# Servidores de transferencia de archivos: FTP

## Servidores y clientes FTP

El protocolo clásico para la transferencia de archivos en Internet se denomina FTP (*File Transfer Protocol*). Con el estado actual de Internet y las múltiples opciones de trasferencia de archivos en la web puede parecer algo innecesario pero sigue siendo una opción sencilla y específica por lo que en ámbitos profesionales continúa gozando de buenísima salud. Por ejemplo sigue siendo el método más habitual para subir archivos, actualizaciones o modificaciones de contenido a un servidor web, especialmente en el modo de hosting.

Los **navegadores web** más modernos también posibilitan la conexión con un servidor FTP, aunque sólo para ver su contenido y realizar descargas, usando una URL con el siguiente formato: `ftp://usuario:contraseña@servidorftp/recurso`

Algunos de las aplicaciones **clientes FTP** más utilizadas son ***Filezilla Client*** (gratuito, para Windows, Linux y Mac), ***gFTP*** (gratuito, instalado por defecto en las principales distribuciones Linux), o ***WinSCP*** (gratuito, para Windows).

En Linux existen muchos servidores FTP diferentes, no hay más que echar un vistazo al paquete virtual de Ubuntu. Los dos más populares actualmente son ***ProFTPd*** y ***vsFTPd***. El primero es más sencillo de utilizar y sus archivos de configuración y estructura similar hacen que se parezca mucho a Apache. Sin embargo el segundo es el servidor FTP por defecto en las principales distribuciones de Linux lo que hace que sea más sencillo de instalar y además se considera más seguro.

En Linux, un usuario FTP tiene su propia carpeta de usuario asociada (en la URI `/home/nombre-usuario-ftp/`) por lo que la integración es muy alta y FTP se beneficia de la completa y potente gestión de usuarios de Linux.

Debes tener en cuenta que también es posible conectarte a un servidor **FTP desde el símbolo del sistema o terminal**, que en ocasiones puede ser de utilidad si no se dispone de un cliente FTP o un interfaz gráfico.

Para usarlo, tan sólo debes escribir el comando ***ftp*** y luego deberás usar las distintas **opciones** que ofrece esa herramienta. Usa los siguientes enlaces para ver una **descripción detallada** de las distintas opciones que ofrece la **herramienta ftp por línea de comandos para *Windows* y *Ubuntu***.

## Puertos y modos de conexión

Por defecto, el servidor FTP utiliza los puertos 20 y 21 para realizar las comunicaciones por FTP. El **puerto 21** se usa para la transmisión de comandos de **control** y el **puerto 20** para los **datos** que se encarga de la transmisión de los ficheros. Se diseñó así para poder enviar comandos sin la necesidad de detener la transmisión de datos ni de encolarlos tras estos.

El protocolo FTP puede funcionar en modo activo **(PORT)** o pasivo **(PASV)**, lo cual determina cómo se establece la conexión de datos.

**MODO ACTIVO (PORT)**

Es el modo predeterminado para las conexiones FTP, de hecho, fue el primero en desarrollarse. En este modo el servidor utilizará el puerto 20 para la transferencia de datos, mientras que transmitirá los comandos utilizando el puerto 21. El cliente en cambio utilizará un puerto aleatorio P superior al 1023 para la transferencia de comandos, y un puerto P+1 para la transferencia de datos.

Históricamente el principal inconveniente de este modo ha radicado en que, si no hay una directiva específica en los firewalls del lado del cliente, la conexión es propensa a ser bloqueada. Esto se debe a que existen dos conexiones independientes: una de salida en la que el cliente establece la conexión del canal de control y otra de entrada en la que el servidor hace lo propio con el de datos. Esta última conexión es iniciada por el servidor en el puerto previamente negociado, y en ocasiones es bloqueada por el firewall del cliente al ser identificada como un intento de conexión externa no autorizada. Este sistema se diseñó en los años 70, cuando la democratización de los sistemas informáticos y los problemas de seguridad eran completamente distintos a los que podemos encontrar hoy en día.

**MODO PASIVO (PASV)**

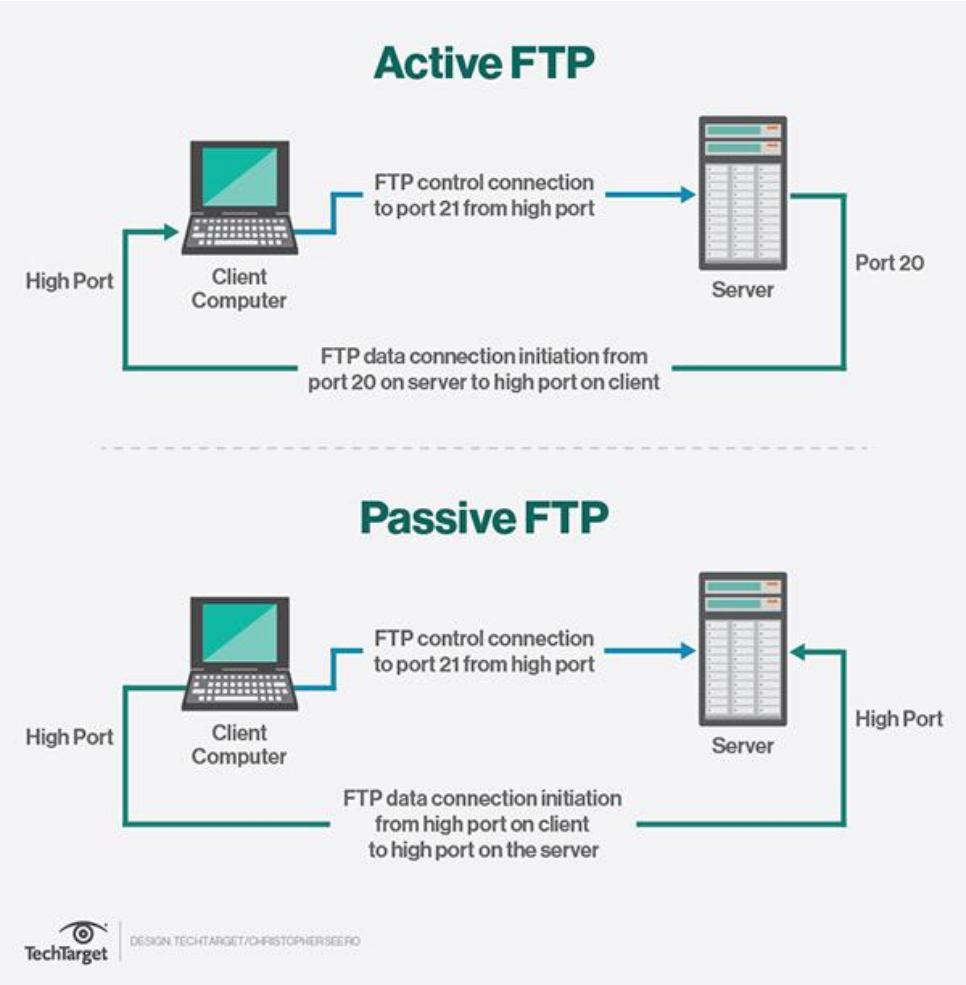
El modo de FTP pasivo surge como consecuencia de los problemas de conexión del modo activo. Este modo mantiene los dos canales (control y datos) pero en este caso es el cliente el encargado de establecer las dos conexiones.

El servidor sigue manteniendo el puerto 21 como puerto de comandos, el puerto de datos en cambio difiere del modo activo y pasa a ser un rango de puertos Q superior a 1023. Del lado del cliente seguimos manteniendo el puerto P superior a 1023 para control, y el puerto P+1 para datos.

El principal inconveniente de habilitar el modo pasivo en un servidor FTP está asociado con el riesgo extra en materia de seguridad que conlleva la apertura de un rango de puertos extra con respecto al modo activo. Notemos que para que el modo pasivo funcione adecuadamente estos puertos deben ser abiertos tanto en el servidor como en el firewall.

Para minimizar este riesgo la estrategia pasa por definir un rango de puertos lo más pequeño posible en función del número de conexiones concurrentes que esperemos tener, pero ojo, porque un cliente no se corresponde con una única conexión, es muy probable que cada cliente abra múltiples conexiones concurrentes.

En las figuras de abajo podemos ver con más detalle el proceso de conexión en ambos modos.



## Tipos de usuarios y acceso al servicio

En FTP existen dos tipos básicos de usuario, los corrientes y los anónimos. Realmente definen el tipo de acceso porque indican si te estás autenticando con un usuario concreto o estás utilizando una cuenta anónima que generalmente no requiere autenticación. Un servidor FTP puede servir ambos tipos simultáneamente.

- **FTP anónimo**: Este modo se utiliza generalmente cuando el servidor FTP se usa para distribuir cualquier tipo de archivo o archivos a un número muy elevado de usuarios en una situación en la que la identificación no es muy importante. Si por ejemplo hemos realizado una aplicación de software libre y queremos distribuirla es una buena opción. En este tipo de conexión sólo se le pide al cliente un nombre de usuario anónimo (generalmente y por defecto es anonymous) y si acaso (no siempre) una contraseña que se refiere a cualquier dirección de correo electrónico válida. Una vez nos hemos conectado al servidor tendremos acceso al directorio anónimo y sus subdirectorios.

- **FTP corriente**: En este caso los usuarios de FTP son los que existen en la máquina en la que instalamos el servidor. Estos usuarios podrán leer y copiar a su directorio personal archivos remotamente. Las mismas credenciales que tienen en la máquina serán las que necesiten para conectarse a mediante FTP. *vsFTPd* permite que este tipo de conexión se restrinja a los usuarios de un grupo determinado. El uso de este tipo de conexión es muy habitual en los hostings web, aunque no es práctico cuando tenemos muchísimos usuarios como puede ser el caso. Para ello se utilizan los denominados **usuarios virtuales** que tendrán credenciales FTP pero no cuenta en el servidor Linux.

## Permisos y cuotas

Hemos visto en los puntos anteriores cómo se configuran los permisos generales de usuarios anónimos y corrientes. Los permisos específicos en el caso que estamos tratando se concretan con los permisos del usuario o grupo al que pertenece en Linux para las carpetas correspondientes. Es importante recordar que si al ir a conectarnos con un usuario al servidor FTP nos aparece el error 500, es porque no se le pueden dar permisos de escritura al usuario en su carpeta raíz.
Establecer límites a los usuarios es de vital importancia en un sistema FTP para evitar que unos pocos consuman demasiados recursos. Aunque no es parte de este módulo, quiero

destacar que es posible limitar el ancho de banda y número de conexiones simultáneas que puede usar el mismo usuario. El tema que aquí vamos a discutir es el más importante porque un solo usuario puede ocupar demasiado espacio de disco y no dejar nada para otros. Para evitar esto se establece una cuota de espacio en disco.

## FTP seguro

La comunicación a través de FTP de manera segura se puede establecer a través de FTPS o SFTP. El primero de ellos conlleva el uso de una capa SSL/TLS debajo del protocolo estándar FTP para cifrar los canales de control y/o datos, de manera similar a la utilizada por el protocolo HTTPS. En cambio SFTP utiliza el protocolo SSH para proporcionar la seguridad a los datos, aunque permite ser usado con otros protocolos de seguridad.

## FTP en el proceso de despliegue de aplicaciones web

El despliegue de aplicaciones Web mediante FTP implica instalar un servidor FTP en la misma máquina en la que tengamos el servidor Web y/o de Aplicaciones y habilitar la carpeta de la aplicación para que se puedan conectar determinados usuarios con opciones para modificar el contenido. De esta forma podremos desarrollar toda la aplicación en una máquina de desarrollo y pruebas y subir la versión final y probada al servidor sin necesidad de acceso físico a él. Evidentemente para un desarrollador web que trabaje para muchos clientes esta opción es completamente vital.