EL SERVICIO FTP

Descargar PDF

PROTOCOLO FTP

INTRODUCCIÓN

El **File Transfer Protocol (FTP)** es un protocolo de la capa de aplicación diseñado para permitir la transferencia de archivos entre sistemas conectados a una red basada en TCP/IP. Su finalidad principal es proporcionar un mecanismo estándar que permita a un usuario acceder a archivos almacenados en un sistema remoto y realizar operaciones como listar directorios, descargar archivos o subir contenido.

FTP fue durante muchos años uno de los protocolos más utilizados en Internet, especialmente para la publicación de páginas web, la distribución de software y el intercambio de información en redes académicas y empresariales. Aunque en la actualidad ha sido sustituido en muchos entornos por alternativas más seguras, sigue siendo un protocolo fundamental desde el punto de vista formativo, ya que permite comprender claramente el funcionamiento de los servicios de red clásicos.

Desde una perspectiva didáctica, FTP resulta especialmente interesante porque hace explícitos muchos aspectos que en protocolos modernos quedan ocultos, como el uso de múltiples conexiones TCP, la negociación de puertos o el intercambio de comandos en texto plano.

CONTEXTO HISTÓRICO Y ESTANDARIZACIÓN

El protocolo FTP se desarrolló en los primeros años de ARPANET, en una época en la que la seguridad no era una preocupación prioritaria y las redes estaban formadas por un número reducido de equipos de confianza. Su especificación principal está recogida en el **RFC 959**, publicado en 1985, que define el funcionamiento básico del protocolo tal y como sigue utilizándose hoy en día.

A lo largo del tiempo se han publicado otros RFC que añaden extensiones o aclaran determinados comportamientos, pero el núcleo del protocolo apenas ha cambiado. Esto explica por qué FTP conserva decisiones de diseño que hoy pueden parecer poco adecuadas, como la ausencia de cifrado o el uso de múltiples conexiones.

FTP pertenece a la **capa de aplicación** del modelo TCP/IP y utiliza **TCP** como protocolo de transporte, lo que garantiza una comunicación fiable y orientada a conexión.

ARQUITECTURA Y PRINCIPIOS DE FUNCIONAMIENTO

FTP utiliza una arquitectura **cliente-servidor**. El cliente inicia la conexión y solicita servicios, mientras el servidor espera y responde a esas peticiones.

La clave de FTP es que emplea **dos canales de comunicación independientes**:

- **Canal de control**: Se usa para enviar comandos y recibir respuestas (por ejemplo, iniciar sesión, pedir un archivo, cambiar de carpeta, etc.). Este canal permanece abierto durante toda la sesión y normalmente utiliza el **puerto TCP 21** del servidor. La **comunicación es en texto plano (ASCII)**, lo que permite observar fácilmente el intercambio de comandos usando herramientas como `telnet`.
- **Canal de datos**: Se utiliza exclusivamente para transferir archivos o listados de directorios. La **comunicación puede ser en texto plano o binaria**, dependiendo del modo de transferencia. A diferencia del canal de control, esta conexión se abre solo cuando es necesario transferir información y se cierra al

terminar. El **puerto** usado para el canal de datos **varía según el modo de conexión (activo o pasivo)**, lo que puede complicar la configuración en redes con firewalls o NAT.

Esta separación entre canales permite que los comandos y las transferencias de archivos no se mezclen, facilitando la gestión y el control de la sesión FTP.

SESIÓN FTP

El proceso típico de una sesión FTP incluye las siguientes fases:

1. **Conexión inicial**: El cliente abre una conexión TCP al puerto 21 del servidor (canal de control) y recibe un mensaje de bienvenida.
2. **Autenticación**: El cliente envía su usuario (`USER`) y contraseña (`PASS`). El servidor valida las credenciales y permite el acceso.
3. **Comandos y transferencias**: El cliente envía comandos FTP por el canal de control (como `LIST`, `RETR`, `STOR`). Cuando se necesita transferir datos, se abre el canal de datos:
   - **Modo activo**: el servidor se conecta al cliente para transferir datos.
   - **Modo pasivo**: el cliente se conecta al servidor para transferir datos.
     Cuando la transferencia termina, se cierra el canal de datos.
4. **Finalización**: El cliente envía el comando `QUIT` y el servidor cierra la sesión y libera los recursos.

USUARIOS FTP

En el contexto del protocolo **FTP (File Transfer Protocol)**, un *usuario* representa una **identidad lógica** utilizada por el servidor para **autenticar**, **autorizar** y **aplicar políticas de acceso** sobre los recursos compartidos.

Es importante destacar que **FTP no define internamente cómo se gestionan los usuarios**, sino que especifica:

- El mecanismo de autenticación (comandos `USER` y `PASS`)
- El control de acceso a las operaciones disponibles

Por este motivo, los tipos de usuarios FTP se clasifican de forma **conceptual**, independientemente del sistema operativo o de la implementación concreta del servidor. La implementación concreta de los usuarios, sus credenciales y permisos depende del sistema operativo y del software del servidor FTP utilizado. Nos centraremos en las categorías más comunes de usuarios FTP, basándonos en dos criterios principales:

1. El **método de autenticación**
2. La **relación del usuario con el sistema subyacente**

USUARIOS SEGÚN MÉTODO DE AUTENTICACIÓN

Esta clasificación describe **cómo un usuario demuestra su identidad** ante un servidor FTP. FTP permite dos métodos principales de autenticación de usuarios:

USUARIO ANÓNIMO

El **usuario anónimo** permite el acceso al servicio FTP sin una identidad individual verificable.

- Utiliza credenciales genéricas, típicamente:
  - Usuario: `anonymous` o `ftp`

    - Contraseña: valor simbólico (por ejemplo, una dirección de correo electrónico)
- No existe una validación real de la identidad del cliente.
- Todos los accesos anónimos son tratados como una misma entidad lógica.

Características:

- Acceso generalmente restringido a **lectura**.
- Escritura permitida únicamente en configuraciones muy controladas.
- Ausencia de trazabilidad individual de acciones.

Casos de uso:

- Distribución pública de software.
- Repositorios abiertos de documentación.
- Servicios académicos o históricos.

Limitaciones:

- Bajo nivel de seguridad.
- Difícil auditoría.
- Uso cada vez menos frecuente en entornos modernos.

USUARIO AUTENTICADO

El **usuario autenticado** se identifica mediante un par **usuario/contraseña**, lo que permite establecer una identidad individual diferenciada.

- Requiere credenciales únicas.
- Permite aplicar políticas específicas por usuario.
- El backend de autenticación es transparente desde el punto de vista del protocolo FTP.

Características:

- Identidad individual bien definida.
- Posibilidad de auditoría y control detallado.
- Base de la mayoría de configuraciones FTP actuales.

Casos de uso:

- Acceso privado a servidores de archivos.
- Intercambio de información entre usuarios autenticados.
- Automatización de transferencias (especialmente con FTPS o SFTP).

Consideraciones de seguridad:

- Requiere protección del canal de comunicación.
- La seguridad depende de la robustez de las credenciales.

USUARIOS SEGÚN SU RELACIÓN CON EL SISTEMA SUBYACENTE

Esta clasificación describe **qué representa realmente un usuario FTP dentro del entorno donde se ejecuta el servicio**, con independencia del sistema operativo o de la implementación concreta del servidor FTP.

A diferencia del método de autenticación, que define **cómo se verifica la identidad**, este criterio se centra en **la naturaleza de la identidad** y en su grado de acoplamiento con el sistema anfitrión.

USUARIO DEL SISTEMA

Un **usuario del sistema** es una identidad FTP que corresponde directamente a una cuenta real existente en el sistema subyacente sobre el que se ejecuta el servicio FTP.

En este modelo, el servidor FTP reutiliza las credenciales y mecanismos de autenticación del sistema anfitrión, estableciendo una correspondencia directa entre la identidad FTP y la identidad del sistema.

Características principales:

- La identidad FTP coincide con una cuenta real del sistema.
- El control de acceso suele alinearse con los permisos del sistema de archivos.
- La gestión de usuarios depende del propio entorno anfitrión.

Casos de uso habituales:

- Entornos pequeños o tradicionales.
- Sistemas donde FTP actúa como servicio complementario.
- Accesos técnicos o administrativos.

Limitaciones y riesgos:

- Mayor superficie de ataque si la misma cuenta se reutiliza para otros servicios.
- Menor aislamiento entre servicios.
- Dependencia directa de la correcta gestión de cuentas del sistema.

USUARIO VIRTUAL

Un **usuario virtual** es una identidad que **existe exclusivamente dentro del servicio FTP** y no corresponde a ninguna cuenta real del sistema subyacente.

El servidor FTP gestiona internamente estas identidades y aplica las políticas de acceso correspondientes sin que el sistema anfitrión tenga conocimiento explícito del usuario. El acceso a los recursos se realiza mediante un mapeo lógico interno.

Características principales:

- Independencia total del sistema operativo.
- Mayor escalabilidad en entornos multiusuario.
- Aislamiento efectivo entre identidades.

Casos de uso habituales:

- Servicios de alojamiento (hosting).
- Plataformas multiusuario.
- Infraestructuras gestionadas o en la nube.

Ventajas principales:

- Menor impacto sobre la gestión del sistema anfitrión.
- Mejor control de seguridad y segmentación.

- Modelo preferido en arquitecturas modernas.

GRUPOS DE USUARIOS FTP

Los **grupos de usuarios** constituyen un mecanismo de **organización y autorización** que permite aplicar políticas comunes a múltiples identidades dentro de un servicio FTP.

A diferencia de los usuarios, los grupos:

- No representan identidades individuales
- No participan en el proceso de autenticación
- No establecen sesiones FTP propias

Su función principal es facilitar la **gestión colectiva de permisos y restricciones**, actuando como una **capa intermedia entre el usuario y los recursos**. Desde un punto de vista conceptual, los grupos permiten:

- Agrupar usuarios con características o necesidades similares
- Simplificar la administración de permisos
- Aplicar políticas de acceso de forma homogénea
- Reducir la complejidad de la gestión individual de usuarios

Los grupos influyen directamente en los mecanismos de **autorización**, pero no en los de autenticación.

AUTORIZACIÓN Y PERMISOS EN FTP

La **autorización** en FTP es el proceso por el cual el servidor decide qué puede hacer cada usuario una vez que ha iniciado sesión. No es lo mismo que la autenticación (identificarse), sino que se trata de los permisos y limitaciones que tiene cada usuario.

Los permisos en FTP, entre otros, determinan:

- A qué carpetas y archivos puede acceder cada usuario.
- Qué acciones puede realizar: ver (leer), subir (escribir), borrar o modificar archivos.
- Cuánto espacio puede usar (cuotas).
- Cuántas conexiones puede tener abiertas al mismo tiempo.
- Qué velocidad máxima de transferencia puede usar.

El administrador del servidor FTP puede establecer estos permisos para cada usuario o grupo, y también puede aplicar restricciones adicionales, como:

- Limitar el acceso según la dirección IP del cliente.
- Permitir el acceso solo en ciertos horarios.
- Restringir el tipo de archivos que se pueden transferir.

Por último, los servidores FTP suelen guardar un registro (log) de todas las acciones realizadas, para poder revisar el uso del servicio y detectar problemas o usos indebidos.

MODOS DE CONEXIÓN FTP

Existen dos modos de conexión que determinan **cómo se establece la conexión** y **cómo se transfieren los datos**.

Comprender estos modos es fundamental para:

- Configurar correctamente firewalls
- Evitar problemas de conectividad
- Garantizar la seguridad y el rendimiento del servicio

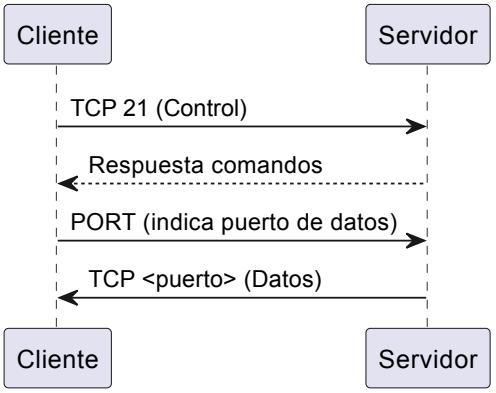
MODO ACTIVO

En el **modo activo**, la conexión funciona así:

1. El cliente abre una **conexión TCP al puerto 21** del servidor (canal de control) para enviar comandos.
2. Cuando el cliente quiere realizar una transferencia de datos le indica al servidor el puerto local al que debe conectarse (comando `PORT`).
3. El **servidor abre la conexión TCP hacia ese puerto del cliente** (canal de datos) para transferir la información.

Características:

- El cliente escucha un puerto para la conexión de datos.
- Puede causar problemas con firewalls o NAT, porque el servidor inicia la conexión de datos hacia el cliente.
- Tradicionalmente es el modo original del protocolo FTP.



MODO PASIVO

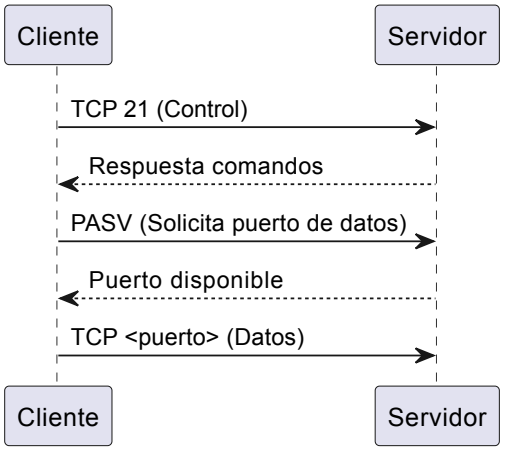
En el **modo pasivo**, la conexión de datos funciona de manera diferente al modo activo y está diseñada para facilitar el acceso desde redes protegidas por **firewalls o NAT**.

Su funcionamiento es el siguiente:

1. El cliente abre la **conexión TCP al puerto 21** del servidor (canal de control) para enviar comandos FTP.
2. Cuando el cliente necesita realizar una transferencia de datos se lo comunica al servidor (comando `PASV`).
3. El **servidor responde indicando un puerto disponible** para la transferencia de datos.
4. El cliente **abre la conexión TCP hacia ese puerto del servidor** y se realiza la transferencia de archivos.

Características:

- El cliente inicia ambas conexiones, lo que evita problemas con firewalls que bloqueen conexiones entrantes hacia el cliente.
- Muy utilizado en entornos modernos y corporativos.
- Garantiza mayor compatibilidad cuando se usan NAT o redes privadas.



COMANDOS Y RESPUESTAS DEL PROTOCOLO FTP

Los comandos FTP son instrucciones enviadas por el cliente al servidor a través del canal de control para realizar diversas operaciones relacionadas con la gestión y transferencia de archivos. Estos comandos son en texto plano y siguen una sintaxis específica definida por el protocolo FTP. Se les conoce comúnmente como **comandos en crudo** (*raw commands*) para diferenciarlos de las órdenes que los usuarios pueden introducir en los clientes FTP.

El servidor responde a cada comando con un código numérico y un mensaje que indica el resultado de la operación solicitada.

COMANDOS FTP (RAW COMMANDS)

La siguiente tabla recoge los principales comandos definidos por el protocolo FTP y su función real dentro de una sesión:

Comando	**Tipo**	**Descripción**
USER <usuario>	Autenticación	Indica el nombre de usuario con el que se desea iniciar sesión.
PASS <contraseña>	Autenticación	Envía la contraseña asociada al usuario.
QUIT	Control	Finaliza la sesión FTP y cierra la conexión de control.
PWD	Navegación	Muestra el directorio de trabajo actual en el servidor.
CWD	Navegación	Cambia el directorio de trabajo.
LIST	Transferencia	Solicita un listado detallado de archivos y directorios.
NLST	Transferencia	Solicita un listado simple de nombres de archivos.
RETR <archivo>	Transferencia	Descarga un archivo desde el servidor al cliente.
STOR <archivo>	Transferencia	Sube un archivo desde el cliente al servidor.

Comando	Tipo	Descripción
DELE <archivo>	Gestión	Elimina un archivo del servidor.
TYPE <tipo>	Configuración	Establece el tipo de transferencia (ASCII o binaria).
PORT <ip,puerto>	Modo activo	Indica al servidor el puerto del cliente para la conexión de datos.
PASV	Modo pasivo	Solicita al servidor que abra un puerto para la conexión de datos.
SYST	Información	Solicita información sobre el sistema operativo del servidor.
NOOP	Control	No realiza ninguna acción; se usa para mantener viva la conexión.

Estos comandos son independientes del sistema operativo y forman parte del estándar FTP.

RESPUESTAS DEL SERVIDOR FTP

Las respuestas del servidor FTP son mensajes enviados al cliente en respuesta a los comandos recibidos. Cada respuesta consta de un **código numérico de tres dígitos** seguido de un mensaje descriptivo. El código indica el resultado de la operación solicitada y se clasifica en varias categorías según su valor:

Categoría	Significado
1xx	Respuesta preliminar (acción en curso)
2xx	Acción completada con éxito
3xx	Se necesita información adicional
4xx	Error temporal
5xx	Error permanente

La siguiente tabla recoge algunos de los códigos de respuesta más habituales en una sesión FTP:

Código	Significado
220	Servicio FTP listo
331	Usuario correcto, se necesita contraseña
230	Autenticación completada
150	Se va a abrir la conexión de datos
226	Transferencia completada
530	Autenticación fallida
550	Archivo no disponible o acceso denegado

EL COMANDO PORT

El comando `PORT` es utilizado en el **modo activo** de FTP para indicar al servidor la dirección IP y el puerto TCP en el que el cliente está escuchando para la conexión de datos. La sintaxis del comando es la siguiente:

```
PORT h1,h2,h3,h4,p1,p2
```

Donde:

- `h1,h2,h3,h4`: representan los cuatro octetos de la dirección IP del cliente.
- `p1,p2`: representan el puerto TCP en el que el cliente está escuchando, calculado como `p1*256 + p2`.

Comando PORT

En el siguiente ejemplo se muestra un comando `PORT` típico:

```
PORT 192,168,1,50,8,69
```

En este ejemplo, la dirección IP del cliente es `192.168.1.50` y el puerto es `2077` (8*256 + 69). Tras recibir este comando, el servidor FTP intentará establecer una conexión TCP al puerto `2077` de la dirección `192.168.1.50`.

EL COMANDO PASV

El comando `PASV` es utilizado en el **modo pasivo** de FTP para solicitar al servidor que abra un puerto TCP para la conexión de datos. Cuando el cliente envía el comando `PASV`, el servidor responde con un mensaje que incluye la dirección IP y el puerto que el cliente debe usar para establecer la conexión de datos. La respuesta del servidor tiene la siguiente estructura:

```
227 Entering Passive Mode (h1,h2,h3,h4,p1,p2)
```

Donde:

- `h1,h2,h3,h4`: representan los cuatro octetos de la dirección IP del servidor.
- `p1,p2`: representan el puerto TCP que el servidor ha abierto para la conexión de datos, calculado como `p1*256 + p2`.

Comando PASV

En el siguiente ejemplo se muestra un comando `PASV` típico:

```
PASV
```

En este ejemplo, el servidor responde con un mensaje como `227 Entering Passive Mode (192,168,1,50,8,69)`, indicando que el servidor está escuchando en la dirección `192.168.1.50` y el puerto `2077` (8*256 + 69). Tras recibir la respuesta del servidor, el cliente FTP intentará establecer una conexión TCP al puerto `2077` de la dirección `192.168.1.50`.

USO DE LOS COMANDOS FTP

Los comandos FTP se envían a través del canal de control utilizando una conexión TCP establecida entre el cliente y el servidor. Cada comando debe ser seguido por una secuencia de caracteres de nueva línea (CRLF) para indicar el final del comando. Herramientas como `telnet` y/o netcat (`nc`) pueden utilizarse para enviar comandos FTP manualmente y observar las respuestas del servidor.

Sesión FTP en modo activo

El siguiente ejemplo ilustra una sesión FTP en modo activo utilizando `telnet` para la conexión de control y `netcat` para la conexión de datos. Nosuestra cómo un cliente FTP se conecta a un servidor (`192.168.10.251`), autentica al usuario (`jose` con contraseña `pepe`), solicita un listado de archivos (`LIST`) y cierra la sesión.

Conexión de control con Telnet

Cliente-LX:~# telnet 192.168.10.251 21
Connected to 192.168.10.251
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (`tim.kosse[at]filezilla-project.org`)
220 Please visit `filezilla-project.org`
USER jose
331 Password required for jose
PASS pepe
230 Logged on
PORT 192,168,10,66,200,1
200 Port command successful
LIST
150 Opening data channel for directory listing of "/"
226 Successfully transferred "/"
QUIT
221 Goodbye
Connection closed by foreign host
Cliente-LX:~#

Antes de ejecutar el comando `LIST` , es necesario abrir una conexión de datos utilizando `netcat` en el puerto especificado en el comando `PORT` (en este caso, puerto `51201` , que se calcula como `200*256 + 1`).

Conexión de datos con netcat

Cliente-LX:~# nc -l -p 51201
X1-L1:*# nc -l -p 51201
-rw-r-xr-x 1 ftp ftp 136774728 Jan 12 09:59 Autofirma_64_v1_9_installer.exe
-rw-r--r-- 1 ftp ftp 89 Jan 17 23:44 datos.txt
-rw-r-xr-x 1 ftp ftp 2241216 Jan 17 22:18 filezilla-server-0-9-60-2.exe

Sesión FTP en modo pasivo

El siguiente ejemplo ilustra una sesión FTP en modo pasivo utilizando `telnet` para la conexión de control y `telnet` nuevamente para la conexión de datos. Muestra cómo un cliente FTP se conecta a un servidor (`192.168.10.251`), autentica al usuario (`jose` con contraseña `pepe`), descarga un archivo (`RETR datos.txt`) y cierra la sesión.

Conexión de control con Telnet

Cliente-LX:~# telnet 192.168.10.251 21
Connected to 192.168.10.251
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (`tim.kosse@filezilla-project.org`)
220 Please visit `https://filezilla-project.org/`
USER jose
331 Password required for jose
PASS pepe
230 Logged on
PASV
227 Entering Passive Mode (192,168,10,251,230,126)
RETR datos.txt
150 Opening data channel for file download from server of "/datos.txt"
226 Successfully transferred "/datos.txt"
QUIT
221 Goodbye
Connection closed by foreign host
Cliente-LX:~#

Antes de ejecutar el comando `RETR` , es necesario abrir una conexión de datos utilizando `telnet` en la dirección IP y puerto especificados en la respuesta al comando `PASV` (en este caso, dirección `192.168.10.251` y puerto `58878` , que se calcula como `230*256 + 126`).

Conexión de datos con Telnet

Cliente-LX:~# telnet 192.168.10.251 59006
Connected to 192.168.10.251
Fichero de texto contiene 3 líneas
Esta es la segunda línea
Esta es la tercera línea
Connection closed by foreign host
Cliente-LX:~#

SEGURIDAD DEL PROTOCOLO FTP

El protocolo FTP presenta varios **problemas de seguridad** debido a que fue diseñado sin tener en cuenta las amenazas actuales de las redes.

PRINCIPALES PROBLEMAS DE SEGURIDAD

- **Credenciales en texto plano**:
  El usuario y la contraseña se envían sin cifrar, por lo que pueden ser capturados fácilmente.
- **Datos sin cifrar**:
  Los archivos transferidos pueden ser interceptados y leídos por terceros.
- **Vulnerabilidad a ataques de red**:
  FTP es susceptible a ataques como interceptación de tráfico o ataques de fuerza bruta.
- **Uso de múltiples puertos**:
  El uso de varios puertos dificulta la configuración segura de firewalls y aumenta la superficie de ataque.

MEJORAS DE SEGURIDAD

Para solucionar o reducir estos problemas se pueden aplicar las siguientes mejoras:

- **Uso de FTPS**:
  Añade cifrado mediante TLS a FTP, protegiendo credenciales y datos.
- **Uso de SFTP**:
  Protocolo alternativo basado en SSH, más seguro y recomendado.
- **Restricción de accesos**:
  Limitar permisos de usuarios y desactivar el acceso anónimo si no es necesario.
- **Contraseñas seguras y control de intentos**:
  Evitar ataques de fuerza bruta.
- **Monitoreo y auditoría**:
  Registrar actividades para detectar usos indebidos.
- **Configuración adecuada de firewalls**:
  Permitir solo el tráfico necesario y proteger los puertos utilizados por FTP.