

## Práctica 7 – Protocolos de nivel de aplicación

### 1- Objetivos

En esta práctica estudiaremos diversos protocolos de la capa de aplicación: HTTP, FTP, Telnet y SMTP; los comandos existentes y problemas que se presentan. Capturaremos los paquetes transmitidos por la red con la herramienta Ethereal para luego extraer la información de la capa de aplicación y analizarla. Por último, nos familiarizaremos con la lectura de RFCs.

En esta primera sesión veremos los protocolos HTTP y FTP.

### 2- Avisos generales

Si quieren conservar cualquier fichero entre sesiones guárdenlo en una memoria USB, dado que no se asegura que los ficheros creados o modificados durante una sesión de prácticas se mantengan para la siguiente.

### 3- Introducción

Conviene recordar que aunque desde un punto de vista general, todos los servicios de Internet implican tráfico de algún tipo de ficheros, cuando estos son de tipos determinados, los servicios y programas que los ejecutan, reciben nombres especiales. Por ejemplo, un navegador es en cierta forma un programa FTP que recibe un tipo especial de documentos (HTML), y que además es capaz de mostrarlos en pantalla. En este caso el protocolo de transferencia utilizado es muy específico (HTTP). Un programa de correo electrónico es también un caso especial de transferencia de ficheros de una clase muy concreta (e-mail); el protocolo específico es SMTP, etc. Sin embargo, FTP se reserva para un uso genérico y es sinónimo de transferencia de cualquier tipo de ficheros: ejecutables; imagen; multimedia, etc, sin ninguna elaboración posterior. Es decir, el mero hecho de transferirlos entre máquinas (enviar por la Red una copia de un fichero contenido en el servidor, y guardarlo en el disco de la máquina cliente).

Dispone de información adicional en las RFCs correspondientes en: [www.faqs.org/rfc](http://www.faqs.org/rfc)

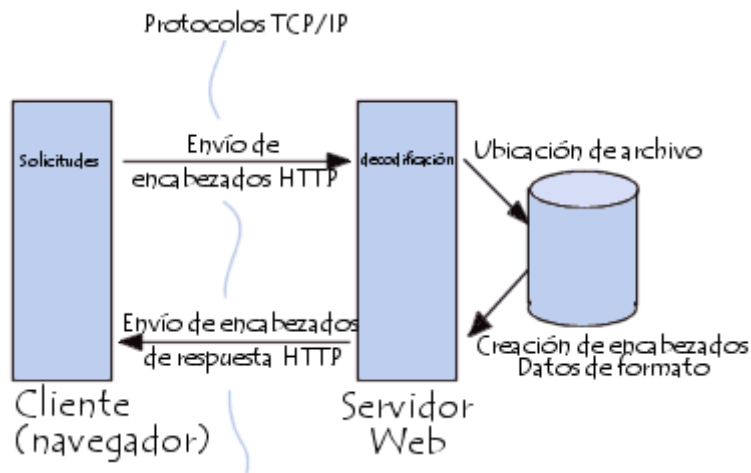
### 4- Protocolo HTTP

Desde 1990, el protocolo HTTP (Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 sólo tenía la finalidad de transferir los datos a través de Internet (en particular páginas Web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML), entre un navegador (el cliente) y un servidor web (denominado, entre otros, httpd en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL.

## Comunicación entre el navegador y el servidor

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:



1. El navegador realiza una solicitud HTTP
2. El servidor procesa la solicitud y después envía una respuesta HTTP

En realidad, la comunicación se realiza en más etapas si se considera el procesamiento de la solicitud en el servidor. Sólo nos ocupamos del protocolo HTTP.

## Solicitud HTTP

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

Una línea de solicitud: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:

- El método
- La dirección URL
- La versión del protocolo utilizada por el cliente (por lo general, HTTP/1.0)

Los campos del encabezado de solicitud: son un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

El cuerpo de la solicitud: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

Por lo tanto, una solicitud HTTP posee la siguiente sintaxis (<crLf> significa retorno de carro y avance de línea):

```
MÉTODO VERSIÓN URL<crLf>
ENCABEZADO: Valor<crLf>
. . . ENCABEZADO: Valor<crLf>
Línea en blanco <crLf>
CUERPO DE LA SOLICITUD
```

A continuación se encuentra un ejemplo de una solicitud HTTP:

```
GET http://www.google.es HTTP/1.0 Accept : Text/html If-Modified-Since :
Saturday, 26-May-2009 14:37:11 GMT User-Agent : Mozilla/4.0 (compatible; MSIE
5.0; Windows XP)
```

## Comandos

Comando	Descripción
GET	Solicita el recurso ubicado en la URL especificada
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada
POST	Envía datos al programa ubicado en la URL especificada
PUT	Envía datos a la URL especificada
DELETE	Borra el recurso ubicado en la URL especificada

## Encabezados

Nombre del encabezado	Descripción
Accept	Tipo de contenido aceptado por el navegador (por ejemplo, <i>texto/html</i> ). Consulte <a href="#">Tipos de MIME</a>
Accept-Charset	Juego de caracteres que el navegador espera
Accept-Encoding	Codificación de datos que el navegador acepta
Accept-Language	Idioma que el navegador espera (de forma predeterminada, inglés)
Authorization	Identificación del navegador en el servidor
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Language	Tipo de idioma en el cuerpo de la solicitud
Content-Length	Extensión del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, <i>texto/html</i> ). Consulte <a href="#">Tipos de MIME</a>
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor

From	Permite especificar la dirección de correo electrónico del cliente
From	Permite especificar que debe enviarse el documento si ha sido modificado desde una fecha en particular
Link	Vínculo entre dos direcciones URL
Orig-URL	Dirección URL donde se originó la solicitud
Referer	Dirección URL desde la cual se realizó la solicitud
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre y la versión del navegador y el sistema operativo

## Respuesta HTTP

Una respuesta HTTP es un conjunto de líneas que el servidor envía al navegador. Está constituida por:

Una línea de estado: es una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio:

- La versión del protocolo utilizada
- El código de estado
- El significado del código

Los campos del encabezado de respuesta: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado. Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

El cuerpo de la respuesta: contiene el documento solicitado.

Por lo tanto, una respuesta HTTP posee la siguiente sintaxis (<crLf> significa retorno de carro y avance de línea):

```

VERSION-HTTP CÓDIGO EXPLICACIÓN <crLf>
ENCABEZADO: Valor<crLf>
. . . ENCABEZADO: Valor<crLf>
Línea en blanco <crLf>
CUERPO DE LA RESPUESTA

```

A continuación se encuentra un ejemplo de una respuesta HTTP:

```

HTTP/1.0 200 OK Date: Tue, 26 May 2009 14:37:12 GMT Server : Microsoft-IIS/2.0
Content-Type : text/HTML Content-Length : 1245 Last-Modified : Tue, 26 May
2009 08:25:13 GMT

```

## Encabezados de respuesta

Nombre del encabezado	Descripción
Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Language	Tipo de idioma en el cuerpo de la respuesta
Content-Length	Extensión del cuerpo de la respuesta
Content-Type	Tipo de contenido del cuerpo de la respuesta (por ejemplo, <i>texto/html</i> ). Consulte <a href="#">Tipos de MIME</a>
Date	Fecha en que comienza la transferencia de datos
Expires	Fecha límite de uso de los datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
Location	Redireccionamiento a una nueva dirección URL asociada con el documento
Server	Características del servidor que envió la respuesta

## Los códigos de respuesta

Son los códigos que se ven cuando el navegador no puede mostrar la página solicitada. El código de respuesta está formado por tres dígitos: el primero indica el estado y los dos siguientes explican la naturaleza exacta del error.

Código	Mensaje	Descripción
<b>10x</b>	<b>Mensaje de información</b>	<b>Estos códigos no se utilizan en la versión 1.0 del protocolo</b>
<b>20x</b>	<b>Éxito</b>	<b>Estos códigos indican la correcta ejecución de la transacción</b>
200	OK	La solicitud se llevó a cabo de manera correcta
201	CREATED	Sigue a un comando <a href="#">POST</a> e indica el éxito, la parte restante del cuerpo indica la dirección <a href="#">URL</a> donde se ubicará el documento creado recientemente.
202	ACCEPTED	La solicitud ha sido aceptada, pero el procedimiento que sigue no se ha llevado a cabo
203	PARTIAL INFORMATION	Cuando se recibe este código en respuesta a un comando de <a href="#">GET</a> indica que la respuesta no está completa.
204	NO RESPONSE	El servidor ha recibido la solicitud, pero no hay información de respuesta
205	RESET CONTENT	El servidor le indica al navegador que borre el contenido en los campos de un formulario
206	PARTIAL CONTENT	Es una respuesta a una solicitud que consiste en el encabezado <i>range</i> . El servidor debe indicar el encabezado <i>content-Range</i>

30x	Redirección	Estos códigos indican que el recurso ya no se encuentra en la ubicación especificada
301	MOVED	Los datos solicitados han sido transferidos a una nueva dirección
302	FOUND	Los datos solicitados se encuentran en una nueva dirección URL, pero, no obstante, pueden haber sido trasladados
303	METHOD	Significa que el cliente debe intentarlo con una nueva dirección; es preferible que intente con otro método en vez de <b>GET</b>
304	NOT MODIFIED	Si el cliente llevó a cabo un comando <b>GET</b> condicional (con la solicitud relativa a si el documento ha sido modificado desde la última vez) y el documento no ha sido modificado, este código se envía como respuesta.
40x	Error debido al cliente	Estos códigos indican que la solicitud es incorrecta
400	BAD REQUEST	La sintaxis de la solicitud se encuentra formulada de manera errónea o es imposible de responder
401	UNAUTHORIZED	Los parámetros del mensaje aportan las especificaciones de formularios de autorización que se admiten. El cliente debe reformular la solicitud con los datos de autorización correctos
402	PAYMENT REQUIRED	El cliente debe reformular la solicitud con los datos de pago correctos
403	FORBIDDEN	El acceso al recurso simplemente se deniega
404	NOT FOUND	Un clásico. El servidor no halló nada en la dirección especificada. Se ha abandonado sin dejar una dirección para redireccionar... :)
50x	Error debido al servidor	Estos códigos indican que existe un error interno en el servidor
500	INTERNAL ERROR	El servidor encontró una condición inesperada que le impide seguir con la solicitud (una de esas cosas que les suceden a los servidores...)
501	NOT IMPLEMENTED	El servidor no admite el servicio solicitado (no puede saberlo todo...)
502	BAD GATEWAY	El servidor que actúa como una puerta de enlace o proxy ha recibido una respuesta no válida del servidor al que intenta acceder
503	SERVICE UNAVAILABLE	El servidor no puede responder en ese momento debido a que se encuentra congestionado (todas las líneas de comunicación se encuentran congestionadas, inténtelo de nuevo más adelante)
504	GATEWAY TIMEOUT	La respuesta del servidor ha llevado demasiado tiempo en relación al tiempo de espera que la puerta de enlace podía admitir (excedió el tiempo asignado...)

## Analizando HTTP

Lance en su PC-SC el analizador de protocolos Ethereal y póngalo a capturar tramas Ethernet. Para un mejor análisis de la información capturada, aplique el siguiente filtro:

`ip.src==10.1.1.XY o ip.dst==10.1.1.XY`

Abra su navegador y escriba como URL: <http://10.1.1.XY>

Verá la página de inicio del servidor Apache que está corriendo en su máquina virtual.

Pare la captura de Ethereal y analice cada una de las tramas, identificando los protocolos presentes. Guarde su captura, la necesitará enseguida.

Vuelva a activar Ethereal.

Conéctese mediante `telnet` a su máquina virtual en el puerto correspondiente al servicio `http`. Para ver los puertos asociados a los distintos servicios, consulte el fichero `/etc/services` en el propio PC-SC.

Una vez conectado, teclee el comando: `GET / http/1.1` y pulse `ENTER` dos veces.

Guarde su captura y compárela con la obtenida al utilizar el navegador, ¿Qué diferencias encuentra? ¿A qué se deben?

Utilicen la información proporcionada por la página genérica del servidor Apache de su máquina virtual para ubicar en ésta, una página `html` sencilla que pueda descargar desde el PC-SC. Si no está familiarizado con el `html`, siga los siguientes pasos:

```
#vi pagina1.html
```

Pulse la tecla “Insert” para activar el modo edición del editor `vi`. Teclee el siguiente texto:

```
<html>
<body>
Laboratorio de Telemática I
</html>
</body>
```

Pulse la tecla “Esc”, teclee `:wq` y pulse “ENTER”.

Ya tiene un página `html`, ahora sólo tiene que llevarla a su máquina virtual y ubicarla en la carpeta indicada por el servidor Apache. Utilice el servicio que crea conveniente para ello.

Checkpoint 7.1: Muestra al profesor de prácticas que es capaz de cargar la página creada mediante una conexión `Telnet` a su máquina virtual desde el PC-SC. Descargue también su encabezado.

Pruebe a conectarse mediante `Telnet` a [www.google.es](http://www.google.es) y descargue su página principal. Redireccione la salida de `Telnet` a un fichero de nombre `google.html` y compruebe el resultado obtenido abriendo dicho fichero en su navegador.

## 4- Protocolo FTP

El protocolo `FTP` (Protocolo de transferencia de archivos) es, como su nombre lo indica, un protocolo para transferir archivos.

La implementación del FTP se remonta a 1971 cuando se desarrolló un sistema de transferencia de archivos (descrito en RFC141) entre equipos del Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology). Desde entonces, diversos documentos de RFC (petición de comentarios) han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

El protocolo FTP está definido por la RFC 959 y diversas actualizaciones en materia de seguridad.

### La función del protocolo FTP

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP.

El objetivo del protocolo FTP es:

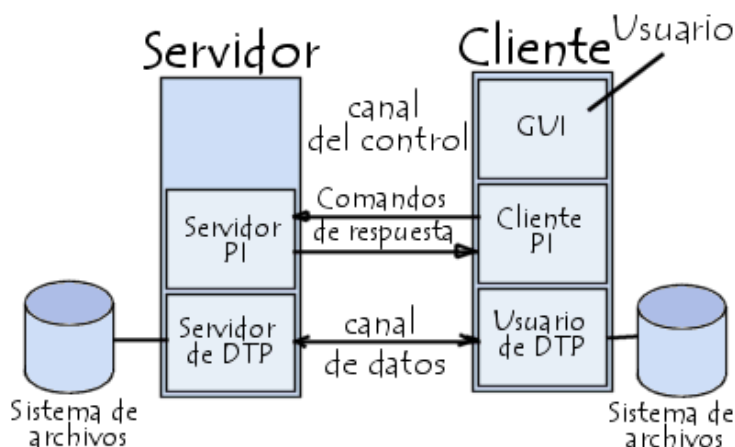
- Permitir que equipos remotos puedan compartir archivos
- Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor
- Permitir una transferencia de datos eficaz

### El modelo FTP

El protocolo FTP está incluido dentro del modelo cliente-servidor, es decir, un equipo envía órdenes (el cliente) y el otro espera solicitudes para llevar a cabo acciones (el servidor).

Durante una conexión FTP, se encuentran abiertos dos canales de transmisión:

- Un canal de comandos (canal de control)
- Un canal de datos

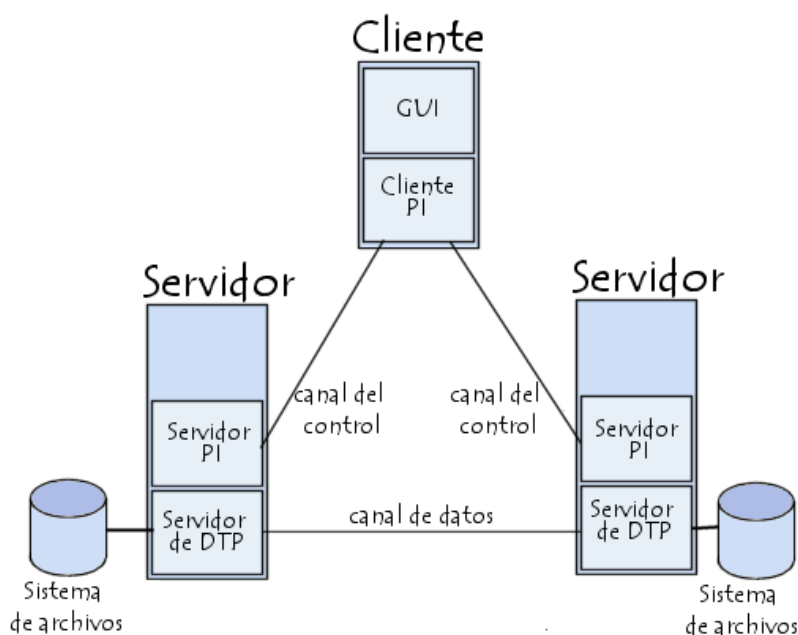


Por lo tanto, el cliente y el servidor cuentan con dos procesos que permiten la administración de estos dos tipos de información:



- DTP (Proceso de transferencia de datos) es el proceso encargado de establecer la conexión y de administrar el canal de datos. El DTP del lado del servidor se denomina SERVIDOR DE DTP y el DTP del lado del cliente se denomina USUARIO DE DTP.
- PI (Intérprete de protocolo) interpreta el protocolo y permite que el DTP pueda ser controlado mediante los comandos recibidos a través del canal de control. Esto es diferente en el cliente y el servidor:
- El SERVIDOR PI es responsable de escuchar los comandos que provienen de un USUARIO PI a través del canal de control en un puerto de datos, de establecer la conexión para el canal de control, de recibir los comandos FTP del USUARIO PI a través de éste, de responderles y de ejecutar el SERVIDOR DE DTP.
- El USUARIO PI es responsable de establecer la conexión con el servidor FTP, de enviar los comandos FTP, de recibir respuestas del SERVIDOR PI y de controlar al USUARIO DE DTP, si fuera necesario.

Cuando un cliente FTP se conecta con un servidor FTP, el USUARIO PI inicia la conexión con el servidor de acuerdo con el protocolo Telnet. El cliente envía comandos FTP al servidor, el servidor los interpreta, ejecuta su DTP y después envía una respuesta estándar. Una vez que se establece la conexión, el servidor PI proporciona el puerto por el cual se enviarán los datos al Cliente DTP. El cliente DTP escucha el puerto especificado para los datos provenientes del servidor. Es importante tener en cuenta que, debido a que los puertos de control y de datos son canales separados, es posible enviar comandos desde un equipo y recibir datos en otro. Entonces, por ejemplo, es posible transferir datos entre dos servidores FTP mediante el paso indirecto por un cliente para enviar instrucciones de control y la transferencia de información entre dos procesos del servidor conectados en el puerto correcto.



En esta configuración, el protocolo indica que los canales de control deben permanecer abiertos durante la transferencia de datos. De este modo, un servidor puede detener una transmisión si el canal de control es interrumpido durante la transmisión.

### Los comandos FTP

Toda comunicación que se realice en el canal de control sigue las recomendaciones del protocolo Telnet. Por lo tanto, los comandos FTP son cadenas de caracteres Telnet (en código NVT-ASCII) que finalizan con el código de final de línea Telnet (es decir, la secuencia <CR>+<LF>, Retorno de carro seguido del carácter Avance de línea indicado como <CRLF>). Si el comando FTP tiene un parámetro, éste se separa del comando con un espacio (<SP>).

Los comandos FTP hacen posible especificar:

- El puerto utilizado
- El método de transferencia de datos
- La estructura de datos
- La naturaleza de la acción que se va a realizar (Recuperar, Enumerar, Almacenar, etc.)

Existen tres tipos de comandos FTP diferentes:

1. Comandos de control de acceso
2. Comandos de parámetros de transferencia
3. Comandos de servicio FTP

Comandos de control de acceso	
Comando	Descripción
USER	Cadena de caracteres que permite identificar al usuario. La identificación del usuario es necesaria para establecer la comunicación a través del canal de datos.
PASS	Cadena de caracteres que especifica la contraseña del usuario. Este comando debe ser inmediatamente precedida por el comando <i>USER</i> . El cliente debe decidir si esconder la visualización de este comando por razones de seguridad.
ACCT	Cadena de caracteres que especifica la cuenta del usuario. El comando generalmente no es necesario. Durante la respuesta que acepta la contraseña, si la respuesta es 230, esta etapa no es necesaria; Si la respuesta es 332, sí lo es.
CWD	<i>Change Working Directory (Cambiar el directorio de trabajo)</i> : este comando permite cambiar el directorio actual. Este comando requiere la ruta de acceso al directorio para que se complete como un argumento.
CDUP	<i>Change to Parent Directory (Cambiar al directorio principal)</i> : este comando permite regresar al directorio principal. Se introdujo para resolver los problemas de denominación del directorio principal según el sistema (generalmente "..").
SMNT	<i>Structure Mount (Montar estructura)</i> :

REIN	<i>Reinitialize (Reinicializar):</i>
QUIT	Comando que permite abandonar la sesión actual. Si es necesario, el servidor espera a que finalice la transferencia en progreso y después proporciona una respuesta antes de cerrar la conexión.
Comandos de parámetros de transferencia	
Comando	Descripción
PORT	Cadena de caracteres que permite especificar el número de puerto utilizado.
PASV	Comando que permite indicar al servidor de DTP que permanezca a la espera de una conexión en un puerto específico elegido aleatoriamente entre los puertos disponibles. La respuesta a este comando es la dirección IP del equipo y el puerto.
TYPE	Este comando permite especificar el tipo de formato en el cual se enviarán los datos.
STRU	Carácter Telnet que especifica la estructura de archivos (F de <i>File [Archivo]</i> , R de <i>Record [Registro]</i> , P de <i>Page [Página]</i> ).
MODE	Carácter Telnet que especifica el método de transferencia de datos (S de <i>Stream [Flujo]</i> , B de <i>Block [Bloque]</i> , C de <i>Compressed [Comprimido]</i> ).
Comandos de servicio FTP	
Comando	Descripción
RETR	Este comando ( <i>RETRIEVE [RECUPERAR]</i> ) le pide al servidor de DTP una copia del archivo cuya ruta de acceso se da en los parámetros.
STOR	Este comando ( <i>store [almacenar]</i> ) le pide al servidor de DTP que acepte los datos enviados por el canal de datos y que los almacene en un archivo que lleve el nombre que se da en los parámetros. Si el archivo no existe, el servidor lo crea; de lo contrario, lo sobrescribe.
STOU	Este comando es idéntico al anterior, sólo le pide al servidor que cree un archivo cuyo nombre sea único. El nombre del archivo se envía en la respuesta.
APPE	Gracias a este comando ( <i>append [adjuntar]</i> ) los datos enviados se concatenan en el archivo que lleva el nombre dado en el parámetro si ya existe; si no es así, se crea.
ALLO	Este comando ( <i>allocate [reservar]</i> ) le pide al servidor que reserve un espacio de almacenamiento lo suficientemente grande como para recibir el archivo cuyo nombre se da en el argumento.
REST	Este comando ( <i>restart [reiniciar]</i> ) permite que se reinicie una transferencia desde donde se detuvo. Para hacer esto, el comando envía en el parámetro el marcador que representa la posición en el archivo donde la transferencia se había interrumpido. Después de este comando se debe enviar inmediatamente un comando de transferencia.
RNFR	Este comando ( <i>rename from [renombrar desde]</i> ) permite volver a nombrar un archivo. En los parámetros indica el nombre del archivo que se va a renombrar y debe estar inmediatamente seguido por el comando <i>RNTO</i> .
RNTO	Este comando ( <i>rename to [renombrar a]</i> ) permite volver a nombrar un archivo. En los parámetros indica el nombre del archivo que se va a renombrar y debe estar inmediatamente seguido por el comando <i>RNFR</i> .
ABOR	Este comando ( <i>abort [cancelar]</i> ) le indica al servidor de DTP que abandone todas las transferencias asociadas con el comando previo. Si no hay conexión de datos abierta, el servidor de DTP no realiza ninguna acción; de lo contrario, cierra la conexión. Sin embargo, el canal de control permanece abierto.

DELE	Este comando ( <i>delete [borrar]</i> ) permite que se borre un archivo, cuyo nombre se da en los parámetros. Este comando es irreversible y la confirmación sólo puede darse a nivel cliente.
RMD	Este comando ( <i>remove directory [eliminar directorio]</i> ) permite borrar un directorio. El nombre del directorio que se va a borrar se indica en los parámetros.
MKD	Este comando ( <i>make directory [crear directorio]</i> ) permite crear un directorio. El nombre del directorio que se va a crear se indica en los parámetros.
PWD	Este comando ( <i>print working directory [mostrar el directorio actual]</i> ) hace posible volver a enviar la ruta del directorio actual completa.
LIST	Este comando permite que se vuelva a enviar la lista de archivos y directorios presentes en el directorio actual. Esto se envía a través del DTP pasivo. Es posible indicar un nombre de directorio en el parámetro de este comando. El servidor de DTP enviará la lista de archivos del directorio ubicado en el parámetro.
NLST	Este comando ( <i>name list [lista de nombres]</i> ) permite enviar la lista de archivos y directorios presentes en el directorio actual.
SITE	Este comando ( <i>site parameters [parámetros del sistema]</i> ) hace que el servidor proporcione servicios específicos no definidos en el protocolo FTP.
SYST	Este comando ( <i>system [sistema]</i> ) permite el envío de información acerca del servidor remoto.
STAT	Este comando ( <i>Estado: [estado]</i> ) permite transmitir el estado del servidor; por ejemplo, permite conocer el progreso de una transferencia actual. Este comando acepta una ruta de acceso en el argumento y después devuelve la misma información que LISTA pero a través del canal de control.
HELP	Este comando permite conocer todos los comandos que el servidor comprende. La información se devuelve por el canal de control.
NOOP	Este comando ( <i>no operations [no operación]</i> ) sólo se utiliza para recibir un comando OK del servidor. Sólo se puede utilizar para no desconectarse después de un período de inactividad prolongado.

## Las respuestas FTP

Las respuestas FTP garantizan la sincronización entre el cliente y el servidor FTP. Por lo tanto, por cada comando enviado por el cliente, el servidor eventualmente llevará a cabo una acción y sistemáticamente enviará una respuesta.

Las respuestas están compuestas por un código de 3 dígitos que indica la manera en la que el comando enviado por el cliente ha sido procesado. Sin embargo, debido a que el código de 3 dígitos resulta difícil de leer para las personas, está acompañado de texto (cadena de caracteres Telnet separada del código numérico por un espacio).

Los códigos de respuesta están compuestos por 3 números, cuyos significados son los siguientes:

- El primer número indica el estado de la respuesta (exitosa o fallida)
- El segundo número indica a qué se refiere la respuesta.
- El tercer número brinda un significado más específico (relacionado con cada segundo dígito).

Primer número		
Dígito	Significado	Descripción
1yz	Respuesta preliminar positiva	La acción solicitada está en progreso. Se debe obtener una segunda respuesta antes de enviar un segundo comando.
2yz	Respuesta de finalización positiva	La acción solicitada se ha completado y puede enviarse un nuevo comando.
3yz	Respuesta intermedia positiva	La acción solicita está temporalmente suspendida. Se espera información adicional del cliente.
4yz	Respuesta de finalización negativa	La acción solicitada no se ha realizado debido a que el comando no se ha aceptado temporalmente. Se le solicita al cliente que intente más tarde.
5yz	Respuesta permanente negativa	La acción solicitada no se ha realizado debido a que el comando no ha sido aceptado. Se le solicita al cliente que formule una solicitud diferente.
Segundo número		
Dígito	Significado	Descripción
x0z	Sintaxis	La acción tiene un error de sintaxis o sino, es un comando que el servidor no comprende.
x1z	Información	Ésta es una respuesta que envía información (por ejemplo, una respuesta a un comando STAT).
x2z	Conexiones	La respuesta se refiere al canal de datos.
x3z	Autenticación y cuentas	La respuesta se refiere al inicio de sesión (USUARIO/CONTRASEÑA) o a la solicitud para cambiar la cuenta (CPT).
x4z	No utilizado por el protocolo FTP.	
x5z	Sistema de archivos	La respuesta se relaciona con el sistema de archivos remoto.

## Analizando FTP

Lance en su PC-SC el analizador de protocolos Ethereal y póngalo a capturar tramas Ethernet. Aplique el siguiente filtro: `ip.src==10.1.1.XY` o `ip.dst==10.1.1.XY`

Inicie una sesión FTP en su máquina virtual. Consulte el manual del comando `ftp` (`man ftp`) y averigüe la utilidad de la opción `-d`. Utilícela.

Transfiera un archivo del equipo local PC-SC al equipo remoto(máquina virtual).

Renombre dicho archivo transferido a la máquina virtual y por último elimínelo.

¿Qué comandos FTP ha utilizado?

Pare la captura de Ethereal y analice cada uno de estos comandos, así como los protocolos involucrados en la sesión FTP. ¿Puede ver la contraseña en la captura obtenida?

Guarde la captura en su PC-SC.

Vuelva a lanzar Ethereal. Aplique los mismos filtros.

Conéctese mediante Telnet al servicio FTP de su máquina virtual.

Mediante los comandos apropiados, al menos, cree un directorio y bórralo.

Detenga la captura de Ethereal y compárela con la guardada anteriormente. ¿Qué diferencias encuentra? ¿Cuál es el mayor inconveniente que observa en el protocolo FTP? ¿Conoce alguna RFC en la que se incremente la seguridad de dicho protocolo? ¿Qué protocolo usaría para una transferencia segura de archivos? Repita la operación anterior mediante dicho protocolo. ¿Cuál es ahora la diferencia respecto FTP?

**Checkpoint 7.2: Muestra al profesor de prácticas los puertos que utiliza el servicio FTP tanto en el cliente como en el servidor. ¿Cuántos puertos se utilizan? ¿Por qué?**

Encuentre la diferencia entre los modos activo y pasivo de un servidor FTP. ¿Qué utilidad tienen? ¿Cuál considera que es más indicado para garantizar la compatibilidad en la arquitectura cliente-servidor?