INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN

USERS

Argentina $ 22.- // México $ 49.-

Técnico en REDES &SEGURIDAD

18

SERVIDORES WEB Y FTP

En este fascículo aprenderemos sobre los servidores utilizados para alojar sitios web y transferir archivos mediante el protocolo FTP. Además, conoceremos la manera de configurarlos y administrarlos correctamente.

- INSTALACIÓN Y ADMINISTRACIÓN
- HOSTING PROPIO VERSUS CONTRATADO
- SERVICIOS RECOMENDADOS
- SEGURIDAD EN SERVIDORES
- DIFERENCIAS ENTRE LOS PROTOCOLOS HTTP Y HTTPS

USERS

Técnico en REDES &SEGURIDAD

Coordinador editorial
Paula Budris

Asesores técnicos
Federico Pacheco
Javier Richarte

Nuestros expertos
Valentín Almirón
José Bustos
Gustavo Cardelle
Rodrigo Chávez
Alejandro Gómez
Javier Medina
Gustavo Martín Moglie
Pablo Pagani
Gerardo Pedraza
Ezequiel Sánchez

Impreso en Sevagraf S.A. Impreso en Argentina.

Técnico en redes y seguridad / coordinado por Paula Budris. - 1a ed. - Buenos Aires: Fox Andina, 2013
576 p. ; 28 x 20 cm. (Users; 22)

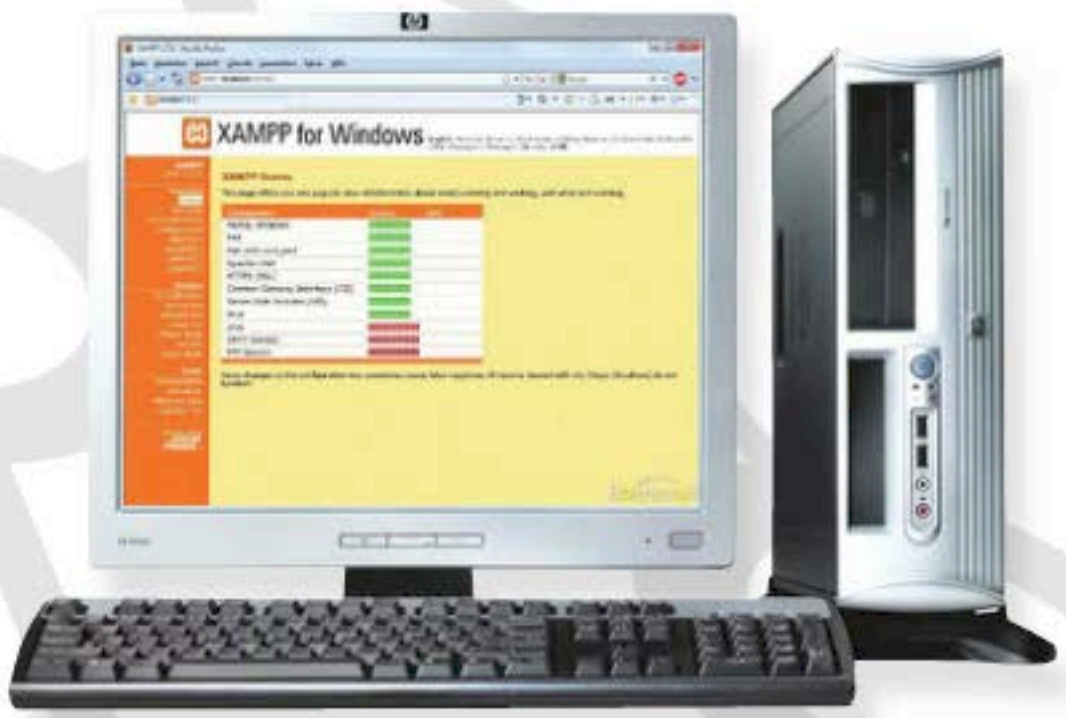
ISBN 978-987-1857-78-4

1. Informática. 2. Redes. I. Budris, Paula, coord.
CDD 004.68

En esta clase veremos...

Las características y ventajas de contar con un servidor web y FTP; además, los pasos necesarios para administrarlos, la seguridad y los protocolos asociados.

En la clase anterior, revisamos las tareas de administración remota. Comenzamos realizando una introducción a esta tecnología y vimos las opciones de software que pueden ayudarnos. Detallamos los usos de la administración remota, y aprendimos a instalar un cliente y servidor UltraVNC. Luego, analizamos las ventajas de TeamViewer y aprendimos a ponerlo en marcha; también conocimos algunas aplicaciones similares y nos adentramos en la tecnología SSH, plataforma Citrix e Intel vPro.
En esta clase, revisaremos las características de los servidores web y FTP, conoceremos qué son y qué ventajas nos entregan. Aprenderemos a instalar estos servidores y revisaremos las consideraciones que debemos tener en cuenta para administrarlos. Para continuar, veremos veremos algunos conceptos importantes sobre seguridad en servidores web y en servidores FTP, luego conoceremos los protocolos HTTP y HTTPS, para analizar las diferencias y opciones que los caracterizan.



18

2
Qué es un servidor web

10
Paso a paso: Instalar un servidor FTP en Windows

20
Seguridad en servidores web

24
HTTP y HTTPS

Qué es un servidor web

Es una aplicación de software que cumple el rol de servidor y se ejecuta en una computadora a la espera de peticiones web de los clientes.

Un **servidor web**, también denominado servidor **HTTP**, es una aplicación o software que se ejecuta en una computadora que cumple con el rol de servidor en una arquitectura cliente-servidor. Esta realiza conexiones bidireccionales o unidireccionales, sincronizadas o no, con uno o varios clientes, recibiendo peticiones y respondiendo dichas solicitudes en un lenguaje de programación determinado del lado del cliente.
Las respuestas recibidas por el cliente son compiladas y ejecutadas por un navegador web. Para la transmisión de datos entre el servidor y el cliente, por lo general, se utiliza el protocolo de red HTTP que emplea el puerto TCP 80 y se encuentra en la capa de aplicación del modelo OSI. El término de servidor web también es utilizado para referirse a la computadora que ejecuta la aplicación o software de servidor web.

Sitios web populares como Google poseen clústeres de servidores que se usan para satisfacer las peticiones diarias producidas.

Funcionamiento

La aplicación de servidor web se ejecuta en una computadora servidor, a la espera de peticiones de clientes (navegadores web que se ejecutan en los clientes). Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición **GET**, en la que una página o un sitio se solicita, a través de la **URL** (*Uniform Resource Locator* o localizador uniforme de recursos) al servidor web. Por ejemplo, cuando ingresamos la dirección **www.organizacion.com/index.html** en la barra de direcciones del navegador, estamos ejecutando la petición GET que presentamos a continuación:

```
GET / index.html HTTP / 1.1 HOST: www.organizacion.com
```

Los formularios presentes en una página web se usan para el envío de datos desde el cliente al servidor. Por lo general, utilizan peticiones del tipo POST.

Las peticiones **POST** son el segundo tipo de petición HTTP más utilizado; su objetivo es el envío de datos. Los datos por enviar al servidor se incluyen en el cuerpo de la misma petición con las cabeceras HTTP asignadas correspondientes respecto al tipo de petición. Casi siempre se asocia con los formularios web en los que los datos suelen ser cifrados para ser enviados de manera segura al servidor. Todo navegador web provee a sus usuarios de una interfaz para poder realizar una o varias solicitudes web. La interfaz está conformada por aquellos elementos del navegador que permiten realizar la petición de forma activa. Estas peticiones pueden ser realizadas también por aplicaciones que no sean un navegador web.

Elementos

A continuación, vamos a detallar algunos elementos que poseen las interfaces de los navegadores para realizar peticiones.

- **Hipervínculo**, enlace o link: se trata de una parte o porción de contenido de una página o sitio web, ya sea texto, imagen u otro elemento que direcciona a una URL. Al pulsar un hipervínculo, el navegador genera una petición GET automática a la dirección URL de dicho link.
- **Formulario:** es un contenedor de datos presente en una página, que captura información ingresada por el usuario para ser almacenada después. Al momento de confirmar el envío de un formulario, el navegador web genera una petición GET o POST (comúnmente POST) automática para enviar los datos al servidor a una dirección particular.
- **Barra de direcciones:** todos los navegadores web incluyen una barra de direcciones mediante la cual podemos acceder, en forma manual, a cualquier dirección URL, de modo que el navegador generará una petición GET automática a dicha URL cada vez que el usuario lo indique ingresando la dirección.
- **Script (activo o pasivo):** cualquier aplicación JavaScript tiene acceso al estado del navegador y puede modificar los datos que describen tal estado. Por eso, un elemento de este tipo puede solicitar una dirección de forma pasiva (sin la intervención del usuario) o de forma activa (mediante alguna acción del usuario). También puede enviar datos de forma pasiva o activa.

El servidor web, luego de recibir una solicitud, la responde devolviendo una página web que va a ser visible en nuestro navegador o un mensaje de error que también será visible como una página. Por ejemplo, al ingresar la URL **www.organizacion.com** en la barra de direcciones de nuestro navegador, este realiza una petición HTTP al servidor de dicha dirección.

> HTML es el lenguaje por excelencia utilizado para el desarrollo de páginas web.

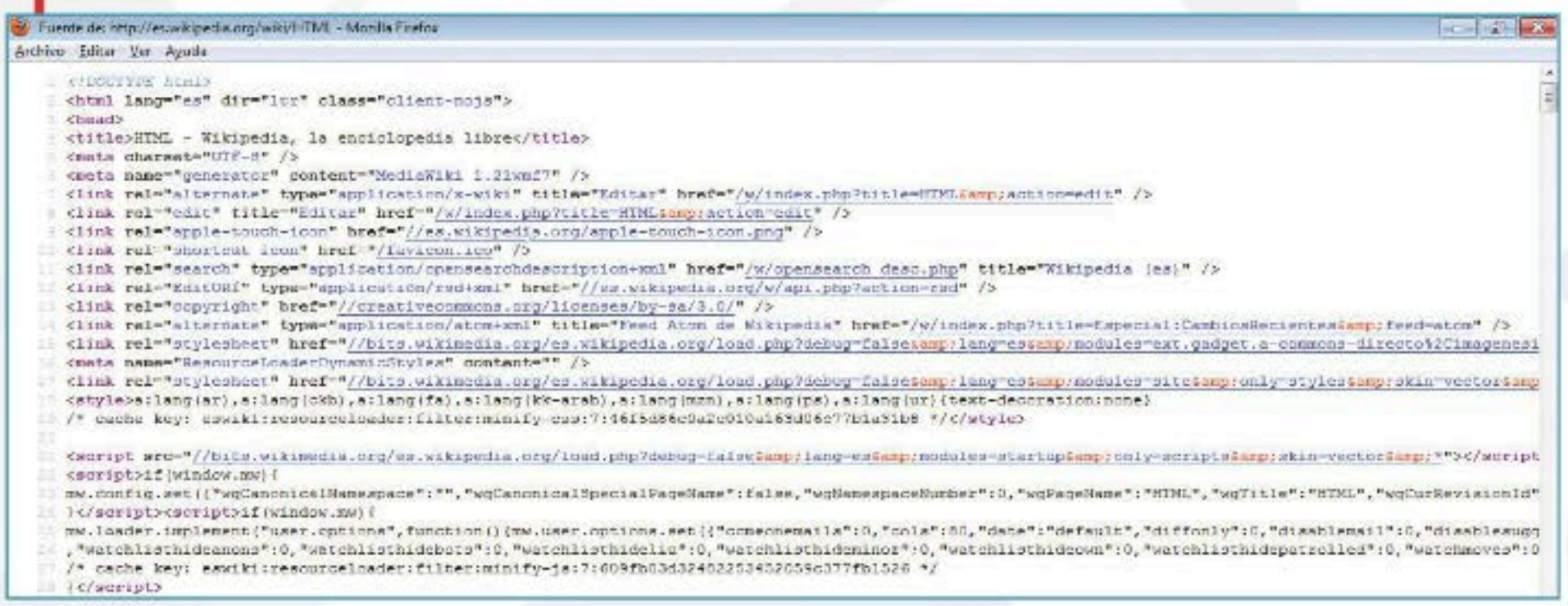
El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan solo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de ella.

LOS NAVEGADORES WEB OFRECEN UNA INTERFAZ PARA REALIZAR UNA O VARIAS SOLICITUDES WEB.

Aplicaciones

Además de la transferencia de código HTML, los servidores web pueden entregar aplicaciones web. Estas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- **Aplicaciones en el lado del cliente:** el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java **applets** o JavaScript: el servidor proporciona el código de las aplicaciones al cliente, y este, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje JavaScript y Java, aunque pueden añadirse más lenguajes mediante el uso de **plugins**.
- **Aplicaciones en el lado del servidor**: el servidor web ejecuta la aplicación; esta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP. Es más práctico que las aplicaciones se encuentren del lado del servidor ya que, al ejecutarse en el servidor y no en los clientes, estos últimos no requieren de ningún agregado para poder interpretar Java o JavaScript. Solo es necesario un navegador web básico. ■



Qué es un servidor FTP

Es una aplicación de software que se utiliza para realizar el intercambio de archivos bajo la arquitectura de computadoras cliente-servidor.

Antes comenzar con la definición de un servidor FTP, debemos recordar qué es **FTP** (*File Transfer Protocol*) o protocolo de transferencia de archivos. Como bien lo especifica su nombre, consiste en un protocolo de red que se utiliza para el intercambio de archivos entre dos o más computadoras. Un servidor FTP es una aplicación o software que se ejecuta en una computadora que cumple con el rol de servidor en una arquitectura cliente-servidor, que utiliza el protocolo antes mencionado y que se encuentra conectada generalmente a Internet (puede que se encuentre conectada a otros tipos de redes, como redes LAN, MAN, etc.). Esta realiza conexiones bidireccionales o unidireccionales entre uno o varios clientes para intercambiar archivos. Las computadoras dispuestas al intercambio de archivos deben encontrarse conectadas a una red **TCP** (*Transmission Control Protocol*) o protocolo de control de transmisión. Los programas servidores FTP no suelen encontrarse en los ordenadores personales, por lo que un usuario casi siempre utilizará el FTP para conectarse remotamente a uno y, así, intercambiar información con él. Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes; o como **servidor de backup** (copia de seguridad) de los archivos importantes que pueda tener una empresa. Existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el **SFTP** (*Secure File Transfer Protocol*) o protocolo de transferencia de archivos seguro. Una computadora cliente puede conectarse a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada nodo.

Con el surgimiento de Internet, es común el uso de servidores FTP web, a los cuales se accede a través del navegador.

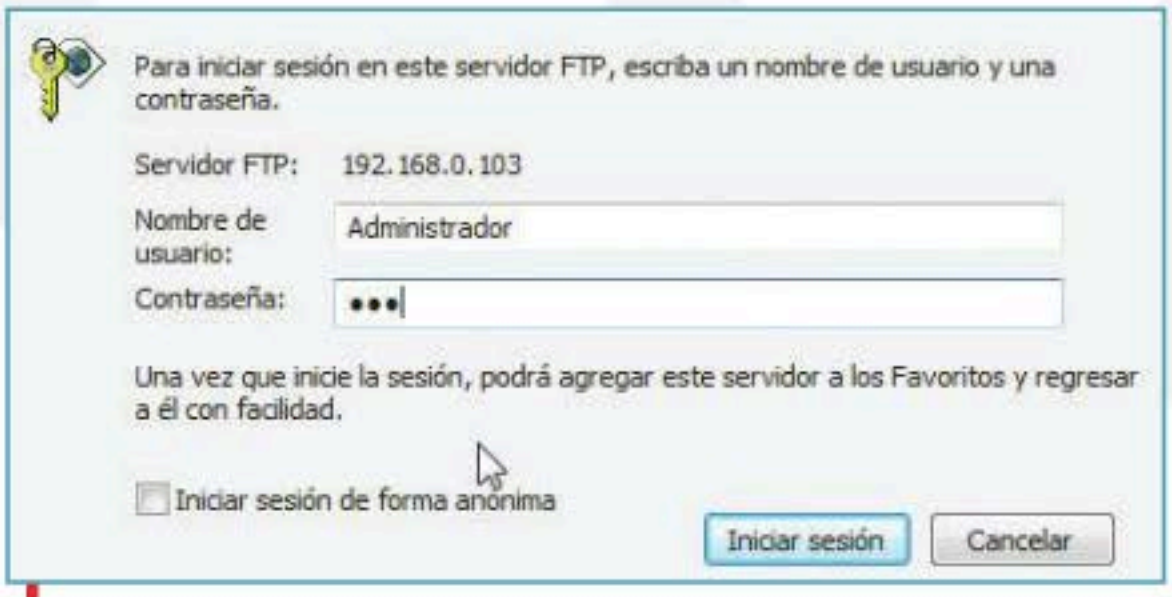
Funcionamiento

El servicio FTP se encuentra dentro de la capa de aplicación del modelo de red TCP/IP y utiliza los puertos TCP 20 y 21.

La principal desventaja que presenta es que está pensado para ofrecer la máxima velocidad, pero no la máxima seguridad, ya que todo el intercambio de información, desde el *login* y *password* del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor y apropiarse de los archivos transferidos. Para solucionar este problema, son de gran utilidad aplicaciones como **SCP** y **SFTP**, incluidas en el paquete SSH, que permiten transferir archivos, pero cifrando todo el tráfico.

FileZilla

FileZilla es una aplicación cliente FTP multiplataforma, de código abierto y libre, licenciada bajo la Licencia Pública General de GNU. Soporta los protocolos de transferencia de archivos **FTP**, **SFTP** (*Secure FTP*) y **FTP** sobre **SSL/TLS**. Como características principales podemos resaltar que posee un administrador de sitios, permite crear y almacenar listados de sitios, FTP con la información de conexión, registro de mensajes y, también, muestra en modo consola las instrucciones enviadas al servidor así como las respuestas obtenidas.



En el momento de acceder a un servidor FTP, se nos solicitará el ingreso de un nombre de usuario y una contraseña válidos.

Clientes FTP

Cuando un navegador no posee integrada la función FTP, o si deseamos subir archivos a una computadora remota, necesitaremos de una aplicación cliente FTP. Un cliente FTP es un programa que se instala en el ordenador del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.
Para utilizar un cliente FTP, necesitamos conocer el nombre del archivo, la computadora en la que se encuentra (servidor, en el caso de descarga de archivos), la computadora a la que deseamos transferir el archivo (en caso de querer subirlo nosotros al servidor) y la carpeta en la que se encuentra.
Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Microsoft Windows, GNU/Linux y Unix. Sin embargo, hay clientes disponibles con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrada la función FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

Funcionamiento

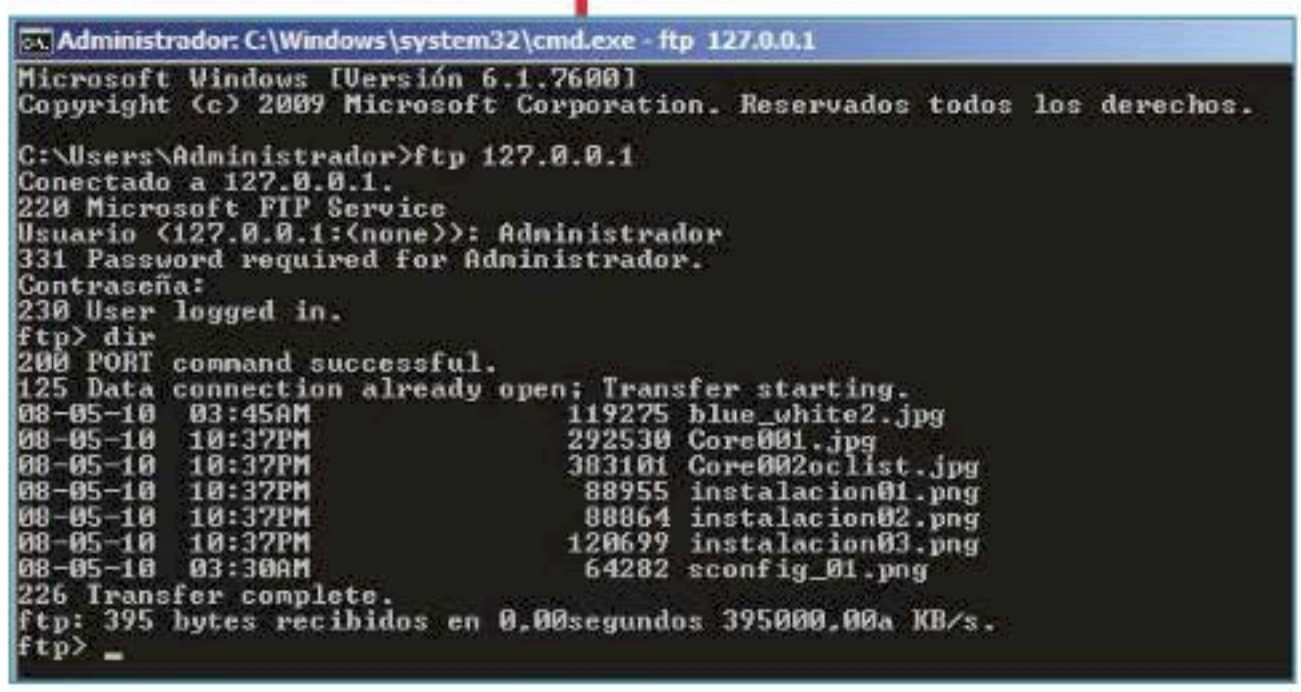
El intérprete de protocolo de usuario (IP de usuario) inicia una conexión en el puerto 21. Luego, genera las instrucciones estándares FTP y las transmite hacia el servidor a través de la conexión de control establecida. Las respuestas a las instrucciones son generadas en el intérprete de protocolo de servidor (IP de servidor) y enviadas al IP de usuario a través de la misma conexión.
Las **instrucciones FTP** definen parámetros para la conexión de datos (puerto de datos, modo de transferencia, tipo de representación y estructura) y la naturaleza de la operación sobre el sistema de archivos (almacenar, recuperar, añadir, borrar, etc.). El proceso de transferencia de datos (DTP) de usuario u otro proceso en su lugar debe esperar a que el servidor inicie la conexión al puerto de datos especificado (puerto 20 en modo activo o estándar) y transferir los datos en función de los parámetros que se hayan especificado.
La comunicación establecida entre cliente y servidor es independiente del sistema de archivos utilizado en cada computadora, por lo que no importa si sus sistemas operativos son distintos. Ambos utilizan el protocolo estandarizado FTP. Otro punto para tener en cuenta consiste en que la conexión de datos es bidireccional, o sea, se puede usar simultáneamente para enviar y para recibir datos. En sus inicios, el problema era la localización de los servidores en la red.
El usuario debía conocer la localización en donde se alojaban los archivos. Con el arribo de Internet y los potentes motores de búsqueda actuales, como Google, la localización de los servidores FTP dejó de ser un problema. En la actualidad, cuando el usuario se descarga un archivo a partir de un enlace de una página web, desconoce que lo está haciendo desde un servidor FTP. El servicio FTP ha evolucionado a lo largo del tiempo y, hoy en día, es muy utilizado en Internet, en redes corporativas, Intranets, etc. Soportado por cualquier sistema operativo, existe gran cantidad de software basado en el protocolo FTP.

Tipos de usuarios

Veremos los tipos de usuarios en un servicio de FTP.

- **Usuario anónimo**: permite que cualquier persona acceda a un servicio FTP sin que se deba realizar la creación de la cuenta específica.
- **Usuario común**: debemos contar con una cuenta de este tipo si necesitamos privilegios de acceso, creación o modificación, a cualquier parte restringida del sistema de archivos del servidor FTP.
- **Usuario invitado**: se utiliza para permitir el acceso a entornos restringidos, como ocurre con el usuario anónimo, pero con más privilegios. ■

La consola de comandos de Windows posee el comando FTP, para conectarnos, autenticarnos e intercambiar archivos.



Hosting propio versus hosting contratado

En estas páginas conoceremos las características, ventajas y desventajas de las opciones de hosting propio y contratado, cada una de las cuales nos permiten darnos a conocer en la Web.

El término **hosting**, en forma literal, significa hospedaje; en informática, con él se hace referencia al lugar virtual donde se almacenarán los datos de nuestra página web. Además, no se trata solo de almacenar los archivos del sitio web, sino también de las herramientas para su edición, configuración y administración. El lugar físico donde se encuentran los servidores dependerá de la empresa dedicada a prestar el servicio.

Hosting propio

Si administramos la red de una empresa, y nos piden la tarea de implementar un hosting propio dentro de la misma empresa, mientras los costos lo permitan, las ventajas son varias.

Si queremos implementar un hosting propio a nivel personal, la primera gran desventaja con la que nos encontraremos es el costo para su implementación física. Antes de empezar con la configuración del equipo que utilizaremos como hosting propio, debemos realizar en el hardware todas las pruebas que consideremos importantes. Además, debemos colocar en una bahía disponible un soporte para discos removibles. Conviene realizar una copia exacta del disco duro en otro disco (si no contamos con algún sistema RAID); de esta forma, en caso de que el disco principal se dañe, podremos reemplazarlo inmediatamente y poner operativo el servidor en un corto tiempo. Una vez terminadas las pruebas sobre el hardware, instalaremos el sistema operativo y solo el software que se requiera para el funcionamiento del hosting, como soporte para base de datos **MySQL**; consultas PHP; una interfaz para la administración del sitio, como Joomla; servicios FTP para permitir la carga remota de archivos. Hay que tener presente que no debemos sobrecargar de aplicaciones o servicios que no serán utilizados; por ejemplo, incluir soporte para varios lenguajes PHP, ASP, JavaScript, etc., de los cuales ya sabemos que utilizaremos uno solo, aumenta el riesgo de vulnerabilidades que puede tener cada complemento que procedemos a instalar.

Zoby Host es una alternativa de hosting gratuito o pago.

HOSTING DEFINE EL LUGAR VIRTUAL DONDE SE ALMACENA UN SITIO O PÁGINA WEB.

La ventaja que nos brinda el tener nuestro propio hosting es la libertad y dominio total para su configuración y modificación; pero, a su vez, la desventaja que eso conlleva es que debemos prestar suma atención a las actualizaciones pertinentes del software que hayamos instalado, en especial, a las vulnerabilidades que se vayan conociendo y puedan poner en peligro su funcionalidad.

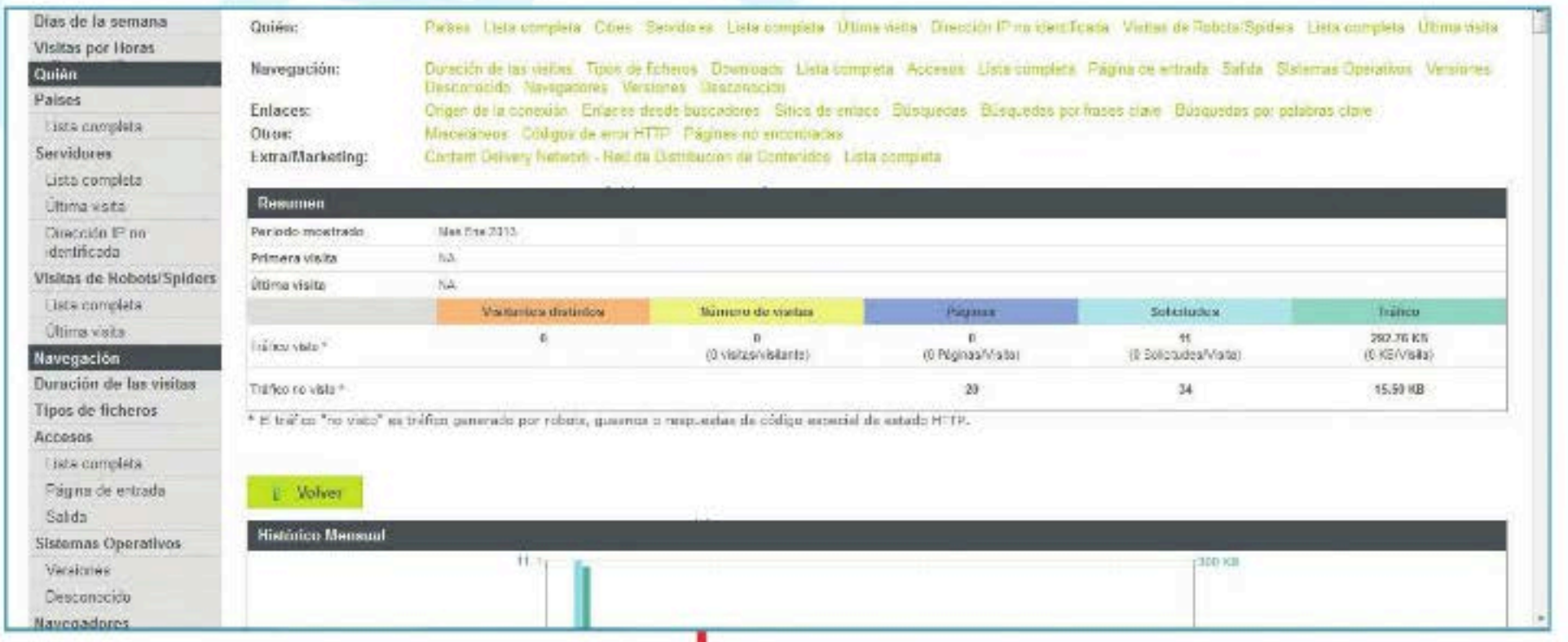
La ubicación física del equipo no debe ser un lugar aislado, pero sí seguro. Generalmente, se encuentra con otros servidores internos, de manera que solo el personal autorizado pueda ingresar.

Tampoco podemos olvidarnos de realizar un **backup**.

Es posible realizarlo de forma manual, conectando un disco externo, o realizándolo a través de una tarea programada, para que se compriman todas las carpetas de las cuales deseamos tener un backup y, luego, se copien a otro equipo como copia de resguardo.

Hosting contratado

A diferencia del modo anterior, en este caso, alquilaremos un espacio virtual para almacenar nuestra página web, con las herramientas necesarias para su creación, edición o carga de



Al tener un hosting contratado, este nos proveerá de un panel de estadísticas sobre el estado del sitio.

archivos editados. Además, los servicios que serán contratados, variarán mucho en precio según sus características. La ventaja que nos provee este modelo es que solo nos tendremos que focalizar directamente en las características del sitio web y, según las características con las que deseemos contar, serán los servicios que contrataremos. Así por ejemplo, si vamos a utilizar varias bases de datos relacionadas, debemos asegurarnos antes de contratar los servicios qué cantidad de bases de datos nos permiten utilizar y qué versiones están soportadas. Por una cuestión de seguridad, como las bases de datos llevan un nombre de usuario y contraseña para que se puedan agregar y consultar datos, es muy probable que ese nombre sea distinto de `admin` o `root`. Además de contratar el hosting y los servicios que se ajusten a nuestro requerimiento, también aparecerá qué tráfico mensual está incluido en nuestro plan. Debemos prestar atención a este punto ya que, si contamos con un gran volumen de clientes, el tráfico contratado deberá poder cubrir esa demanda y, además, tener un margen para los futuros y potenciales clientes.

¿Cuál utilizar?

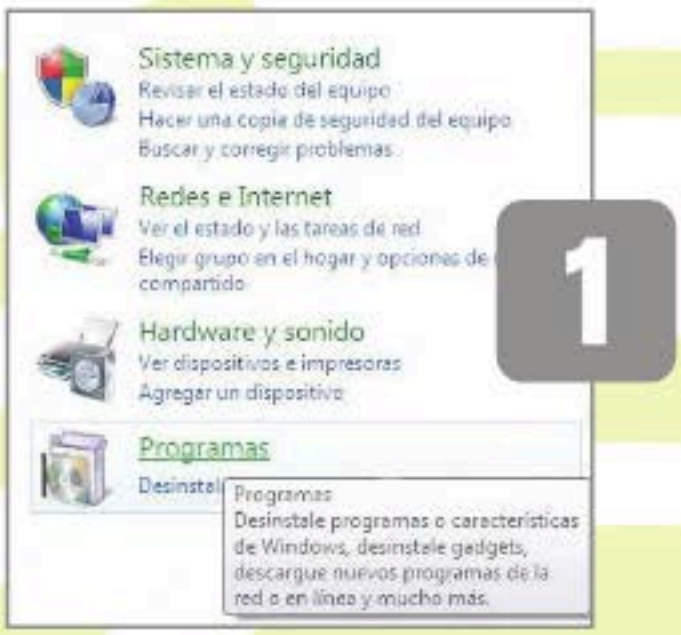
En ambos casos, para ayudarnos a decidir cuál conviene más, debemos evaluar todas las necesidades y dudas que tenemos. ¿A qué público queremos llegar? ¿Qué servicios necesitamos? De nada nos sirve pagar un hosting que nos provea de herramientas que no utilizaremos, como tampoco elegir un hosting económico y estar muy limitados por las pocas herramientas disponibles. Si lo que deseamos hacer es un proyecto personal, no conviene invertir directamente en un hosting contratado ni propio, ya que las tasas de transferencias para subidas que nos provee nuestro ISP son bajas. En este caso, nos conviene al principio probar con alguna alternativa gratuita, como **Google Sites**, o, si queremos tener nuestro propio blog, podemos contar con un hosting gratuito como **WordPress.com** o **Blogger.com**. Debemos tener presente que un hosting gratuito, si bien es una gran alternativa para nuestro proyecto personal, insertará en nuestro sitio publicidades que no podremos configurar. Si no las deseamos, es posible contratar en el mismo hosting una opción para que no aparezcan las publicidades; además, contaremos con algún servicio adicional, y los precios resultan casi siempre más económicos que los de un hosting contratado. Si el hosting será para nuestro cliente que manejará e interactuará con sus clientes vía web, el hosting contratado puede considerarse una alternativa. Pero los servidores que tienen las empresas de servicios de hosting albergan cientos de sitios web en sus servidores, con lo cual, un ataque dirigido a uno de esos sitios puede afectar en forma indirecta al sitio de nuestro cliente y dejarlo inoperativo por algún tiempo indeterminado. En este caso, por la actividad de nuestro cliente, convendrá poner un hosting propio, para tener un mayor control y dominio de él. ■

Desventajas

Una de las mayores desventajas de implementar un hosting propio es afrontar el gasto económico principal para su implementación. Debemos contar con un equipo que se encuentre preparado para estar prendido todo el tiempo, con lo cual el tema de su refrigeración será un factor importante. Además, debemos contratar una mayor tasa de transferencia para subidas con nuestro ISP; esto permitirá, a nuestros potenciales visitantes, cargar nuestra página web con mayor fluidez y permitir consultas simultáneas.

Instalar un servidor web en Windows

Internet Information Server (IIS) es la solución de software de servidor web que ofrece Microsoft para sus clientes.

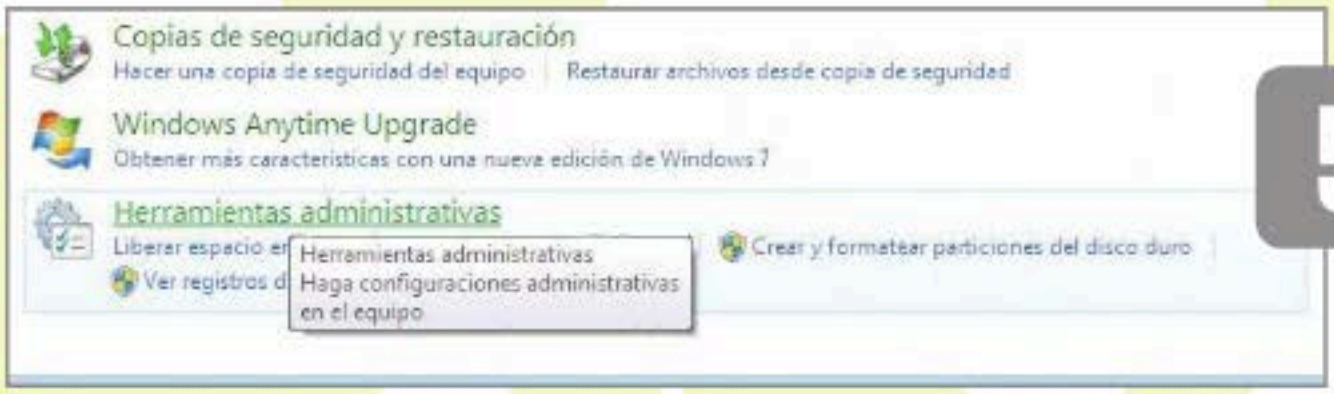


1 Como primer paso para poder instalar **Internet Information Server** en nuestra computadora con Windows 7, debemos acceder a **Panel de Control/Programas**. Este último menú es el que nos permite instalar y desinstalar programas, al igual que características de Windows.

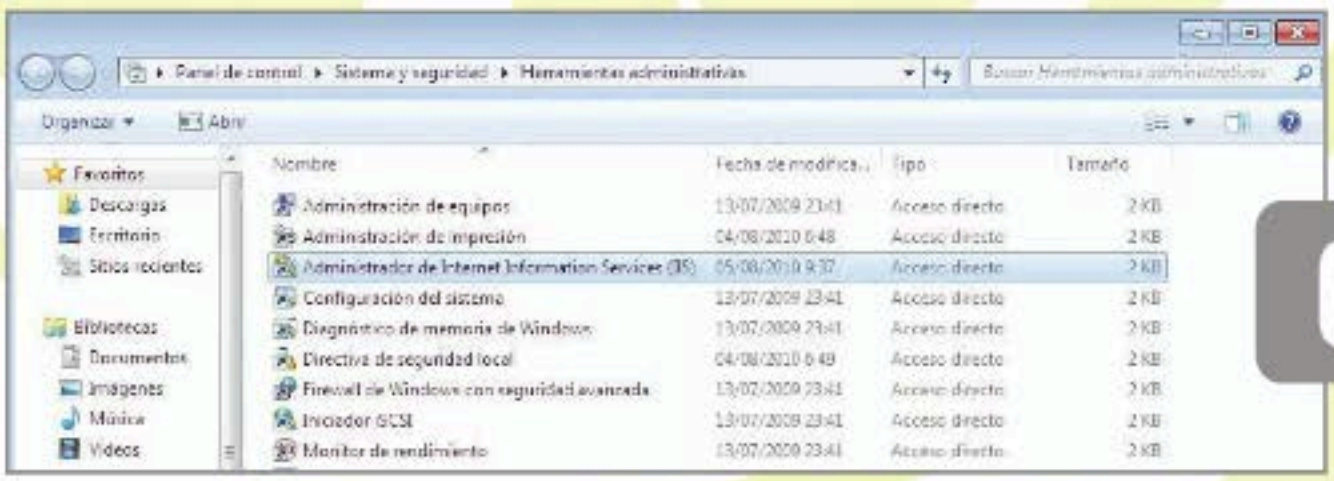
2 A continuación, seleccionamos la opción denominada **Programas y características/Activar o desactivar las características de Windows**, que se encuentra entre las opciones presentadas en la ventana; luego esperamos mientras aparece la ventana de características.

3 Se abre una ventana en la cual el sistema nos permite seleccionar las características que deseamos activar. Marcamos **Internet Information Services**, además de las opciones que consideremos necesarias.

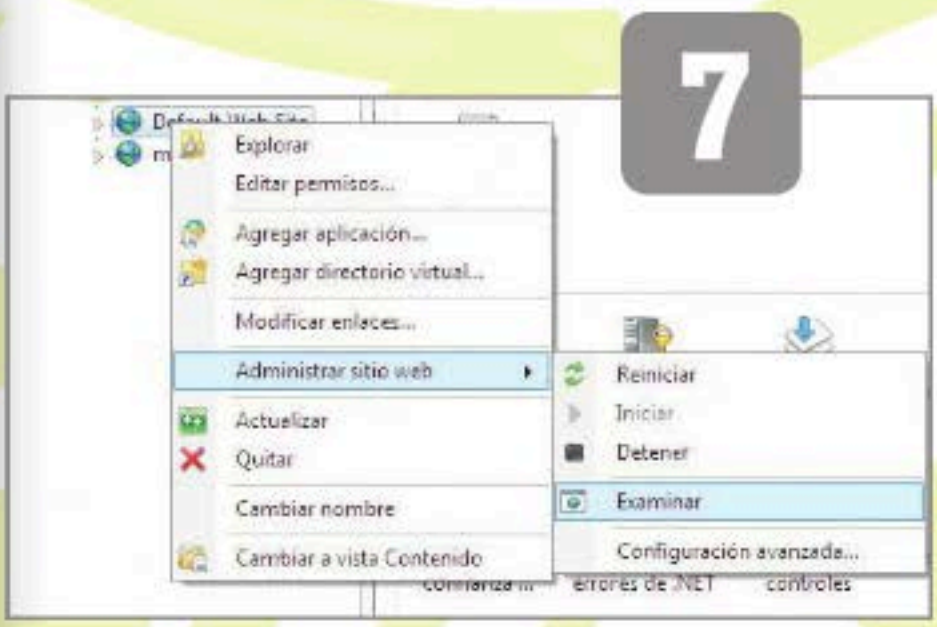
4 Una vez instalada la opción **Internet Information Server**, procedemos a configurar el servicio. Primero seleccionamos **Panel de Control/Sistema y Seguridad**.



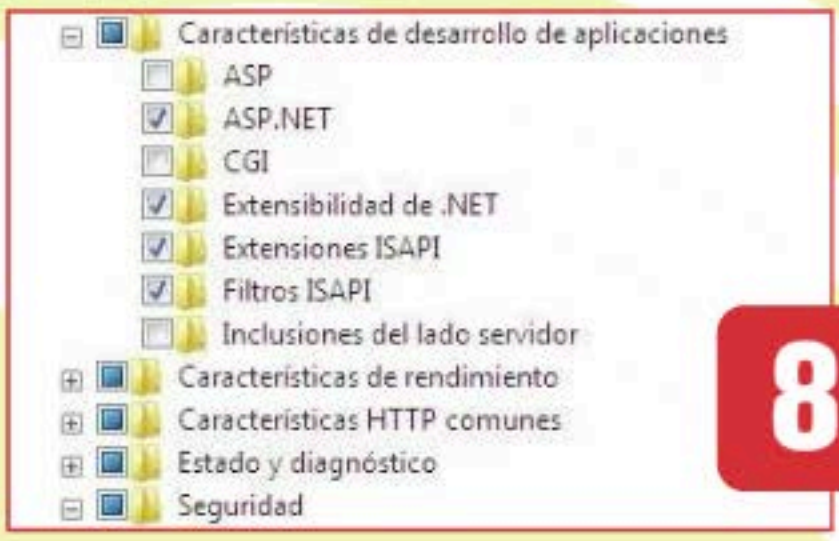
5



6



7



8

5 Una vez llevado a cabo el paso anterior, será necesario que seleccionemos la opción **Herramientas administrativas**, la cual se encuentra al final de la ventana presentada.

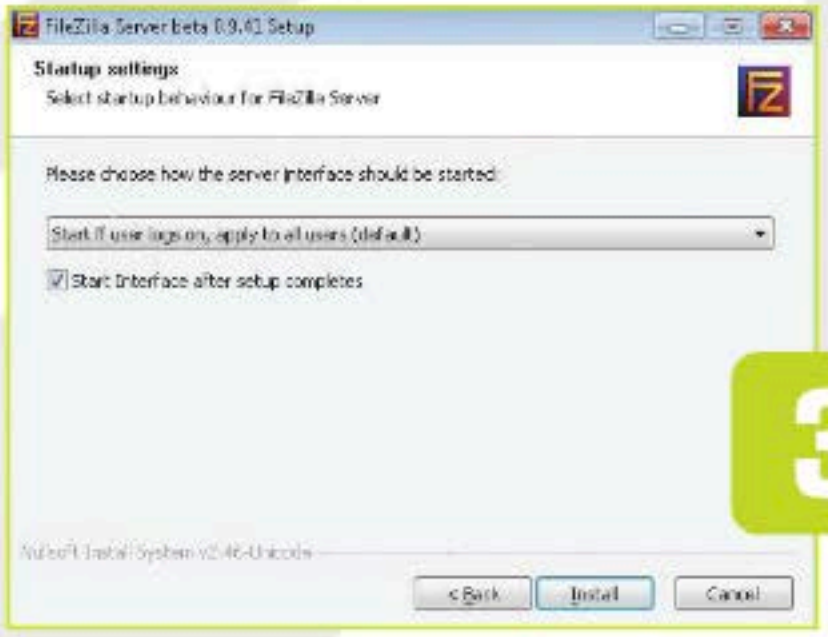
6 Siguiendo con el proceso de instalación, ubicamos el icono llamado **Administrador de Internet Information Services (IIS)**, posteriormente debemos hacer doble clic para que esta herramienta se ejecute.

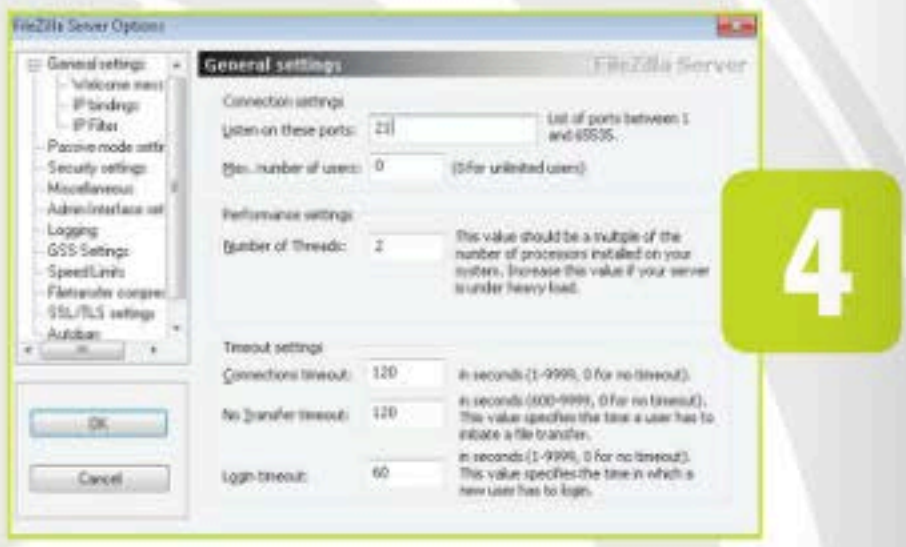
7 Buscamos el sitio web almacenado dentro de la PC local, desplegamos el menú contextual del sitio y seleccionamos la opción **Examinar** para corroborar que la aplicación se encuentra instalada en forma correcta.

8 Debemos analizar a conciencia qué características de **Internet Information Server** vamos a requerir, ya que activar características que no utilicemos, por ejemplo, para el desarrollo de aplicaciones, puede repercutir en forma negativa en el desempeño de nuestro servidor.

Instalar un servidor FTP en Windows

FileZilla es un servidor y cliente FTP libre. Analizaremos en detalle el proceso de instalación y conoceremos sus principales características.



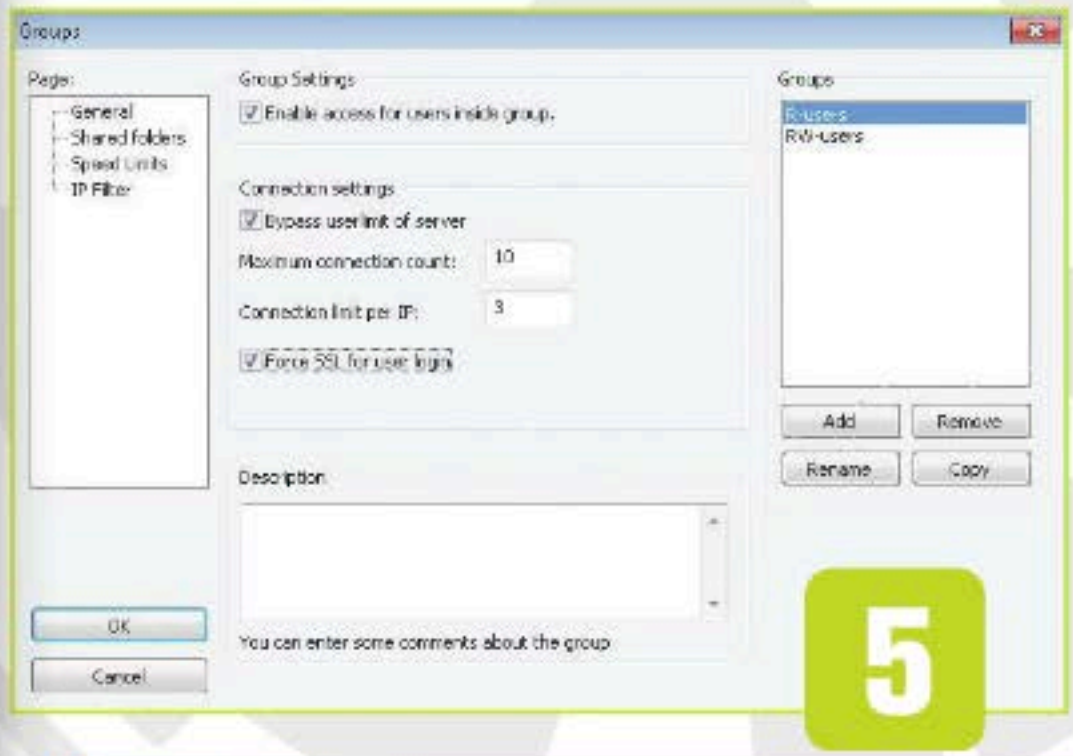


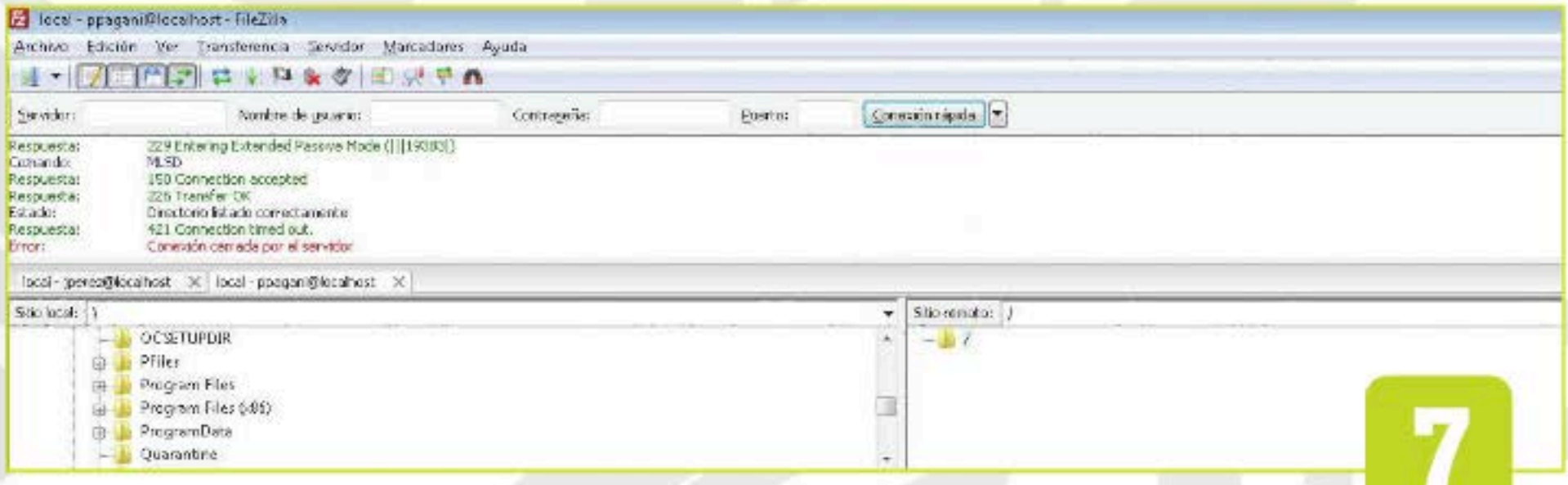
1 Descargamos la última versión disponible del servidor para Windows y el cliente para la plataforma deseada en **http://filezilla-project.org**. Para descargar binarios, iremos a **http://sourceforge.net**. También es posible descargar el código fuente de los checksums de cada archivo, desde la misma dirección web de la aplicación.

2 Debemos seleccionar la instalación del **FileZilla Server**, que se ejecuta como un servicio de Windows, y la consola administrativa. Esta última no es necesaria para que el servicio se ejecute, puede estar instalada en otro equipo y accederla de forma remota a través del puerto 14147.

3 Al instalar la interfaz administrativa, la opción por defecto es que esta se inicie para todos los usuarios y quede disponible minimizada. Desde allí, podemos realizar las configuraciones necesarias para el funcionamiento del servidor.

4 Finalizada la instalación de los componentes, abrimos la configuración y definimos las opciones de seguridad, como la IP del servicio, los logs de seguridad, la funcionalidad SSL/TLS que requiere definir certificados **pub/priv**, entre otras posibilidades.





5 Luego de configurado el servidor, debemos generar los grupos requeridos estableciendo los permisos que se aplican para cada directorio, ya sea de lectura, escritura o borrado. Recomendamos asignar los permisos a los usuarios únicamente mediante grupos para facilitar la administración.

6 Instalamos el software cliente, cuyos componentes disponibles son: iconos adicionales, idiomas complementarios (español y portugués, entre otros), el **shell extension**, que permite integrar el cliente de FileZilla al explorador de Windows para mover archivos desde el Explorador de Windows hacia la aplicación.

7 Finalizada la instalación, iniciamos el cliente y, desde el archivo, abrimos el gestor de sitios. Probamos la instalación realizando una conexión a **localhost** en el servidor utilizando el usuario y la contraseña inicial definidos. El cliente permite conectar varias sesiones usando pestañas.

8 Debemos ser cuidadosos y no compartir unidades del sistema, ya que un usuario, de manera impensada, podría causar problemas en el servidor. Recomendamos bloquear los usuarios que realicen intentos de conexión fallidos, para evitar ataques de fuerza bruta.

Administración de un servidor web

Un mantenimiento y monitoreo de calidad de nuestro servidor web es la actividad más importante para mantener el servicio de nuestro sitio en línea.

En la actualidad, instalar y poner en línea un **servidor web** no posee una complejidad alta asociada. Puede ser un trabajo que consuma solo algunas horas de nuestro tiempo, incluso para personas inexpertas, dada la gran cantidad de información que se encuentra circulando en Internet al respecto. Además, existen empresas y organizaciones que se dedican a alquilar espacio en servidores dedicados, lo que simplifica la necesidad de montar nuestro propio servidor web.

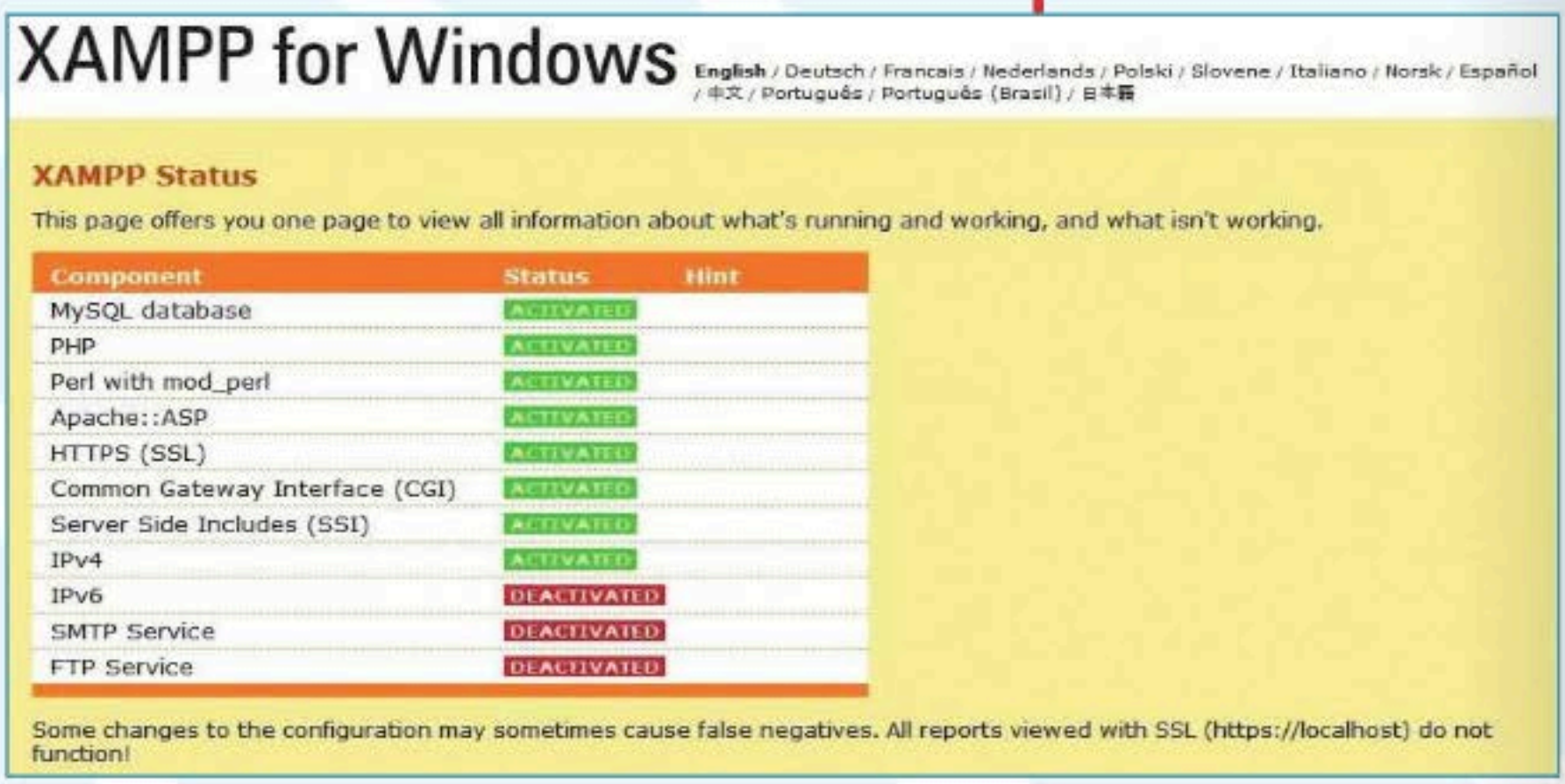
A diferencia de la instalación y puesta en línea, el **mantenimiento y monitoreo** de nuestro servidor es una tarea ardua que implica prevención, configuración, actualización del software relacionado, escalado del hardware y de los medios de conexión a medida que el tráfico aumenta, etc. La estabilidad de un servidor web (entendiendo como *estabilidad* el brindar un servicio continuo en el tiempo, libre de cuelgues y fallos) va a depender de la calidad de las actividades de mantenimiento y monitoreo que se realizan de manera periódica. El éxito de mantener un servidor activo y en línea depende directamente de la previsión que tengamos por adelantado.

Previsión

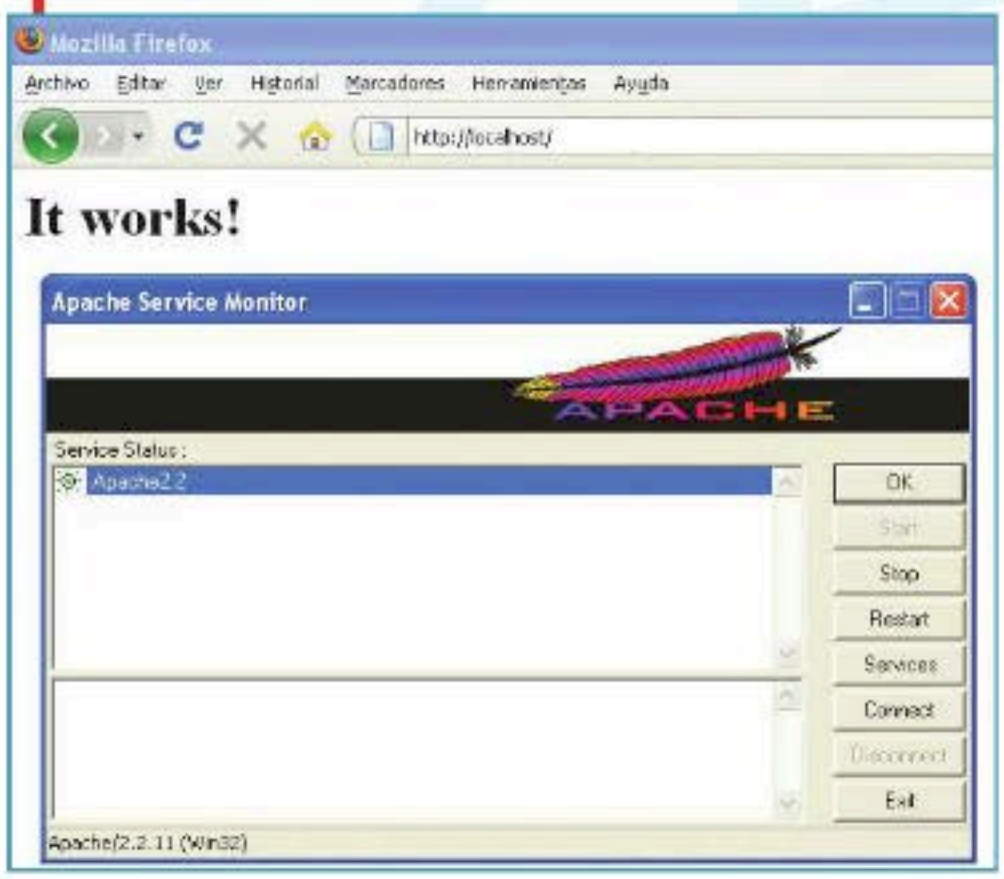
Aunque la falta de previsión es un problema difícil de detectar en etapas tempranas, puede que sea visible luego de meses o, incluso, de años de la instalación y puesta en línea del servidor web, cuando ya sea demasiado tarde para realizar acciones correctivas con el objetivo de restablecer el servicio en el corto plazo. Puede salir a la luz en momentos críticos cuando se produce un tráfico elevado repentino, cuando falla un disco duro o al sufrir un ataque informático (**hackeo**).

Cabe aclarar que la previsión tiene sus límites, y resulta imposible tener en cuenta todos los escenarios posibles; lo saludable es proyectar una evolución del servicio para poder ir escalando el servidor a medida que el tráfico aumenta, estar al tanto de fallos de seguridad y actualizaciones de software para corregir defectos en los sistemas, implementar

La aplicación **XAMPP** es la solución más completa para implementar un servidor web y cuenta con Apache entre sus componentes.



QtdSync es excelente cliente-servidor para poder implementar la utilización de **rsync** en la sincronización de archivos y directorios.



Apache Service Monitor nos permite administrar nuestro servidor web de manera centralizada, gráfica e intuitiva.

políticas de respaldo y restauración de la información y de las aplicaciones, y monitorear el ecosistema en busca de fallos de hardware y ataques, para aplicar acciones correctivas lo antes posible.

Consideraciones

A continuación, analizaremos algunos aspectos importantes para tener en cuenta en la administración de un servidor web.

- **Implementación de políticas de respaldos (backups):** si bien este aspecto resulta un poco obvio, en la práctica no se le suele asignar la importancia que requiere. Si carecemos de un plan de respaldos concreto, es hora de que nos sentemos cuanto antes a delinearlo y a ponerlo en práctica. No hay otra forma de asegurarnos de que la información perteneciente a nuestro servidor web, la que se genera como resultado de la operatoria cotidiana, esté a salvo si se produce una caída del servicio por fallos de hardware o software. Cuando hablamos de respaldos, tenemos dos objetos genéricos que requieren de nuestra atención: los directorios y archivos físicos, y las bases de datos, si es que nuestro servidor web las posee.
Para respaldar los directorios y archivos, podemos utilizar software como **rsync**, **tar**, y similares. Este tipo de software se encarga de replicar y sincronizar cambios en directorios y archivos en una computadora remota. Para respaldar bases de datos, debemos utilizar las herramientas específicas que ofrecen los distintos motores de bases de datos. En ambos casos, lo ideal es que implementemos un proceso automático o semiautomático de backup para estar seguros de que los respaldos se van a realizar independientemente de nuestra disponibilidad para llevarlos a cabo. Características que debe poseer una correcta política de respaldos:

- Debe poseer una periodicidad acorde con el dinamismo con el que se produzcan los cambios en la información por respaldar (diaria, semanal, mensual, etc.). Supongamos que en nuestro servidor web se registran un millón de usuarios por hora, realizar respaldos diarios no va a ser muy útil ya que, si se produce un fallo, habremos perdido demasiada información de manera definitiva; a esto nos referimos con establecer un período de tiempo de respaldo acorde.
- Debe mantener varias versiones del respaldo con distintos períodos de tiempo (un respaldo diario, uno semanal y uno mensual, por ejemplo). De esta manera, podemos satisfacer la necesidad de tener acceso a información que no se encuentra en el último respaldo porque ha sido borrada, pero esta puede ser encontrada en otra versión del respaldo.

A DIFERENCIA DE LA INSTALACIÓN, EL MANTENIMIENTO Y MONITOREO DE UN SERVIDOR ES UNA TAREA ARDUA.

- Debe establecer un período de tiempo límite de antigüedad para guardar respaldos de manera de optimizar el espacio físico de almacenamiento. Por ejemplo, podríamos solamente mantener los respaldos con un año de antigüedad y borrar todos los respaldos que sean más viejos.

- Los respaldos se deben realizar en una ubicación física diferente de donde se encuentra el servidor web, es decir, fuera de la infraestructura principal. Supongamos que el datacenter (centro de datos) donde se encuentra nuestro servidor web se inunda o se incendia, es importante que los respaldos se encuentren a salvo de estos acontecimientos.
- La información de respaldo debe estar igual de segura que la información que se encuentra en el servidor principal.
- Debemos asegurarnos de que el respaldo contenga toda la información crítica necesaria para poder recuperar nuestro servidor de forma íntegra, de manera que el usuario que consume nuestro servicio pueda seguir operando en forma normal.

- **Verificación de los respaldos realizados**: una vez que el proceso de respaldo de información se encuentra operativo, es decir, en funcionamiento, es necesario que validemos, con periodicidad, que los archivos de respaldo generados estén correctamente conformados (no se encuentren corruptos) y se puedan restaurar con éxito. Es muy posible que necesitemos de un entorno de prueba, es decir, una infraestructura similar a la principal en donde restaurar los backups y verificar que nuestro servidor funcione en forma correcta. Frente a una eventualidad real, sabremos cómo proceder gracias a las restauraciones de prueba realizadas.

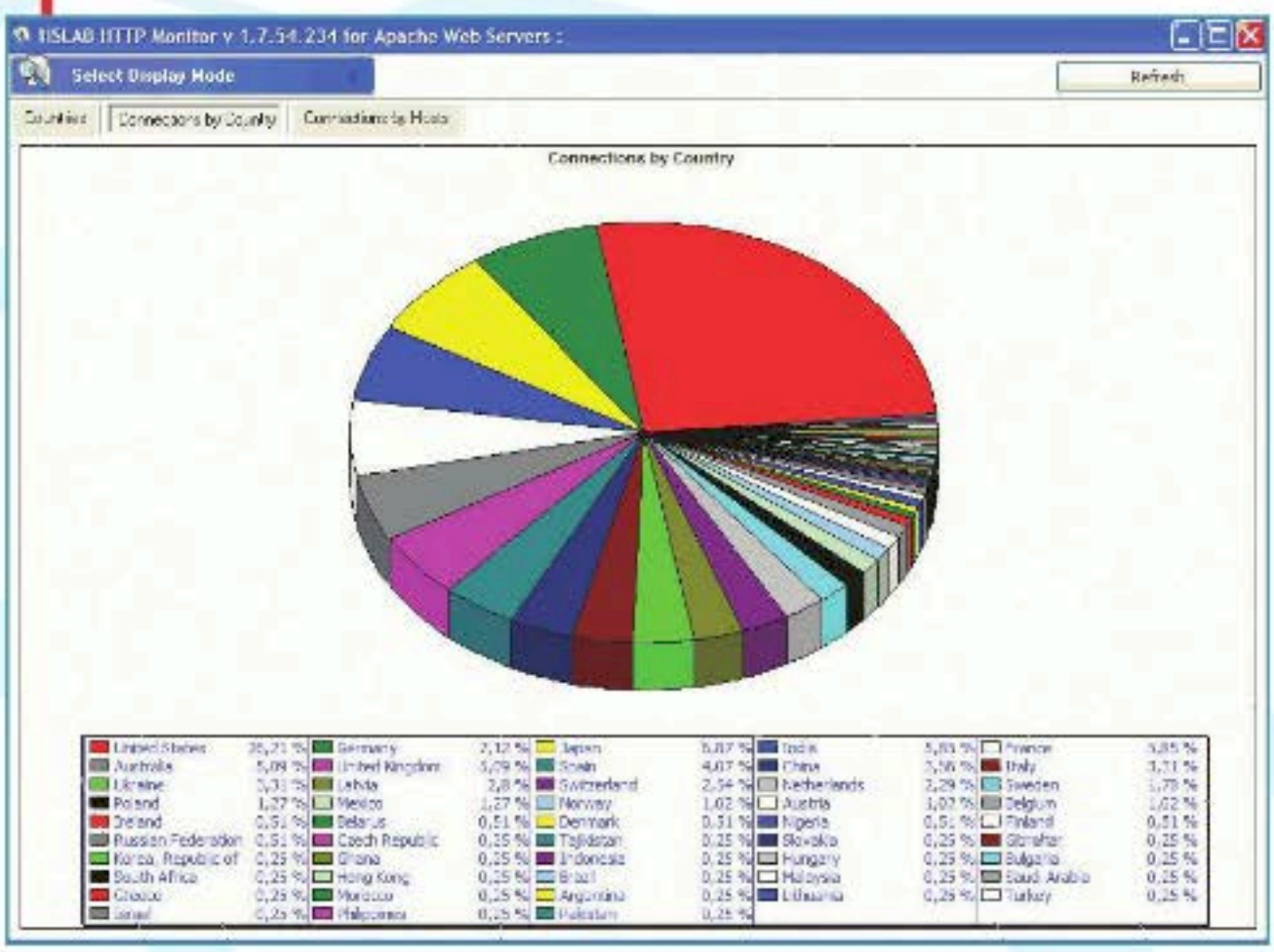
- **Limpieza de archivos de auditoría (logs)**: otro punto para tener en cuenta es la limpieza periódica de archivos logs generados que posean una antigüedad definida con anterioridad. Estos deberían ser limpiados de forma automática cada cierto tiempo para optimizar la utilización del espacio físico. Cuando la cantidad de transacciones que se ejecutan contra el servidor asciende a un número alto, puede que los archivos logs generados terminen por utilizar todo el espacio en disco disponible para nuestro servidor y produzcan una caída del servicio.

- **Monitoreo de la utilización de los recursos**: así como es importante implementar una política de respaldos, monitorear la utilización de los recursos también es clave. Debemos verificar periódicamente la carga de procesos a la que es sometida la CPU, el uso de memoria RAM, el espacio disponible en disco y el ancho de banda de la conexión a Internet que se consume. De esta manera, vamos a poder determinar con antelación cuándo realizar una actualización (upgrade) de hardware. Cuando mencionamos *upgrade de hardware*, nos referimos a instalar más memoria RAM, agregar un disco duro nuevo, contratar un ancho de banda mayor, etc. Si no tomamos este tipo de precauciones, puede que un día el crecimiento en la demanda del servicio termine haciendo colapsar nuestro servidor.

- **Monitoreo de procesos y servicios**: mantener en ejecución software como Apache, Internet Information Server (IIS), MySQL, servicios de correo electrónico (pop, smtp, imap) y demás puede ser crucial para mantener a nuestros usuarios satisfechos. Deberíamos utilizar alguna herramienta de software que automatice el proceso de monitoreo de los servicios para que nos informe en caso de que se produzca algún fallo y nos evite enterarnos de este a través de la queja de un usuario.

- **Endurecer nuestro servidor (hardening)**: el proceso de *hardening* o endurecimiento consiste en mantener nuestro servidor seguro frente a las diferentes amenazas que se encuentran en la Web, tanto las conocidas como las futuras. Por esta razón, resulta importante que conozcamos qué procesos se ejecutan en nuestro servidor, para así detectar procesos anormales, qué puertos tenemos abiertos para interiorizarnos de los peligros que ello conlleva, mantenernos informados sobre los distintos síntomas que producen las principales amenazas, para poder actuar con celeridad frente a un ataque, etc. El mantenernos informados sobre las novedades en materia de seguridad nos va a permitir aplicar nuevas tecnologías a nuestro sistema y reducir los riesgos.

Existen varias herramientas de software para generar y consultar estadísticas de funcionamiento de un servidor web. HSLAB HTTP Monitor es una de ellas.



▶ **Actualizaciones de seguridad**: ningún software se encuentra ciento por ciento libre de fallos y errores que pueden aprovechar las amenazas para vulnerar nuestro servidor. Partiendo de esta regla, debemos estar al tanto sobre las actualizaciones que se liberan con el fin de solucionar estos defectos, para descargarlas e instalarlas lo antes posible. No aplicar las actualizaciones a nuestro sistema es definitivamente un comportamiento negligente, ya que las vulnerabilidades de las distintas aplicaciones se divulgan muy rápido por Internet y, a medida que pasa el tiempo, más atacantes se ponen al tanto de ellas. Los riesgos crecen exponencialmente con el tiempo.

Alternativas

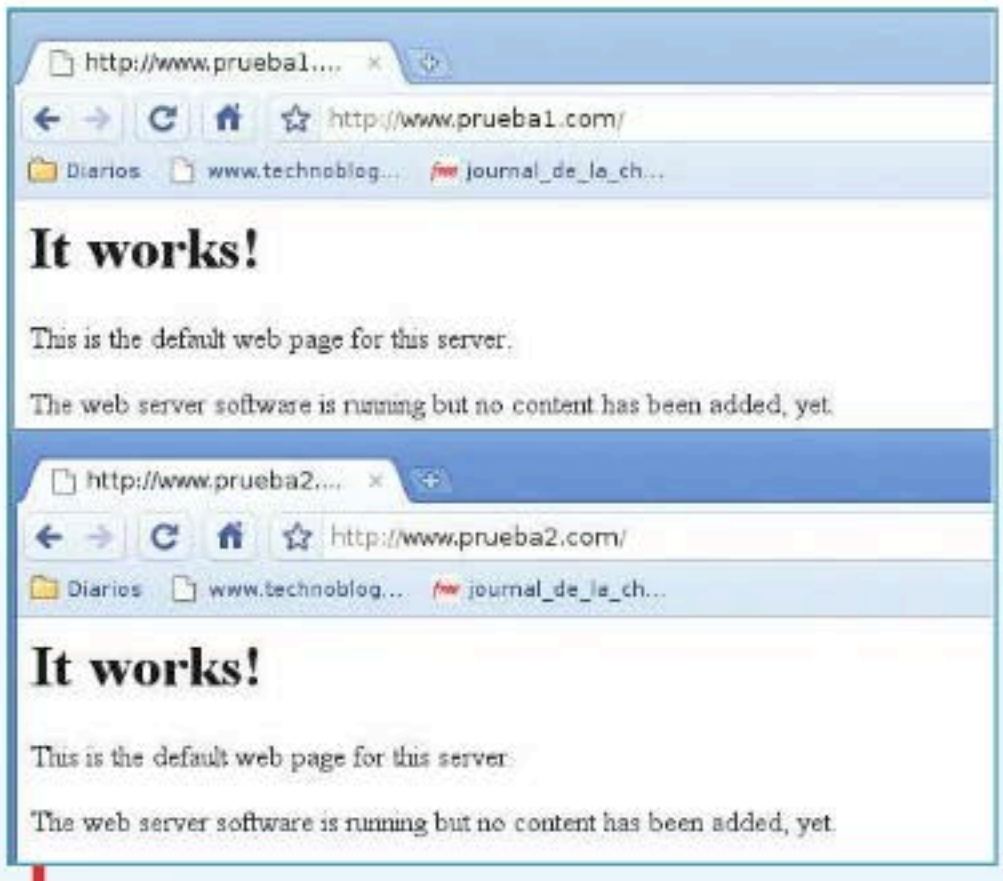
Dos de los servidores web más conocidos en la actualidad, hablando de software, son **Apache Web Server**, de **Apache Foundation**, e **Internet Information Server (IIS)**, de **Microsoft**. Dado que el primero es la solución más popular, a continuación vamos a describir algunas consideraciones para tener en cuenta y configurarlo correctamente.

Partiendo del hecho de que poseemos una capacidad de hardware limitada y que Apache Web Server está diseñado para soportar múltiples peticiones simultáneas de decenas o centenares de usuarios (si nuestro sitio es popular), es fácil darse cuenta de que el hardware existente puede no ser suficiente para que el software de servidor web funcione a su máxima capacidad. Nuestro servidor permanecerá expectante, a la espera de peticiones de conexión realizadas por usuarios y, a medida que vayan llegando, destinará un proceso (o un hilo de ejecución) para atenderlas.

MONITOREAR LA UTILIZACIÓN DE LOS RECURSOS DEL SERVIDOR ES UNA TAREA MUY IMPORTANTE.

Mientras mayor sea la cantidad de peticiones que se formulen de manera simultánea, más procesos/hilos necesitaremos para satisfacer la demanda, y consumiremos más recursos del servidor, sobre todo memoria. Un servidor web consume mucha memoria RAM cuando se encuentra en funcionamiento y a medida que el tráfico aumenta. Este va a ser el recurso clave que debemos aprender a gestionar: la RAM que no se utiliza se desperdicia y provoca que perdamos visitantes o que estos sean atendidos con mayor lentitud. Pero, si no controlamos el consumo de memoria y el servidor la agota, comenzará a hacer swapping a disco, y evitar esto es la regla de oro: si llegamos a esta instancia, se habrá caído el servicio y, posiblemente, deberemos reiniciarlo. Los principales ajustes que necesitamos realizar se encuentran en el archivo `httpd.conf` que se halla en la ruta `C:\Apache\conf`.

▶ `Timeout` establece la cantidad de tiempo medido en segundos que Apache esperará a determinados eventos antes de cerrar o abortar una conexión. En servidores con pocos recursos, 300 segundos puede ser una cantidad bastante elevada. Reducir ese tiempo sustancialmente ayudará a gestionar mejor la memoria del servidor. Valores de 30 o 40 son óptimos en servidores con pocos recursos. Un valor de 10, incluso, podría mejorar el rendimiento en determinados contextos.



Apache Web Server nos permite definir hosts virtuales para alojar más de un sitio en una misma instalación física.

▶ Los tres parámetros, `KeepAlive`, `MaxKeepAliveRequest` y `KeepAliveTimeout`, definen la posibilidad de usar conexiones persistentes y la forma de tratarlas; el número de solicitudes que se permitirán sobre cada conexión (`MaxKeepAliveRequest`), y el tiempo de espera sin solicitudes antes de abortarlas. Tener habilitadas este tipo de conexiones reduce la carga del servidor y los tiempos de respuesta.

▶ Los parámetros restantes, `StartServers`, `MinSpareServers` y `MaxSpareServers`, ajustan la cantidad de procesos Apache que se crearan al inicio, y el mínimo y máximo de estos que mantendremos inactivos a la espera de que lleguen solicitudes. Los procesos inactivos consumirán memoria, pero nos permitirán dar una respuesta más rápida a los usuarios.

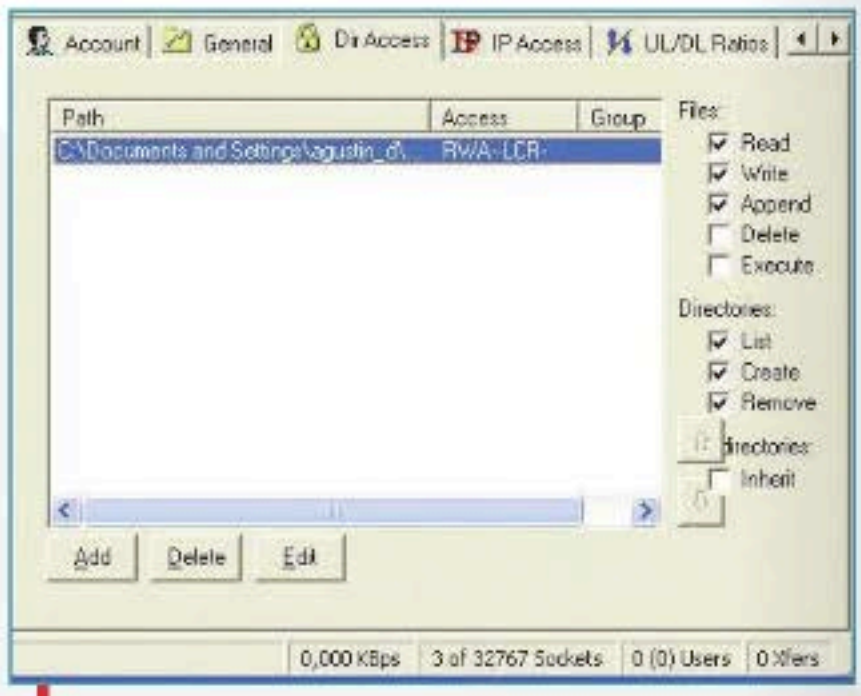
▶ `MaxClients` es, quizás, el parámetro más importante. Define la cantidad máxima de procesos simultáneos que nuestro servidor podrá crear para atender solicitudes. Un número más pequeño del que podemos permitimos desperdiciará memoria y ralentizará las solicitudes de muchos usuarios (que permanecerán a la espera de que uno de estos procesos se libere para atenderlo) mientras que un número demasiado elevado agotará la memoria del servidor y lo obligará a hacer swapping a disco.

▶ Tenemos un parámetro más, `MaxRequestsPerChild`, que define la cantidad de solicitudes que atenderá cada proceso antes de reciclarse. El valor predeterminado es cero, que indica un número ilimitado. Definir un valor elevado pero no ilimitado ayudará a que el servidor libere y limpie su memoria. Un valor entre 500 y 1000 es adecuado para una pequeña PC. ■

Administración de un servidor FTP

La administración de un servidor FTP es una tarea compleja y requiere de previsión y minimización de riesgos para asegurar un servicio estable.

Un **servidor FTP** es un programa de software que se ejecuta sobre una computadora que cumple el rol de servidor y generalmente conectado a Internet (existe la posibilidad de que esté conectado a otros tipos de redes, como redes LAN, MAN, etc.). Su función es la de proveer un servicio de intercambio de archivos. Las aplicaciones de servidor FTP no suelen instalarse en computadoras personales. Casi siempre, un usuario de un servicio FTP se conecta a un servidor FTP que aloja el servicio a través de un software cliente FTP instalado en una computadora cliente. Una aplicación ampliamente extendida del servicio FTP es el alojamiento de páginas web; se utiliza para actualizar y agregar archivos en sitios de Internet y realizar respaldos. En la actualidad, instalar un servidor FTP resulta bastante sencillo dada la gran cantidad de información que se encuentra en Internet al respecto. Tutoriales paso a paso, guías visuales con capturas de pantalla y muchas opciones gratuitas o pagas en lo que se refiere al software. No obstante, la tarea de administrar un servidor FTP no resulta tan sencilla, ya que debemos tener una serie de consideraciones para poder mantener el servicio en línea y sin problemas a lo largo del tiempo.



Los privilegios asignados a los usuarios de un servicio FTP delimitan las acciones que pueden realizar dentro del servidor.

Filezilla Server

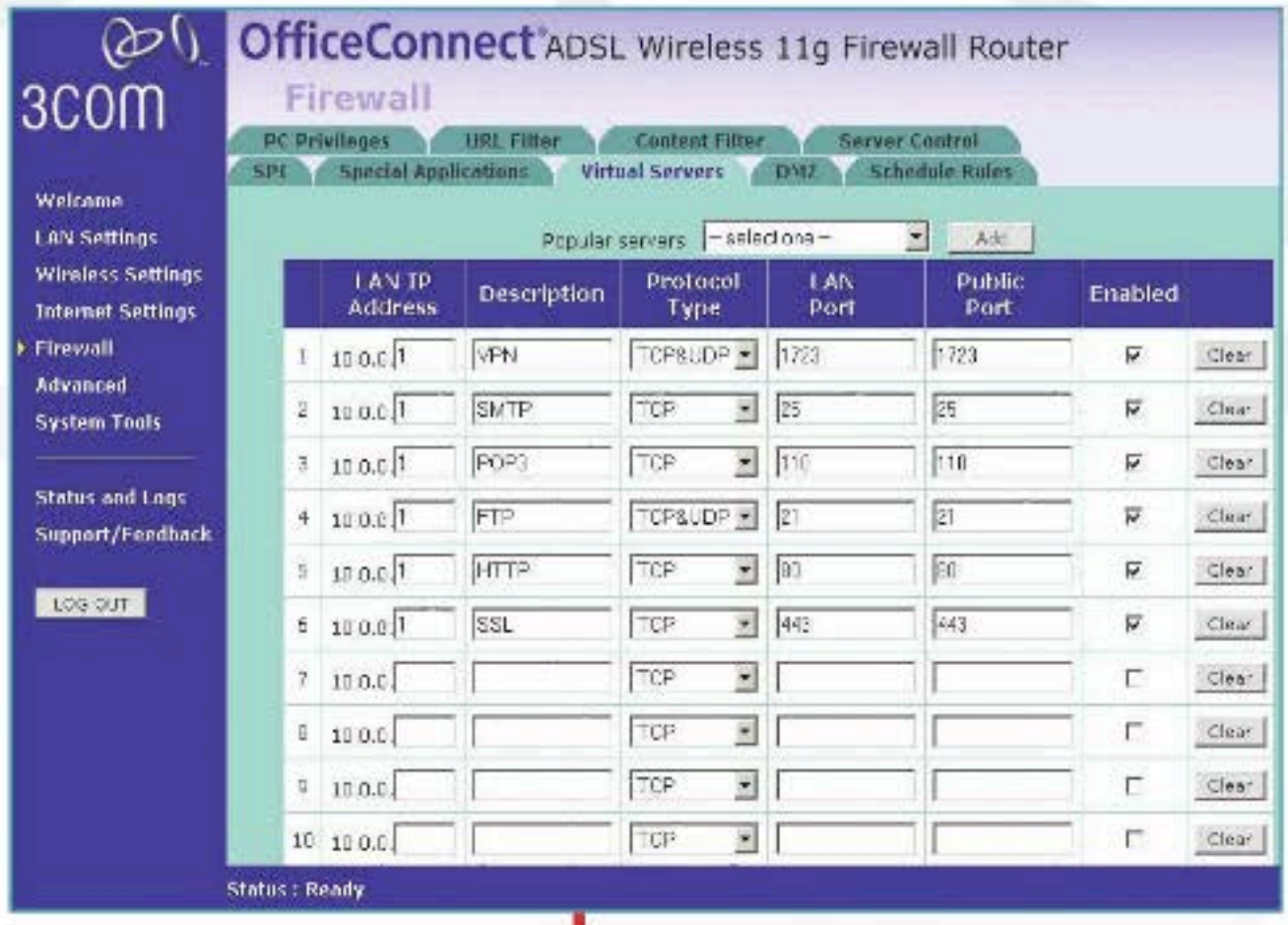
Uno de los servidores FTP más utilizado en la actualidad es Filezilla Server. Se trata de una aplicación libre y multiplataforma que nos ofrece una gran cantidad de características orientadas a simplificar la creación y administración de un servidor FTP. Entre sus funciones encontramos un árbol gráfico de directorios, navegación por carpetas, la posibilidad de crear certificados de seguridad y, además, su proceso de instalación es muy sencillo, gracias al práctico asistente que nos ayudará en cada etapa del proceso de instalación y configuración inicial.

Usuarios

Antes que nada, debemos definir a quién vamos a dirigir el servicio; si a un conjunto de usuarios bien definidos, al público en general, a usuarios ocasionales, etc. Esta elección va a delimitar y a encuadrar la administración futura de usuarios.

- **Acceso anónimo (Anonymous):** los servidores FTP anónimos ofrecen sus servicios de forma libre y gratuita a todos los usuarios (público en general), les permiten acceder a los archivos alojados en ellos sin la necesidad de poseer una cuenta de usuario definida. Es una forma práctica de permitir que todos los usuarios asiduos y potenciales tengan acceso con privilegios definidos sin que para ello el administrador del servidor deba crear una cuenta para cada usuario y asignarle los permisos correspondientes (la administración no requiere de la intervención del administrador). Si un servidor posee un servicio **FTP anonymous**, solo debemos ingresar `anonymous` cuando este solicite el usuario durante el proceso de autenticación. No se requiere de ninguna contraseña

preestablecida, aunque tendremos que ingresar una contraseña durante el logeo, la cual no será sometida a proceso de validación alguno. Por lo general, se suele utilizar la dirección de correo electrónico propia. Solamente con eso se consigue acceso a los archivos del FTP, aunque con menos privilegios que un usuario normal. Casi siempre, solo tendremos privilegios de lectura y copia sobre los archivos que sean públicos, así indicados por el administrador del servidor al que nos conectemos. En la mayoría de los casos, se utiliza un servicio FTP anónimo para almacenar archivos de gran tamaño que no tienen utilidad mientras no sean transferidos a la computadora del usuario, como por ejemplo programas, y de esta forma se reservan los servidores de páginas web (HTTP) para almacenar información textual destinada a ser leída en línea.



Para poder montar un servidor FTP hogareño, es necesario mapear el puerto **TCP 21** en la puerta de enlace a Internet.

para iniciar una sesión FTP, debemos introducir una cuenta de usuario (**login**) y una contraseña (**password**) válidas que nos identifiquen unívocamente. La administración de este tipo de usuarios requiere de la intervención del administrador. Para simplificar la gestión de usuarios, cuando el grupo de estos se encuentra bien delimitado al igual que sus permisos, podemos heredarlos de un dominio dependiendo de la aplicación servidor que utilicemos.

► **Acceso de usuario común:** si vamos a interactuar con usuarios que necesitan contar con privilegios de acceso a cualquier parte del sistema de archivos del servidor FTP, de modificación de archivos existentes y la posibilidad de subir archivos propios, es común que se establezcan usuarios de este tipo. En el servidor, se almacena la información de las distintas cuentas de usuario que pueden acceder a él con sus privilegios correspondientes, de manera que,

► **Acceso de invitado (Guest):** este tercer tipo de usuario es una combinación de los dos anteriores. El objetivo de esta clasificación o tipificación es permitir que cada usuario se conecte al servidor utilizando su cuenta y contraseña, pero evitar de forma general que tengan acceso a partes del sistema y directorios que no necesitan para realizar su labor. De esta manera, estamos creando un entorno restringido genérico que se puede aplicar a un conjunto de cuentas y, así, disminuimos la intervención requerida del administrador. En la práctica, un servidor FTP suele lidiar con los tres tipos de cuentas. Luego de definir el tipo de usuario, necesitamos estimar la cantidad de usuarios que utilizarían el servicio en forma periódica para poder dimensionarlo, así como también un aproximado sobre el volumen de datos que se intercambiará, para poder definir una cantidad de espacio en disco necesario y el ancho de banda que va a consumir el servicio.

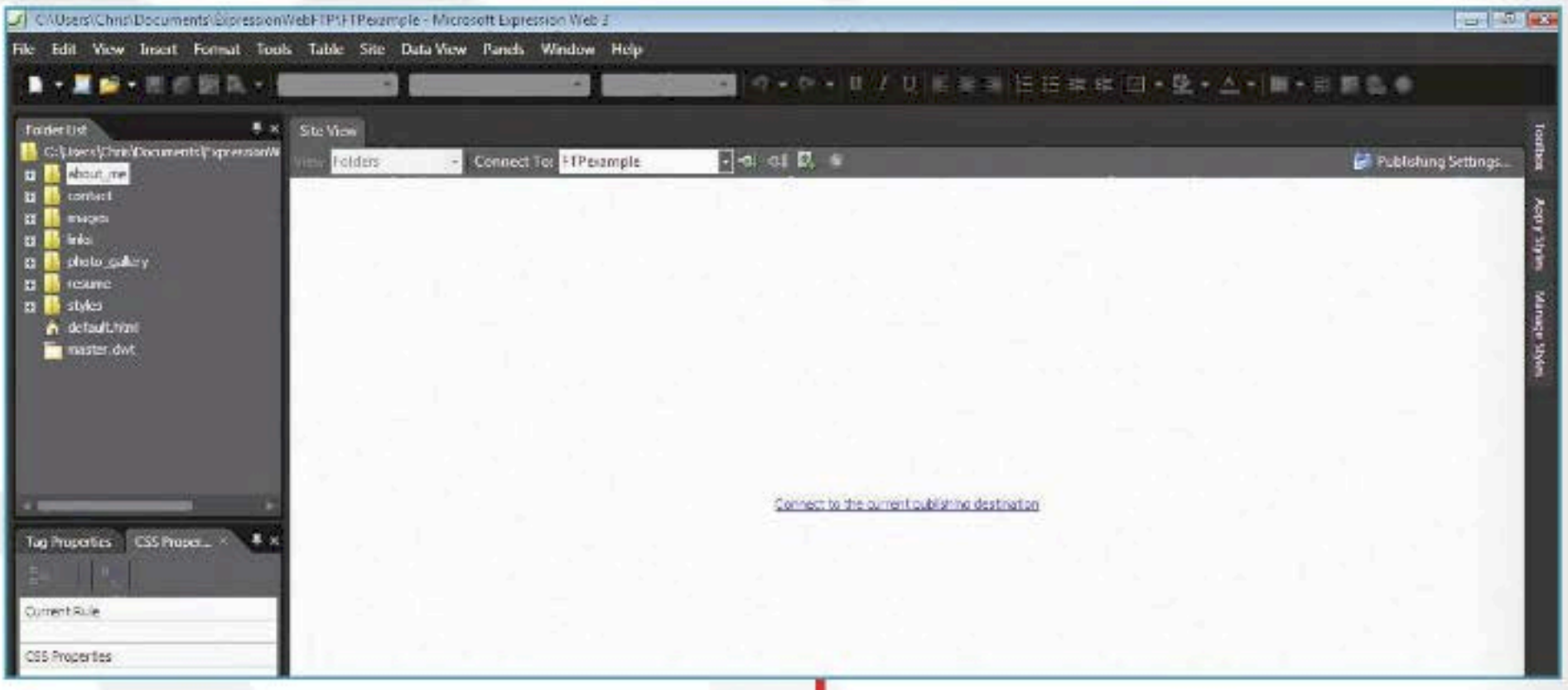
```
Administrador: C:\Windows\system32\cmd.exe - ftp 127.0.0.1
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Administrador>ftp 127.0.0.1
Conectado a 127.0.0.1.
220 Microsoft FTP Service
Usuario (127.0.0.1:(none)): Administrador
331 Password required for Administrador.
Contraseña:
230 User logged in.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-05-10  03:45AM               119275 blue_white2.jpg
08-05-10  10:37PM               292530 Core001.jpg
08-05-10  10:37PM               383101 Core002oclist.jpg
08-05-10  10:37PM                88955 instalacion01.png
08-05-10  10:37PM                88864 instalacion02.png
08-05-10  10:37PM               120699 instalacion03.png
08-05-10  03:30AM                64282 sconfig_01.png
226 Transfer complete.
ftp: 395 bytes recibidos en 0,00segundos 395000,00a KB/s.
ftp> _
```

No es necesario acceder a un servidor FTP a través de una GUI. Podemos utilizar instrucciones básicas a través de una ventana de comandos.

Privilegios

Una vez definidos los usuarios, debemos asignarles los privilegios con los que van a contar. Para ello, es necesario que asociemos los directorios con los



Además de los clientes de escritorio, existen clientes web para poder acceder a los servicios de un servidor FTP.

que va a trabajar el usuario, al usuario propiamente dicho. Y luego definir los permisos con los que contará por cada directorio y para los archivos contenidos dentro de ellos. Los permisos para los directorios pueden ser:

- **Listar:** para poder listar el contenido de los directorios.
- **Crear:** para crear directorios dentro del directorio asociado.
- **Remover:** este permiso se utiliza para realizar la eliminación de directorios dentro del directorio asociado.

Estos permisos se asignan por cada directorio que se asocia al usuario. Los permisos que se pueden asignar para el conjunto de archivos contenidos en un directorio son los siguientes:

- **Lectura:** para descargar archivos existentes dentro del directorio remoto a la computadora local.
- **Escritura:** para poder subir archivos nuevos dentro del directorio.
- **Eliminación:** al igual que en los directorios, se usa para poder eliminar archivos dentro del directorio especificado.

Cabe aclarar que los privilegios se pueden asignar para grupos de usuarios, por lo que la actividad del administrador se simplifica.

Cuota de disco

Mediante el establecimiento de cuotas de disco, podemos limitar el espacio de disco disponible para utilizar por los usuarios, de manera de prevenir un posible colapso del sistema en caso de que estos agoten el total de espacio en disco que posee el servidor. De esta manera, cuando un usuario agota su cuota de disco, debe o bien eliminar archivos existentes para ganar espacio libre o bien solicitarle al administrador que amplíe su cuota.

ADMINISTRAR UN SERVIDOR FTP NO RESULTA TAN SENCILLO, YA QUE DEBEMOS CONSIDERAR MUCHAS OPCIONES.

Ratios UL/DL

Este parámetro se utiliza para definir el ancho de banda de la conexión para la subida de archivos y para la descarga de archivos de forma separada, por cada usuario. De esta manera,

FTPS

También conocido como **FTP sobre SSL**, es un protocolo de transferencia de archivos basado en FTP que le agrega seguridad al encriptar las conexiones de transferencia utilizando el protocolo SSL. De esta manera, se solucionan las falencias o carencias que posee el protocolo FTP original en este aspecto. Existe otra variante del protocolo FTP, denominada **SFTP**, que también aumenta la seguridad encriptando las conexiones de transferencia, pero, a diferencia de FTPS, utiliza el protocolo SSH para realizar dicha tarea.

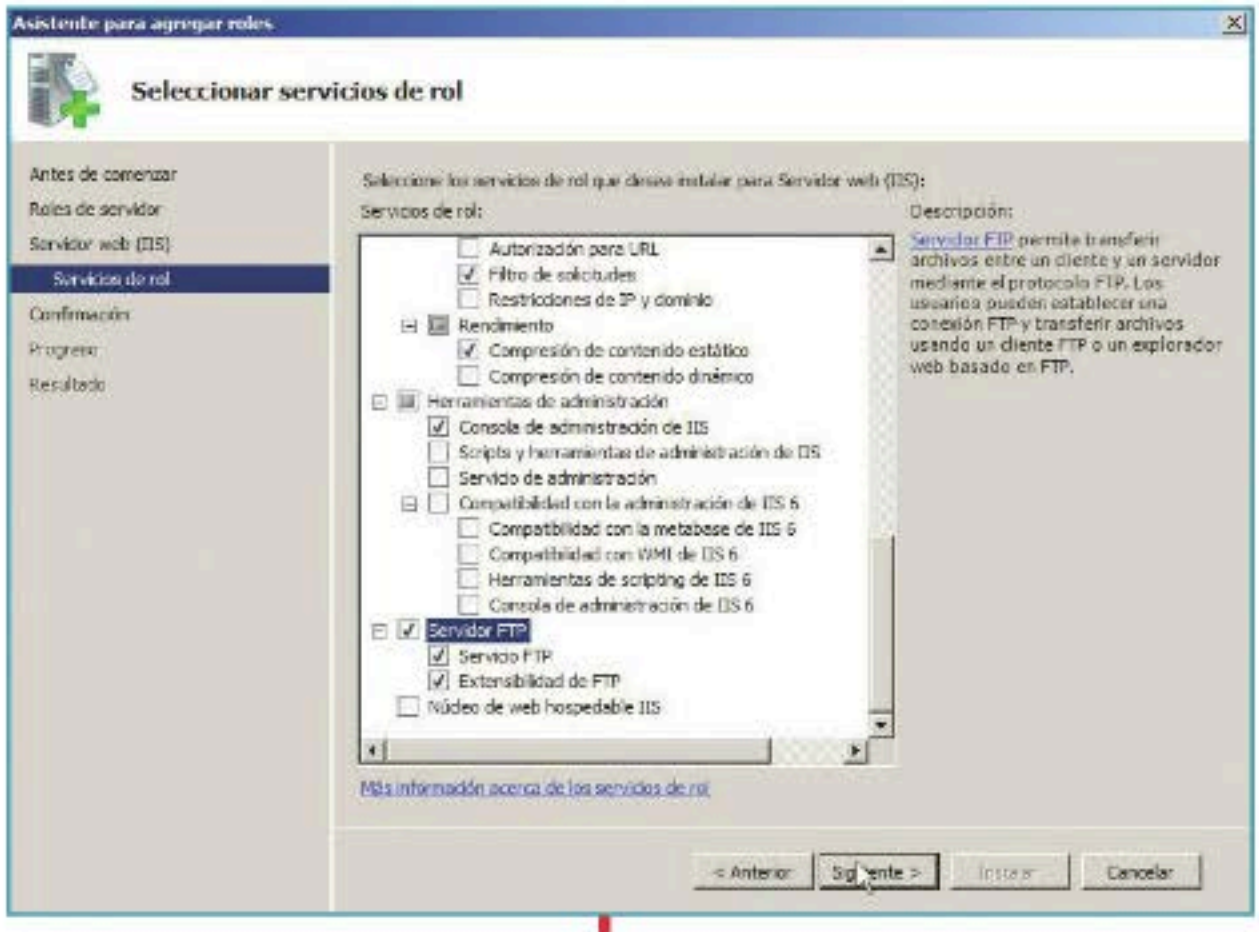
podemos limitar el consumo de ancho de banda a un máximo en horarios pico para asegurar que el servidor no se colapse como ocurriría si el ancho de banda fuera ilimitado para cada usuario. También nos permite asegurar una cierta calidad de conexión para cada usuario. Debemos mencionar que, como desventaja, puede suceder que un usuario posea ancho de banda ocioso, es decir, que no lo esté utilizando en su totalidad, y otro usuario necesite más ancho de banda del que haya sido asignado como máximo.

Modos de Conexión

Debemos especificar con qué modo de conexión va a trabajar nuestro servidor. FTP soporta dos modos de conexión: modo activo y modo pasivo.

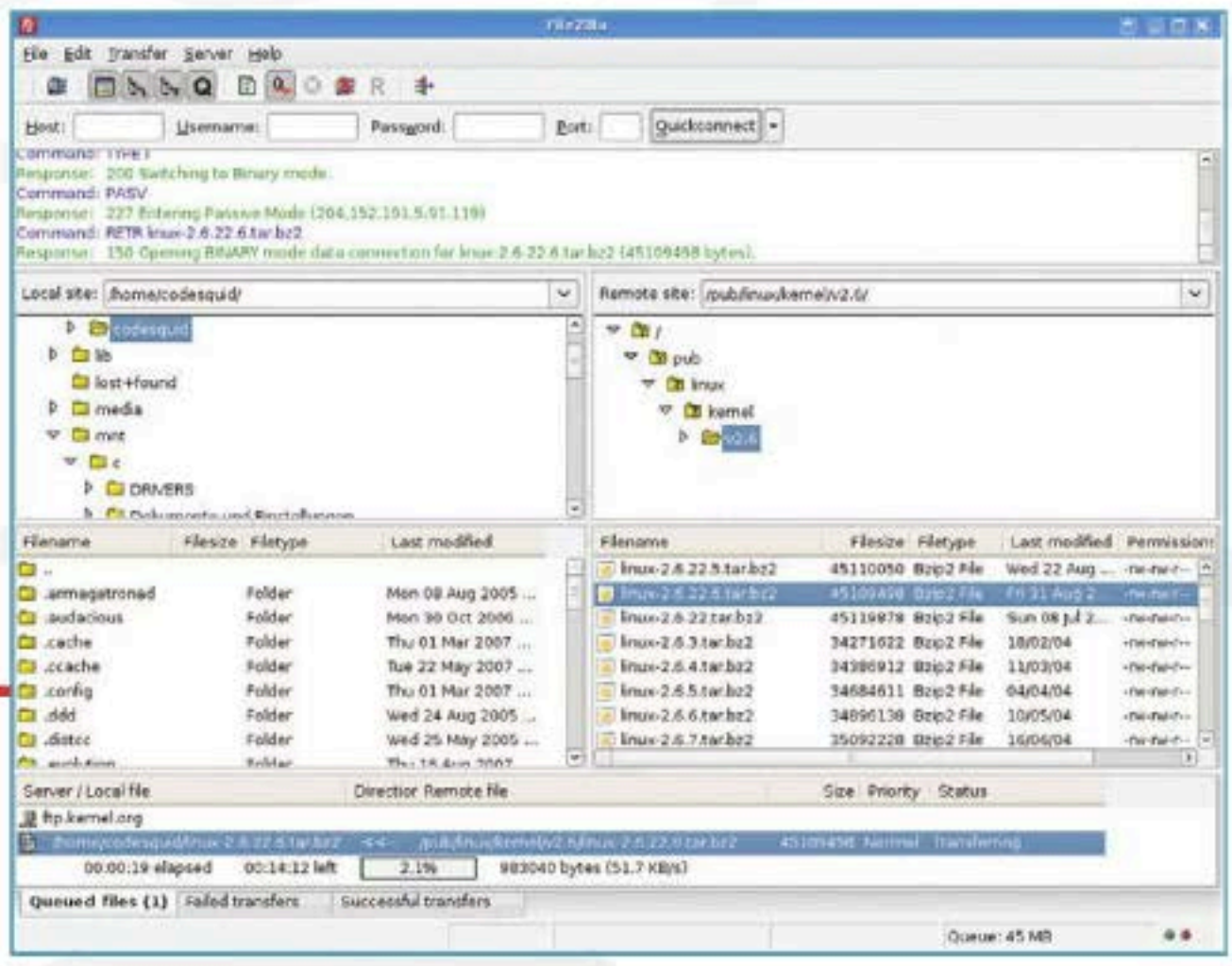
▶ **Modo activo**: en el modo activo, el servidor siempre crea la conexión en su propio puerto TCP 20, mientras que del lado del cliente la conexión se asocia a un puerto aleatorio mayor al puerto TCP 1024. Para ello, el cliente le envía un mensaje al servidor indicándole el número de puerto, de forma tal que el servidor pueda abrir una conexión de datos por donde se transferirán los archivos y los listados, en el puerto especificado. La principal desventaja que presenta este modo es que el cliente debe estar dispuesto a aceptar cualquier conexión entrante a un puerto TCP superior al puerto TCP 1024 con los riesgos que esto trae aparejado si consideramos que estamos conectados a una red insegura, como lo es Internet. Además, por lo general, los firewalls rechazan las conexiones aleatorias. La solución es el modo pasivo.

▶ **Modo pasivo**: cuando un cliente envía una solicitud de conexión, el servidor FTP le indica un puerto TCP específico (mayor al puerto TCP 1023 del servidor) al que debe conectarse. En este modo de conexión, el cliente inicia una conexión hacia el puerto TCP que ha sido indicado por el servidor en forma específica.



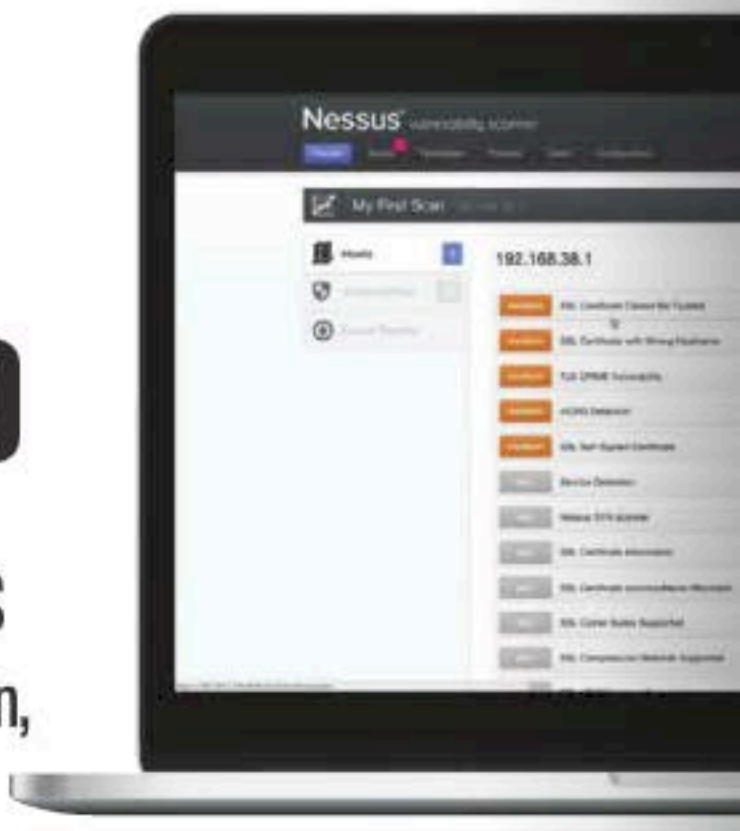
La solución del servidor FTP de Microsoft forma parte de la suite de servicios **Internet Information Services (IIS)**.

Para finalizar, es necesario mencionar algunas soluciones concretas para montar un servidor FTP, en primer lugar el servicio que forma parte de la suite **Internet Information Services (IIS)** y también **FileZilla Server**. Como sabemos, la primera opción se encuentra integrada en los sistemas operativos de Microsoft, y la segunda alternativa se puede descargar en forma sencilla y gratuita, accediendo al sitio web que se encuentra en la dirección **http://filezilla-project.org**. ■



FileZilla es un cliente FTP muy popular en el mercado. Es de código abierto y multiplataforma.

Seguridad en servidores web



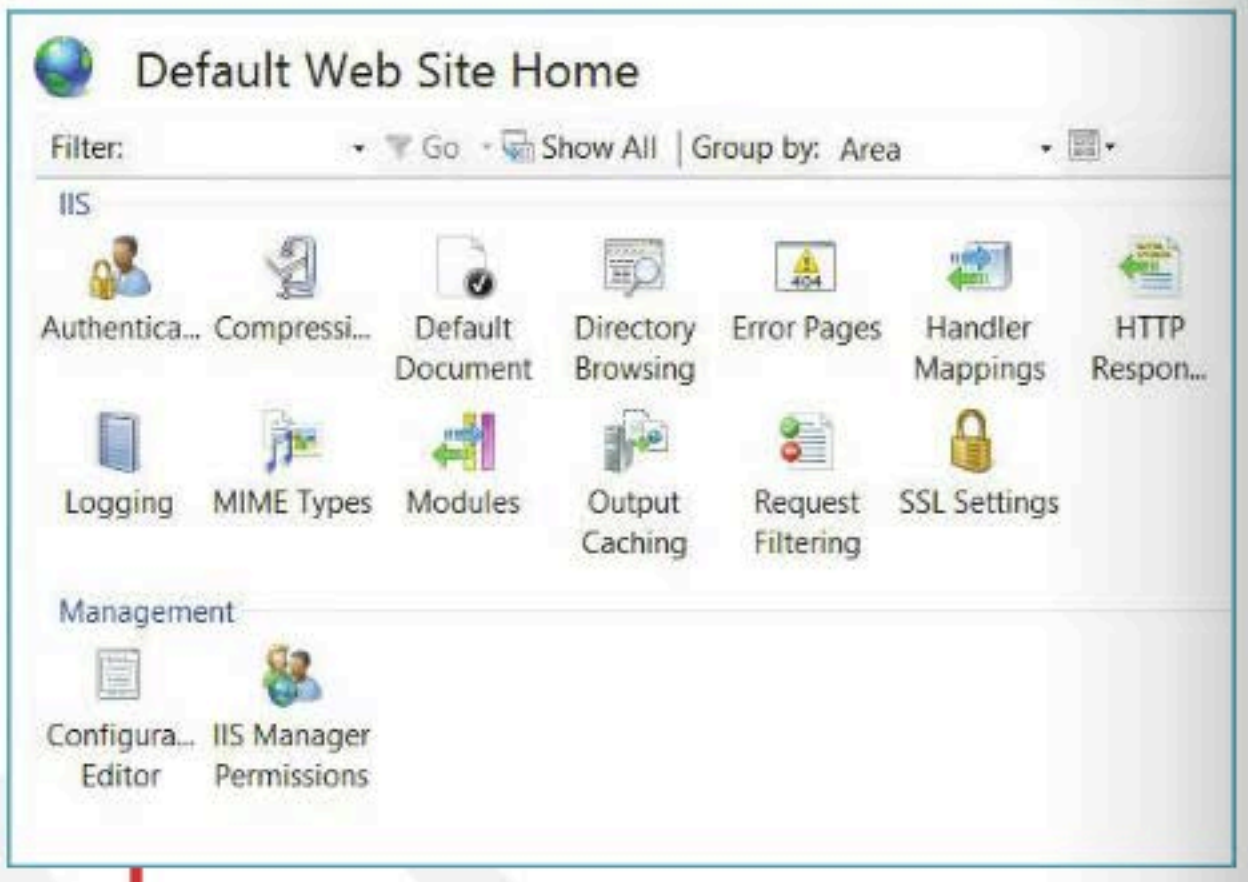
Analizamos la seguridad de los gestores web IIS y Tomcat, los más utilizados en el mercado; también, recomendamos las configuraciones por considerar.

Al evaluar la seguridad de un servidor web, debemos tener en cuenta todos los componentes que forman parte del servicio.

Es necesario considerar: la seguridad física del servidor, la seguridad lógica en el sistema operativo, los componentes de infraestructura que permiten el acceso a este, como por ejemplo: firewall, router y switch. Una vez analizados estos, debemos estimar la seguridad de la aplicación y el gestor web.

En cuanto a la aplicación, hace falta que la codificación sea realizada considerando las cuestiones de seguridad. Para este fin, pueden utilizarse las guías y aplicaciones del proyecto **OWASP** para orientar el desarrollo seguro.

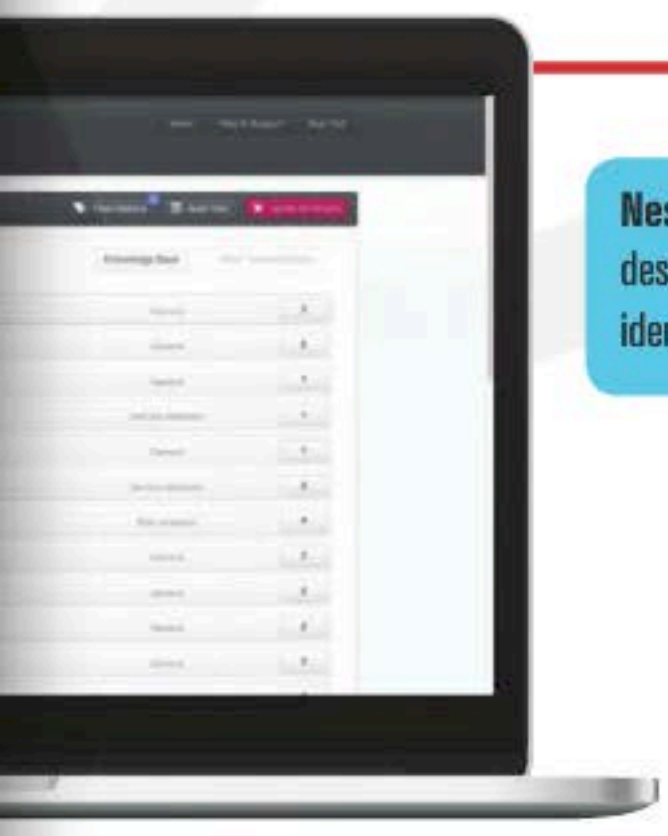
El gestor web consiste en la aplicación que sirve el contenido a los usuarios; los más comunes son **Apache**, **IIS** y **WebSphere Host Publisher**. Debemos definir las configuraciones que permitan una menor



Microsoft Internet Information Services (IIS) Manager concentra todas las configuraciones que son necesarias para administrar las aplicaciones web.

URL Scan

Es una herramienta gratuita ofrecida por Microsoft, que permite restringir los HTTP request que procesará el servidor. Para esto, escucha los requerimientos y realiza filtros basados en reglas. Estos filtros minimizan las posibilidades de ser víctima de diversos ataques como: denegación de servicio, SQL Inyections, Cross-site scripting, **etc. En la instalación por defecto, se generan reglas que pueden ser modificadas en el archivo:** C:\Windows\system32\inetsrv\urlscan\UrlScan.ini. **Cuando se utilicen bases de datos, se debe configurar urlScan para que filtre los strings utilizados en** SQL Inyection.



Nessus, el escáner de vulnerabilidades desarrollado por **Tenable**, permite identificarlas en numerosos productos web.

superficie de contacto, lo que puede ser logrado deshabilitando los servicios no requeridos y restringiendo los permisos de los servicios que sí son requeridos.

Internet Information Services (IIS)

Comenzando con IIS (el gestor web integrado en Windows), es recomendable deshabilitar los servicios del sistema operativo no requeridos, por ejemplo, deberían deshabilitarse: **Alerter**, **Computer Browser**, **DHCP Client**, etc.
El producto Internet Information Server puede instalarse en el directorio por defecto, pero la información para ser publicada (por ejemplo las páginas HTML) deben ubicarse en una partición NTFS que sea distinta a la del sistema operativo.
Dentro del IIS, debemos deshabilitar todos los componentes que no sean utilizados, como por ejemplo: **FTP Server**, **SMTP Server**, **Internet Printing**, etc.
En la configuración de la aplicación, debemos deshabilitar la opción **ParentPaths** ya que podría permitir el acceso no autorizado al directorio inmediato superior. Se debe verificar que no se visualice la dirección IP en el campo **Content-Location** de la información devuelta vía HTTP (puerto 80) del servidor. Para ello debemos:

- Realizar telnet al puerto 80 del servidor.
- Ejecutar el comando **GET/HTTP/1.0<CR><CR>**.
- Verificar que no se visualice la dirección IP en el campo **Content-Location**.

Para deshabilitarlo, usamos este comando: **cscript.exe c:\inetpub\adminscripts\adsutil.vbs set w3svc/UseHostName True** y reiniciamos el servicio **w3svc**.
Sobre los directorios virtuales que contenga la web App, debemos deshabilitar los siguientes permisos: **Script sourceaccess**, **Write** y **Directorybrowsing**.

Tomcat

En cuanto a **Apache Tomcat**, en el momento de instalarlo se debe utilizar la última versión estable disponible.
En servidores productivos, no se deben instalar los componentes **native**, **documentation**, **examples**, **webapps**, etc. Se debe crear un usuario y grupo no privilegiado que iniciará el servicio (por ejemplo: **tomctusr** y **tomctgrp**).
Modificar el **ownership** de **USUARIO_HOME** al usuario y grupo definido.
Modificar los archivos en **USUARIO_HOME/conf** a **readonly**. Verificar que **tomctusr** tenga permisos **rw** en **/tmp** y solo **wx** (300) en **USUARIO_HOME/logs**.
Asignar los permisos mínimos necesarios (**rx**) sobre la aplicación por utilizar.
Se debe analizar si la aplicación requiere permisos de escritura sobre el sistema de archivos, lo cual no es aconsejable.
Por otra parte, se recomienda intercambiar los archivos de configuración de Tomcat, para simplificar y reducir los componentes innecesariamente habilitados por defecto. Para esto, renombrar (mv): **USUARIO_HOME/conf/server.xml** a **USUARIO_HOME/conf/server-original.xml** y **USUARIO_HOME/conf/server-minimal.xml** a **USUARIO_HOME/conf/server.xml**. Se debe evitar el uso de los puertos inferiores a **1024** ya que, en Unix, estos requieren ejecutarse con privilegios de **root**. Por defecto, Tomcat se instala configurado para escuchar en el puerto **8080**. La modificación del puerto de escucha se hace desde el archivo de configuración **USUARIO_HOME/conf/server.xml**. Luego de modificar el puerto de escucha, es necesario reiniciar el servicio. El puerto **8005** nos permite detener el servicio Tomcat y sus aplicaciones. El script para detener el servicio provisto por Tomcat realiza una conexión a este puerto, enviando un **string** que indica el **shutdown**.
Es recomendable modificar este string en el archivo **USUARIO_HOME/conf/server.xml** para evitar apagados no autorizados.
Asimismo, es necesario que todos los puertos (salvo los usados por el servicio, ej.: **8080**, **8443**) estén filtrados por un firewall.
Tomcat posee conectores definidos en el archivo **server.xml**. Algunos están habilitados, y otros deshabilitados.

El proyecto **Open Source OWASP Live CD**, basado en Linux, permite realizar verificaciones de seguridad sobre aplicaciones web.

Se recomienda deshabilitar todos los conectores no utilizados por la aplicación. Se requiere por lo menos un conector que esté escuchando algún tipo de tráfico, de lo contrario, Tomcat no podrá servir ninguna petición. Si no deshabilitamos los conectores no usados, estamos permitiendo que Tomcat reciba peticiones en ese puerto. Es recomendable que se use el conector HTTPS para toda la información sensible generada. Para ocultar la versión del servidor, se debe reempaquetar **USUARIO_HOME/server/lib/usuario.jar** con una versión actualizada del archivo **ServerInfo.properties**; desempaquetar usuario.jar y modificar la **default error page** para evitar que muestre el **stacktrace**, y volver a empaquetar. ■

Seguridad en servidores FTP

Examinaremos en profundidad el antiguo protocolo de intercambio de archivos, también revisaremos sus características de seguridad.

Al planear los elementos de seguridad que integran un servidor FTP, debemos tener en cuenta todos los componentes que hacen a la seguridad del servicio. Debemos considerar el entorno físico del servidor, la seguridad en el sistema operativo y los componentes de infraestructura que permiten el acceso, como por ejemplo: firewalls, routers y switches. Una vez asegurados estos componentes, debemos verificar la configuración del servicio FTP. Los servidores FTP implementan el **RFC 114** de IETF, cuyo origen data del año 1971 con sucesivas modificaciones hasta la actualidad. Originalmente, el protocolo FTP no consideraba mayores cuestiones de seguridad, pero, en sus actualizaciones, se agregó soporte para **TLS/SSL** (**FTPS**), autenticación fuerte, integridad y confidencialidad, entre otros.

La configuración por defecto del IIS 6.0 permite el acceso anónimo al servidor FTP.

Seguridad del sistema

En cuanto a la seguridad del sistema operativo en general, debemos definir las configuraciones que permitan una menor superficie de contacto, lo que puede ser logrado deshabilitando los servicios no requeridos y restringiendo los permisos de los servicios que sí son requeridos. Uno de los servidores FTP más comúnmente utilizados en Microsoft Windows es el **Internet Information Server (IIS),** que se encuentra integrado a este. Solo debería habilitarse el servicio de FTP cuando fuera necesario y no pudiera ser reemplazado con algún otro servicio seguro como SSH, SFTP o similar. Es recomendable habilitar la encriptación SSL/TLS siempre que sea posible, lo que generará que el servicio se comporte como FTPS, garantizando la confidencialidad e integridad de la información transferida.

Carpetas y permisos

La carpeta que se publique en el FTP deberá ser distinta de las que pertenecen al sistema operativo y, en lo posible, diferentes de cualquier sitio web publicado. Nunca se debería compartir un disco entero del sistema operativo; es recomendable utilizar una unidad completamente separada del SO. Los permisos para lectura/publicación de archivos deben otorgarse a los usuarios mediante utilización de grupos, de esa manera se facilita la administración y es posible definir roles (RBAC). Para el caso del acceso de lectura a recursos FTP públicos, no es necesario autenticar a los usuarios, por lo que es recomendable que el acceso sea con el usuario anónimo IUSR_Servername que posee ciertas restricciones. Cabe mencionar que, independientemente de los permisos

File eXchange Protocol

FXP es un método para transferir datos desde un servidor FTP a otro sin rutear datos al cliente. El FTP convencional establece una conexión solo entre servidor y cliente. En la sesión FXP, el cliente inicia la conexión e indica que la transferencia de datos debe realizarse a otro destino. La ventaja de utilizar FXP radica en que el cliente no consume ancho de banda. Habilitar FXP es riesgoso, ya que vuelve vulnerable el ataque **FTP bounce** que permite, entre otros, realizar escaneo de puertos de manera silenciosa utilizando un servidor víctima. La popular herramienta **Nmap** permite realizar este ataque.

que se asignen en el servicio FTP, se deben configurar las ACL necesarias en el sistema de archivos (NTFS) para cada uno de los grupos/usuarios. Para el acceso a recursos FTP públicos, no es obligatorio restringir el acceso a nivel de direcciones IP. Pero sí puede realizarse en intranets para restringir los segmentos de red no autorizados o en caso de detectarse ataques de fuerza bruta.

Restricciones

Es posible configurar las restricciones por equipos (**Single computer**) o grupos de equipos (**Group of computers**). Se debe implementar una política de auditoría a fin de registrar los logins exitosos y fallidos. Es recomendable definir un tamaño máximo de log para no comprometer el espacio en disco.

Si el servidor cuenta con distintas placas de red, es aconsejable definir una por la cual se aceptarán las conexiones FTP.

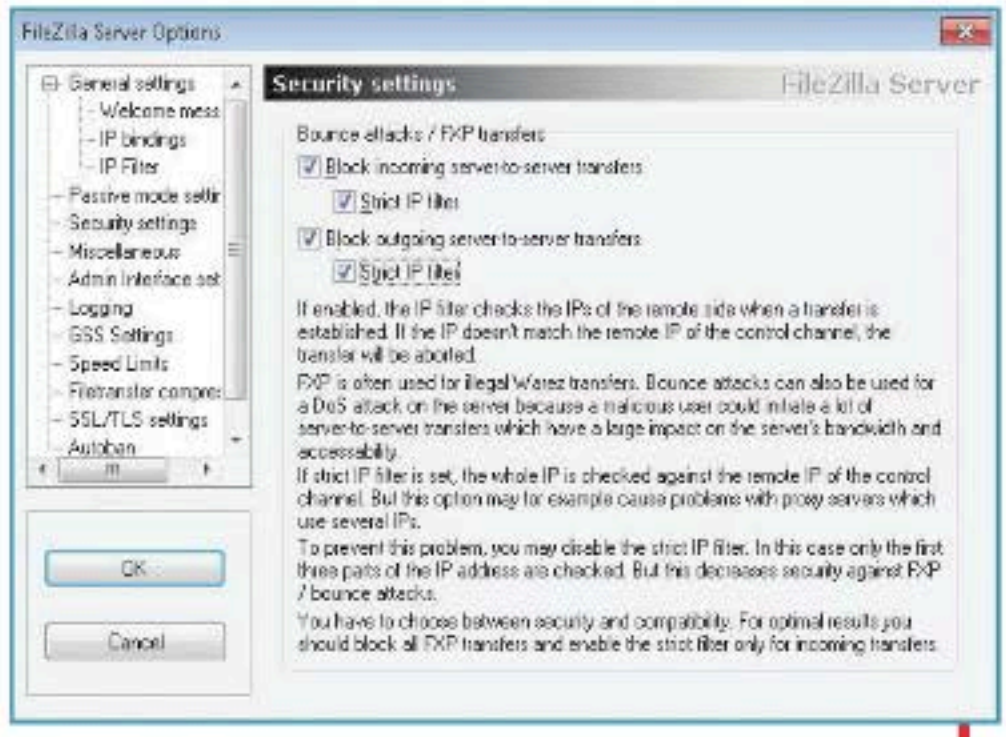
Muchos servidores permiten bloquear las transferencias FXP, potencialmente peligrosas, pero hacerlo puede implicar que los usuarios que utilicen un proxy no puedan descargar contenido.

El mecanismo de control que impide las transferencias **FXP** consiste en controlar que la IP que inicia la transferencia sea la misma que descarga el contenido. Algunos servidores FTP, como por ejemplo **FileZilla** y **Gene6**, soportan **MODE Z**, que permite comprimir el tráfico **onthefly** para ahorrar ancho de banda. Esta funcionalidad podría limitar la capacidad de un proxy o IPS de detectar malware. Solo una vez descomprimido en el servidor/cliente, podrá escanearse en busca de virus.

Ya sea que esta funcionalidad esté o no habilitada, es recomendable instalar un antivirus completo y mantenerlo actualizado en el servidor FTP para evitar la distribución de contenido malicioso, aun cuando el servidor no sea Windows.

Sea cual fuere el servidor FTP que instalemos, es necesario revisar las configuraciones iniciales. Por ejemplo, IIS permite por defecto el acceso anónimo al servidor. Si mantenemos acceso anónimo al servidor, es recomendable crear un directorio para solo escritura y un directorio para solo lectura. De esa forma, garantizamos que los archivos presentes en la carpeta de solo lectura no sean alterados con fines maléficos.

Una solución intermedia para poder enviar información confidencial por la red usando un FTP convencional es encriptar la información antes de transferirla. Para esto, pueden utilizarse diversa técnicas o herramientas. Otra opción es forzar el tráfico FTP a través de una VPN, la cual se encarga de otorgar altos niveles de confidencialidad e integridad.



Protección contra Bounce Attack presente en FileZilla Server. Previene el escaneo de puertos y ataques DoS.

Otras opciones

Si decidimos remover el acceso anónimo e implementar autenticación vía cuentas NT, debemos considerar que el servicio FTP envía la información en texto plano, incluso las contraseñas, por lo que, si alguien hace un **sniff**, podrá ganar acceso con facilidad. Por último, es necesario que mantengamos el sistema operativo y el servidor FTP actualizados, sobre todo si va a estar expuesto en Internet. Algunas de los exploits comúnmente utilizados contra servidores FTP son: **Bounce Attack**, **buffer overflow**, **DoS Attack** y envío de comandos inválidos o cadenas largas. Estos **exploits** permiten que un atacante local o remoto ejecute código arbitrario, gane acceso privilegiado y acceda a rutas del sistema no publicadas en el FTP, entre otros problemas. ■

HTTP y HTTPS

HTTP y HTTPS son protocolos de red para transferencia de hipertexto. La diferencia radica en que el segundo cifra los datos antes de transferir.

El protocolo **HTTP** (*Hypertext Transfer Protocol*) o protocolo de transferencia de hipertexto es un protocolo de red que especifica la sintaxis y la semántica de lenguaje que deben utilizar los componentes de software de una arquitectura web (clientes, servidores, etc.) durante una comunicación entre ellos.
Cada transacción dentro de la **World Wide Web** utiliza este protocolo.
Es un protocolo orientado a transacciones y sigue el esquema solicitud-respuesta entre un cliente y un servidor.
El cliente (navegador web por ejemplo) que realiza una solicitud se denomina **agente del usuario**. La información que intercambian un cliente y un servidor se denomina recurso y se identifica mediante un localizador uniforme de recursos (**URL** o *Uniform Resource Locator*).
Los recursos pueden ser archivos, el resultado de la ejecución de un proceso o la respuesta de una consulta realizada a una base de datos, entre otros.

Un ataque man-in-the-middle podría apoderarse de nuestra cuenta de home banking al interceptar nuestro usuario y contraseña si la conexión no está cifrada.

HTTPS SE REFIERE AL USO DEL PROTOCOLO HTTP COMÚN SOBRE UNA CONEXIÓN CIFRADA.

Características

Es un protocolo sin estado, es decir, no almacena ningún tipo de información sobre las conexiones anteriores.
Las aplicaciones web, con frecuencia, necesitan mantener un estado; para ello se hace uso de las cookies. Estas se pueden utilizar para el rastreo de usuarios y se guardan en los clientes por tiempo indeterminado.
Este protocolo es inseguro, ya que carece de un mecanismo de cifrado para los datos que viajan por una conexión entre un cliente y un servidor. Si un atacante tiene acceso a la conexión, puede apropiarse de la información que se intercambia y hacer uso de ella, ya que es perfectamente legible. Es vulnerable a ataques como **man-in-the-middle** o **eavesdropping**.

HTTPS

Para suplir esta falencia, surge **HTTPS** (*Hypertext Transfer Protocol Secure*) o protocolo de transferencia de hipertexto seguro. Esta variante del protocolo utiliza cifrado basado en SSL / TLS. De esta manera, si un atacante accede a una comunicación y se apropia de datos intercambiados, no va a poder hacer uso de ellos ya que no van a ser legibles.
Para el protocolo HTTP, las URLs comienzan con **http://** y utilizan por defecto el puerto **80**. Las URLs del protocolo **HTTPS** comienzan con **https://** y utilizan el puerto **443** por defecto.

Comparación

HTTP opera en la capa de aplicación del modelo, pero el protocolo de seguridad opera en una subcapa más baja, cifrando un mensaje HTTP previo a la transmisión y descifrando un mensaje una vez recibido. Si realizamos un análisis en profundidad, HTTPS no es un protocolo separado, pero se refiere al uso del protocolo HTTP común sobre una conexión cifrada **SSL** (*Secure Sockets Layer*) o capa de socket segura o una conexión con **TLS** (*Transport Layer Security*) o seguridad de la capa de transporte. ■

PRÓXIMA ENTREGA

19

SERVIDORES DE MAIL

En el próximo número veremos las características de los servidores de correo tanto en sistemas Windows como en GNU/Linux. Analizaremos también las opciones de seguridad y las mejores aplicaciones.

Técnico en
REDES &SEGURIDAD

PROFESORES EN LÍNEA
profesor@redusers.com

SERVICIOS PARA LECTORES
usershop@redusers.com

SOBRE LA COLECCIÓN

CURSO VISUAL Y PRÁCTICO QUE APORTA LOS SABERES NECESARIOS PARA FORMAR TÉCNICOS EXPERTOS EN REDES Y SEGURIDAD. INCLUYE UNA GRAN CANTIDAD DE RECURSOS DIDÁCTICOS COMO INFOGRAFÍAS, GUÍAS VISUALES Y PROCEDIMIENTOS REALIZADOS PASO A PASO.

Con la mejor metodología para llevar adelante el montaje y mantenimiento de las redes informáticas y con los aspectos clave para brindarles la protección necesaria, esta obra es ideal para aquellos aficionados que deseen profundizar sus conocimientos y para quienes quieran profesionalizar su actividad.

REDES &SEGURIDAD 00018
9 789871 857784

CONTENIDO DE LA OBRA

1 Introducción a las redes informáticas
2 Tipos de redes y topologías
3 Dispositivos de red
4 Instalación de redes cableadas
5 Puesta en marcha de una red cableada
6 Configuración de redes cableadas
7 Instalación de redes inalámbricas
8 Configuración de redes inalámbricas
9 Seguridad en redes cableadas e inalámbricas
10 Configuración avanzada de routers
11 Recursos compartidos y dispositivos multimedia
12 Seguridad física de la red
13 Impresoras de red
14 Hardware de servidores
15 Administración de Windows Server
16 Administración de sistemas Linux
17 Administración y asistencia remota
18 SERVIDORES WEB Y FTP
19 Servidores de mail
20 Servidores de archivos e impresión
21 Servidores adicionales
22 VLAN, VPN y trabajo remoto
23 Telefonía IP
24 Cámaras IP