DESPLIEGUE DE APLICACIONES WEB

CFGS. Diseño de Aplicaciones Web (DAW). 18-19

IES FRANCESC DE BORJA MOLL

TEMA 4. INSTALACIÓN Y ADMINISTRACIÓN DE SERVIDORES DE TRANSFERENCIA DE ARCHIVOS.

ÍNDICE DE CONTENIDOS:

1. SERVICIO DE TRANSFERENCIA DE FICHEROS.

Normalmente para subir archivos en Internet, ya sean de texto, imágenes, vídeo... es necesario emplear algún método de transferencia de archivos para ubicarlos.

Uno de los métodos empleados como servicio de transferencia de archivos se realiza mediante el servicio ftp. Éste utiliza el protocolo FTP (*File Transfer Protocol*) empleando la arquitectura cliente-servidor. Así el servidor ftp esperará peticiones para transferir los archivos y el cliente ftp, ya sea por terminal o de modo gráfico, realizará esas peticiones.

Uno de los principales problemas, a pesar de ser uno de los métodos más utilizados del protocolo FTP es la no seguridad de la información, esto ocurre porque la transferencia tiene lugar sin cifrar la información transferida. Éste no sólo es un problema del protocolo FTP, sino de muchos de los protocolos utilizados en Internet (HTTP, TELNET, POP, etc) puesto que en el comienzo de Internet no se preveía su expansión actual y no se pensaba en asegurar la información mediante cifrado, sino simplemente asegurar el buen funcionamiento. Hoy en día existen extensiones sobre el protocolo FTP que aseguran el cifrado en la transferencia, como FTPS, empleando el cifrado SSL/TLS.

No confundir FTPS con SFTP, ya que este último es implementado con otro servicio, el servicio SSH, y es utilizado para conexiones remotas seguras a través de un terminal de comandos.

1.1. ¿Cómo funciona?

El protocolo FTP emplea una **arquitectura cliente/servidor**, siendo el cliente ftp quien solicita la transferencia de archivos y el servidor ftp quien ofrece los archivos. Pertenece a la familia de protocolos de red TCP y por lo tanto es un **protocolo orientado a conexión**, esto es, el cliente ftp necesita establecer una conexión con el servidor para empezar la transferencia de ficheros. Si no se establece la conexión ésta no tiene lugar.

Puesto que FTP es un protocolo que no utiliza una autenticación de usuarios y contraseña cifrada, se considera un protocolo inseguro y no se debería utilizar a menos que sea absolutamente necesario. Existen otras alternativas al FTP, como por ejemplo el protocolo FTPS, para mantener comunicaciones cifradas. Aún así, el protocolo FTP está muy extendido en Internet, ya que a menudo los usuarios necesitan transferir archivos entre máquinas sin importar la seguridad.

El protocolo FTP requiere de dos puertos TCP en el servidor para su funcionamiento, a diferencia de la mayoría de los protocolos utilizados en Internet que solamente requieren un puerto en el servidor. **Un puerto es necesario para establecer el control de la conexión y otro se utiliza para el control de la transmisión, es decir, un puerto se utiliza para establecer la conexión entre el cliente y el servidor y otro para la transferencia de datos.**

Los puertos TCP del servidor en cuestión, suelen ser el 21 para el control de la conexión y otro a determinar según el modo de conexión: podría ser el 20 o incluso uno mayor de 1024. Hay que tener en cuenta que estos puertos pueden ser modificados en la configuración del servidor, así no es obligatorio que los puertos 21 y 20 sean los asignados al servidor FTP, pero sí son los que éste maneja por defecto. El puerto 21 también es conocido como puerto de comandos y el puerto 20 como puerto de datos.

La ventaja que supone utilizar el protocolo FTP se basa en su alto rendimiento y sencillez, que lo hacen una opción conveniente para la transferencia de archivos a través de Internet.

1.2. Cliente FTP.

Para poder establecer una conexión con el protocolo FTP son necesarias dos partes: un **servidor** y un **cliente**.

Existen múltiples tipos de clientes ftp, desde clientes en terminal de comandos, como ftp o lftp, clientes gráficos como gftp o FileZilla, hasta un cliente ftp en los navegadores mediante ftp://

¿Cuál elegir? Depende:

- ¿Se conoce la consola ftp? Si se maneja con soltura en la consola ftp, se puede pensar en un cliente ftp de comandos que permita utilizar la tecla "tabulado" después de escribir unos caracteres para complementar los nombres de archivos.

- ¿Cuál es el uso que se necesita? ¿Para qué se va a utilizar? A lo mejor solamente se quiere visitar un servidor ftp y descargar un archivo sin tener que andar instalando nuevos programas. En este caso se puede utilizar el cliente ftp del navegador, ftp://

- ¿Se quiere reanudar la conexión en caso de corte en la misma? En este caso mejor un cliente tipo gráfico.

- ¿Se sesea facilidad de manejo? Un cliente terminal de comandos suele ser menos interactivo que uno gráfico, se debe saber manejar con comandos en la consola ftp, mientras que en un cliente gráfico se puedemanejar a través de clics del ratón. Los clientes gráficos suelen ser más amigables y por lo tanto más utilizados.

- ¿Qué tipo de conexión se quiere establecer? ¿cifrada? ¿no cifrada? Dependiendo del tipo de conexión se debe emplear un cliente u otro, ya que no todos los clientes ftp permiten conexiones cifradas.

- ¿Se desea recordar conexiones (sitios)? El mismo caso, no todos los clientes ftp lo permiten.

Un cliente ftp muy recomendable es el cliente gráfico ftp FileZilla, ya que posee las siguientes características:

- Fácil de usar.
- Soporta FTP, FTP sobre SSL / TLS (FTPS) y SFTP.
- Compatibilidad con múltiples plataformas: se ejecuta en Windows, Linux, BSD, Mac OS X y más.
- Soporte Ipv6.
- Disponible en varios idiomas.
- Soporta y reanuda la transferencia de archivos de gran tamaño (mayores de 4 GB).
- Interfaz de usuario con pestañas.
- Potente administrador de sitios y cola de transferencia.
- Marcadores.
- Arrastrar y soltar.
- Permite configurar límites de velocidad de transferencia.
- Nombre de filtros.
- Directorio de comparación.
- Asistente de configuración de la red.
- Edición de archivos remoto.
- Automantenimiento de la conexión.
- HTTP/1.1, SOCKS5 y soporte de FTP-Proxy.
- Fichero de registro.
- Sincronización de directorios de navegación.
- Búsqueda de archivos remoto.

Para saber más acerca de FileZilla:

https://filezilla-project.org/

1.3. Tipos de usuarios.

¿Qué usuarios se pueden conectar al servidor ftp? ¿cualquiera? ¿sólo los usuarios del sistema?

Típicamente existen dos tipos de usuarios:

- **Usuarios anónimos**: usuarios que tienen acceso y permisos limitados por el sistema de archivos. Al conectarse al servidor FTP sólo deben introducir una contraseña simbólica, normalmente cualquier dirección de correo -real o ficticia-, por ejemplo: a@ .

- **Usuarios del sistema**: aquellos que disponen de una cuenta en la máquina que ofrece el servicio FTP. Al conectarse al servidor FTP deben introducir su contraseña de sistema.

Pero en ciertos servidores, como el servidor ProFTPD, existe una tercera posibilidad muy interesante: usuarios virtuales. Los usuarios virtuales poseen acceso y permisos al servidor FTP sin necesidad de ser usuarios del sistema, por lo tanto si un usuario virtual quisiera acceder al sistema operativo como si fuese un usuario del sistema, ya sea de forma local o remota no podría, pues su cuenta de usuario no existe en el sistema. Los usuarios virtuales tienen definida una contraseña propia y pueden estar definidos en ficheros de autenticación (de texto) con el mismo formato que los del sistema operativo GNU/Linux /etc/passwd, directorios LDAP, bases de datos SQL y servidores RADIUS.

Dependiendo del servidor ftp, se podrá tener unos métodos de autenticación de usuarios u otros, por ejemplo en el servidor ftp ProFTPD se permite los siguientes métodos:

- Ficheros de autenticación del sistema operativo: /etc/passwd y /etc/shadow: Para ello se usan las directivas **AuthUserFile** y **AuthGroupFile**.

- Usuarios virtuales definidos mediante ficheros de autenticación (de texto) propios, distintos de los del sistema operativo: para ello también se usan las directivas **AuthUserFile** y **AuthGroupFile**.

- Autenticación PAM: Es necesario establecer la directiva **AuthPAMAuthorative** a '**on**'.

- Bases de datos SQL, tales como MySQL o Postgres. Para ello emplear el módulo **mod_sql**.

- LDAP: Para ello emplear el módulo mod_ldap.

- RADIUS: Para ello emplear el módulo mod_radius.

Mediante la directiva **UserPassword** se puede crear una contraseña para un usuario particular que sobreescribe la contraseña del usuario en /etc/passwd (o /etc/shadow), esta contraseña es solamente efectiva dentro del contexto en el cual la directiva es aplicada, esto es, no se modifica el fichero /etc/passwd (o /etc/shadow) sino que se da la posibilidad de que el usuario emplee otra contraseña distinta de la definida en los ficheros del sistema operativo.

1.4. Modos de conexión del cliente.

Se ha comentado que el servidor FTP a diferencia de otros servidores necesita dos puertos TCP para hacer posible la transferencia de archivos. Ahora bien, ¿son estos puertos siempre los mismos o no? ¿son independientes del tipo de cliente y servidor o no? Básicamente, depende de dos factores: del modo de conexión del cliente ftp y de la configuración del servidor ftp.

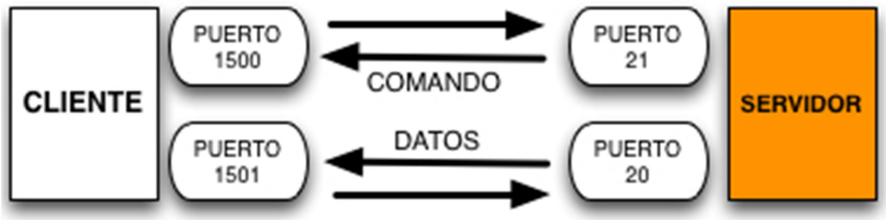
A priori, si no se modifica la configuración del servidor ftp, éste otorgará siempre el puerto TCP 21 para el canal de conexión de control. Es el puerto del canal de transmisión de datos, el que varía según el modo de conexión del cliente ftp, que puede ser activo o pasivo.

Cuando una aplicación cliente FTP inicia una conexión a un servidor FTP, abre el puerto 21 en el servidor. Se utiliza este puerto para arrojar todos los comandos al servidor. Cualquier petición de datos desde el servidor se devuelve al cliente a través de otro puerto TCP del servidor dependiendo del modo de conexión del cliente. Así:

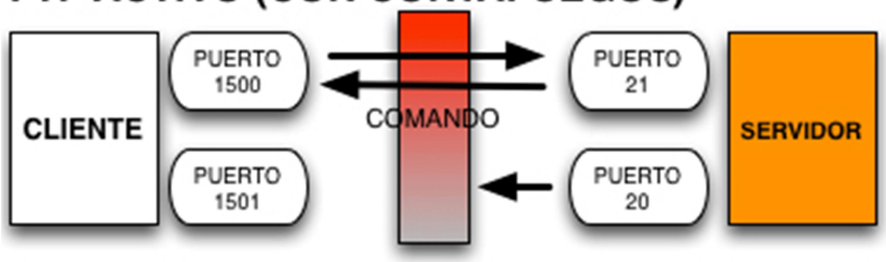
- El **modo activo** *(comando PORT)* es el método original utilizado por el protocolo FTP para la transferencia de datos a la aplicación cliente. **Cuando el cliente FTP inicia una transferencia de datos, el servidor abre una conexión desde el puerto 20 en el servidor para la dirección IP y un puerto aleatorio sin privilegios (mayor que 1024) especificado por el cliente**. Esta manera de establecer la conexión implica que la **máquina cliente debe poder aceptar conexiones en cualquier puerto superior al 1024**. Con el crecimiento de las redes inseguras, tales como Internet, es muy común el uso de cortafuegos para proteger las máquinas cliente. Debido a que estos cortafuegos en el lado del cliente normalmente rechazan las conexiones entrantes desde servidores FTP en modo activo, se creó el modo pasivo.

- La aplicación FTP cliente es la que inicia el **modo pasivo** *(comando PASV)*, de la misma forma que el modo activo. El cliente FTP indica que desea acceder a los datos en modo pasivo y el servidor proporciona la dirección IP y el puerto aleatorio, sin privilegios (mayor que 1024) en el servidor. Luego, el cliente se conecta al puerto en el servidor y descarga la información requerida.

FTP ACTIVO

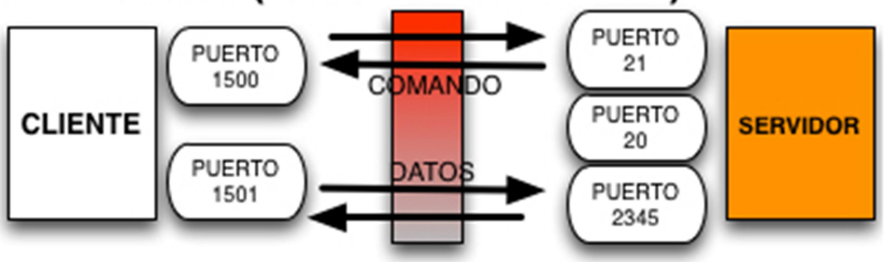


FTP ACTIVO (CON CORTAFUEGOS)

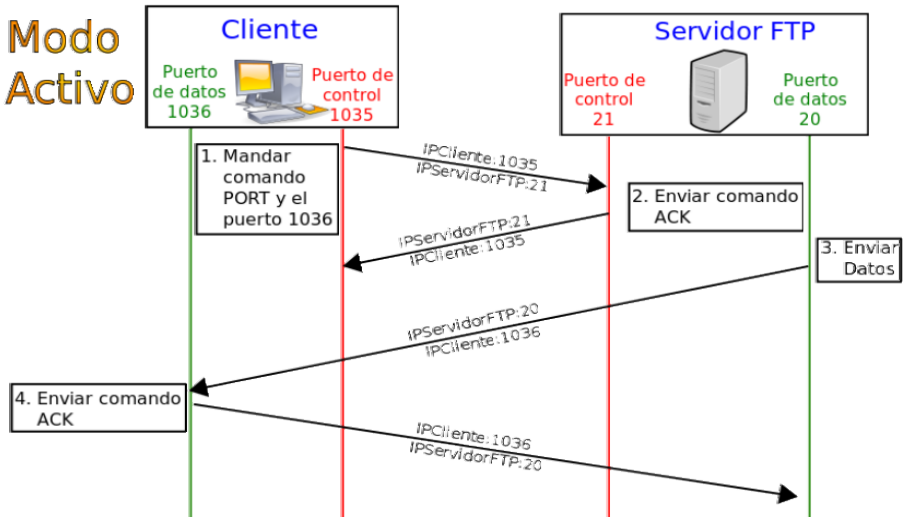


El cortafuegos bloquea el intento de comunicación del servidor con el cliente. Esto se debe a que el servidor usa un puerto diferente al de la conexión inicial.

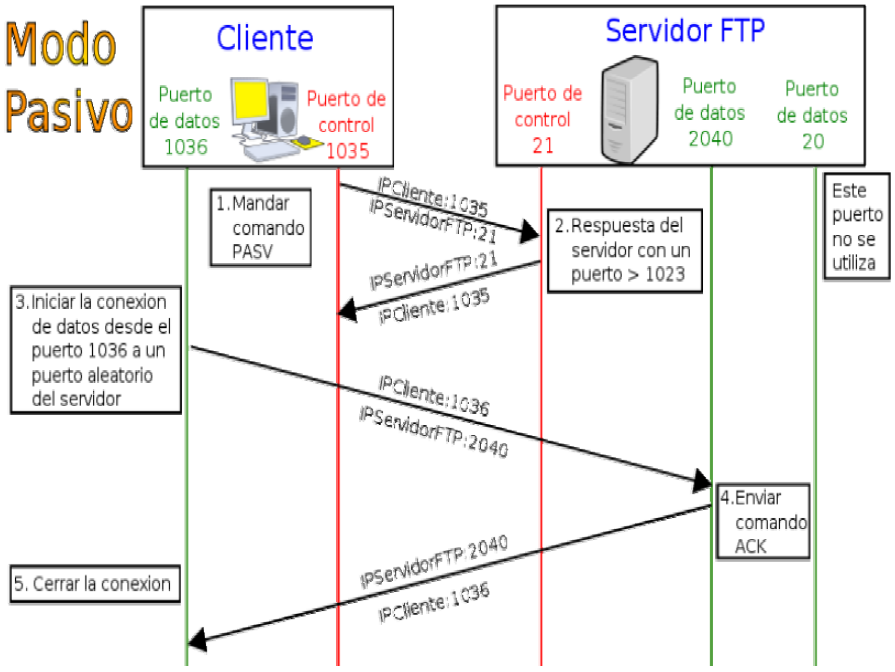
FTP PASIVO (CON CORTAFUEGOS)



El cortafuegos no bloquea el intento de comunicación del servidor con el cliente. Esto se debe a que el cliente inició la comunicación durante ambos intentos.



En modo Activo, el servidor siempre crea el canal de datos en su puerto 20, mientras que en el lado del cliente el canal de datos se asocia a un puerto aleatorio mayor que el 1024. Para ello, el cliente manda un comando PORT al servidor por el canal de control indicándole ese número de puerto, de manera que el servidor pueda abrirle una conexión de datos por donde se transferirán los archivos y los listados, en el puerto especificado.



En modo pasivo, cuando el cliente envía un comando PASV sobre el canal de control, el servidor FTP le indica por el canal de control, el puerto (mayor a 1024 del servidor. Ejemplo:2040) al que debe conectarse el cliente. El cliente inicia una conexión desde el puerto siguiente al puerto de control (Ejemplo: 1036) hacia el puerto del servidor especificado anteriormente (Ejemplo: 2040).

1.5. Tipos de transferencia de archivos.

Desde el punto de vista de FTP, los archivos se agrupan en dos tipos:

- Archivos ASCII: son archivos de texto plano (.txt, .ps, .html...)
- Archivos binarios: todo lo que no son archivos de texto: ejecutables (.exe), imágenes (.jpg, .png ...), archivos de audio (.mp3, .wav ...), vídeo (.avi, .mov …) , etcétera.

Es importante saber con qué tipo de archivos se está trabajando en la transferencia, ya que si no se utilizan las opciones adecuadas se puede destruir la información del archivo. El servidor ftp permite configurar la transferencia de archivos según el tipo del mismo, es por eso que al ejecutar el cliente FTP, antes de transferir un archivo, se debe utilizar uno de los siguientes comandos o poner la correspondiente opción en un programa con interfaz gráfica:

- **ascii** para tipos de archivos ascii.
- **binary** para tipos de archivos binarios.

1.6. Establecer permisos en FTP.

El protocolo FTP sigue los permisos establecidos en entornos de tipo UNIX y sus similares GNU/Linux, con lo cual existen tres grupos de permisos en el siguiente orden: propietario, grupo y otros:

- **Propietario(user=u):** El creador o el que ha subido el archivo al servidor FTP.
- **Grupo(group=g):** Se refiere a un grupo de usuarios que posee la propiedad del archivo, al que probablemente pertenece el propietario.
- **Otros(others=o):** Son el resto de usuarios no propietarios o que no pertenecen al grupo indicado. Son el resto del mundo.

Cada grupo a su vez puede tener tres permisos en el siguiente orden: lectura, escritura y ejecución identificados respectivamente por una 'r', una 'w' y una 'x'. La ausencia de permiso es identificada con el carácter '-'. Cada permiso tiene un equivalente numérico, así: r=4, w=2, x=1 y -=0. Por ejemplo: rw- identifica permiso de lectura y escritura o lo que es lo mismo 4+2+0=6

En un sistema operativo tipo GNU/Linux mediante el comando 'ls -l' se pueden ver los permisos asignados a ficheros y directorios, por ejemplo si la salida del anterior comando es:

```
-rw-r--r-- 1 alumno clase 0 jun 20 01:15 prueba1.txt
```

significa que,

- **prueba1.txt** es un fichero ya que **-rw-r--r--** comienza con '-', si fuese un directorio aparecería un 'd'
- **rw-r--r--** identifica los permisos del fichero prueba1.txt, que divididos 3 a 3 representan de izquierda a derecha: propietario, grupo, otros.
- **rw-** identifican los permisos del usuario propietario, en este caso **alumno**. Por lo tanto alumno posee los permisos de **lectura** y **escritura** sobre el fichero prueba1.txt o lo que es lo mismo 4+2+0=6
- **r--** identifican los permisos del grupo propietario, en este caso **clase**. Por lo tanto clase posee solamente el permiso de **lectura** o lo que es lo mismo **4+0+0=4**
- **r--** identifican los permisos de los **otros** (resto del mundo). Por lo tanto todos los usuarios que no son alumno y aquellos que no pertenecen al grupo clase poseen solamente el permiso de **lectura** o lo que es lo mismo **4+0+0=4**

Por lo tanto los permisos **rw-r- -r- -** equivalen a **644**.

Por otro lado en un sistema GNU/Linux, en principio, no todos los usuarios del sistema tienen acceso por ftp, así existe un fichero **/etc/ftpusers** que contiene una lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: root, bin, uucp, news.

1.7.1. Comandos ftp.

En la consola ftp pueden estar disponibles múltiples comandos, algunos de los más empleados son los recogidos en la siguiente tabla:

ABRIR/CERRAR CONEXIÓN	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
`open servidor`	Inicia conexión remota con un servidor ftp.
`close / disconnect`	Finalizan la sesión ftp sin cerrar la consola ftp.
`bye / quit / exit`	Terminan la sesión ftp y salen de la consola ftp.

!	Sale a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver, teclea exit en la línea de comandos.

AYUDA	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
`? / help`	Muestra una lista de los comandos disponibles.
`? comando / help comando`	Muestra la información relativa al comando.

TRABAJAR CON DIRECTORIOS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
`cd directorio`	Cambia de directorio en el servidor remoto.
`lcd directorio`	Cambiarse de directorio en el equipo local (cliente ftp).
`dir directorio / ls directorio`	Listan el contenido del directorio remoto actual.
`pwd`	Muestra el directorio activo en el servidor.
`lpwd`	Muestra el directorio activo en el equipo local (cliente ftp).
`rmdir` directorio	Elimina un directorio vacío en el servidor.
`mkdir` directorio	Crea un directorio en el servidor.Crea un directorio en el servidor.

TRABAJAR CON FICHEROS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
`delete` archivo	Borrar un archivo en el servidor remoto.
`mdelete` patrón	Borrar varios archivos según un patrón.
`get` archivo	Obtiene archivo en el equipo cliente desde el servidor remoto.
`mget` archivos	Obtiene varios archivos desde el servidor remoto.
`put` archivo	Envía un archivo al servidor remoto.
`mput` archivos	Envía varios archivos al servidor remoto.
`rename` archivo	Cambia el nombre a un archivo en el servidor.
`ascii`	Para configurar y transferir archivos tipo ascii.
`binary`	Para configurar y transferir archivos tipo binario.
`less` archivo	Leer contenido de archivo mediante el comando less.

TRABAJAR CON PERMISOS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
`chmod`	Cambio de permisos en el servidor remoto.
`umask`	Configura el sistema de permisos en el lado remoto.

1.8. Servicio de transferencia de archivos en modo gráfico.

El servicio de transferencia de ficheros obliga a entender el funcionamiento de un servidor ftp mediante el uso de sus comandos. No es una forma muy interactiva. Existe otro método mediante clientes ftp de modo gráfico o mediante el navegador, ya que éste incorpora su propio cliente ftp.

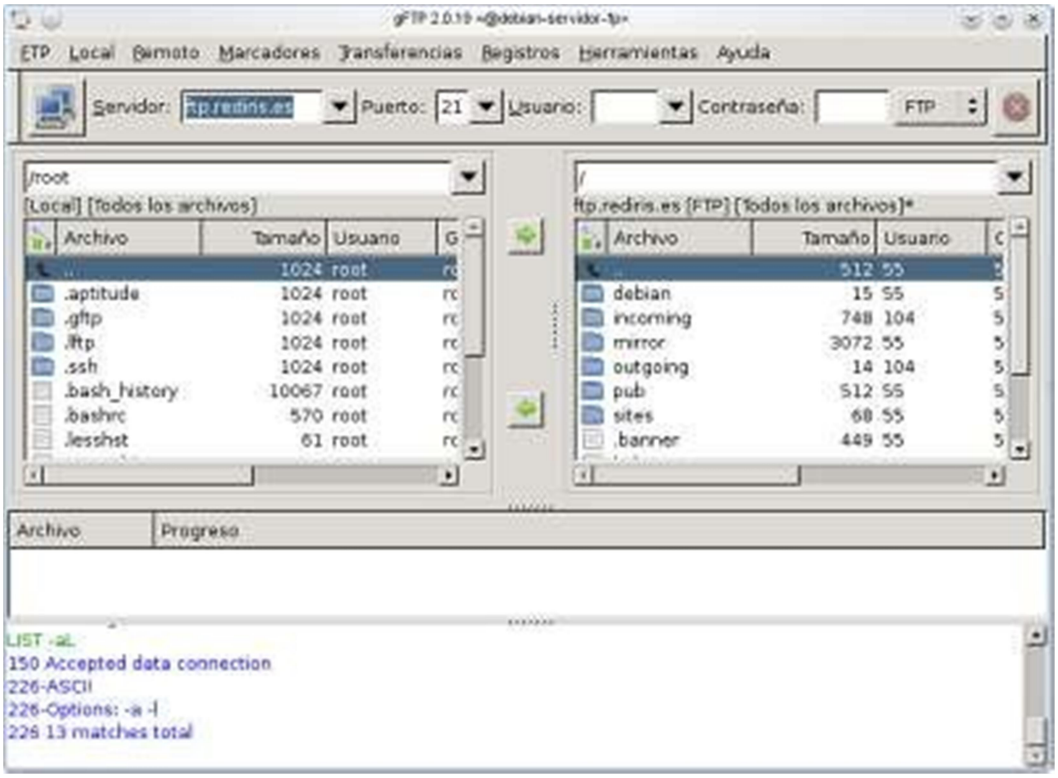
Típicamente los clientes gráficos se comportan todos igual, esto es, tienen una interfaz parecida, básicamente presentan una ventana partida en dos secciones: la de la izquierda suele representar el equipo cliente ftp -desde donde se intenta establecer la conexión- y la de la derecha suele representar el equipo servidor ftp -quién recibe la conexión-. Luego suelen existir, en alguna zona determinada de la ventana: en el centro entre las dos secciones, arriba de las dos secciones, etc una serie de botones, usualmente representados como flechas que indican la posibilidad de subir o descargar archivos. Incluso dependiendo del cliente en modo gráfico es posible guardar los datos de las conexiones como plantillas, de tal forma que la próxima vez que se intente establecer la conexión con un mismo servidor ftp en vez de tener que rellenar los campos referentes a la conexión se puede hacer a través de la plantilla que ya posee el valor de esos campos.

Dentro de los clientes ftp en modo gráfico cabe destacar dos: **gftp** y **filezilla**. A continuación se puede ver un ejemplo de como utilizarlos para establecer una conexión con un servidor ftp, el servidor ftp.rediris.es:

Cliente en modo gráfico **gftp**.

- **Servidor**: Escribir aquí el nombre o IP del servidor FTP: ftp.rediris.es
- **Puerto**: Escribir aquí el puerto TCP de la conexión de control, por defecto: 21. Se puede omitir siempre y cuando sea el 21.
- **Usuario**: Escribir aquí el usuario con permisos de conexión en el servidor ftp. En la imagen se puede ver que no se ha escrito nada, esto es debido a que el servidor ftp.rediris.es permite la entrada a cualquier usuario y el cliente gráfico gftp al intentar conectar pedirá un usuario que tenga permisos para la conexión. Pulsar en cancelar y gftp cubrirá los campos usuario y contraseña, entrando al servidor ftp.

- **Contraseña**: Escribir aquí la contraseña del usuario con permisos de conexión en el servidor ftp. En la imagen se puede ver que no se ha escrito nada, esto es debido a la misma causa que en el campo Usuario.



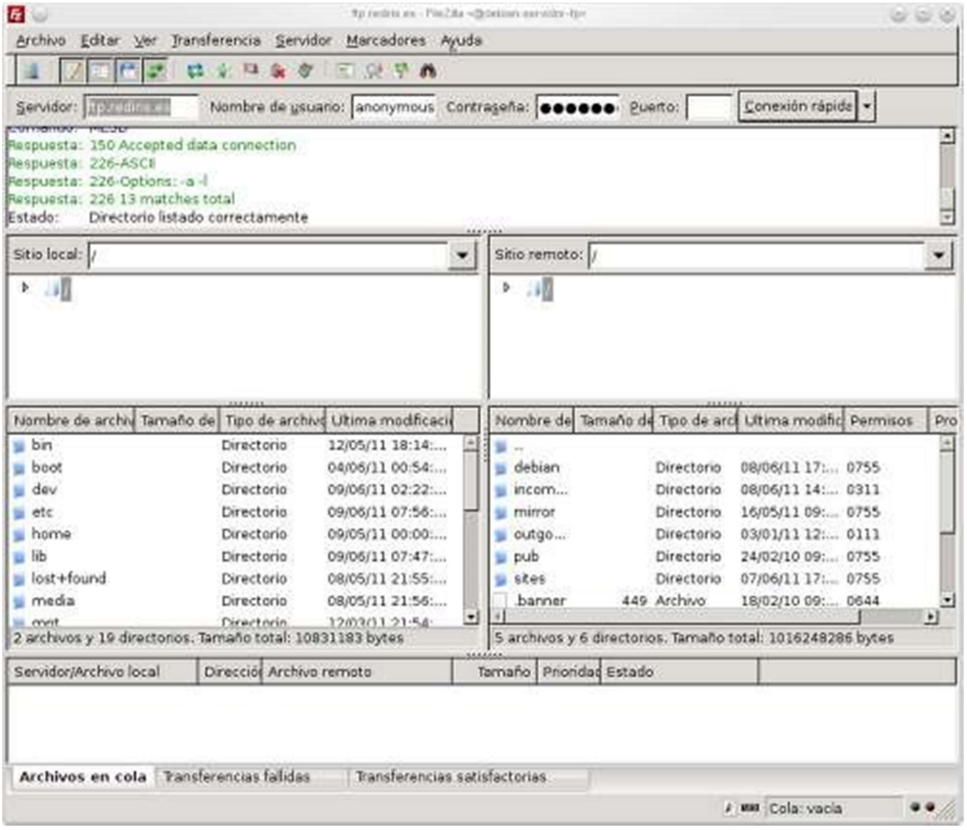
2. Cliente en modo gráfico filezilla.

- **Servidor**: Escribir aquí el nombre o IP del servidor FTP: ftp.rediris.es

- **Nombre de usuario**: Escribir aquí el usuario con permisos de conexión en el servidor ftp. Filezilla, al contrario que gftp, no cubre los datos usuario y contraseña si no se escribe nada en los campos, entonces se debe escribir un nombre de usuario, por ejemplo anonymous, y una contraseña -cualquier secuencia de caracteres-.

- **Contraseña**: Escribir aquí la contraseña del usuario con permisos de conexión en el servidor ftp. En la imagen se puede ver que se ha escrito una secuencia de caracteres punto, lo que significa que a la hora de escribir caracteres en ese campo no se muestra su valor por seguridad. Es necesario escribir una contraseña por lo comentado en el campo anterior: Nombre de usuario.

- **Puerto**: Escribir aquí el puerto TCP de la conexión de control, por defecto: 21. Se puede omitir siempre y cuando sea el 21.



1.9. Servicio de transferencia de archivos desde el navegador.

El navegador web también puede ejercer de cliente ftp y, puesto que la mayoría de los sistemas operativos cuentan con un navegador en su instalación, es una de las herramientas más usadas para transferencia de archivos.

Para poder usar el navegador como cliente ftp se debe escribir en la barra de dirección una dirección URL tipo, como la siguiente:

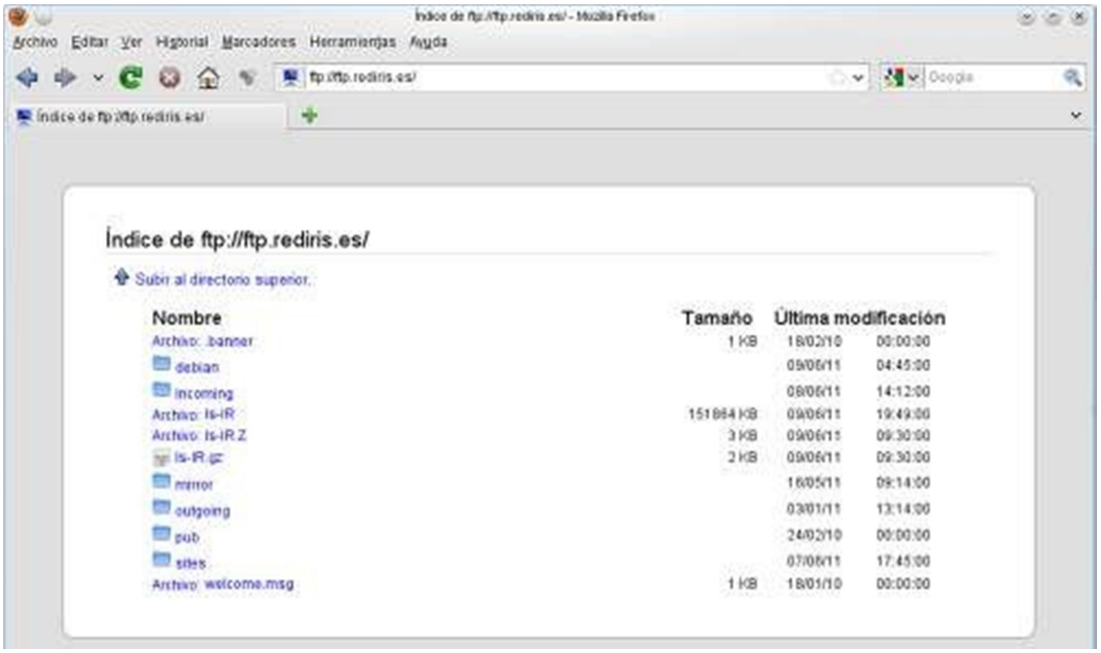
ftp://nombre_servidor_ftp:puerto

donde,

- **ftp://** indica que el protocolo que se desea que interprete el navegador sea el ftp.
- **nombre_servidor_ftp** representa el nombre o la IP del servidor ftp.
- **puerto** indica el puerto TCP, por defecto 21. Se puede omitir siempre y cuando sea el 21.

Si el servidor ftp permite la conexión a un usuario anónimo, al ejecutar **ftp://nombre_servidor_ftp:puerto** se entrará directamente al servidor ftp, esto es, el navegador no preguntará qué usuario y contraseña se necesita para establecer la conexión.

En la siguiente imagen se puede ver como se puede acceder al servidor ftp de rediris utilizando el navegador:



Así, lo único que se tiene que hacer es escribir en la dirección URL: **ftp://ftp.rediris.es** y pulsar **Enter**, con lo cual, automáticamente, conectas con el servidor ftp, pudiendo visitar las carpetas y ver los ficheros como si de un explorador de archivos se tratara.

Para descargar las carpetas o archivos simplemente se debe pulsar con el botón derecho del ratón sobre ellos y elegir la opción **Guardar enlace como…** -que aparece en Firefox y es similar en otros navegadores-.

Pero no todo van a ser ventajas al utilizar el navegador como cliente ftp, puesto que otros clientes tienen la posibilidad de continuar las descargas cuando estás sufrieron algún tipo de interrupción, cosa que no pasa con el cliente ftp del navegador, como por ejemplo el cliente gráfico FileZilla que soporta y reanuda la transferencia de archivos de gran tamaño(> 4 GB).

1.10. Asegurando el servicio de transferencia de archivos.

Bien, pero ¿qué pasa con los datos en la transferencia? ¿viajan cifrados? ¿no? Pues empleando el protocolo ftp cualquiera que tenga acceso al canal de transmisión podrá ver en texto claro todo lo que se transmite, esto es, los datos no se cifran. Esto puede carecer de importancia, o no, según el contexto de la transmisión. Así, puede que a un organismo público no le importe compartir

información a través de ftp y que los datos en la transferencia viajen sin cifrar y, sin embargo, a una empresa si le interese que los datos viajen cifrados.

Entonces, cuando interese asegurar el servicio de transferencia de archivos se debe descartar el protocolo ftp y empezar a pensar en otras alternativas, como: **ftps** o **sftp**.

FTPS es una extensión del protocolo FTP que asegura el cifrado en la transferencia mediante los protocolos SSL/TLS. Permite tres tipos de funcionamiento:

- SSL Implícito:
  - Como conexiones HTTPS.
  - Usa los puertos 990 y 989.
- SSL Explícito
  - El cliente usa los mismos puertos estándar FTP: 20 y 21 pero se efectúa el cifrado en ellos.
  - Usa AUTH SSL.
- TLS Explícito:
  - Similar a SSL Explícito pero usa AUTH TLS.

El cifrado al que se hace referencia es el cifrado de clave pública o asimétrico: **clave pública(kpub)** y **clave privada(kpriv)**. La **kpub** interesa publicarla para que llegue a ser conocida por cualquiera, la **kpriv** no interesa que nadie la posea, solo el propietario de la misma. Ambas son necesarias para que la comunicación sea posible, una sin la otra no tienen sentido, así una información cifrada mediante la **kpub** solamente puede ser descifrada mediante la **kpriv** y una información cifrada mediante la **kpriv** sólo puede ser descifrada mediante la **kpub**.

En el cifrado asimétrico se puede estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el **cliente ftp(A)** y el **servidor ftp(B)**. Ver la siguiente tabla como ejemplo de funcionamiento del cifrado asimétrico:

Funcionamiento del cifrado asimétrico	
 A(inf) → inf cifrada → B [descifrar inf] → B(inf) = A(inf) A(inf) → inf cifrada = [(inf)]kpubB → B [inf. cifrada]kprivB → B(inf) = A(inf)	
Identificación	
A	Cliente ftp.
inf cifrada = [(inf)]kpubB	Información cifrada mediante la clave pública de B obtenida a través de un certificado digital.
[inf. cifrada]kprivB	Información descifrada mediante la clave privada de B.
B	Servidor ftp.

Para saber más acerca de cómo asegurar FTP con TLS:
http://tools.ietf.org/html/rfc4217

Para saber más sobre TLS: http://tools.ietf.org/html/rfc4346

1.11. El servicio de transferencia de archivos en el proceso de despliegue de una aplicación Web.

Suele ser típico que cualquier aplicación web en Internet disponga de la posibilidad de subir archivos mediante una configuración del código fuente de la misma, una aplicación propia o una aplicación de terceros, como los paneles de administración web.

Si se emplea una aplicación web para subir archivos se debe tener en cuenta cuánto tiempo se puede mantener la conexión abierta con el servicio web y cuál es el tamaño máximo de subida de un archivo. Estas cuestiones suelen ser típicas de la configuración del servidor web. Por contra, si se emplea un servidor ftp, dependerá de éste las cuestiones anteriores.

Se suele configurar el servidor web con unos parámetros: tiempo de conexión y tamaño máximo de subidas de archivos diferentes del servidor ftp, de tal forma que para archivos de tamaño no muy grandes se puedan emplear aplicaciones web y no se sufra un corte en la subida de archivos y, para archivos grandes, se emplee el servidor ftp.

Normalmente las empresas de alojamiento web (hosting) permiten la subida de archivos mediante un servidor ftp y poseen documentos sobre cómo operar con éste, esto es, documentación que explica cómo conectarse a sus servidores ftp a través de algún cliente ftp, como por ejemplo: FileZilla, Cute FTP, Fetch o Transmit. También suelen permitir usar SCP o SFTP para transferir ficheros de forma segura mediante un canal cifrado. Por ejemplo en Filezilla se puede establecer la conexión de forma cifrada directamente, sólo con indicar como puerto TCP el número del servidor SSH, por defecto, 22.

Muchos editores web también permiten subir tu aplicación web al servidor con el protocolo FTP, esto puede resultar más sencillo que el uso de una aplicación de FTP independiente.

Eso si, sea cual sea el método ftp que se utilice para subir archivos y actualizar un sitio web, se desaconseja el uso de aplicaciones no actualizadas que podrían comprometer la seguridad de ese sitio web.

A continuación se pueden ver errores típicos, junto con sus soluciones, que se puede encontrar al subir tu aplicación mediante un servidor FTP:

- El cliente FTP muestra el error **access denied**, o similar, cuando se sube o se borra ficheros y carpetas: Comprobar que el usuario FTP tenga permisos suficientes sobre la carpeta o fichero en la que se desea subir o que se desea borrar.

- Las páginas no son reconocidas de forma automática al acceder al dominio: Los servidores GNU/Linux son sensibles a mayúsculas y minúsculas por lo que verifica el nombre de los archivos.

- El cliente de FTP muestra el mensaje **too many connections from your IP address**: Esto quiere decir que existen más conexiones abiertas con el servidor FTP desde la misma dirección IP de las permitidas. En ese caso, asegurarse que no exista ninguna aplicación, como un cortafuegos, que pueda estar bloqueando las conexiones abiertas, y provocando, de esta forma, que se establezcan más intentos de conexión de los necesarios.