

Introducción al servicio de transferencia de ficheros FTP

- Introducción al servicio de transferencia de ficheros FTP
 - ¿Por qué estudiar FTP?
 - Objetivos de FTP
 - Funcionamiento
 - Modelo cliente-servidor
 - Comandos FTP
 - Respuestas FTP
 - Conexiones de control y de datos
 - Modos de conexión
 - Modo Activo y Modo Pasivo
 - Modo activo
 - Modo pasivo
 - Tipos de transferencias
 - Servidores de FTP
 - Clientes de FTP
 - 1. Clientes de consola de comandos
 - 2. Clientes gráficos
 - 3. Navegadores web
 - 4. Sistemas operativos y otros

El objetivo de este servicio es la **transmisión de archivos** entre un **servidor** y los **clientes**.

- Si desarrollamos una **página web** probablemente queramos subir los archivos html, css, js, png, gif, jpg... que conforman nuestra página a un servidor. Podemos hacerlo a través del **servicio FTP**.
- Quizás queramos descargarnos la última versión de nuestra distribución de Linux, y quizá se distribuya a través de FTP. Con un cliente podemos descargarnos la imagen ISO para grabarla a un DVD o pendrive.
- Si disponemos de un servidor web, es posible que nos interese que cada noche se generen copias de seguridad de los archivos y de las bases de datos. Pero no es buena idea almacenar las copias de seguridad en el mismo disco duro que se encuentran los ficheros. Por FTP podemos transmitir esas copias.

Estas son algunas de las situaciones donde el **protocolo FTP** puede resultar útil. Sin embargo, hoy en día, hay alternativas **más modernas y convenientes** y, especialmente, **más seguras**.

El **protocolo FTP** es un protocolo de red para la transferencia de ficheros entre sistemas conectados a una red **TCP**, y está basado en una arquitectura **cliente-servidor**.

Desde un equipo cliente nos conectamos a un servidor y podemos **enviar** o **descargar** archivos, sin importar el **sistema operativo** utilizado en cada equipo.

El protocolo **FTP** es uno de los más antiguos de Internet. Se empezó a utilizar en abril de 1971 [RFC 114](#) antes de que existiera la pila TCP/IP. Fue modificado varias veces y se le añadieron nuevos comandos y funcionalidades. Lo detallado en el [RFC 959](#) es lo que se usa hoy en día.

El servicio FTP es ofrecido por la **capa de aplicación** del modelo de capas **TCP/IP** y se utilizan los puertos bien conocidos **20** para **transporte de datos** y **21** para **comando y control**. Aunque ya veremos más adelante que el puerto de transporte de datos no siempre es el 20.

FTP fue diseñado pensando en la **velocidad**, pero no se tuvo en cuenta la seguridad y **todo el intercambio de información se realiza en texto plano** sin ningún tipo de cifrado. Desde el **usuario y clave** hasta **cualquier tipo de archivo** por lo que un atacante puede capturar este tráfico, obtener las credenciales y hacerse con los archivos transferidos.

Para solucionar este problema de seguridad, se utilizan aplicaciones como **SCP** o **SFTP**, incluidas en el paquete **SSH** que permiten **transferir archivos de forma segura cifrando el tráfico**. O **FTPS**, que es **FTP** sobre una capa **TLS** o **SSL**.

FTPS no debe de ser confundido con **SFTP**, que es un subsistema de transferencia segura de archivos para Secure Shell (SSH).

Hay ciertas situaciones en las que no importa que nos capturen el tráfico, por ejemplo cuando descargamos archivos que están disponibles de forma pública como por ejemplo una distribución de Linux. Pero incluso en estos casos, puede que sea más cómodo realizar la descarga a través del protocolo **HTTP**.

Tradicionalmente los **navegadores web** incluían **soporte del protocolo FTP** (podían actuar como clientes FTP), pero recientemente muchos navegadores han optado por eliminar este soporte por considerar el **protocolo obsoleto** al haber formas mejores y más seguras de transferir ficheros. Si usas un navegador que no soporte FTP y te interesa utilizarlo, quizá tengas que activarlo a través de **extensiones**.

Básicamente lo que nos permite el protocolo FTP desde el punto de vista del cliente es **gestionar archivos de forma remota**, es decir, subir y bajar ficheros, borrar y crear carpetas, listar directorios, etc. Y desde el punto de vista del servidor: gestionar permisos y compartir archivos.

¿Por qué estudiar FTP?

Comprendo que puede resultar desmotivante estudiar un protocolo que está obsoleto. Así que intentaré justificar por qué puede ser interesante estudiar el protocolo FTP más allá de que esté recogido en el plan de estudios y yo, como profesor, deba impartir este tema y tú, como estudiante, debas estudiarlo.

1. Es un protocolo histórico, que se ha utilizado durante muchos años, y que todo profesional de redes debe conocer.
2. Sirve para repasar conceptos como puertos, TCP, UDP, etc.
3. Puede resultar útil a la hora de transferir ficheros entre máquinas. Por ejemplo, para descargar copias de seguridad mediante un programa.
4. Es un protocolo sencillo que está recogido en la [RFC 959](#). Lo utilizaremos para familiarizarnos con las RFC y perderles el miedo.
5. Sirve para concienciarnos en cuanto a la necesidad de utilizar protocolos seguros, utilizando métodos de cifrado. Capturar una contraseña de FTP transmitida en texto plano es algo fácil de realizar con un sniffer como el Wireshark o con programas dedicados como el Cain & Abel.
6. Vale para ver cómo se pueden comunicar clientes y servidores a través de comandos.
7. Muchos clientes de FTP implementan además otros protocolos más actuales como SSH, conexión con Amazon Cloud, Dropbox, Azure, etc.

Objetivos de FTP

Están detallados en la **RFC 959** y son:

- Promover la compartición de archivos entre equipos.
- Animar al uso indirecto o implícito (a través de programas) de servidores remotos.
- Establecer transferencias de archivos que tengan total independencia de los sistemas de archivos usados en cliente y servidor.
- Transferir archivos de forma eficaz y fiable.

Funcionamiento

Modelo cliente-servidor

Se basa en el modelo **cliente-servidor**:

1. Un cliente inicia una conexión hacia un servidor.
2. Se autentica (usuario y contraseña).
3. Obtiene acceso a los directorios a los que tenga permisos de acceso y podrá realizar las operaciones para las que esté autorizado.

Los servidores FTP ponen a disposición del cliente los **directorios remotos** y los **archivos remotos**, almacenados en estos directorios. A los archivos y directorios del ordenador del cliente les llamaremos **archivos locales** y **directorios locales**.

Si la acción de que realiza el cliente es una transferencia de archivo, se iniciará un proceso de **transferencia de datos**. Por ejemplo, si el cliente solicita descargar un fichero o si el cliente solicita subir un fichero.

Comandos FTP

Los **comandos FTP** son unas cadenas de caracteres formadas por el nombre del comando y los parámetros, separados por espacios.

Existen tres tipos de comandos:

1. Comandos de control de acceso: Son los comandos que se emplean para autenticar el usuario, como **USER** y **PASS**.
2. Comandos de parámetros de transferencia: Para especificar el puerto de conexión, los modos y los tipos de transferencia.
3. Comandos de servicio FTP: Para realizar acciones sobre directorios remotos: subir y bajar archivos, crear directorios, borrar archivos...

Se pueden consultar en detalle el punto **4.1 FTP COMMANDS** de la [RFC 959](#).

Respuestas FTP

Los servidores envían respuestas a los comandos de usuario con un código numérico de 3 dígitos que indica cómo fue procesado el comando al que corresponde la respuesta.

El primer dígito indica si el programa comando se procesó o no con éxito. Los otros indican con mayor detalle el contenido de la respuesta.

- **1xy**: La acción no terminó. Debe obtenerse una nueva respuesta para poder enviar otro comando.
- **2xy**: La acción terminó con éxito. Puede enviarse otro comando.
- **3xy**: Se está a la espera de que el cliente envíe información adicional para poder completar la acción.
- **4xy**: Se indica que la acción solicitada no se pudo realizar ahora, pero puede realizarse más adelante.
- **5xy**: Se indica que la acción solicitada no se puede realizar.

Se pueden consultar en detalle en el punto **4.2. FTP REPLIES** de la [RFC 959](#).

Conexiones de control y de datos

Una conexión FTP entre un cliente y un servidor se lleva a cabo sobre dos conexiones **TCP** de nivel de transporte:

1. **Conexión de control**: Es a través de la cual el cliente envía los comandos y recibe las respuestas a los comandos por parte del servidor tras ser procesados.
2. **Conexión de datos**: Es la que se usa para transferir archivos entre cliente y servidor. Y no solo se envían archivos, sino también información sobre el listado de directorios, por ejemplo.

Un ejemplo sería el siguiente:

1. Un cliente solicita descargar un archivo utilizando el comando **RETR** seguido del nombre de archivo a través de la **conexión de control**.
2. El servidor procesa el comando y envía el archivo a través de la conexión de datos (si es necesario, se abre esa conexión de datos).
3. Cuando termina de enviar el archivo, el servidor envía una respuesta al cliente a través de la conexión de control indicando que el archivo se envió con éxito.

Las **conexiones de datos** se cierran una vez termina la transferencia de archivos y se inician cada vez que se va a llevar a cabo una transferencia.

La **conexión de control** se inicia cuando el cliente solicita la conexión al servidor y se cierra cuando el cliente decide terminar la conexión. O la cierra el servidor tras un tiempo de inactividad por parte del cliente.

Modos de conexión

Los **servidores FTP** utilizan el puerto **21/TCP** para atender a las **conexiones de control**: reciben los comandos por el puerto 21 y envían las respuestas por su puerto 21.

- Si un servidor tiene varias conexiones abiertas, usa para todas ellas el puerto 21 como puerto de escucha de la conexión de control.

El cliente abre un puerto superior a 1024 para cada conexión de control que establece con un servidor FTP.

- Si un cliente tiene varias conexiones de control hacia un servidor FTP, tiene un puerto abierto superior a 1024 para cada una de las conexiones iniciadas. Incluso si fueran simultáneas al mismo servidor.

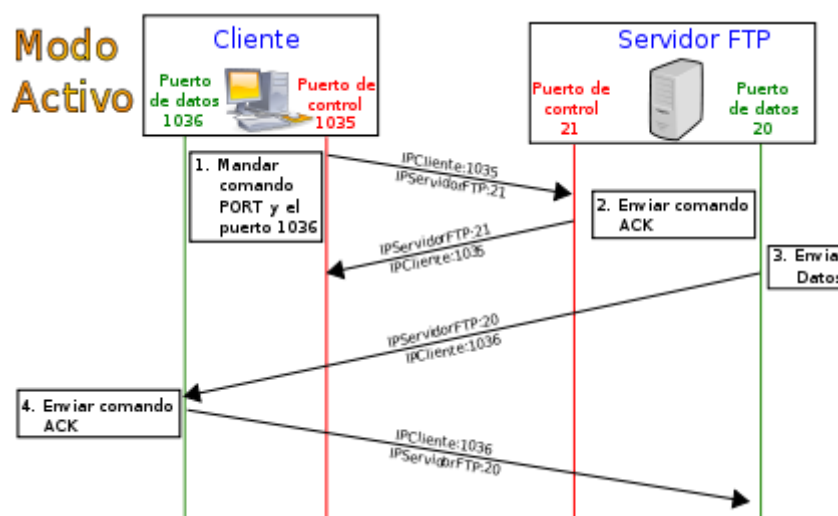
Ejemplo:

Un cliente se conecta a un servidor FTP. Para ello abre el puerto 4000 y se conecta al puerto 21 del servidor. Ahora quiere abrir otra conexión de control contra el mismo servidor, pues abre el puerto 4001 y se conecta al servidor al puerto 21 del servidor. De esta forma, se pueden distinguir las conexiones.

Modo Activo y Modo Pasivo

Existen dos modos de establecer **conexiones de datos** entre cliente y servidor: **modo activo** y **modo pasivo**.

Modo activo



En el **modo activo**, el servidor FTP utiliza el puerto **20/TCP** como **puerto de conexión de datos**.

Este modo es el predeterminado para las conexiones FTP y fue el primero en desarrollarse.

En el **modo activo**, una vez se establece una conexión de control, para establecer una **conexión de datos** se hace lo siguiente:

1. El cliente envía al servidor un comando PORT a través de la conexión de control indicando un número de puerto TCP superior a 1024 que pretende abrir para la conexión de datos.
2. El servidor inicia la conexión de datos (no transferencia) entre su puerto **TCP/20** y el puerto que le indicó el cliente.
3. La conexión de datos se utiliza para la transferencia de archivos.

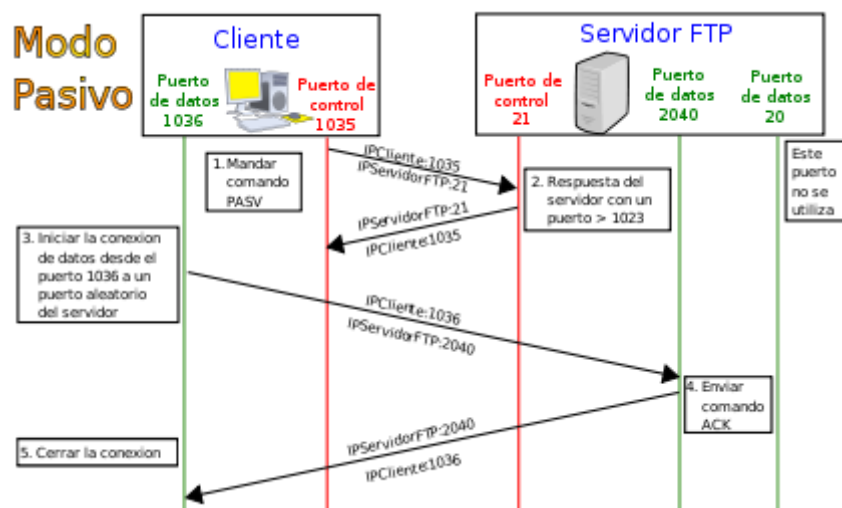
Pero aquí tenemos un **problema de seguridad** y es que **el servidor inicia la conexión hacia un puerto del cliente**. Por lo tanto la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior a 1024, lo que es inaceptable en redes inseguras como Internet.

- Esto puede ser interpretado como una **amenaza** por un **cortafuegos** que rechazará este tipo de conexiones aleatorias (intento de conexión externa no autorizada).
- Los clientes pueden encontrarse tras una **NAT**, y si bien existen mecanismos que podrían interpretar y permitir este tipo de conexiones, no es lo recomendable.

Cabe recordar que FTP se desarrolló en los años 70, donde los problemas de seguridad son distintos a los de hoy en día.

Para solucionar este problema, se desarrolló el **modo pasivo**.

Modo pasivo



En el **modo pasivo**, el servidor FTP **no** utiliza el puerto **20/TCP** como **puerto de conexión de datos**.

En el **modo pasivo**, una vez se establece una conexión de control, para establecer una **conexión de datos** se hace lo siguiente:

1. El cliente envía al servidor un comando PASV, indicando que va a establecer conexión en **modo pasivo**.
2. El servidor responde con un número de puerto TCP superior a 1024 (**¡aquí no es el puerto 20/TCP!**) que pretende abrir para la conexión de datos.
3. El cliente inicia la conexión de datos abriendo un puerto TCP disponible y mayor que 1024 y conectándose con el puerto que le indicó el servidor.
4. Se transmiten los archivos por la conexión de datos.

En este caso, el problema de seguridad se traslada del cliente al servidor. Es posible definir un rango de puertos para este tipo de conexiones.

Tipos de transferencias

Para transmitir archivos entre clientes y servidores hay dos modos diferentes:

1. **MODO ASCII**
2. **MODO BINARIO**

El **modo ASCII** (predeterminado) se utiliza para transmitir **archivos de texto**. El modo binario, para todo lo demás.

Los sistemas operativos UNIX y Windows representan los finales de línea de los archivos de texto de diferente manera, y en el modo ASCII se realizan conversiones de caracteres hacia y desde el juego de caracteres estándar de red. Por ejemplo, los caracteres de fin de línea se convierten según sea necesario, en función del sistema operativo de destino.

Normalmente los clientes FTP gráficos realizan esta función sin que sea necesario especificar el modo. En el cliente ftp de consola debremos especificar el modo de transferencia:

Para establecer el modo ascii:

```
ascii
```

Para establecer el modo binario:

```
binary
```


Servidores de FTP

Existen una gran cantidad de servidores FTP para distintos sistemas operativos:

- [vsftpd](#): Servidor seguro y rápido para sistemas Unix. La configuración es sencilla y está centrado en la seguridad y eficiencia.
- [Filezilla Server](#): Servidor FTP para Windows con licencia GPL.
- [FTP Serv-u](#): Servidor FTP para Windows y Linux. Es comercial, aunque dispone de una versión de evaluación de 14 días que cuando caduca se transforma en una versión limitada.
- Otros:
 - Microsoft FTP Server.
 - [ProFTPd](#)
 - [PureFTPd](#)

Clientes de FTP

Existen un montón de clientes FTP. Vamos a distinguir 4 tipos:

1. Clientes de consola de comandos

```
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls *.c
200 Port command successful
150 Opening data channel for directory listing of "/*.c"
-rw-r--r-- 1 ftp ftp      115693 Apr 27 10:56 gc.c
-rw-r--r-- 1 ftp ftp      14289 Apr 27 10:57 gtk_functions.c
-rw-r--r-- 1 ftp ftp       902 Apr 27 10:57 map_sources.c
-rw-r--r-- 1 ftp ftp     21701 Apr 27 10:57 olc.c
-rw-r--r-- 1 ftp ftp      2993 Apr 27 10:57 os_coord_ordinance_survey.c
-rw-r--r-- 1 ftp ftp      7519 Apr 27 10:57 os_coord_transform.c
226 Successfully transferred "/*.c"
ftp> get gc.c
local: gc.c remote: gc.c
200 Port command successful
150 Opening data channel for file download from server of "/gc.c"
226 Successfully transferred "/gc.c"
115693 bytes received in 0.01 secs (17.5355 MB/s)
ftp> █
```

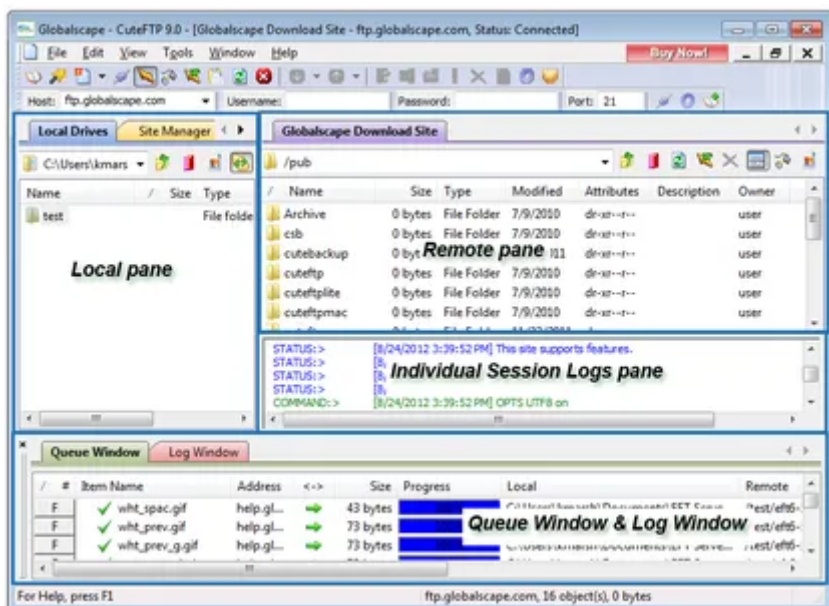
Los sistemas operativos Windows y Linux integran un cliente **ftp** que se ejecuta en modo comando.

En [este enlace](#) se pueden consultar los comandos del cliente ftp de Windows, que es similar al de Linux aunque no es igual (los comandos pueden coincidir).

2. Clientes gráficos

Existen multitud de clientes FTP con entorno gráfico:

- [FileZilla](#): Disponible para Windows, Mac y Linux con licencia GPL.
- [Smart FTP](#): Solo disponible para Windows, con licencia de evaluación de 30 días. Dispone de muchas opciones de configuración.
- [gFTP](#): Disponible para Linux, con licencia GPL. Fácil de utilizar.
- Otros:
 - [Cyberduck](#)
 - [Transmit](#)



Los clientes de FTP suelen mostrar los archivos locales, los archivos remotos, un panel con la cola de tranferencias, y un panel con los comandos de sesión.

3. Navegadores web

Los navegadores web históricamente incluían un cliente FTP integrado, pero recientemente muchos lo han declarado obsoleto.

En este [documento PDF](#) explican cómo activar el soporte FTP en el navegador en las versiones:

- Firefox version 88 (or newer)
- Microsoft Edge version 90 (or newer)

- Google Chrome version 90 (or newer)

Otra opción, es utilizar alguna extensión del navegador que actúe como cliente FTP.

4. Sistemas operativos y otros

Los sistemas operativos pueden tener soporte FTP integrado y pueden permitirnos montar carpetas de servidores FTP como si de una carpeta de nuestro sistema de archivos se tratase.

Más información: [Chrome and Firefox Killed FTP Support: Here's an Easy Alternative](#)

También hay editores de texto para programación web que incluyen un cliente FTP integrado y que son capaces de subir los archivos una vez el autor guarda los cambios, o librerías para distintos lenguajes de programación que pueden automatizar las acciones a realizar en un servidor.