

EL SERVICIO FTP

 [Descargar PDF](#)

- [EL SERVICIO FTP](#)
- ▼ [Protocolo FTP \(File Transfer Protocol\)](#)
 - [Índice](#)
- ▼ [Protocolo FTP \(File Transfer Protocol\)](#)
 - [Índice](#)
 - [1. Introducción al protocolo FTP](#)
 - [2. Contexto histórico y estandarización](#)
 - [3. Arquitectura y principios de funcionamiento](#)
- ▼
 - [4. Canales de comunicación: control y datos](#)
 - [Canal de control](#)
 - [Canal de datos](#)
- ▼
 - [5. Modos de conexión FTP: activo y pasivo](#)
 - [Modo activo](#)
 - [Modo pasivo](#)
- ▼
 - [6. Autenticación y tipos de acceso](#)
- ▼
 - [7. Comandos y clientes FTP](#)
 - [7.1 Clientes FTP](#)
 - [7.2 Comandos FTP \(raw commands\)](#)
 - ▼ [7.3 Respuestas del servidor FTP](#)
 - [Categorías de códigos de respuesta](#)
 - [Códigos habituales](#)
 - [8. Transferencia de archivos en FTP](#)
- ▼
 - [9. Análisis de sesiones FTP y funcionamiento interno](#)
 - **Diagrama mejorado de sesiones FTP**
 - **Diagrama secuencial detallado mejorado (Telnet + Netcat)**
- [10. Limitaciones técnicas y problemas habituales](#)
- [11. Consideraciones de seguridad](#)

▼ Ejemplos prácticos

- Modo activo (Active FTP)
- Modo pasivo (Passive FTP)

Protocolo FTP (File Transfer Protocol)

Índice

1. Introducción al protocolo FTP
2. Contexto histórico y estandarización
3. Arquitectura y principios de funcionamiento
4. Canales de comunicación: control y datos
5. Modos de conexión FTP: activo y pasivo
6. Autenticación y tipos de acceso
7. Comandos y respuestas del protocolo FTP
8. Transferencia de archivos en FTP
9. Análisis de sesiones FTP y funcionamiento interno
10. Limitaciones técnicas y problemas habituales
11. Consideraciones de seguridad
12. Protocolos alternativos y evolución

Protocolo FTP (File Transfer Protocol)

Índice

1. Introducción al protocolo FTP
2. Contexto histórico y estandarización
3. Arquitectura y principios de funcionamiento
4. Canales de comunicación: control y datos
5. Modos de conexión FTP: activo y pasivo
6. Autenticación y tipos de acceso

1. Introducción al protocolo FTP

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.

2. 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.

3. Arquitectura y principios de funcionamiento

FTP sigue una arquitectura **cliente-servidor** claramente definida. El cliente es siempre quien inicia la comunicación, mientras que el servidor permanece a la espera de conexiones entrantes y atiende las peticiones recibidas.

Una de las características más importantes del protocolo FTP es que utiliza **dos conexiones TCP independientes** durante una sesión:

- Una conexión para el **control de la sesión**.
- Una conexión para la **transferencia de datos**.

Esta separación permite que los comandos y las transferencias de archivos se gestionen de forma independiente. Mientras el canal de control permanece abierto durante toda la sesión, el canal de datos se abre y se cierra cada vez que es necesario transferir información.

Aunque este diseño aporta flexibilidad, también introduce complejidad adicional, especialmente en redes modernas con firewalls y dispositivos NAT.

4. Canales de comunicación: control y datos

Canal de control

El canal de control se establece al inicio de la sesión FTP y normalmente utiliza el **puerto TCP 21** del servidor. Permanece abierto durante toda la sesión y se utiliza para el intercambio de comandos y respuestas entre el cliente y el servidor.

Toda la comunicación del canal de control se realiza en **texto ASCII**, lo que permite observar directamente el funcionamiento del protocolo mediante herramientas como `telnet`. Los comandos enviados por el cliente y las respuestas del servidor siguen un formato textual claramente definido.

Canal de datos

El canal de datos se utiliza exclusivamente para la transferencia de información, como archivos o listados de directorios. A diferencia del canal de control, esta conexión no es persistente, sino que se establece y se cierra para cada operación.

El puerto utilizado para el canal de datos no es fijo y depende del modo de conexión que se esté utilizando (activo o pasivo). Esta característica es una de las principales fuentes de problemas de FTP en redes protegidas.

5. Modos de conexión FTP: activo y pasivo

FTP define dos modos distintos para establecer el canal de datos. La diferencia fundamental entre ellos es **quién inicia la conexión de datos**.

Modo activo

En el **modo activo**, el cliente informa al servidor del puerto en el que va a escuchar para la transferencia de datos. A partir de ese momento, es el **servidor** quien inicia la conexión de datos desde su **puerto TCP 20** hacia el cliente.

Este modo refleja el diseño original del protocolo, pero presenta serios inconvenientes en redes modernas, ya que obliga al cliente a aceptar conexiones entrantes, algo que suele estar bloqueado por firewalls o NAT.

Modo pasivo

En el **modo pasivo**, el cliente solicita al servidor que abra un puerto para la transferencia de datos. El servidor responde indicando qué puerto ha abierto, y es el **cliente** quien inicia la conexión de datos hacia ese puerto.

Este modo es el más utilizado en la actualidad, ya que todas las conexiones son iniciadas por el cliente, lo que facilita su funcionamiento en redes protegidas.

6. Autenticación y tipos de acceso

FTP permite diferentes formas de acceso al servidor:

- **Acceso autenticado**, mediante un nombre de usuario y una contraseña definidos en el sistema.
- **Acceso anónimo**, en el que el usuario se identifica normalmente como `anonymous` y no requiere una cuenta personal.

Un aspecto crítico del protocolo FTP es que las credenciales se transmiten **en texto plano**, sin ningún tipo de cifrado. Esto significa que cualquier atacante con acceso al tráfico de red podría capturar fácilmente nombres de usuario y contraseñas.

Esta falta de seguridad es una de las principales razones por las que FTP ha sido reemplazado en la práctica por protocolos más modernos y seguros.

7. Comandos y clientes FTP

El protocolo FTP define un conjunto de comandos en texto plano que se intercambian entre el cliente y el servidor a través del canal de control. Estos comandos forman parte del **protocolo FTP propiamente dicho** y se conocen como **comandos en crudo** (*raw commands*).

Sin embargo, en la práctica, los usuarios no suelen trabajar directamente con estos comandos. Para ello se utilizan **clientes FTP**, que son aplicaciones diseñadas para facilitar el uso del protocolo.

7.1 Clientes FTP

Un **cliente FTP** es un programa que implementa el protocolo FTP y ofrece al usuario una interfaz más sencilla. Existen clientes FTP de distintos tipos:

- Clientes de línea de comandos.
- Clientes gráficos.
- Clientes integrados en navegadores o gestores de archivos.

Estos clientes **no inventan nuevos comandos**, sino que actúan como intermediarios: traducen las órdenes del usuario a los comandos FTP reales definidos por el protocolo.

Por ejemplo:

- El usuario escribe `ls` → el cliente envía `LIST`.
- El usuario escribe `get archivo.txt` → el cliente envía `RETR archivo.txt`.
- El usuario escribe `put archivo.txt` → el cliente envía `STOR archivo.txt`.

Cuando se analiza una sesión FTP con herramientas como `telnet`, `nc` o Wireshark, **solo aparecen los comandos raw**, nunca los comandos abreviados del cliente.

7.2 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	Autenticación	Indica el nombre de usuario con el que se desea iniciar sesión.
PASS	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	Transferencia	Descarga un archivo desde el servidor al cliente.
STOR	Transferencia	Sube un archivo desde el cliente al servidor.
DELE	Gestión	Elimina un archivo del servidor.
TYPE	Configuración	Establece el tipo de transferencia (ASCII o binaria).
PORT	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.

7.3 Respuestas del servidor FTP

Cada comando enviado por el cliente genera una respuesta del servidor formada por un **código numérico de tres dígitos** y un mensaje textual.

Categorías de códigos de respuesta

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

Códigos habituales

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

8. Transferencia de archivos en FTP

La transferencia de archivos en FTP se realiza siempre a través del **canal de datos**, que es independiente del canal de control. Cada operación de transferencia implica la apertura de una conexión TCP adicional que se cierra al finalizar la operación.

FTP define dos tipos principales de transferencia:

- **ASCII**, pensada para archivos de texto, con posible conversión de finales de línea.
- **Binaria (Image)**, utilizada para archivos binarios, sin modificación de los datos.

Seleccionar el tipo adecuado es esencial para evitar la corrupción de archivos.

9. Análisis de sesiones FTP y funcionamiento interno

Una **sesión FTP** se establece cuando un cliente se conecta a un servidor FTP. El funcionamiento interno implica varias fases:

1. Conexión inicial (handshake)

- El cliente abre una conexión TCP al **puerto 21** del servidor (canal de control).
- El servidor responde con un **mensaje de bienvenida** indicando que está listo.

2. Autenticación

- El cliente envía su **usuario** (`USER`) y **contraseña** (`PASS`).
- El servidor valida las credenciales y establece la sesión.

3. Comandos y transferencias

- A través del canal de control, el cliente envía **comandos FTP** (`LIST`, `RETR`, `STOR`).
- Para transferir archivos, se abre el **canal de datos**, que puede ser:
 - **Activo**: el servidor se conecta al cliente.
 - **Pasivo**: el cliente se conecta al servidor.

4. Finalización de la sesión

- El cliente envía `QUIT`.
- El servidor libera los recursos asociados.

Diagrama mejorado de sesiones FTP

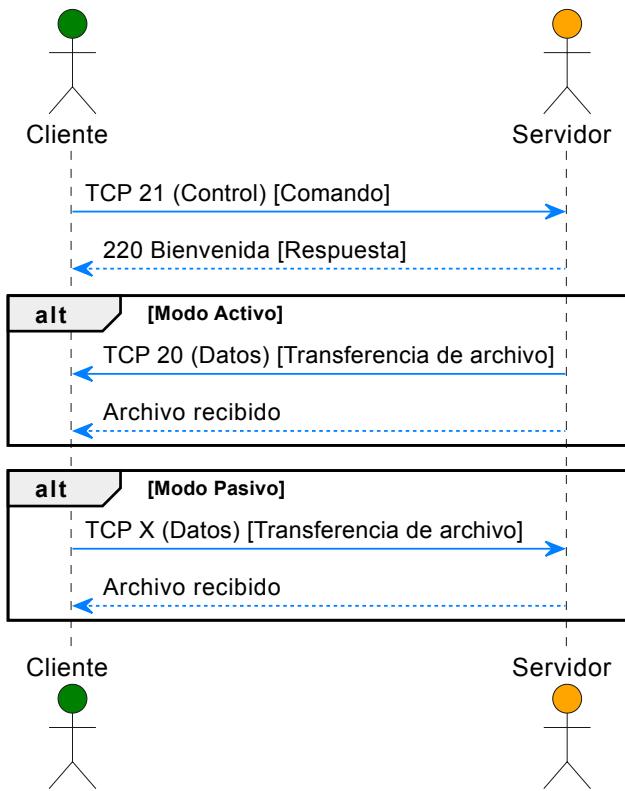
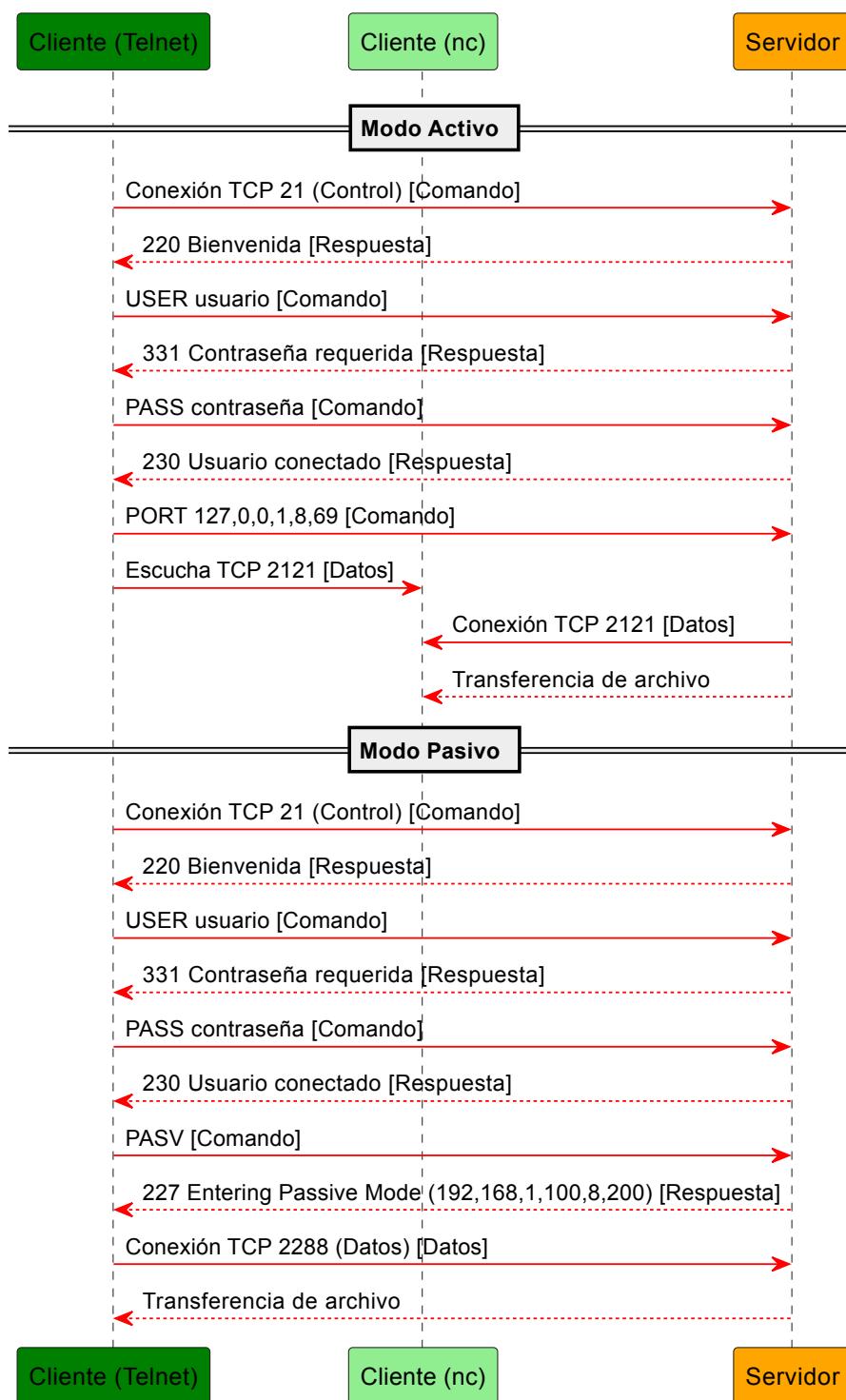


Diagrama secuencial detallado mejorado (Telnet + Netcat)



10. Limitaciones técnicas y problemas habituales

- **Transmisión en texto plano:** usuario, contraseña y archivos sin cifrar.
- **Problemas con firewalls y NAT:**
 - Modo activo puede fallar si el cliente está detrás de NAT/firewall.
- **Gestión de permisos limitada:** sin control granular por archivo o carpeta.
- **Ineficiencia en operaciones pequeñas o múltiples:** cada transferencia requiere abrir y cerrar el canal de datos.

- **Falta de verificación de integridad:** FTP no comprueba automáticamente la corrupción de archivos durante la transferencia.

11. Consideraciones de seguridad

- Evitar usar FTP estándar en redes públicas.
- Usar alternativas seguras:
 - **FTPS (FTP + SSL/TLS)**
 - **SFTP (FTP sobre SSH)**
- Configurar correctamente **firewalls** según el modo (activo o pasivo).
- **Restringir acceso** solo a usuarios y directorios autorizados.
- Registrar logs de actividad para **auditoría y detección de anomalías**.
- Usar **contraseñas fuertes** y, si es posible, **cifrado de archivos** sensibles.

Ejemplos prácticos

Modo activo (Active FTP)

Conexión de control con Telnet

```
telnet ftp.ejemplo.com 21
# 220 Servidor FTP listo
USER usuario
PASS contraseña
230 Usuario conectado
```

Conexión de datos con Netcat

```
# En cliente
nc -l 2121
# En sesión de control
PORT 127,0,0,1,8,69
RETR archivo.txt
```

Modo pasivo (Passive FTP)

Conexión de control con Telnet

```
telnet ftp.ejemplo.com 21
USER usuario
PASS contraseña
PASV
# 227 Entering Passive Mode (192,168,1,100,8,200)
```

Conexión de datos con Telnet (segunda sesión)

```
telnet 192.168.1.100 2288
RETR archivo.txt
```

Notas finales

- Los diagramas mejorados muestran **visualmente la diferencia entre comandos y transferencia de datos**.
- Esta versión con colores facilita la comprensión del flujo en **modo activo y pasivo** y es útil para presentaciones o documentación técnica.