IES N1 Gijón

Principado de Asturias
Consejería de Educación

UNIDAD DE TRABAJO 3

Servicios FTP

Celia C. Pellico Llera

Cofinanciado por la Unión Europea

Contenidos

1. Introducción
2. Servicio de transferencia de ficheros (FTP)
3. Funcionamiento
4. Cliente FTP
5. Servidores FTP
6. Protocolo FTP
7. Tipos de acceso al servidor FTP
8. Tipos de conexión: de control y de datos
9. Modos de conexión del cliente
10. Tipos de transferencia de archivos
11. Servicios de transferencia de archivos en modo texto
12. Servicios de transferencia de archivos en modo gráfico
13. Navegadores
14. Asegurar el servicio de transferencia de archivos
15. El servicio de transferencia de archivos en el despliegue de una aplicación Web
16. Establecer permisos en FTP

1. Introducción

Una de las principales ventajas que ofrecen las redes basadas en TCP/IP es la posibilidad de transferir información entre los equipos que forman parte de ellas.

Existen múltiples servicios de red que permiten a los usuarios enviar ficheros de un sistemas a otro como servicios de correo electrónico (adjuntando archivos en los mensajes), servicios web (a través de hipervínculos que apuntan a ficheros localizados en un servidor web), servicios P2P (Peer to Peer) de descarga, etc. Algunos de estos servicios no están diseñados exclusivamente para transferir ficheros y no suelen estar optimizados para ello.



- FTP es un servicio muy antiguo (existe desde 1971).
- Es fácil de configurar y mantener.
- Existen múltiples implementaciones.
- No es un protocolo cifrado.

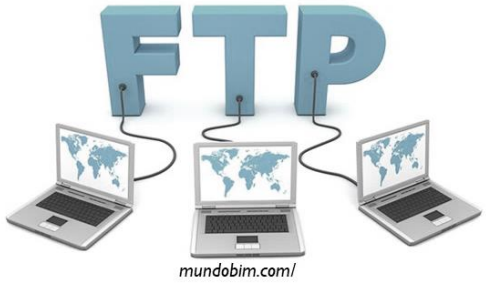
2. Servicio de transferencia de ficheros (FTP)

Normalmente para subir archivos en Internet, ya sean de texto, imágenes, vídeo... hay 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 que utiliza el protocolo FTP. **FTP** son las siglas de **F**ile **T**ransfer **P**rotocol (Protocolo de Transferencia de Archivos). Es un protocolo de red para la transferencia de archivos en redes basadas en 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.

Desde un equipo cliente podemos conectarnos al servidor para descargar archivos de él o enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.



Características

FPT permite a los usuarios:

–Acceder a sistemas remotos y listar directorios y ficheros.

–Transferir ficheros desde o hacia el sistema remoto, es decir, subir (*upload*) o bajar (*download*) ficheros.

–Realizar acciones adicionales en el sistema remoto como renombrar, borrar, crear archivos o carpetas, cambiar permisos, descomprimir…

Seguridad

FTP **no es un protocolo seguro**. Fue diseñado para **ofrecer velocidad pero no seguridad**, a pesar de ser uno de los métodos más utilizados.

Se utilizan mecanismos de autenticación de usuarios para determinar los privilegios de acceso y transferencia en el servidor, pero:

–No se usan mecanismos para garantizar que los equipos involucrados en la transferencia son quienes dicen ser. Es vulnerable a ataques de suplantación de identidad (*spoofing*).

–Todo el intercambio de información, incluyendo el usuario, password y la transferencia de cualquier archivo, se realiza en "texto plano", sin ningún tipo de cifrado. Es vulnerable a ataques de análisis de tráfico de red (*sniffing*).

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.

Pese a todo, constituye una de las herramientas más útiles para el intercambio de ficheros entre diferentes ordenadores y es la forma habitual de publicación en Internet, ya que a menudo los usuarios necesitan transferir archivos entre máquinas sin importar la seguridad.

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.

3. Funcionamiento

El protocolo FTP emplea una arquitectura **cliente/servidor**, siendo el cliente ftp quien solicita la transferencia de archivos y el servidor ftp quien los ofrece.

Pertenece a la familia de protocolos de red TCP (capa de *Aplicación*) 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.

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 uno. Estos puertos TCP suelen ser el **21** para el **control de la conexión** y otro a determinar según el modo de conexión que 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.



Componentes

El servicio FTP está formado por:

- **Clientes FTP**. Acceden al sistema de ficheros del equipo donde están instalados y establecen conexiones con los servidores FTP para subir o descargar archivos.
- **Servidores FTP**. Acceden al sistema de ficheros del sistema donde están instalados, manejan las conexiones con los clientes y en función de los privilegios definidos permiten la descarga y/o la subida de ficheros.
- **Protocolo FTP**. Conjunto de normas y reglas en base a las cuales "dialogan" los clientes y los servidores FTP. Usa TCP como protocolo de transporte.



4. Cliente FTP

Es un programa que se instala en el ordenador y utiliza el protocolo FTP para conectarse a un servidor y transferir archivos (descarga o subida).

El cliente FTP necesita conocer:

- El nombre del archivo.
- El ordenador en el que reside.
- El ordenador al que se quiere transmitir el archivo.
- La carpeta en la que se encuentra.

Existen muchos programas clientes FTP básicos tanto en Linux como en Windows que trabajan en modo comando como, por ejemplo, **ftp** o **lftp**. Y también existen programas clientes FTP con opciones añadidas e interfaz gráfica como **gftp** o **FileZilla**.

Aunque algunos navegadores tienen integrado el servicio FTP, es más confiable a la hora de conectarse a un servidor FTP no anónimo utilizar un programa cliente.

¿Qué cliente elegir?

- **¿Conocemos la consola ftp?**

Si nos manejamos con soltura en la consola ftp podemos optar por un cliente ftp de comandos que permita utilizar la tecla "tabulador" después de escribir unos caracteres para complementar los nombres de archivos.

- **¿Cuál es el uso que necesitamos?**

Si solamente queremos visitar un servidor ftp y descargar un archivo sin tener que instalar nuevos programas, en este caso podemos utilizar el cliente ftp del navegador, ftp://

- **¿Queremos reanudar la conexión en caso de corte en la misma?**

En este caso mejor un cliente tipo gráfico.

- **¿Deseamos facilidad de manejo?**

Un cliente terminal de comandos suele ser menos interactivo que uno gráfico, debemos saber manejarnos con comandos en la consola ftp, mientras que en un cliente gráfico se puede manejar a través de clics del ratón. Los clientes gráficos suelen ser más amigables y por lo tanto más utilizados.

- **¿Qué tipo de conexión queremos establecer? ¿Cifrada? ¿No cifrada?**

Dependiendo del tipo de conexión debemos emplear un cliente u otro, ya que no todos los clientes ftp permiten conexiones cifradas.

- **¿Deseamos recordar conexiones (sitios)?**

No todos los clientes ftp lo permiten.

5. Servidores FTP

Un servidor FTP es un programa que atiende y procesa las conexiones de los clientes FTP y que puede acceder al sistema de ficheros del equipo donde está instalado permitiendo la subida y descarga de archivos.



Ofrecen múltiples opciones de configuración para establecer privilegios de los usuarios, limitaciones de subida y descarga, tiempos de conexión y espera, etc.

Uno de los usos más comunes de los servidores FTP suele ser el **alojamiento de páginas web**. Los clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes al servidor a través de un programa FTP.

Los programas servidores FTP no suelen encontrarse en los ordenadores personales sino en máquinas que cumplen esta función. El usuario o usuaria normalmente utilizará FTP para conectarse remotamente a uno y así intercambiar información con él.

Existen muchos programas de servidor ftp, tanto libres como propietarios para diferentes sistemas operativos.

Software de servidor FTP para Unix/Linux

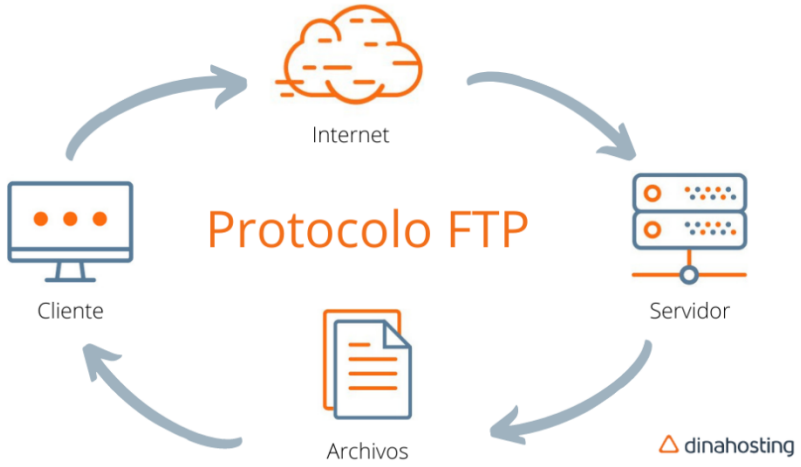
- Vsftpd
- Proftpd
- Wu-ftpd

Software para sistemas Windows

- IIS (servidor FTP incluido en IIS)
- FileZilla Server
- Provide Server

6. Protocolo FTP

Es el conjunto de normas que regulan el diálogo entre cliente y servidor del servicio de transferencia de ficheros.



- La comunicación se basa en el envío de mensajes de texto con órdenes y respuestas.
- El protocolo de transporte es TCP.
- Los comandos son cadenas de caracteres terminadas en CR+LF.
- Las respuestas del servidor son la consecuencia de la acción ejecutada al recibir un comando. Se componen de un código de 3 dígitos que indica el resultado de la orden y un texto descriptivo. Los 3 dígitos indican:
  - Primer dígito: Si la orden ha tenido éxito o no
  - Segundo dígito: Indica a qué se refiere la respuesta
  - Tercer dígito: Información más específica sobre el segundo dígito.

7. Tipos de acceso al servidor FTP

Los servidores FTP permiten, dependiendo de cómo se configuren, dos tipos básicos de acceso desde los clientes: **anónimo** y **autorizado**.

Acceso anónimo

El cliente FTP sin autenticar y sin ningún privilegio, se conecta al servidor con un usuario especial anónimo. Como nombres para este usuario se emplean de forma estándar **anonymous** y/o **ftp**. Por etiqueta, la contraseña solía ser la dirección de correo del usuario. Ahora no suele necesitarse contraseña.



Lo habitual (aunque el administrador del servidor FTP puede decidir otra cosa) es que el cliente anónimo solo pueda descargar archivos y su acceso esté limitado a un directorio público del servidor.

Acceso autorizado

El cliente que solicita la conexión tiene una cuenta con ciertos privilegios en el servidor y, tras autenticarse, se le deja entrar a su directorio predeterminado.
Desde esa área puede descargar ficheros y, si la política del sistema se lo permite, también escribir, aunque normalmente se limita su espacio mediante una cuota de disco.
Puede estar autorizado a recorrer parte del árbol de directorios y listar su contenido o escribir en ellos, dependiendo del tipo de privilegios que posea.

- El cliente ftp se identifica con unas credenciales conocidas por el servidor, ya sean usuarios locales del sistema, del directorio (LDAP, AD, Etc.), de una BD de credenciales, etc.
- Las personas usuarias autorizadas pueden tener acceso de escritura en carpetas personales y/o en carpetas de *upload*.
- Las personas usuarias pueden estar "enjauladas" de forma que no pueden recorrer toda el sistema de ficheros del host servidor, sino sólo la estructura visible por el servidor FTP.
- Según los privilegios configurados, las personas usuarias podrán descargar, subir, borrar, y pueden tener una cuota de uso de disco o de velocidad de transferencia.

Usuarios virtuales

Disponible en algunos servidores, como el servidor **ProFTPD**, es una tercera posibilidad interesante. 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 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** y servidores **RADIUS**.



Dependiendo del servidor ftp, habrá unos métodos de autenticación de usuarios u otros.

Por ejemplo en ***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***.
  Más información: 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 ***AuthPAMAuthorative*** a '**on**'.
  Más información: Directiva AuthPAMAuthorative.
- Bases de datos **SQL**, tales como MySQL o Postgre. Para ello emplea el módulo **mod_sql**.
  Más información: 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.

https://www.redeszone.net/tutoriales/servidores/proftpd/

8. Tipos de conexión: de control y de datos

Como ya se comentó anteriormente, los servidores y los clientes mantienen conexiones TCP independientes para el control y la transferencia de datos.

Conexión de control (TCP 21)

Inicialmente el cliente establece una conexión con el servidor para "*dialogar*" con él. Le envía comandos de descarga (***get***), subida (***put***), listado (***ls***), etc. Y recibe respuestas del servidor que le informan de cómo atiende las peticiones.

Esta conexión permanece activa hasta que el usuario cierra la sesión o hasta que el servidor la finalice porque caduca el tiempo de espera a causa de la inactividad (***timeout***).

Los servidores pueden atender a varias conexiones de control simultáneamente, tantas como se configuren en el servidor para evitar su sobrecarga.

Conexión de datos (TCP 20)

Cuando el cliente solicita una transferencia de información se crea una nueva conexión (conexión de datos) que se cierra al finalizar la transmisión.

Asociadas a una conexión de control pueden existir múltiples conexiones de datos simultáneas, tantas como se configure en el servidor para evitar su sobrecarga.

Las conexiones de datos, originalmente realizadas por el puerto 20 del servidor, son realizadas últimamente por puertos no privilegiados (***puerto > 1023***).

La **conexión de control** (**puerto 21**) se encarga de iniciar la comunicación entre el cliente y el servidor y mantenerla activa.

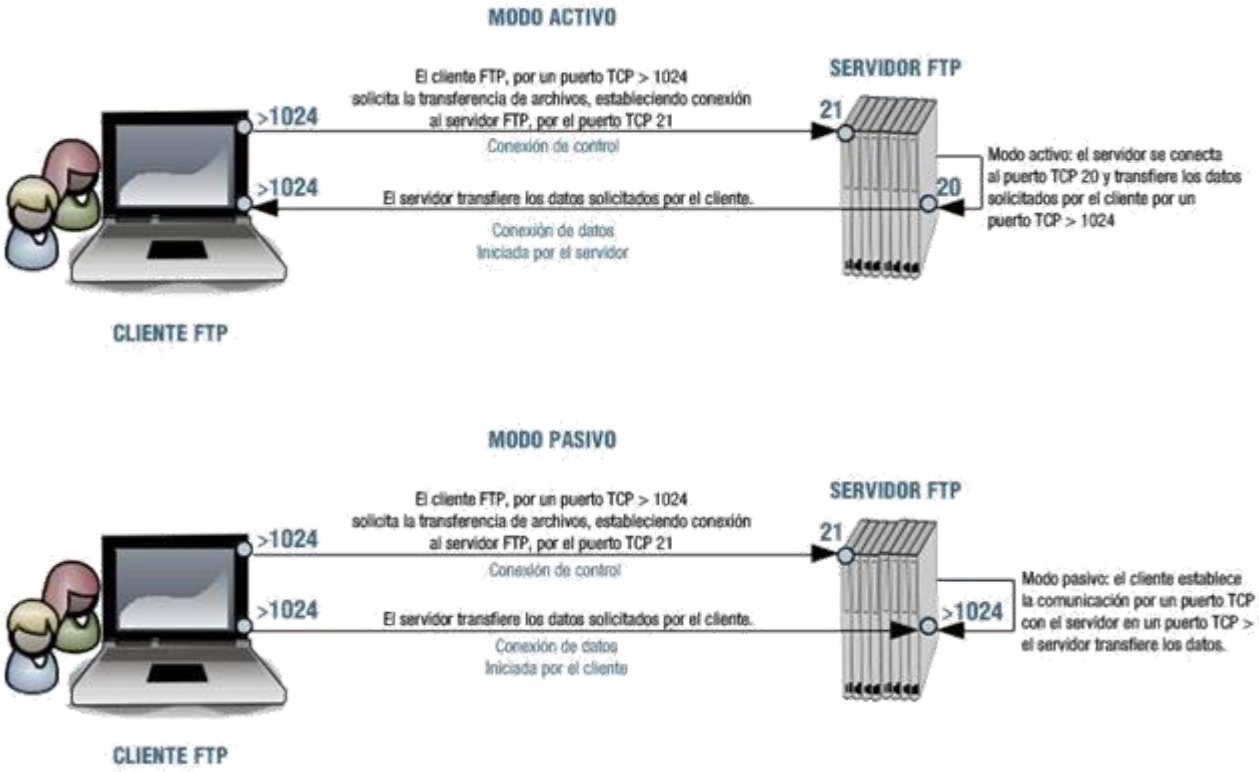
La **conexión de transferencia de datos** (**puerto 20**) existe sólo cuando hay datos que transmitir.

9. Modos de conexión del cliente

El servicio FTP tiene dos modos de funcionamiento, que son:

- Modo estándar o **activo**.
- Modo **pasivo**.

Un cliente FTP puede iniciar la conexión en ambos modos de funcionamiento y los dos utilizan el puerto **21** del servidor como canal de control.



Modo activo

En este modo, el cliente se conecta con el servidor e inicia una conexión de control:

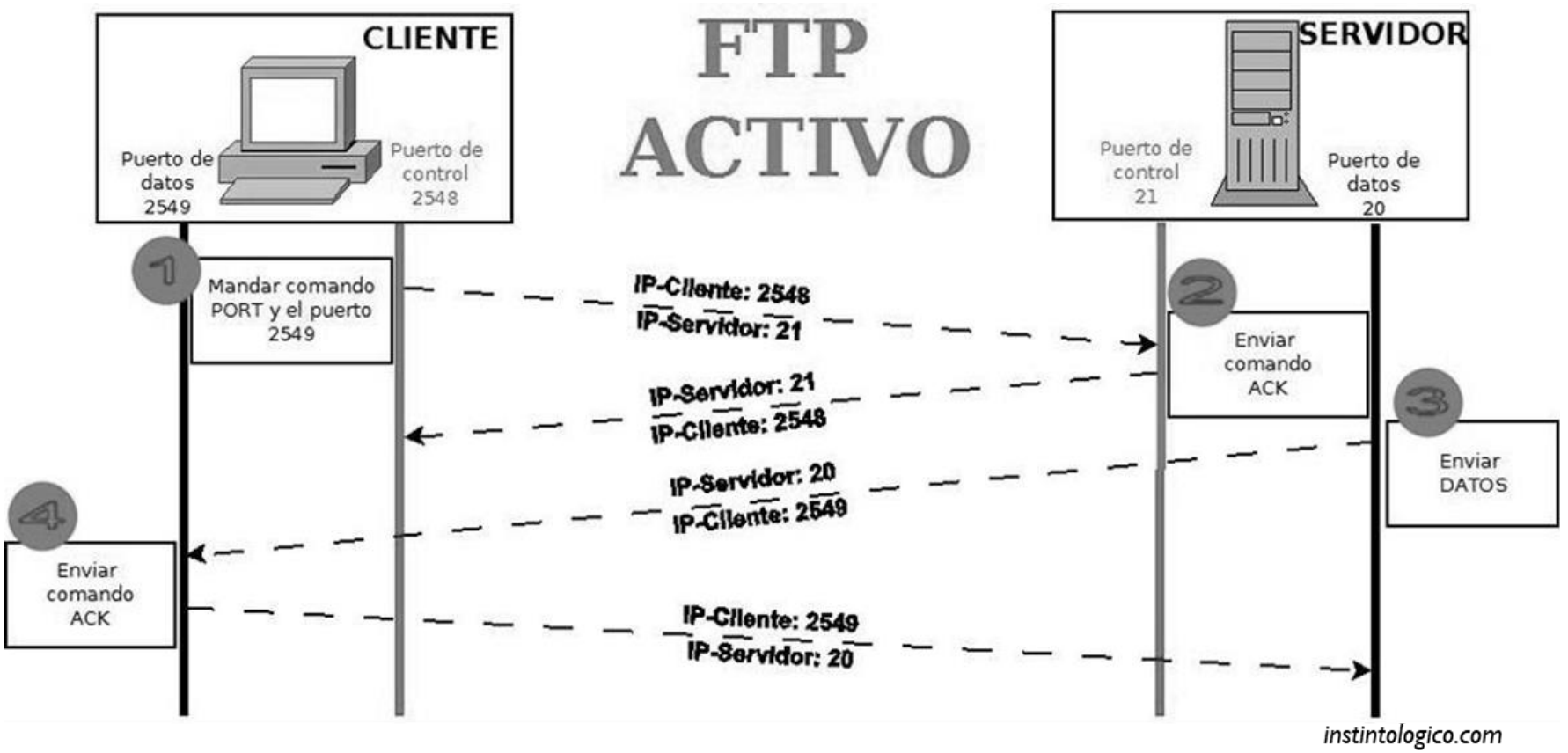
- Abre un puerto local **P** superior a **1024** (por ejemplo el 3000).
- Establece una conexión TCP con el puerto **21** del servidor FTP.

Cuando se solicita una transferencia de ficheros:

- El cliente envía el comando **PORT** al servidor en el que se especifica su dirección IP y su número de puerto (normalmente **P+1**) que *abrirá* para usar en la conexión de datos (por ejemplo el 3001)
- El servidor inicia una conexión TCP desde su puerto **20** hacia el puerto que le ha indicado el cliente
- Se utiliza la conexión de datos para realizar la transferencia de información.

Este modo de trabajo tiene un problema de seguridad. El cliente, al tener que aceptar una conexión de entrada en un puerto superior al **1024**, implica muchos problemas sobre todo si está conectado a una red insegura como lo es Internet. De hecho, los cortafuegos instalados en los equipos es muy probable que rechacen esas conexiones aleatorias.

Por eso se desarrolló el modo **pasivo**.


FTP
ACTIVO
CLIENTE
Puerto de datos 2549
Puerto de control 2548
SERVIDOR
Puerto de control 21
Puerto de datos 20
1
Mandar comando PORT y el puerto 2549
IP-Cliente: 2548
IP-Servidor: 21
2
Enviar comando ACK
IP-Servidor: 21
IP-Cliente: 2548
3
Enviar DATOS
IP-Servidor: 20
IP-Cliente: 2549
4
Enviar comando ACK
IP-Cliente: 2549
IP-Servidor: 20
instintologico.com

Modo pasivo

En el modo pasivo es siempre el cliente el que inicia las conexiones con el servidor. **El puerto 20 del servidor no se utiliza**.

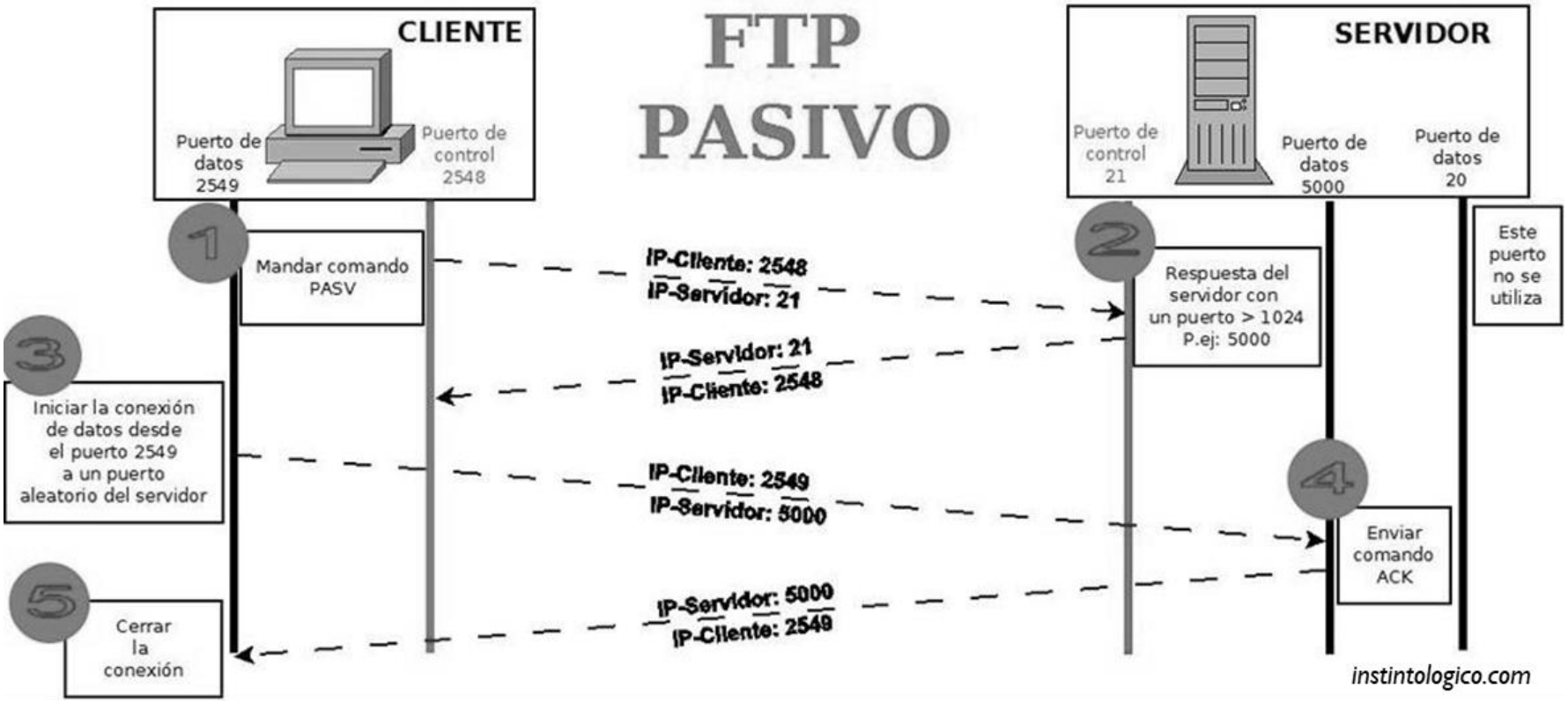
El cliente se conecta con el servidor e inicia una conexión de control:

- Abre un puerto local **P** superior a 1024 (por ejemplo el 3000).
- Establece una conexión TCP con el puerto **21** del servidor.

Cuando se solicita una transferencia de ficheros:

- El cliente envía el comando **PASV** para activar el modo pasivo. Como respuesta a este comando, el servidor retorna un número de puerto superior a 1024 que tenga disponible (por ejemplo el 4000).
- El cliente inicia una conexión TCP, abre un puerto (normalmente **P+1**) superior al 1024 (por ejemplo 3001) hacia el puerto que le ha indicado el servidor (en el ejemplo 4000).
- Se utiliza la conexión de datos para realizar la transferencia de información.

La **diferencia fundamental** es que en el modo pasivo es el cliente quién inicia ambas conexiones. Por ello el modo pasivo resuelve el problema de que el cliente tenga que aceptar conexiones en puertos mayores a 1024, pero lo traslada al servidor, que debe tener un rango de puertos abierto sin protección de su cortafuegos.



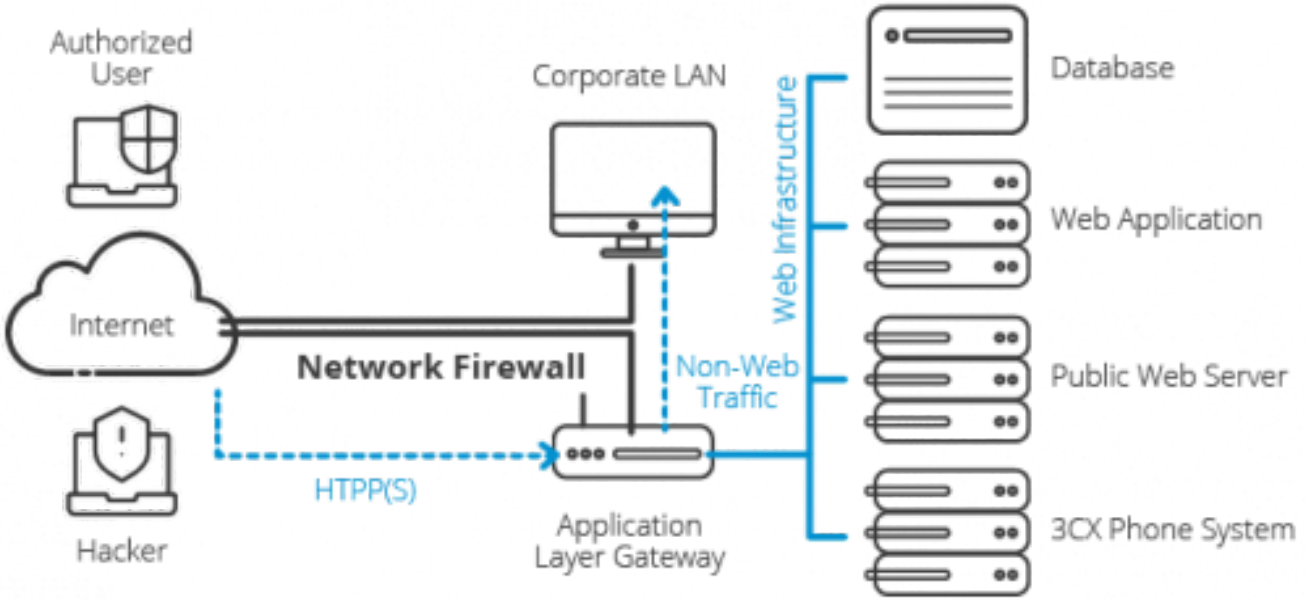
En los puertos bien conocidos (<1024) root tiene acceso a ellos, por lo que si estamos "*hablando*" con un puerto bien conocido sabemos que está hablando con **root**.

Cortafuegos y NAT

El uso de dos conexiones TCP hace que este protocolo tenga que tener un tratamiento especial respecto a reglas de cortafuegos y al habitual uso de NATp (NAPT Network Address Translation) en los routers, con particularidades para clientes y servidores.

- **Cliente FTP en modo activo**
  - El cortafuegos del cliente debe permitir conexiones salientes al puerto 21 y entrantes desde el puerto 20.
  - Si el cliente está detrás de un cortafuegos corporativo y/o un router con NATp, éste debe implementar FTP ALG (*Application Layer Gateway*) para tener en cuenta este protocolo respecto al NAT.
- **Servidor FTP con conexiones en modo activo**
  - Los cortafuegos hacia el servidor deben permitir conexiones TCP de entrada al puerto 21 del servidor y salientes desde el puerto 20 del servidor hacia puertos >1024 de clientes remotos.
- **Servidor FTP con conexiones en modo pasivo**
  - Los cortafuegos hacia el servidor deben permitir conexiones entrantes al puerto 21 y a cierto rango de puertos sobre >1024
  - Si el servidor está tras NATp, el cortafuegos debe permitir conexiones TCP entrantes al puerto 21, redirigir el puerto 21 del router al servidor, permitir cierto rango de puertos >1024 y redirigirlo al servidor. (No se debería configurar un servidor FTP en estas circunstancias).

*La Puerta de Enlace de Capa de Aplicación (del inglés 'Application Layer Gateway' – **ALG**) es un componente de Software que administra protocolos de aplicación específicos como el Protocolo de Inicio de Sesión (del inglés 'Session Initiation Protocol' – SIP) y del Protocolo de Transferencia de Archivos (del inglés 'File Transfer Protocol' – FTP). Un ALG actúa como intermediario entre Internet y un servidor de aplicaciones que pueda entender el protocolo de aplicación. El ALG aparece como servidor de punto final y controla si permite o deniega el tráfico al servidor de aplicaciones. Para ello, intercepta y analiza el tráfico especificado, asigna recursos y define políticas dinámicas para permitir que el tráfico pase a través del Gateway.*



10. Tipos de transferencia de archivos

FTP es un protocolo muy antiguo en el que se diferenciaba el tipo de transferencia de ficheros de texto (en ASCII) y de ficheros binarios.

- **Transferencia ASCII** (*type ascii*). Se transmite carácter a carácter (byte a byte). Se utiliza para archivos de texto (*txt, html, java*…)
- **Transferencia en formato binario** (*type bin*). Se transmite bit a bit. Se utiliza para archivos que no son de texto (ejecutables, imágenes, videos…)

EXTENSIÓN DE ARCHIVO	TIPO DE TRANSFERENCIA
doc (documento)	BINARIO
Z (comprimido)	
ZIP (comprimido)	
Sit (comprimido)	
shar (comprimido)	
ARC (comprimido)	
tar (empaquetado)	
txt (texto)	ASCII
html (página WEB)	
ps (poscript)	
hqx (comprimido)	

Los clientes FTP permiten definir el formato de transmisión a utilizar en función del tipo de archivo a transferir.

En los clientes en modo texto, se debe establecer el formato de forma explícita. Actualmente, muchos clientes ofrecen modo automático que detecta el tipo de archivo y establece el tipo de transferencia adecuado.

11. Servicios de transferencia de archivos en modo texto

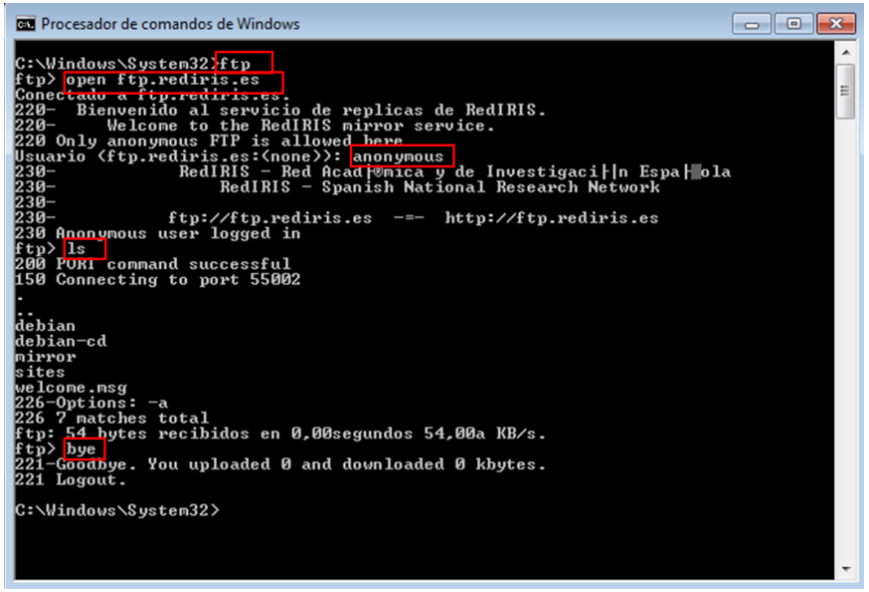
(Clientes en línea de comandos)

Nos comunicaremos con el servidor FTP a través de una **ventana del sistema** si es un sistema operativo Windows, o a través de un **Terminal** si estamos en un sistema operativo Linux. En primer lugar se establece una conexión con el servidor. Una vez autenticada la persona usuaria, a través de una serie de comandos podemos realizar las operaciones que se deseen.

Para iniciar una conexión se emplea la sintaxis ***ftp servidor*** (por ejemplo *ftp www.rediris.es, ftp www.debian.com*).

 # ftp ftp.rediris.es

A continuación pedirá usuario y contraseña para establecer la conexión. En el caso del servidor de *rediris* podemos conectar mediante un usuario cualquiera y una contraseña cualquiera.



Es típico en servidores ftp que exista un usuario anónimo, cuya contraseña sea cualquier dirección de correo -real o ficticia- como, por ejemplo, a@.

Comandos

Una vez establecida la conexión, el cliente pone a disposición de la persona usuaria una serie de comandos (*ls, get, put, cd,* …) para listar el contenido de los directorios del servidor, iniciar bajadas y subidas de ficheros, etc. Dependiendo del sistema operativo pueden variar los comandos que podemos utilizar. Con el comando ***help*** o ***?*** podemos consultar los comandos disponibles.

Si se quiere ejecutar un comando del equipo local se precede del símbolo **!**. Por ejemplo, *ls* muestra un listado del directorio actual del servidor y !ls o !dir muestran un listado del directorio actual del cliente.

Hay una **excepción**: para ejecutar el comando *cd* en local hay que usar ***lcd*** y no !cd.

Comando/s y argumentos	Explicación
ABRIR/CERRAR CONEXIÓN	
open [servidor]	Inicia conexión remota con un servidor ftp.
user [login]	Permite autenticarse como un usuario o usuaria determinado.
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, tecleamos exit en la línea de comandos.
AYUDA	
? / help	Muestra una lista de los comandos disponibles.
? comando **help comando**	Muestra la información relativa al comando.

Comando/s y argumentos	Explicación
TRABAJAR CON DIRECTORIOS	
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.
!<comando>	Ejecuta un comando en un shell en el cliente.
TRABAJAR CON FICHEROS	
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 [patrón]	Obtiene todos archivos que satisfagan cierto patrón, desde el servidor remoto.
put archivo	Envía un archivo al servidor remoto.
mput [patrón]	Envía todos archivos que satisfagan cierto patrón, 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	
chmod	Cambio de permisos en el servidor remoto.
umask	Configura el sistema de permisos en el lado remoto.

12. Servicios de transferencia de archivos en modo gráfico

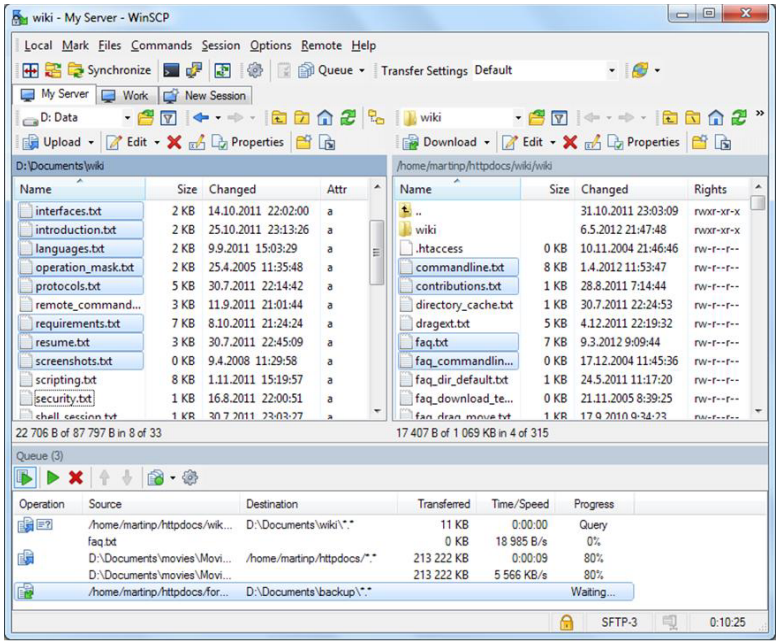
(Clientes gráficos)

Internamente funcionan igual que en el modo comando, pero ofrecen transparencia en las operaciones. Cuentan con un interfaz gráfico de carpetas y ficheros donde se pueden copiar los ficheros entre el equipo cliente y el equipo servidor sin necesidad de conocer los comandos. Esto lo realiza el programa de forma transparente.

Dependiendo del cliente en modo gráfico puede ser posible guardar los datos de las conexiones como plantillas, de tal forma que la próxima vez que intentemos establecer la conexión con un mismo servidor ftp no sea necesario rellenar los campos referentes a la conexión.

Algunos de los más utilizados son:

- Filezilla Client
- WinSCP
- Fire FTP
- SmartFTP
- CuteFTP
- gFtp



www.redeszone.net

FileZilla

FileZilla es válido para plataformas Windows y Linux.



Página web de Filezilla

Posee las siguientes características:

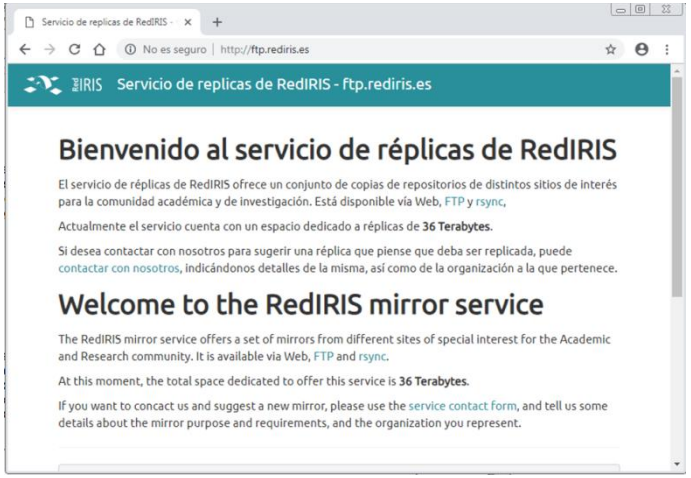
- *Fácil de usar.*
- *Soporta FTP, FTP sobre SSL / TLS (FTPS) y SFTP.*
- *Compatibilidad con múltiples plataformas: se ejecuta en Windows, Linux, BSD, Mac OS X…*
- *Soporte Ipv6.*
- *Disponible en varios idiomas.*
- *Soporta y reanuda la transferencia de archivos de gran tamaño (mayores de 4 GB).*
- *Interfaz de usuario con pestañas.*
- *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.*
- *Fichero de registro.*
- *Sincronización de directorios de navegación.*
- *Búsqueda de archivos remoto.*

13. Navegadores

La mayoría de los navegadores incluyen un cliente ftp, en algunos casos limitado, pero fácil de utilizar. Permiten instalar complementos adicionales que incluyen clientes FPT más completos.

Para acceder, basta con incluir en la barra de dirección la dirección FTP a la que queramos acceder. La sintaxis será del estilo ***ftp://direcciónftp***. Si para acceder al servidor ftp es necesario hacerlo con nombre de usuario y contraseña, la sintaxis es entonces:
ftp://nombreusuario:contraseña@direcciónftp.

Por ejemplo:
ftp://ftp.rediris.es
ftp://alumno@192.168.0.44
Para iniciar conexiones con el usuario **anonymous** no hay que indicar ni usuario ni contraseña.



Pero hay algunas desventajas en utilizar el navegador como cliente ftp. Otros clientes tienen la posibilidad de continuar las descargas cuando se sufre algún tipo de interrupción, lo que no pasa con el cliente ftp del navegador.

El acceso a servidores FTP desde navegadores está en desuso. Hay navegadores que ya no lo admiten.

14. Asegurar el servicio de transferencia de archivos

FTP, como protocolo antiguo (SMTP, Telnet, POP, DNS, etc.) fue diseñado en una época en la que los estándares de seguridad no eran los actuales. Internet era un entorno seguro y la seguridad aplicada se basaba en evitar errores de comunicación, sin pensar que pudiera haber usuarios /as con malas intenciones.

Es un protocolo **no cifrado** lo que, en la actualidad, es una brecha de seguridad respecto al uso de contraseñas y la transmisión de datos sensibles. Por otro lado, tampoco tiene fuertes mecanismos de autenticación para evitar suplantación de identidad.

Empleando el protocolo ftp, cualquiera que tenga acceso al canal de transmisión podrá ver en texto claro todo lo que se transmite puesto que los datos no se cifran. Esto puede carecer de importancia, o no, según 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.



Es por ello que este protocolo está en desuso siendo sustituido por protocolos equivalentes en funcionalidad que aseguren la autenticación y la privacidad de la información, como FTPS, FXP y SFTP/SCP.

FTPS (o FTP/SSL)

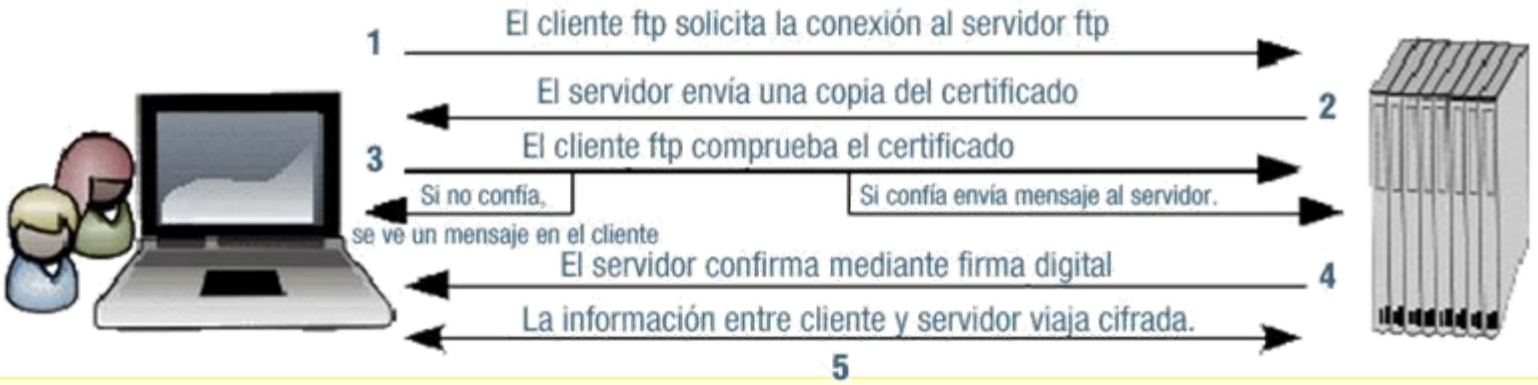
Es el conjunto de especificaciones que determinan cómo encapsular FTP en **SSL** (Secure Sockets Layer) o en **TLS** (Transport Layer Security) para ofrecer comunicaciones seguras.
Consiste en la utilización de algoritmos de cifrado y certificados digitales para garantizar la confidencialidad, integridad y autenticación de los extremos implicados en la comunicación.
Hay dos tipos de funcionamiento:

FTPS Implícito

- El cliente establece una conexión de control cifrada si el servidor soporta FTPS. Todas las transferencias son cifradas.
- Se usan puertos específicos (990 y 989) para que clientes que no soporten FTPS puedan comunicarse de forma estándar no cifrada.
- No hay negociación cliente/servidor respecto al cifrado. Todo se transmite de forma segura.
- Son conexiones tipo **HTTPS**.

FTPS Explícito

- El cliente inicia la comunicación por el puerto estándar (20 y 21) y envía el comando *AUTH SSL* o ***AUTH TLS***.
- Si el servidor soporta el protocolo, se establece la comunicación cifrada y autenticada por certificados.
- Si el servidor no soporta este protocolo, se ofrece el uso de FTP convencional.
- Cliente y Servidor pueden negociar qué partes se transmiten de forma cifrada y cuáles no.



El cifrado al que nos referimos es el cifrado de clave pública o asimétrico. La **clave pública(kpub)** interesa publicarla para que llegue a ser conocida por cualquiera, la **clave privada(kpriv)** no interesa que nadie la posea, solo el propietario de la misma.

Ambas son necesarias 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**.

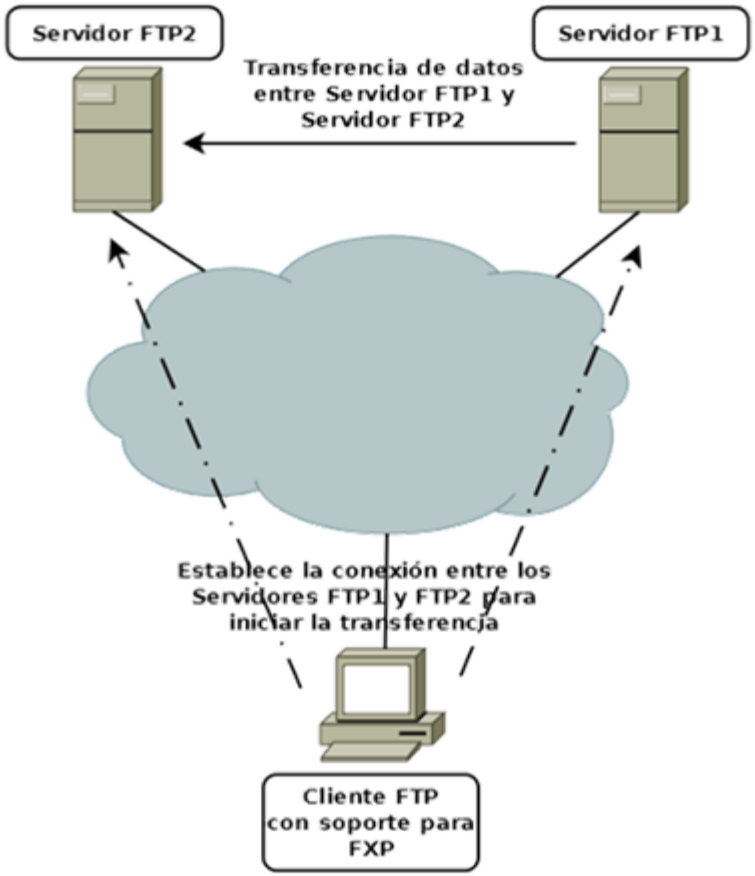
En el cifrado asimétrico podemos estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el **cliente ftp(A)** y el **servidor ftp(B)**.



FXP (File Exchange Protocol)

Es un protocolo que permite la transferencia de ficheros **entre servidores FTP** usando un cliente FTP.

- Los datos se transfieren de un servidor FTP remoto hacia otro servidor sin que la información pase por la conexión del cliente.
- FXP permite una conexión directa entre dos hosts y reduce el tiempo de comunicación entre servidores.
- Al ser la conexión directa entre los servidores, la velocidad de la comunicación es independiente de la conexión a internet del cliente. Y hay que tener en cuenta que la conexión a Internet de los servidores es habitualmente mayor que la de un cliente.
- Lo único que ve el cliente es un mensaje de éxito o fallo de la transferencia.
- Los servidores remotos deben soportar el modo pasivo y permitir comandos PORT para utilizar FXP.



SSH, SFTP y SCP

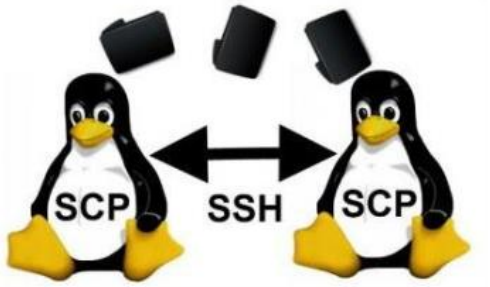
- **SSH** es un protocolo de acceso remoto a terminales de equipos remotos. La característica es que usa un cifrado simétrico de gran rapidez para transferir datos de forma segura.
- SSH integra mecanismos de transferencia de ficheros garantizando autenticación, integridad y privacidad. Se basa en los protocolos SFTP y SCP.



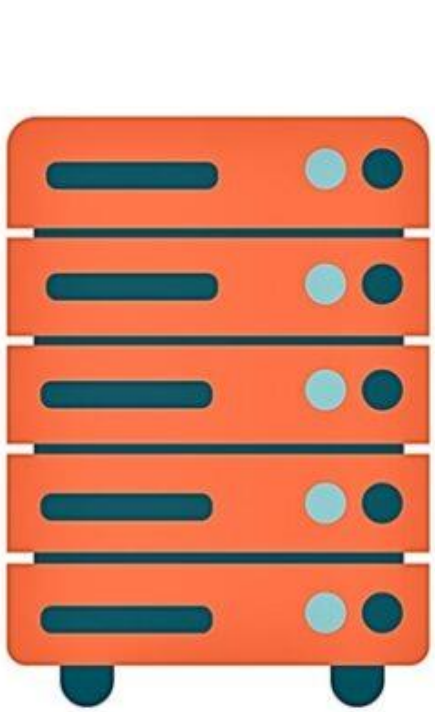
- **SFTP** permite la transferencia de ficheros entre sistemas remotos de una forma muy parecida a FTP con la seguridad que aporta SSH.



- **SCP** es un comando de copia de ficheros entre dispositivos remotos utilizando SSH.
- Existen múltiples clientes SFTP y SCP, tanto en entorno de texto como gráficos.



FTP, FTPS, and SFTP: Security and Port Usage Comparison



FTP

Ports 21 (control) and 20 (data)

No Encryption

Insecure for transmitting sensitive data.

FTPS

Port 21 (explicit connections) and 989 and 990 (implicit connections)

Adds Encryption Using TLS/SSL

Secures data and credentials.

SFTP

Port 22

Uses SSH for Encryption

Provides strong security for all data transfers.

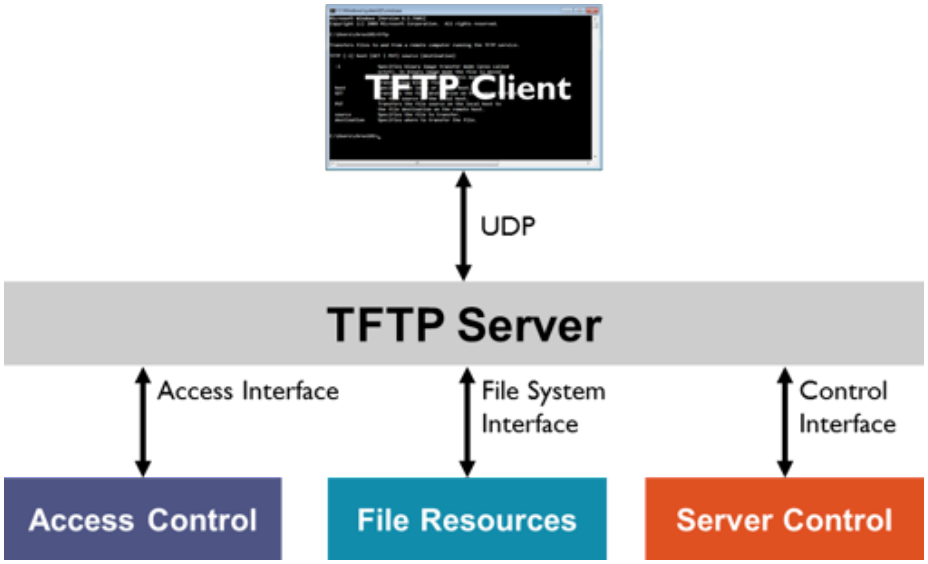
cheap$$L SECURITY

TFTP (Trivial File Transfer Protocol)

Protocolo de aplicación diseñado para ofrecer un servicio de transferencia de ficheros simple y rápido, basado en el modelo cliente/servidor.

Características

- Utiliza UDP en el puerto 69 como protocolo de la capa de transporte.
- No usa mecanismos de autenticación o cifrado.
- Al usar UDP tiene mayor velocidad de transferencia que FTP (TCP), pero no tiene mecanismos de seguridad frente a errores.
- Es muy utilizado como método de transferencia de ficheros de firmware para actualización de software de dispositivos de red (routers, switches).
- Otra utilidad importante es, como opción de DHCP en arranques por PXE (**P**reboot e**X**ecution **E**nvironment) para enviar el sistema operativo a equipos sin unidad de disco (*thin clients*).



15. El servicio de transferencia de archivos en el despliegue de una aplicación Web

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

Es habitual 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 empleamos una aplicación web para subir archivos debemos tener en cuenta cuánto tiempo se 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. En cambio, si empleamos un servidor ftp dependerán de éste las cuestiones anteriores.

Se suele configurar el servidor web con unos parámetros como 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 archivos 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 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** (Secure Copy Protocol de Linux, protocolo de transferencia de archivos en red) 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 servidor SSH, por defecto, 22.

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

Sea cual sea el método ftp que utilicemos para subir archivos y actualizar nuestra web, se desaconseja el uso de aplicaciones no actualizadas que podrían comprometer la seguridad.

A continuación se muestran algunos **errores típicos** que podemos encontrar al subir la aplicación mediante un servidor FTP:

Error	Solución
El cliente FTP muestra el error *access denied*, o similar, cuando subimos o borramos ficheros y carpetas.	Comprobar que el usuario FTP tenga permisos suficientes sobre la carpeta o fichero en la que deseamos subir borrar.
Nuestras páginas no son reconocidas de forma automática al acceder a nuestro dominio.	Los servidores GNU/Linux son sensibles a mayúsculas y minúsculas por lo que hemos de verificar el nombre de los archivos.
El cliente de FTP 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, debemos asegurarnos 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.

16. Establecer permisos en FTP

El protocolo FTP sigue los permisos establecidos en entornos de tipo UNIX y similares, 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 indicada 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'** se muestran los permisos asignados a ficheros y directorios.
Por ejemplo si la salida del anterior comando es:
-rw-r--r-- 1 alumn clase 0 jun 20 01:15 prueba1.txt

Esto significa que:

- ***prueba1.txt*** es un fichero ya que -rw-r--r-- comienza con '-', si fuese un directorio aparecería una 'd'
- ***rw-r--r--*** identifica los permisos del fichero prueba1.txt, que divididos de 3 en 3 representan de izquierda a derecha: propietario, grupo, otros.
  - ***rw-*** identifican los permisos del usuario propietario, en este caso *alumn*. Por lo tanto alumn 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 alumn y aquellos que no pertenecen al grupo clase poseen solamente el permiso de lectura o lo que es lo mismo 4+0+0=4

  Así que los permisos ***rw-r- -r- -*** equivalen a ***644***.

REFERENCIAS

Bibliografía

Servicios de red e internet. García Sánchez, Álvaro. Sanz Rodríguez, Javier. González Sotillo, Álvaro. Enamorado Sarmiento, Luis. Garceta Grupo Editorial.

Servicios en Red. Andreu Gómez, Joaquín . Editorial Editex

Webgrafía

Wikipedia

recursostic.educacion.es

Otros…

