MEDAC
Instituto Oficial de Formación Profesional

DESPLIEGUE DE APLICACIONES WEB
TÉCNICO EN DESARROLLO DE APLICACIONES WEB

El servicio de transferencia de archivos

09

ÍNDICE

 ISBN: 978-84-18983-26-9

OBJETIVOS

Conocer qué es el servicio de transferencia de archivos.

Conocer los diferentes protocolos utilizados para transferir archivos.

Realizar pruebas de funcionamiento en los diferentes sistemas.

/ 1. Introducción y contextualización práctica

En esta unidad, estudiaremos qué son y para qué sirven los protocolos de transferencia de archivos.

Para comprender mejor el funcionamiento desde el lado del cliente, analizaremos las características de cada uno de estos protocolos y aprenderemos a establecer conexiones con un servidor remoto.

Planteamiento del caso práctico inicial

A continuación, vamos a plantear un caso práctico a través del cual podremos aproximarnos de forma práctica a la teoría de este tema.

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema, encontrarás su resolución en el apartado Resumen y Resolución del caso práctico.

Fig. 1. Administración de servidores de aplicaciones web

Audio intro. "Caso práctico inicial"

https://bit.ly/3hvuMxh

/ 2. El servicio de transferencia de archivos en el despliegue de una aplicación web

Cuando se trata de **instalar o mantener un servidor**, independientemente del papel que desempeñe, tendremos la necesidad de intercambiar información con dicha máquina. Como consecuencia, la transferencia de archivos en un servidor se convierte en algo imprescindible, tanto para tareas de mantenimiento y copias de seguridad, como por motivos de desarrollo y despliegue de aplicaciones. Para el adecuado desarrollo o implantación de una aplicación web, será necesario subir, actualizar o modificar archivos. En el caso de los servidores virtuales, asumimos que conectar una unidad USB para intercambiar información no es una opción.

Para esta finalidad existen **protocolos y servicios de transferencia** de datos. Estos servicios nos ofrecen la posibilidad de establecer una conexión segura y eficiente, implementando los mecanismos de autenticación y seguridad necesarios para realizar operaciones con las carpetas y archivos que se encuentren en los servidores.

Como es habitual, los protocolos y sistemas utilizados para transferir información entre el cliente y el servidor evolucionan con el tiempo, apareciendo nuevos protocolos que mejoran la **seguridad y la eficiencia de la conexión.** Entre los diferentes servicios de transferencia que podemos implementar, nos centraremos en tres: **FTP, SCP-SSH y Webdav.** Para empezar, vamos a conocer el funcionamiento básico, modos de conexión y comandos necesarios para entender cada uno de estos protocolos desde el punto de vista del cliente.

/ 3. FTP: Fundamentos y modos de conexión

FTP (File Transfer Protocol) es un protocolo de transferencia de archivos que se sitúa en la capa de aplicación del modelo OSI. FTP utilizando los puertos 20 y 21 del protocolo TCP para establecer las conexiones basándose en el modelo **cliente-servidor**. Las conexiones FTP se iniciarán siempre en el cliente.

El protocolo FTP está diseñado para aprovechar el ancho de banda y transmitir información a la **máxima velocidad** que permita la red. Además, este protocolo añade mecanismos de **seguridad y autenticación** para evitar accesos no autorizados.

Como desventaja encontramos que todas las conexiones se realizan en texto plano, por lo que, aunque añada mecanismos de seguridad, **FTP no es un protocolo seguro.**

El protocolo FTP implementa dos modos de funcionamiento:

- **Modo activo:** es el modo nativo de FTP. El cliente envía al servidor la IP y el puerto por el que se establecerá la conexión. Una vez que el servidor acepte la petición se iniciará la transferencia de datos (Fig. 2 superior).
- **Modo pasivo:** el cliente envía una solicitud de conexión pasiva y el servidor contesta con la IP y los puertos que se utilizarán en la conexión. A continuación, el cliente establecerá la sesión con los parámetros indicados. (Fig.2 Inferior)

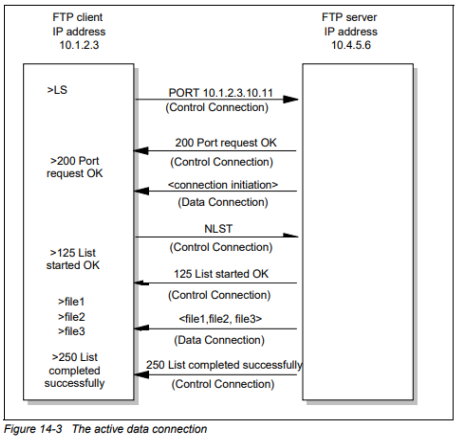

Figure 14-3 The active data connection

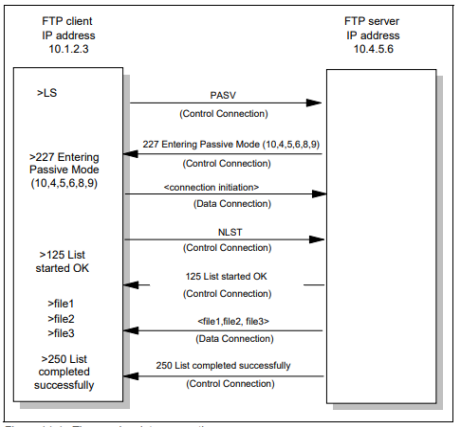

Fig. 2. Modos de conexión FTP. https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf

/ 4. FTP: tipos de usuarios y formatos de archivos

a. **Tipos de usuarios**: Como vimos en el apartado anterior, FTP introduce un sistema de autenticación para gestionar el control de accesos.

A la hora de autenticar contra un servidor FTP podemos encontrar dos opciones:

- **Usuarios de sistema:** para iniciar sesión se utilizarán los usuarios creados en el sistema operativo del servidor FTP. Los usuarios registrados pueden tener acceso a un directorio home o carpeta personal. Además, es posible configurar cuotas, es decir, asignar una cantidad de espacio de disco para cada usuario.

- **Usuarios anónimos:** en FTP es posible configurar carpetas para acceder de manera **pública**. De esta manera, se utilizará el usuario **anonymous** para establecer la conexión. Habitualmente, las conexiones anónimas están restringidas a solo lectura, pero en caso contrario, y si el servicio FTP estuviera expuesto a internet, cualquier persona podría subir archivos maliciosos.

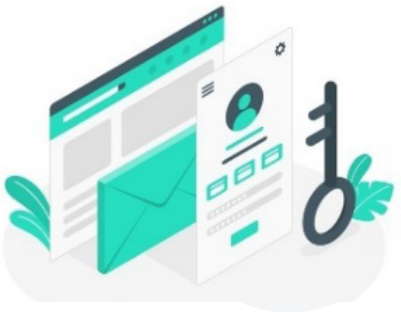

Fig. 3. Administración de servidores de aplicaciones web

b. **Formatos de archivos**: El protocolo FTP utiliza diferentes modos de transmisión según el formato del archivo que queramos transmitir. Se diferencia entre archivos Binarios y ASCII:

- Los archivos **binarios** contienen información codificada de manera que solo pueda interpretarse utilizando el software adecuado.

Extensión	Descripción
zip, rar, gz	Ficheros comprimidos
jpg, png, gif	Ficheros de imágenes
mpg, avi, mp3	Ficheros multimedia
doc, xls, ppt	Ficheros del paquete Microsoft Office

Fig. 4. Ejemplo de un log de eventos de Apache Tomcat.

- Los archivos **ASCII** solo contienen caracteres de texto, incluyendo saltos de línea.

Extensión	Descripción
txt	Ficheros de texto modificables con un editor de texto
html	Páginas web
js	Código Javascript
java	Código fuente Java
xml	Ficheros en un lenguaje basado en etiquetas similares a las empleadas en HTML

Fig. 5. Ejemplo de un log de eventos de Apache Tomcat

Audio 1. "Ataques a servidores FTP – Bounce Attack"
https://bit.ly/3aRCgIw

/ 5. Caso práctico 1: "Conexión anónima a un servidor FTP"

Planteamiento: Andrés está desarrollando una aplicación web para una empresa y necesita descargar sus imágenes corporativas para poder utilizarlas.

El departamento de sistemas de dicha empresa, ha preparado un servidor FTP para que pueda descargar los logos y todo lo necesario. En el correo le han comentado que, como no hay información sensible, lo han configurado para que pueda acceder de forma anónima.

Los datos que ha recibido son los siguientes:

"Buenas tardes,

Ya tiene disponible todos los archivos que ha solicitado. Puede acceder a ellos conectándose de manera anónima a nuestro servidor FTP. La dirección es ftp.empresapriv.com"

Nudo: ¿Qué datos necesitará Andrés para iniciar sesión? ¿Necesitará algún software adicional?

Desenlace: Andrés podrá conectar al servidor de la empresa utilizando un navegador web.

Al especificar que permite conexiones anónimas podrá utilizar el usuario **Anonymous** con contraseña guest, o utilizar la combinación de usuario y contraseña ftp.

Una vez que inicie sesión podrá descargar el fichero haciendo clic.

Fig. 6. Ejemplo de un log de eventos de Apache Tomcat.

/ 6. FTP: modos de conexión

a. **Línea de comandos**

En las conexiones FTP, el cliente se comunica con el servidor utilizando una serie de comandos de consola para transmitir las órdenes. A continuación, listaremos algunos de los comandos más utilizados:

Comando	Descripción	Comando	Descripción
Open	Conecta con un servidor FTP	Mkdir	Crea una carpeta
Quit	Desconecta del servidor cerrando la sesión FTP	Rmdir	Elimina una carpeta
Close	Desconecta del servidor sin cerrar la sesión	Get	Copia un archivo del servidor FTP
Ls	Lista el directorio	Put	Copia un archivo al servidor FTP
Pwd	Muestra la ruta actual	Binary	Activa la transferencia de archivos binarios
Cd	Cambia de directorio	ascii	Activa la transferencia de archivos en modos ascii
Delete	Elimina un archivo		

Tabla 1. Comandos utilizados en las conexiones FTP

b. **Cliente FTP gráfico**

Los gestores de conexiones FTP con entorno gráfico facilitan el proceso de conexión con el servidor remoto. Estos gestores enmascaran los comandos que vimos en el apartado anterior con acciones gráficas.

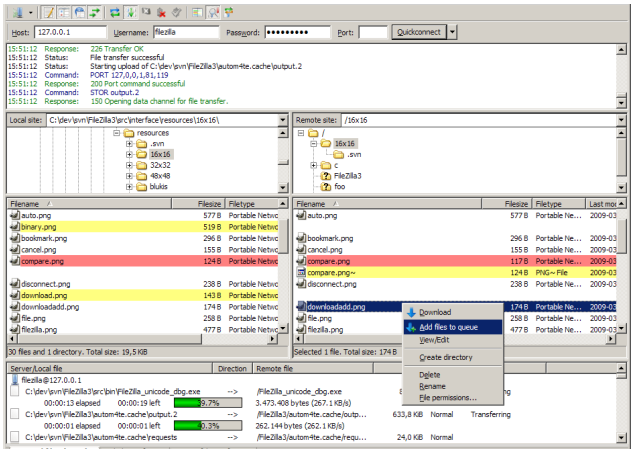

Fig. 7. Filezilla Client - https://filezilla-project.org/images/screenshots/fz3_win_main.png

Video 1. "Conexión ftp por comandos"

https://bit.ly/2FQgoSd

/ 7. FTP SSL

Como veremos en el tema siguiente, el protocolo FTP transmite los datos de inicio de sesión en texto plano. Para mejorar la seguridad de la conexión se cifran los datos enviados mediante un certificado SSL.

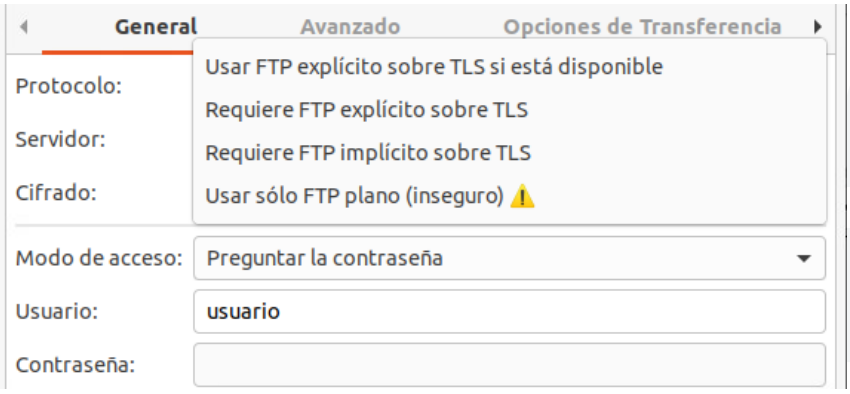

Fig. 8. Tipos de cifrado disponibles en el cliente FTP Filezilla para Ubuntu.

Al utilizar SSL sobre FTP encontramos dos modos de funcionamiento:

- **FTPS Implícito:** en este modo la conexión FTP no se establece contra el puerto por defecto, es decir, el puerto TCP 21. En su lugar se ataca al puerto **TCP 990** y cliente y servidor realizan una negociación TSL para asegurar la comunicación.

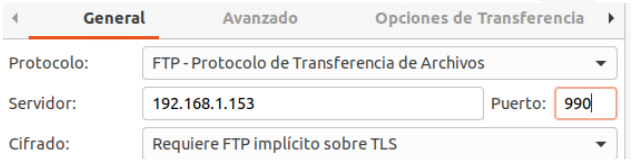

Fig. 9. Ejemplo de una conexión FTP utilizando TSL implícito.

- **FTPS Explícito:** en este modo el cliente FTP debe solicitar al servidor utilizar el protocolo SSL y establecerá la conexión al puerto 21.

  Para utilizar las opciones de conexión seguras, el servidor remoto debe estar configurado correctamente para aceptar SSL.

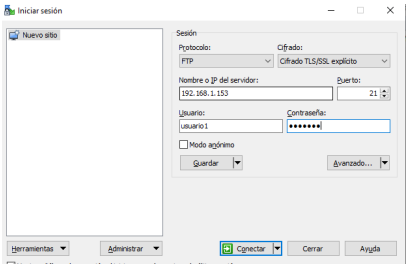

Fig. 10. Ejemplo de una conexión FTP TSL explicita al puerto 21.

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

Fig. 11. Ejemplo de configuración TSL en servidor FTP.

/ 8. SCP

Otra forma de transferir información entre el cliente y el servidor es utilizar el protocolo SCP.

SCP **(Secure Copy Protocol)** es un protocolo de transferencia de datos que utiliza el protocolo SSH (Secure Shell).

Este protocolo nos permite intercambiar archivos mediante línea de comandos, aunque también existen aplicaciones con GUI que nos facilita el trabajo.

a. **Conexión SCP por consola**

Para enviar o recibir archivos utilizando SCP necesitamos conocer exactamente la **ubicación del servidor remoto.**

La forma de utilizar SCP es parecida a establecer una conexión SSH contra un servidor:

```
scp /ruta/local/archivo  usuario@host_remoto:/ruta/al/archivo/archivo
```

Código 1. Ejemplo de una conexión scp para subir un fichero a un servidor.

Para entender mejor el funcionamiento de SCP, veremos el siguiente ejemplo:

Queremos descargar de un servidor remoto con dirección IP 192.168.1.250 un archivo que se encuentra en /home/usr1/archivo1.tar.gz. El comando que deberemos ejecutar sería:

```
scp usuario@192.168.1.250:/home/usr1/archivo1.tar.gz .
```

Código 2. Ejemplo de una conexión scp para descargar un fichero de un servidor..

b. **Conexión SCP con GUI**

Para trabajar a través de una interfaz, utilizaremos **aplicaciones externas** como podrían ser WinSCP para Windows o Filezilla. Estas aplicaciones nos proporcionan un entorno agradable y fácil de usar, ya que nos ofrecen unos entornos de trabajo amigables que pueden sustituir perfectamente a los comandos si no se tiene mucha experiencia con los mismos. La apariencia visual de ambas herramientas sería esta:

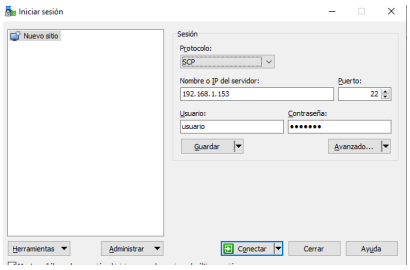

Fig. 12. Conexión SCP utilizando WinSCP para Windows.

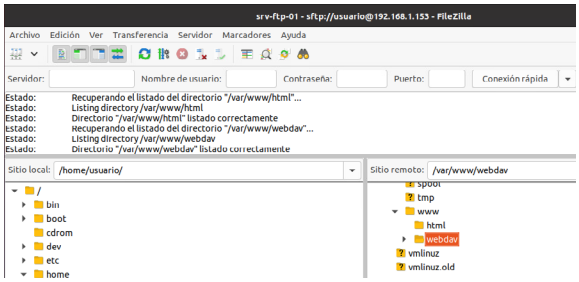

Fig. 13. Conexión SCP utilizando Filezilla para Ubuntu.

/ 9. Caso práctico 2: "Backup de una aplicación web"

Planteamiento: Ana debe realizar una copia de seguridad de archivos en un servidor web. Para ello está investigando qué sistema será el más seguro, eficiente y fácil de utilizar.

El servidor web tiene expuesto a internet el puerto 22 (SSH), permitiendo conexiones solo desde las direcciones IP de la empresa de Ana y del propio cliente; por este motivo, Ana ha pensado en utilizar algún protocolo basado en SSH.

Nudo: ¿Qué solución puede implementar para realizar copias de seguridad utilizando el protocolo SSH? ¿Podrá disponer de un entorno gráfico?

Solo necesita hacer backup de la carpeta /var/www/webapi1 que tiene la siguiente estructura:

```
usuario@srv-ftp-01:/var/www/webapi1$ tree

|-- app.js
|-- css
|   |-- styles.css
|   |-- styles-js.css
|-- images
|   |-- img1.jpg
|   |-- logo.png
|-- index.html
|-- login.php
```

Código 3. Estructura de carpetas.

Desenlace: Ana podrá utilizar el protocolo SSH para utilizar algún software de *backup* como podría ser cobian backup, o directamente descargar la copia usando línea de comandos.

En caso de necesitar una copia de seguridad de manera ocasional, podrá conectarse por SSH, comprimir el directorio en un archivo tar.gz y usar SCP para descargarlo:

```
usuario@srv-ftp-01:/var/www$ sudo tar cvzf webapi1.tar.gz
webapi1/
webapi1/
webapi1/login.php
webapi1/index.html
webapi1/app.js
webapi1/css/
webapi1/css/styles-js.css
webapi1/css/styles.css
webapi1/images/
webapi1/images/img1.jpg
webapi1/images/logo.png
```

Código 4. Compresión de la carpeta desde el servidor

```
usuario@MDHP01:/mnt/c/Users/Miguel$ scp
usuario@192.168.1.153:/var/www/webapi1.tar.gz .
usuario@192.168.1.153's password:
webapi1.tar.gz
100%  315   157.0KB/s   00:00
usuario@MDHP01:/mnt/c/Users/Miguel$
```

Código 5. Descarga del archivo comprimido usando scp

/ 10. Webdav

El protocolo Webdav establece un servidor de transferencia de archivos basado en el protocolo HTTP. Esto permite conectarse a un servidor remoto mediante el navegador web o accediendo desde el gestor de archivos independientemente del sistema operativo que utilicemos.

Si accedemos al servidor de archivos utilizando un navegador web, será tan sencillo como escribir la dirección de dicho servidor. Si la autenticación estuviera activada nos pediría usuario y contraseña:

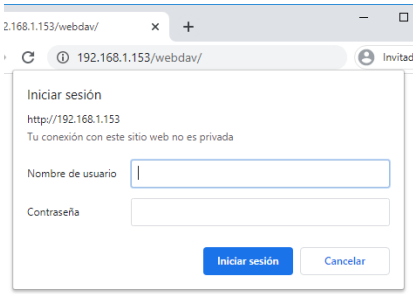

Fig. 14. Inicio de sesión en un servidor Webdav usando un navegador web.

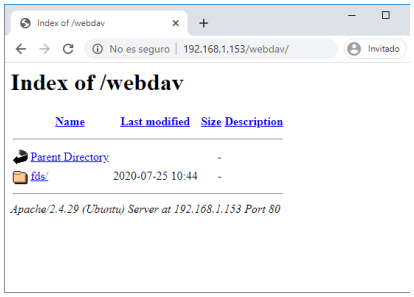

Fig. 15. Listado de archivos en un servidor Webdav usando un navegador web.

Por otro lado, para acceder a recursos compartidos mediante WebDav en sistemas Windows necesitaremos conectar una **unidad de red**.

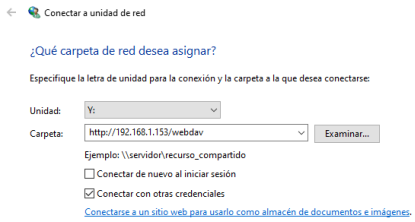

Fig. 16. Conexión de unidad de red al servidor Webdav usando un sistema operativo Windows.

En sistemas basados en Linux, utilizaremos la opción **conectar al servidor.** En este caso, utilizaremos el esquema **dav://** para indicar que queremos utilizar el protocolo Webdav por el puerto 80, o **davs://** para acceder mediante https.

Fig. 17. Conexión de unidad de red al servidor Webdav usando un sistema operativo Linux.

Vídeo 2. « Conectar a carpeta Webdav en Ubuntu »
https://bit.ly/2YwAWWA

/ 11. Resumen y resolución del caso práctico de la unidad

En esta unidad, hemos hablado de los diferentes sistemas de **transferencia de información**, fundamentos teóricos y como utilizarlos.

Respecto a **FTP**, hemos introducido los diferentes tipos de usuarios, formatos de archivos y el listado de comandos. Estos comandos nos permitirán establecer conexiones y poder realizar operaciones desde una consola.

Finalmente, hemos hablado de **SCP**, un protocolo basado en SSH que nos permite intercambiar datos con servidores remotos de forma segura, y de ***Webdav***, que utiliza servicios webs para compartir información.

Fig. 18. Intercambio de información utilizando protocolos de transferencia.

Resolución del caso práctico de la unidad

En el caso práctico inicial, se plantea la siguiente situación: Carlos está desarrollando y desplegando una aplicación web sobre un servidor en la nube y necesita intercambiar información.

Carlos podrá utilizar el protocolo SSH (SCP) para enviar y recibir archivos del servidor remoto de manera sencilla y segura.

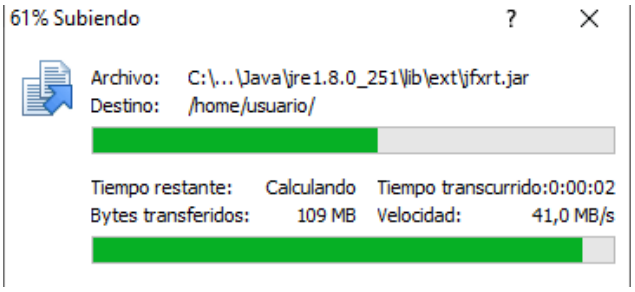

Fig. 19. Transferencia SCP usando WinSCP

/ 12. Bibliografía

Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J., Britt, D. T., & Redbooks, I. B. M. (2006). TCP/IP *Tutorial and Technical Overview*. IBM Redbooks.

Miguel, R. M. (2020). *Administración de Servicios de Transferencia de Archivos y Contenidos Multimedia* (MF0497_3). RA-MA S.A. Editorial y Publicaciones.

/ 13. Webgrafía

https://www.redeszone.net/tutoriales/servidores/ftps-ftpes-sftp-caracteristicas-diferencias/