciones del servidor ProFTPD](#)

3.3.- Configurar el servidor como ftp privado y anónimo.

Igual te interesa contar en la configuración de tu servidor ftp con un usuario anónimo, el cual establecerá la conexión con cualquier contraseña y tendrá permisos diferentes a los usuarios del sistema (privados). Normalmente los permisos del usuario anónimo en un servidor ftp se establecerán para que solamente pueda moverse por los directorios y descargar archivos, nunca subirlos, esto es, normalmente el usuario anónimo no podrá crear ni eliminar ficheros y directorios.



Para hacer que proftpd permita conexiones mediante usuarios del sistema y usuarios anónimos debes modificar el fichero [/etc/proftpd/proftpd.conf](#) (0.01 MB)

La modificación consiste en retocar su configuración activando a mayores la conexión mediante usuarios anónimos, puesto que los usuarios del sistema por defecto ya poseen acceso mediante ftp y se conectan con la misma contraseña del sistema.

Por lo tanto, al final del fichero incorpora las siguientes líneas:

Esta configuración permitirá conectar al servidor ftp mediante el usuario **ftp** o por su alias **anonymous** empleando una contraseña cualquiera. Una vez conectado solamente tendrá acceso al contenido de la carpeta /home/ftp y no podrá subir ni eliminar nada de ella.

3.4.- Configurar el servidor como ftp anónimo.

Puedes configurar proftpd para que permita conexiones mediante usuarios anónimos obligando a conectar sin poseer una contraseña del sistema, esto es, conectando con una contraseña cualquiera. Para ello debes modificar de nuevo el fichero [/etc/proftpd/proftpd.conf](#) (0.01 MB)

Por lo tanto, cambia la configuración del usuario anonymous ftp de tal forma que al final del fichero aparezcan las siguientes líneas:

```
Anonymous-privado
User nobody
Group nobody
#No se necesita tener una shell en la redirección
PermitWriteDir -r
#No se permite escribir en la conexión
AnonymousPassword -r
#No permite DESCARGAR en ninguna dirección al invitado
AllowWriteDir
AllowWriteFile
AllowWritePerm
#No permite
#Anonymous
```

Puedes crear un usuario anónimo con carácter privado, esto es, que requiera contraseña para establecer la conexión. Por ejemplo, en la configuración siguiente se convierte el usuario de sistema 'invitado', para el servidor ftp, en un usuario anónimo que requiere contraseña para establecer la conexión. Además, solamente tendrá permisos de escritura desde cualquier equipo que conecte mediante la dirección de red 192.168.200.

```
Anonymous-privado
User invitado
Group nobody
#No se necesita tener una shell en la redirección
AnonymousPassword -r
#No permite DESCARGAR en ninguna dirección al invitado invitado a no ser que establezca conexión
#de la red 192.168.200
AllowWriteDir
AllowWriteFile
AllowWritePerm
#No permite
#Anonymous
```

Puedes convertir cualquier usuario privado (del sistema) que posea una consola de comandos válida en / etc/shell en un usuario anónimo. Por ejemplo en las configuraciones anteriores sólo tendrías que sustituir el usuario 'ftp' y el usuario 'invitado' por el nombre de un usuario existente en el sistema operativo.



Debes conocer

En el siguiente enlace encontrarás información sobre la directiva Order de ProFTPD.

[Directiva Order de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre la directiva Allow de ProFTPD.

[Directiva Allow de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre la directiva Deny de ProFTPD.

[Directiva Deny de ProFTPD.](#)



Ejercicio resuelto

Según lo visto anteriormente, ¿cómo permitirías al usuario invitado establecer conexión desde las redes: 192.168.200 y 10.0.200?

[Mostrar retroalimentación](#)



Ejercicio resuelto

Y si además, ¿quisieras permitir el acceso desde los dominios **tuhostA.tudominio.edu**, **tuhostB.tudominio.edu** y **tuhostC.tudominio.edu**?

[Mostrar retroalimentación](#)

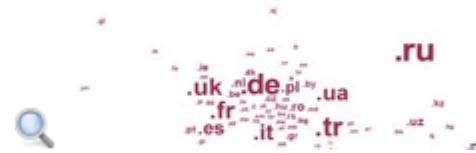
3.5.- Configurar el servidor ftp con múltiples dominios.

Anteriormente hemos visto cómo poder configurar el servidor ftp con múltiples usuarios, pero todos pertenecientes al mismo sitio/dominio, entonces, ¿no se puede configurar usuarios pertenecientes a distintos dominios en el mismo servidor ftp? La respuesta es que sí, sí se puede, ¿cómo?, mediante la configuración de hosts virtuales o virtualhosts. Éstos básicamente lo que hacen es permitir que un mismo servidor ftp pueda alojar múltiples dominios, así configurando hosts virtuales podemos alojar: **empresa1.com**, **empresa2.com**,..., **empresaN.com** en el mismo servidor ftp. Cada empresa tendrá su virtualhost único e independiente de los demás.

Aunque como se ha comentado anteriormente cada virtualhost es único e independiente de los demás, todo aquello que no esté incluido en la definición de cada virtualhost se heredará de la configuración principal: **proftpd.conf** (**/etc/proftpd/proftpd.conf**). Así, si quieres definir una directiva común en todos los virtualhost no debes modificar cada uno de los virtualhosts introduciendo esa directiva sino que debes definir esa directiva en la configuración principal del servidor ftp ProFTPD, de tal forma que todos los virtualhost heredarán esa directiva, por ejemplo en proftpd.conf puedes encontrar la directiva TimeoutIdle 1200, que establece la directiva TimeoutIdle igual a 1200 segundos, esto es, indica el número máximo de segundos que puede estar un usuario sin hacer nada, pasado ese tiempo se cierra la conexión ftp.

En la definición de la directiva VirtualHost podemos poner la IP del servidor FTP ó bien el nombre DNS correspondiente. En nuestro escenario, la IP_Servidor_FTP=192.168.200.250 **ftp.empres1a.com** y **ftp.empres2a.com** identifican a la misma máquina.

Hay que tener en cuenta que si las IP empleadas son **IP privadas**, sin existencia en Internet, siempre que se haga referencia a las mismas a través de nombre de dominios, deberá existir un **servidor DNS** que las resuelva en local o bien, en su defecto, deberán existir las entradas correspondientes en el fichero del sistema local **/etc/hosts**.



Independientemente de si configuras virtualhosts basados en IP o en nombre, puedes utilizar usuarios del sistema, pero también puedes crear los usuarios virtuales que quieras en un fichero similar a **/etc/passwd** y llamarlo en la configuración mediante la directiva AuthUserFile, entonces:

- ✓ Ejecuta el siguiente comando que creará un fichero de autenticación para usuarios virtuales,
`ftpasswd --passwd --name user-empresa1 --file /etc/passwd.usuarios.virtuales --uid 107 --home /var/ftp/empresa1 --shell /bin/false`

donde,

- ◆ `ftpasswd`, es el comando que permite crear los usuarios virtuales.
- ◆ `--passwd`, es el parámetro que pedirá la contraseña del usuario.
- ◆ `--name user-empresa1`, identifica al usuario virtual de nombre `user-empresa1`.
- ◆ `--file /etc/passwd.usuarios.virtuales`, creará, en caso de no existir, o modificará, el caso de existir el fichero de autenticación de usuarios virtuales.
- ◆ `--uid 107`, es el identificador perteneciente al usuario del sistema **ftp**. Se puede saber ejecutando el comando: `id ftp`.
- ◆ `--home /var/ftp/empresa1`, identifica a donde se conecta el usuario.
- ◆ `--shell /bin/false`, identifica una consola de comandos que no permite conexión como usuario del sistema.

- ✓ Ejecuta también el comando:

```
ftpasswd --passwd --name user-empresa2 --file /etc/passwd.usuarios.virtuales --uid 107 --home /var/ftp/empresa2 --shell /bin/false
```

A continuación prosigues, dependiendo si deseas configurar virtualhosts basados en IP o virtualhosts basados en nombre.

3.6.- Virtualhosts basados en nombre.

En la definición de la directiva VirtualHost podemos poner la IP del servidor FTP ó bien el nombre DNS correspondiente. En nuestro escenario, la IP_Servidor_FTP=192.168.200.250, **ftp.empresa1.com** y **ftp.empresa2.com** identifican a la misma máquina y a la misma IP. Ahora si, cada virtualhost, así como el servidor principal, deben servir en un puerto TCP distinto.

¿Cómo lo haces? Sigues el procedimiento:



1. En la configuración de ProFTPD (`/etc/proftpd/proftpd.conf`) debes activar la configuración del fichero **virtuals.conf** descomentando la línea:

```
<Include /etc/proftpd/virtuals.conf>
```

2. Agrega la configuración virtualhost para empresa1 en el fichero **/etc/proftpd/virtuals.conf**

```
<VirtualHost 192.168.200.120>
    Port 2121
    ServerName "Servidor FTP empresa1"
    AuthUserFile /etc/ftpusers
    AuthGroupFile /etc/ftpgroups
    DefaultUser anonymous
    RequireValidShell off
    <RequireAll>
```

3. Agrega la configuración virtualhost para empresa2 en el fichero **/etc/proftpd/virtuals.conf**

```
<VirtualHost IP_empresa2>
    Port 2122
    AuthUserFile /etc/ftpusers
    AuthGroupFile /etc/ftpgroups
    DefaultUser anonymous
    RequireValidShell on
    <RequireAll>
```

4. Configura permisos en las carpetas `/var/ftp/empresa1/` y `/var/ftp/empresa2/` para los usuarios virtuales:

```
<ChrootDirectory /var/ftp/empresa1 /var/ftp/empresa2>
```

5. Recarga la configuración del servidor.

```
<Reload>
```

Explicación fichero virtualhost:

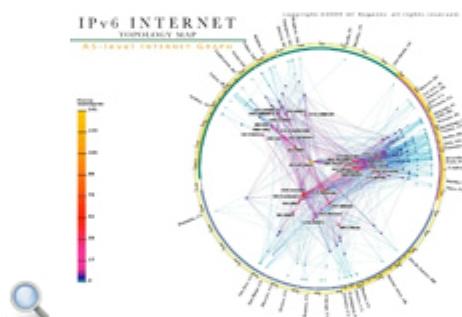
`<VirtualHost IP_Servidor_FTP>` → Inicio etiqueta virtualhost: define la IP del servidor ftp. También puede ser `<VirtualHost Nombre_DNS_Servidor_FTP>`

Port numero → Identifica el puerto TCP por el que espera la conexión el servidor FTP

3.7.- Virtualhosts basados en IP.

En la definición de la directiva VirtualHost podemos poner la IP del servidor FTP o bien el nombre DNS correspondiente. En nuestro escenario, la IP_Servidor_FTP=192.168.200.250 **ftp.empresa1.com** y **ftp.empresa2.com** identifican a la misma máquina y a distintas IP. Ahora es indiferente el puerto TCP en el que sirve cada virtualhost, así como el servidor principal, esto es puede ser el mismo o no ya que ahora cada puerto está relacionado con una IP distinta.

La IP que debemos poner ahora en la definición de la directiva Virtualhost cambia, cada IP corresponde a una interfaz de red del servidor FTP, en nuestro escenario IP_Servidor_FTP=192.168.200.250, **ftp.empresa1.com** se identifica con 192.168.200.251 y **ftp.empresa2.com** con 192.168.200.252



Este método no aporta ventajas sobre el anterior, es más, aún puede ser más difícil de mantener ya que las IP del servidor FTP se modifican con cierta frecuencia.

¿Cómo lo haces? Sigues el mismo procedimiento usado para los virtualhost basados en nombre únicamente se diferencia en la configuración de los virtualhost, así:

1. Modifica la configuración virtualhost para empresa1 en el fichero /etc/proftpd/virtuals.conf

•	<VirtualHost 192.168.200.251>
•	ServerName "Servidor FTP empresa1"
•	AuthUserFile /etc/proftpd/empres1/usuarios_ftp.txt
•	DefaultRoot /var/ftp/empresa1/
•	RequireValidShell off
•	</VirtualHost>

2. Agrega la configuración virtualhost para empresa2 en el fichero /etc/proftpd/virtuals.conf

•	<VirtualHost 192.168.200.252>
•	AuthUserFile /etc/proftpd/empres2/usuarios_ftp.txt
•	ServerName "Servidor FTP empresa2"
•	DefaultRoot /var/ftp/empresa2/
•	RequireValidShell off
•	</VirtualHost>

3. Configura permisos en las carpetas / var/ftp/empresa1/ y / var/ftp/empresa2/ para los usuarios virtuales:

•	chown -R ftpuser:ftpusers /var/ftp/empresa1/
•	chown -R ftpuser:ftpusers /var/ftp/empresa2/

4. Recarga la configuración del servidor.

•	/etc/init.d/proftpd restart
---	-----------------------------

Explicación fichero virtualhost:

<VirtualHost IP_Servidor_FTP> → Inicio etiqueta virtualhost: define la IP del servidor ftp. También puede ser <VirtualHost Nombre_DNS_Servidor_FTP>

ServerName "Servidor FTP empresaX" → Configura el nombre que se muestra en la conexión de los usuarios.

DefaultRoot / var/ftp/empresaX/ → Definición de la ruta que sirve ProFTFD, en este caso /var/ftp/empresaX/ mediante la directiva DefaultRoot, esto es, indica que los usuarios cuando conecten con el servidor ftp, estarán encerrados en la ruta /var/ftp/empresaX/, con lo cual no podrán acceder a otro directorio que no esté contenido dentro de éste.

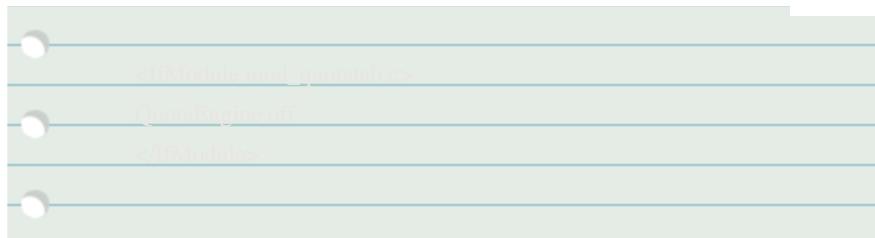
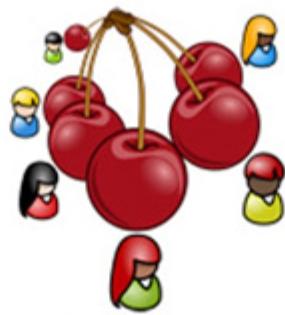
RequireValidShell off → No es necesario tener una shell declarada en el fichero /etc/shells .

</VirtualHost> → Fin de la etiqueta VirtualHost: fin de la definición de este virtualhost para la empresa1.

3.8.- Cuotas de disco para los usuarios (I).

La capacidad de almacenamiento no es infinita, por lo tanto será interesante saber cómo crear cuotas de disco para los usuarios y ya puestos para los usuarios en los virtualhosts.

El archivo **/etc/proftpd/proftpd.conf** llama mediante la directiva **Include** al archivo **/etc/proftpd/modules.conf** en el que están activadas las cuotas (**LoadModule mod_quotatab.c**, **LoadModule mod_quotatab_file.c**), luego para activarlas tienes que sustituir en el archivo **/etc/proftpd/proftpd.conf** el código:



por el código siguiente:



donde,

<IfModule mod_quotatab.c> ... </IfModule> → Indica que si el módulo **mod_quotatab.c** está cargado en el archivo **/etc/proftpd/modules.conf** se realizarán las directivas que contengan.

QuotaEngine on → Activa las cuotas.

QuotaLog /var/log/proftpd/quota.log → Indica el archivo de registro sobre cuotas.

<IfModule mod_quotatab_file.c> ... </IfModule> → Indica que si el módulo **mod_quotatab_file.c** está cargado en el archivo **/etc/proftpd/modules.conf** se realizarán las directivas que contengan.

QuotaLimitTable file:/etc/proftpd/ftpquota.limittab → Indica el archivo sobre el límite de cuota Limit.

QuotaTallyTable file:/etc/proftpd/ftpquota.tallytab → Indica el archivo sobre el límite de cuotas Tally.

Para ProtFTPD existen básicamente dos tipos de cuotas: **limit** y **tally**.

- ✓ **Limit:** Es la cuota que te interesa si estás pensando en restringir el espacio en disco a los usuarios. Éste puede ser soft, cuando existe un espacio de gracia(tamaño en bytes) que puede sobrepasar el límite, o hard cuando no existe un espacio de gracia.
- ✓ **Tally:** Utilizado cuando quieras limitar el número de ficheros que se utilizan.



Para saber más

Para mayor información sobre las cuotas puedes visitar la documentación oficial de ProFTPD sobre mod_quotatab.

[Documentación oficial de ProFTPD sobre mod_quotatab](#)

La forma más sencilla de crear las cuotas es hacer el símil upload=espacio en disco, con lo cual los archivos subidos no pueden ocupar más del que queramos darle como espacio en disco, esto es, los bytes subidos funcionan como espacio en disco, ya que no existe diferencia entre ellos pues los bytes cargados a través de FTP se almacenan automáticamente en el disco, por lo que deberías emplear la cuota tipo limit.

Puedes crear las cuotas mediante el comando [ftpquota](#):

/usr/local/etc/ftpquota -type=limit -name=user-empresa1 -quota=1000M
bytes uploaded <= quota & bytes written <= quota

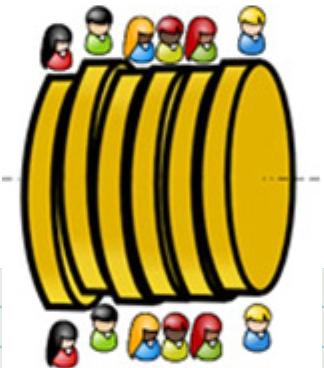
Por ejemplo, si quisieras limitar a un usuario de nombre user-empresa1 el espacio de subida en 1 GB:

/usr/local/etc/ftpquota -type=limit -name=user-empresa1 -quota=1G
bytes uploaded <= quota & bytes written <= quota

Y si quisieras limitar la subida y bajada a 4 GB y 2 GB respectivamente al usuario user-empresa2:

/usr/local/etc/ftpquota -type=limit -name=user-empresa2 -quota=4G
bytes upload=4 & bytes download=2 & quota=0G & write permission denied

3.8.1.- Cuotas de disco para los usuarios (II).



- ✓ Para actualizar la cuota de un usuario, por ejemplo, user-empresa1:

```
# ftpquota -update-record -type=limit -name=user-empresa1 -quota=1000000
```

con lo cual, si compruebas de nuevo los registros, verás que los cambios surgieron efecto:

```
# ftpquota -show-records -type=limit -name=user-empresa1 -quota-type=bytes
```

Name: user-empresa1
Quota Type: bytes
For Session: False
For User: True
Used Disk Space: 101724800000
Downloaded bytes: unlimited
Uploaded bytes: unlimited
Downloaded files: unlimited
Uploaded files: unlimited

- ✓ Para desactivar la cuota de un usuario debes borrar el registro, por ejemplo, user-empresa1:

```
# ftpquota -delete-record -type=limit -name=user-empresa1 -quota-type=bytes
```

con lo cual, si compruebas de nuevo los registros, verás que los cambios surgieron efecto:

```
# ftpquota -show-records -type=limit -name=user-empresa1 -quota-type=bytes
```

Dropzone: empty table



Para saber más

Puedes ver la ayuda del comando `ftpquota` mediante: `ftpquota - -help`.

No olvides recargar la configuración del servidor ProFTPD: /etc/init.d/proftpd restart.

3.9.- Acceso seguro mediante TLS.



En Debian 6 (squeeze) al instalar el paquete proftpd ya se puede establecer la conexión por TLS siempre y cuando se configure el archivo **/etc/proftpd/tls.conf** y procedas como sigue:

1. Edita el archivo **/etc/proftpd/proftpd.conf** y descomenta la línea:

```
#TLSModule /etc/proftpd/tls.conf
```

2. Crea las claves, pública y privada, para la conexión cifrada:

- ✓ Método 1: Instalación del paquete  [openssl](#) y ejecución del comando `openssl`.

```
Generating RSA private key, 1024 bit(s) long
-----
-----
```

You are about to be asked to enter information that will be incorporated
into your certificate request:
What you are about to enter is what is called a Distinctive Name or a DN

Country Name (2 letter code) [AU]:
State or Province Name (abbreviated if possible) [Some-State]:
Locality Name (e.g. city) [San Jose]:
Organization Name (e.g. company) [My Company Ltd.]:
Organization Unit Name (e.g. section) [Software Development]:
Email Address [root@...]:

- ✓ Método 2: Comando `proftpd-gencert`. Los dos comandos anteriores se resumen en uno, con la salvedad que el certificado será válido solamente durante 1 año y no 10:

```
-----
```

3. Modifica los permisos:

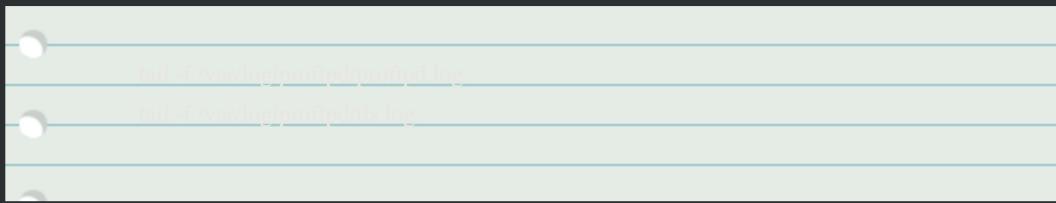
```
-----
```

4. Modifica el fichero **/etc/proftpd/tls.conf** como se indica en el fichero ejemplo [tls.conf](#) (0.01 MB)
5. Recarga la configuración del servidor ProFTPD:

```
-----
```

6. Comprueba la configuración mediante el usuario del servidor ftp **invitado**, creado anteriormente, con un cliente FTPES, es decir, un cliente ftp que permita la conexión por TLS como FileZilla.

Puedes verificar en tiempo real las conexiones con el servidor ftp revisando los archivos de registro mediante los comandos:



7. Si deseas, puedes hacer valer la configuración para todos los usuarios, incluso aquello pertenecientes a virtualhost, modificando el fichero **/etc/proftpd/tls.conf** como se indica en el fichero ejemplo [tls2.conf](#) (0.01 MB).



Debes conocer

Te proponemos el siguiente enlace de un vídeo práctico sobre cómo configurar TLS en el servidor ProFTPD y cómo configurar una plantilla de FileZilla que soporta conexión TLS. La configuración se realiza sobre una distribución GNU/Linux basada en Debian.

[Resumen textual alternativo](#)

Anexo.- Licencias de Recursos.

Licencias de recursos

Recurso (1)	Datos del recurso (1)
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: Montaje sobre: http://www.flickr.com/photos/_rfc_/5954364713/sizes/l/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: Montaje sobre: http://www.flickr.com/photos/_rfc_/5954921098/sizes/s/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: http://www.flickr.com/photos/_rfc_/5954919330/sizes/m/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: http://www.flickr.com/photos/_rfc_/5954919724/sizes/z/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: http://www.flickr.com/photos/_rfc_/5954363473/sizes/l/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: http://www.flickr.com/photos/_rfc_/5954919436/sizes/l/in/photostream/
	Autoría: _rfc_. Licencia: CC BY NC SA 2.0. Procedencia: Montaje sobre: http://www.flickr.com/photos/_rfc_/5954920726/sizes/z/in/photostream/



Autoría: The ProFTPD Project.
Licencia: Copyright © 1999 - 2011, The ProFTPD Project. Derecho de citar.
Procedencia: <http://www.proftpd.org/proftpd.png>.



Autoría: _rfc_.
Licencia: CC BY NC SA 2.0.
Procedencia:
http://www.flickr.com/photos/_rfc_/5954364775/sizes/s/in/photostream/



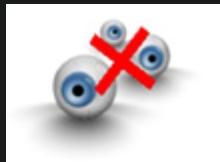
Autoría: _rfc_.
Licencia: CC BY NC SA 2.0.
Procedencia:
http://www.flickr.com/photos/_rfc_/5954364567/sizes/s/in/photostream/



Autoría: querkmachine.
Licencia: CC BY-NC-SA 2.0.
Procedencia:
<http://www.flickr.com/photos/querkmachine/5743377982/sizes/s/in/photostream/>



Autoría: _rfc_.
Licencia: CC BY NC SA 2.0.
Procedencia:
http://www.flickr.com/photos/_rfc_/5954920366/sizes/z/in/photostream/



Autoría: _rfc_.
Licencia: CC BY NC SA 2.0.
Procedencia:
http://www.flickr.com/photos/_rfc_/5954364823/sizes/s/in/photostream/

Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.

Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

Historial de actualizaciones

Versión: 01.02.02	Fecha de actualización: 25/01/16	
Actualización de materiales y correcciones menores.		
Versión: 01.02.00	Fecha de actualización: 07/01/15	Autoría: Jesús Manuel Marín Navarro
Añadido nuevo contenido en apartado 1.4 sobre vulnerabilidades en servidores Web. Corregido apartado 3.4		
Versión: 01.01.00	Fecha de actualización: 02/01/14	Autoría: Jesús Manuel Marín Navarro
Agregado todo el apartado 1 completo de alta disponibilidad y alto rendimiento en GNU/Linux. Se han modificado los contenidos, las orientaciones del alumnado y el mapa conceptual. Pero el mapa conceptual necesita revisión.		
Versión: 01.00.00	Fecha de actualización: 30/10/13	
Versión inicial de los materiales.		