Castillo Martínez Leonel Jafet

Protocolos de capa de aplicación

TELNET (Telecomunication Network)

El protocolo TELNET tiene como propósito proporcionar un servicio de comunicaciones orientado a bytes de 8 bit general y bidireccional. Y su objetivo principal es permitir un método estándar de interfaz de dispositivos terminales y procesos orientados a terminales entre sí. Dicho servicio se puede resumir como una interfaz estandarizada, por medio de la cual un programa de un usuario cliente puede acceder a los recursos de otro usuario.

Al ser un protocolo perteneciente a la capa de aplicación, utiliza la capa de transporte para enviar y recibir paquetes. En lo que a establecimiento de sesión respecta, para el establecimiento de una conexión cliente-servidor, se utiliza el **puerto 23** por defecto. La conexión TCP se establece entre el **puerto U del usuario** y el **puerto L del servidor**.

Este protocolo se basa en tres ideas principales:

- El concepto “Terminal Virtual de Red”: Cuando se inicia una conexión TELNET, se supone que se inicia y finaliza en un NVT, el cual es un dispositivo imaginario que proporciona una representación intermedia de un terminal. Esto elimina la necesidad para los ordenadores “Servidor” y “Usuario” de guardar información de las características del terminal del otro y de las convenciones para manejarlo. Ambos mapean las características del dispositivo local para que a través de la red parezca un NVT y ambos pueden asumir un mapeado similar en el otro extremo. Se pretende que el NVT sea algo intermedio entre ser muy restringido y ser demasiado exigente.
- Principio de opciones negociadas: El principio de las opciones negociadas toma en cuenta el hecho de que muchos hosts desearán proporcionar servicios adicionales además de los disponibles dentro de un NVT, y muchos usuarios tendrán terminales sofisticados y desearían tener servicios elegantes, en lugar de los mínimos. Independientes de, pero aún estructuradas dentro del protocolo TELNET hay varias “opciones” que se votarán y que pueden ser usadas con la estructura “**DO, DON’T, WILL, WON’T**” para permitir al usuario y servidor estar de acuerdo con usar convenciones más elaboradas (o tal vez sólo diferentes) para sus conexiones TELNET. Entre esas opciones se podrían incluir el cambio de set de caracteres, el modo de eco, etc.
- Visión simétrica de terminales y procesos: La simetría de la sintaxis de negociación puede potencialmente llevar a bucles infinitos de reconocimiento. Para evitar esto, prevalecen las siguientes normas.
    - Las partes solo pueden solicitar un cambio del estado de una opción.
    - Si una parte recibe lo que parece una petición para entrar en algún modo en el que ya se encuentra, **la petición no debería reconocerse**. No hacerlo es esencial par evitar bucles infinitos en la negociación.
    - Siempre que una parte envíe una orden de opción a la otra, ya sea una petición o un reconocimiento, y el uso de la opción va a tener algún efecto en el procesado de los datos enviados de la primera parte a la segunda, dicha orden se debe enviar en el punto donde se desee que comience a tener efecto.

En la medida de lo posible, el protocolo TELNET se ha hecho simétrico entre el servidor y el usuario para que de forma natural se adapte a conexiones usuario-usuario y servidor-servidor. Se espera, pero no se requiere en absoluto, que las opciones fomenten la simetría. En cualquier caso, se acepta explícitamente que la simetría es un principio operativo más que una regla inamovible.

El protocolo Telnet ofrece un sistema de negociaciones de opciones que permite el uso de funciones avanzadas en forma de opciones, en ambos lados, al iniciar solicitudes para su autorización desde el sistema remoto.

En los inicios de Internet, la red ARPANET estaba conformada por equipos como teclados, juegos de caracteres, resoluciones. Cuyas configuraciones eran poco homogéneas. Además, las sesiones entre terminales también tenían su propia forma de controlar el flujo de datos de entrada y salida.

Como alternativa a esto, se decidió desarrollar una interfaz estándar denominada NVT. Con la cual se proporcionó una base de comunicación estándar, compuesta de caracteres ASCII de 7 bits, a los cuales se les agregó el código ASCII extendido; tres caracteres de control, cinco caracteres de control opcionales, un juego de señales de control básicas. Con base en esto, TELNET consistió en crear una abstracción del terminal que permite a cualquier host comunicarse con otro sin conocer previamente sus características.

Muchos sistemas de tiempo compartido ofrecen mecanismos para permitir a un terminal recuperar el control de un proceso en ejecución; las funciones IP y AO son ejemplos de estos mecanismos. En sistemas como esos, usados localmente, se tiene acceso a todas las señales generadas por el usuario, ya sean estas caracteres normales o señales "fuera de banda" especiales como las generadas por la tecla "BREAK". Esto no siempre se cumple cuando el terminal se conecta al sistema a través de la red; los mecanismos de control de flujo de la red pueden provocar que una señal de este tipo se almacene en cualquier otra parte, por ejemplo, en el ordenador del usuario. Para evitar este problema, se ha creado el mecanismo "Synch" de TELNET. Una señal Synch está formada por una notificación urgente de TCP junto con la orden TELNET de MARCA DE DATOS. Para utilizar telnet, es muy importante tener en cuenta que el cliente telnet dispone de dos formas de operación:

- En modo comando permite utilizar una serie de órdenes que afectan al modo de operación, incluyendo conectar y desconectar.
- En modo normal nuestro equipo se comporta como si fuese un teclado (modificado y remoto) del ordenador al que estamos conectados. Cada pulsación de tecla es enviada al equipo remoto, y lo que vemos en la pantalla es realmente el eco que, en respuesta a esa señal, nos envía el equipo remoto.

Establecimiento de sesión

Para llevar a cabo esta conexión, lo primero que se tiene que hacer es llevar a cabo la autenticación de usuario, para lo cual se necesitan proporcionar los siguientes datos:

- Nombre: Se refiere a la Dirección IP del Servidor, ya sea en forma numérica o bien en forma de URL. Por ejemplo: 192.168.0.1 o ltu.jovenclub.cu
- Número de puerto: Indica el servicio concreto que se desea acceder dentro del servidor. Una dirección telnet con un número adicional de puerto, permite no

solamente acceder al ordenador remoto, sino también acceder a un servicio o programa específico dentro de él.

- Tipo de terminal: La identificación del terminal es un acuerdo que utilizan tanto la máquina cliente como el servidor, referente a ciertas características y secuencias a utilizar. De esta forma se utiliza un terminal virtual, con independencia de cuál sea realmente el terminal físicamente utilizado. Desde el punto de vista del cliente, la identificación debe hacerse en dos partes.
    - a: Indicando al programa cliente que se comporte como una terminal virtual de cierto tipo.
    - b: Informando al servidor qué tipo se terminal se utilizará. Consiste en una cadena de caracteres que se envía al host durante el proceso de negación de la conexión.
- Login: Conformado por un nombre de usuario. Aunque en algunos servicios no es necesario llevar a cabo esta autenticación dado que el servidor no lo exige para ese servicio en concreto.
- Password: Será la contraseña que autentica al usuario, aunque existen servicios que no necesitan este dato.
- Log-off: Se refiere al cierre remoto de la sesión, este paso es necesario antes de llevar a cabo la desconexión física, dado que no se desea mantener una sesión abierta de forma indefinida en el host anfitrión.

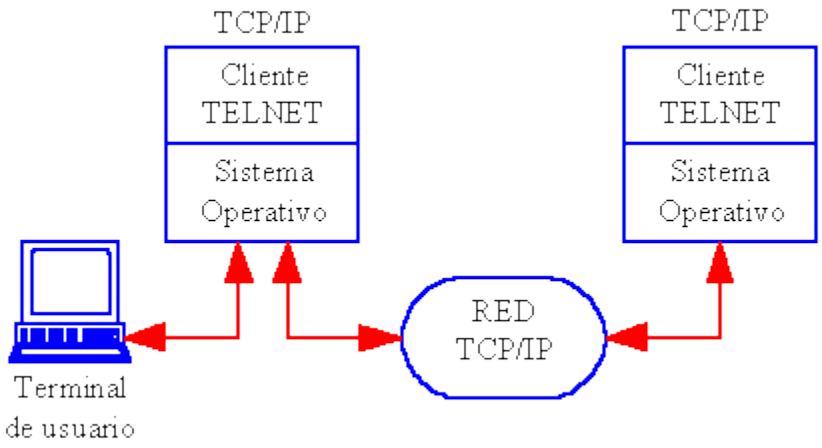

Figura 1. Esquema de conexión TCP para una aplicación TELNET.

TELNET es un servicio que está disponible en varias plataformas, tales como Windows, UNIX y Linux. Un ejemplo de la utilización de comandos de usuario para este servicio es el siguiente:

```
telnet myhost.com
```

Figura 2. Intento de conexión a host remoto.

Aquí se intenta abrir una conexión al host remoto myhost.com. Si se establece una conexión, el host solicitará un nombre de usuario y contraseña.

```
telnet -l myusername myhost.com 5555
```

Figura 3. Intento de conexión a host remoto con puerto 5555.

En el comando mostrado en la figura 3 se intenta abrir una conexión al host remoto myhost.com en el puerto 5555, usando el nombre de usuario myusername. Si tiene éxito, el host le pedirá la contraseña de myusername.

Suponiendo que la plataforma utilizada es UNIX, algunos de los comandos ejecutables son:

Tabla 1. Comandos de servicio TELNET utilizables.

Comando	Descripción
?	Mostrar ayuda
close	Cerrar sesión Telnet
display	Mostrar la configuración de la conexión en pantalla (tipo de terminal y puerto)
entorno	Para definir las variables del entorno del sistema operativo
logout	Para cerrar la sesión
mode	Cambia entre los modos de transferencia ASCII (transferencia de un archivo como texto) y los modos BINARIOS (transferencia de un archivo en modo binario)
open	Abre otra conexión de la actual
quit	Sale de la aplicación Telnet
set	Cambia la configuración de conexión
unset	Carga la configuración de conexión predeterminada

FTP (File Transfer Protocol)

El protocolo de transferencia de archivos permite cargar o descargar archivos de un servidor a través de un cliente. Los objetivos del FTP son

- Promocionar el uso compartido de ficheros
- Animar el uso indirecto o implícito de servidores remotos
- Hacer transparente al usuario las variaciones entre la forma de almacenar ficheros en diferentes ordenadores
- Transferir datos fiable y eficientemente.

El FTP, aunque puede ser utilizado directamente por un usuario en un terminal, está diseñado principalmente para ser usado por programas.

El proceso de transferencia pasiva de datos “escucha” en el puerto de datos hasta que recibe una conexión del proceso de transferencia activa para abrir la conexión de datos. El servicio FTP es ofrecido por la capa de aplicación y utiliza la capa de transporte para enviar y recibir datos. Utiliza normalmente los **puertos 20 y 21**.

El intérprete de protocolo del servidor “escucha” en el **puerto L** hasta que recibe una conexión de un user-PI y establece una conexión de comunicaciones para control. Recibe órdenes FTP estándar desde el user-PI, envía respuestas y controla el server-DTP.

El tipo de representación de datos usado para transferir y almacenar los datos. El tipo implica ciertas transformaciones a la hora de almacenar y enviar los datos. Los tipos de representación definidos en el FTP se describen en la sección titulada “Estableciendo conexiones de datos”.

FTP admite dos modos de conexión del cliente:

- Activo: o Estándar, PORT, debido a que el cliente envía comandos tipo PORT al servidor por el canal de control al establecer la conexión. En este modo 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.
  Lo anterior tiene un grave problema de seguridad, y es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024, con los problemas que ello implica si tenemos el equipo conectado a una red insegura como Internet.

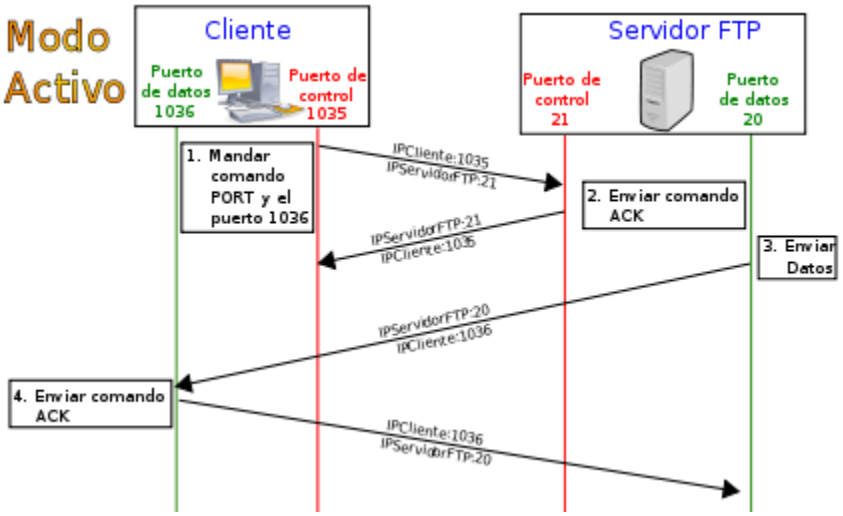

Figura 4. Ilustración modo de conexión activo.

- Pasivo: O PASV, dado que en este caso se envían comandos de tipo PASV. 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) al que debe conectarse el cliente. El cliente inicia una conexión desde el puerto siguiente al puerto de control hacia el puerto del servidor especificado anteriormente.
  Antes de cada nueva transferencia tanto en el modo Activo como en el Pasivo, el cliente debe enviar otra vez un comando de control (PORT o PASV, según el modo en el que haya conectado), y el servidor recibirá esa conexión de datos en un nuevo puerto (aleatorio si es en modo pasivo o por el puerto 20 si es en modo activo).

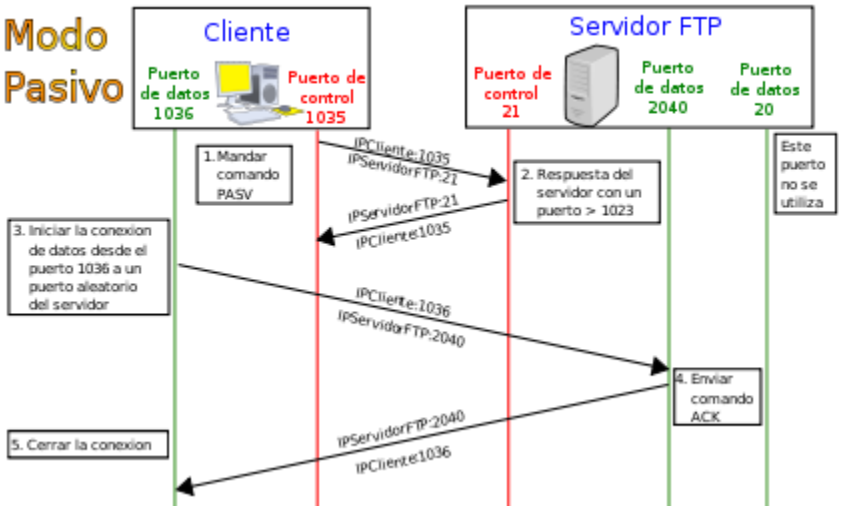

Figura 5. Ilustración modo de conexión pasivo.

En ambos modos el cliente establece una conexión con el cliente mediante el puerto 21 que establece el canal de control.

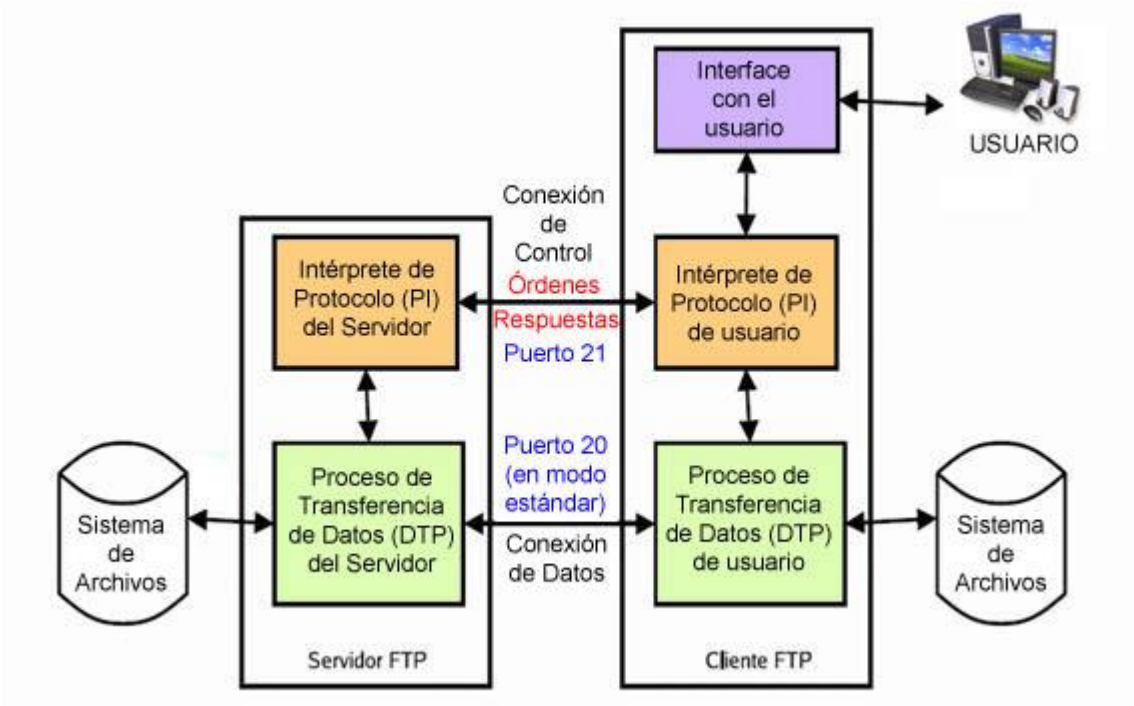

Figura 6. Diagrama de un servicio FTP.

Servidor FTP

Un servidor FTP es un programa especial que se ejecuta en un equipo servidor normalmente conectado a Internet (aunque puede estar conectado a otros tipos de redes). Su función es permitir el intercambio de datos entre diferentes servidores/ordenadores. Un servidor FTP se compone de un intérprete de protocolo (PI) de usuario y proceso de transferencia de datos (DTP) de usuario.

Cliente FTP

Un cliente FTP no es otra cosa que un software que se instala en la máquina del usuario y permite conectar mediante el protocolo FTP hacia el servidor FTP remoto que está en Internet. Se compone de un intérprete de protocolo, de un proceso de transferencia de datos de usuario y de una interfaz con el usuario.

Tipos de transferencia de archivos en FTP

Es importante conocer cómo debemos transportar un archivo a lo largo de la red. Si no utilizamos las opciones adecuadas podemos destruir la información del archivo.

- Tipo ASCII: Adecuado para transferir archivos que sólo contengan caracteres imprimibles (archivos ASCII, no archivos resultantes de un procesador de texto), por ejemplo, páginas HTML, pero no las imágenes que puedan contener.
- Tipo Binario: Este tipo es usado cuando se trata de archivos comprimidos, ejecutables para PC, imágenes, archivos de audio, etc.

Tabla 2. Ejemplos de cómo transferir tipos de archivo dependiendo su extensión.

Extensión de archivo	Tipo de transferencia
txt (texto)	ascii
html (página WEB)	ascii
doc (documento)	binario
ps (poscript)	ascii
hqx (comprimido)	ascii
Z (comprimido)	binario
ZIP (comprimido)	binario
ZOO (comprimido)	binario
Sit (comprimido)	binario
pit (comprimido)	binario
shar (comprimido)	binario
uu (comprimido)	binario
ARC (comprimido)	binario
tar (empaquetado)	binario

Para llevar a cabo un inicio de sesión en un sistema remoto FTP, el comando ftp abre la interfaz de usuario al protocolo de transferencia de archivos de internet. Esta interfaz de usuario, denominada intérprete de comandos, le permite iniciar sesión en un sistema remoto y realizar distintas operaciones con su sistema de archivos. Las operaciones principales se resumen en la siguiente tabla.

Tabla 3. Comandos FTP esenciales.

Comando	Descripción
`ftp`	Accede al intérprete de comandos `ftp`.
`ftp` *sistema remoto*	Establece una conexión `ftp` a un sistema remoto. Para obtener instrucciones, consulte Cómo abrir una conexión `ftp` a un sistema remoto.
`open`	Inicia sesión en el sistema remoto desde el intérprete de comandos.
`close`	Cierra la sesión del sistema remoto y vuelve al intérprete de comandos.
`bye`	Sale del intérprete de comandos `ftp`.
`help`	Muestra todos los comandos `ftp` o, si se proporciona un nombre de comando, se describe brevemente lo que hace el comando.
`reset`	Vuelve a sincronizar la secuenciación de respuesta de comando con el servidor `ftp` remoto.
`ls`	Muestra los contenidos del directorio de trabajo remoto.
`pwd`	Muestra el nombre del directorio de trabajo remoto.
`cd`	Cambia el directorio de trabajo remoto.
`lcd`	Cambia el directorio de trabajo local.
`mkdir`	Crea un directorio en el sistema remoto.
`rmdir`	Elimina un directorio en el sistema remoto.
`get, mget`	Copia un archivo (o varios archivos) del directorio de trabajo remoto al directorio de trabajo local.
`put, mput`	Copia un archivo (o varios archivos) del directorio de trabajo local al directorio de trabajo remoto.
`delete, mdelete`	Elimina un archivo (o varios archivos) del directorio de trabajo remoto.

Ahora bien, para llevar a cabo una conexión FTP a un sistema remoto por medio de la línea de comandos se deben seguir los siguientes pasos:

1. Asegurarse de tener autenticación FTP.
2. Abrir una conexión a un sistema remoto utilizando el comando *ftp*.

```
$ ftp remote-system
```

3. Escribir el nombre de usuario.

```
Name (remote-system:user-name): user-name
```

4. En caso necesario, especificar la contraseña.

```
331 Password required for user-name:
Password: password
```

Si el sistema al que accede tiene una cuenta ftp anónima establecida, se le solicita una dirección de correo electrónico para la contraseña. Si la interfaz ftp acepta la contraseña, muestra un mensaje de confirmación y el indicador (*ftp*>).

Códigos de respuesta

El código de respuesta es un valor de tres dígitos. El primer dígito se utiliza para indicar una de tres posibles resultados-el éxito, el fracaso o para indicar un error o una respuesta incompleta:

- 2yz - respuesta Éxito
- 4yz o 5yz - No hay respuesta
- 1yz o 3yz - Un error o una respuesta incompleta

El segundo dígito define la clase de error:

- x0z - Sintaxis. Estas respuestas se refieren a errores de sintaxis.
- x1z - Información. Las respuestas a las solicitudes de información.
- x2z - Conexiones. Respuestas en referencia al control y las conexiones de datos.
- x3z - Autenticación y contabilidad. Respuestas para el proceso de inicio de sesión y los procedimientos contables.
- x4z - No definido.
- x5z - Sistema de archivos. Estas respuestas transmiten códigos de estado del sistema de archivos del servidor.

El tercer dígito del código de respuesta se utiliza para proporcionar detalles adicionales para cada una de las categorías definidas por el segundo dígito.

HTTP (Hypertext Transfer Protocol)

El Protocolo de transferencia de hipertexto (HTTP) es una aplicación sin estado. protocolo de nivel para información distribuida, colaborativa, de hipertexto sistemas.

Cada mensaje de este protocolo es una solicitud o respuesta. Un servidor escucha en una conexión de una solicitud, analiza cada mensaje recibido, interpreta la semántica del mensaje en relación con el objetivo de solicitud identificado, y responde a esa solicitud con uno o más mensajes de respuesta. Un cliente construye solicitudes de mensaje para comunicar intenciones específicas, examina respuestas recibidas para ver si las intenciones fueron llevadas a cabo y la determina cómo interpretar los resultados.

Al igual que los demás protocolos, HTTP utiliza la capa de transporte para realizar la transferencia y recepción de datos.

La comunicación HTTP usualmente toma lugar a través de conexiones TCP/IP. Y el puerto que se utiliza de forma predeterminada es el puerto TCP 80.

Localizador Uniforme de Recursos (URL)

Las son direcciones únicas que sirven para localizar una página Web y sus contenidos en un servidor de la red.

Estas direcciones se escriben en la barra de direcciones del navegador para que éste envíe una solicitud por la red y como culminación del proceso, sea posible visualizar en los ordenadores personales el contenido de un sitio alojado en un servidor remoto.

Transacción HTTP

Una transacción HTTP está formada por los siguientes componentes:

- Conexión: El cliente realiza una conexión al servidor.
- Solicitud: El cliente solicita información del servidor.
- Respuesta: El servidor elige proveer la información al cliente o negarse a proveer dicha información.
- Cierre: Alguna de las dos entidades termina la transacción.

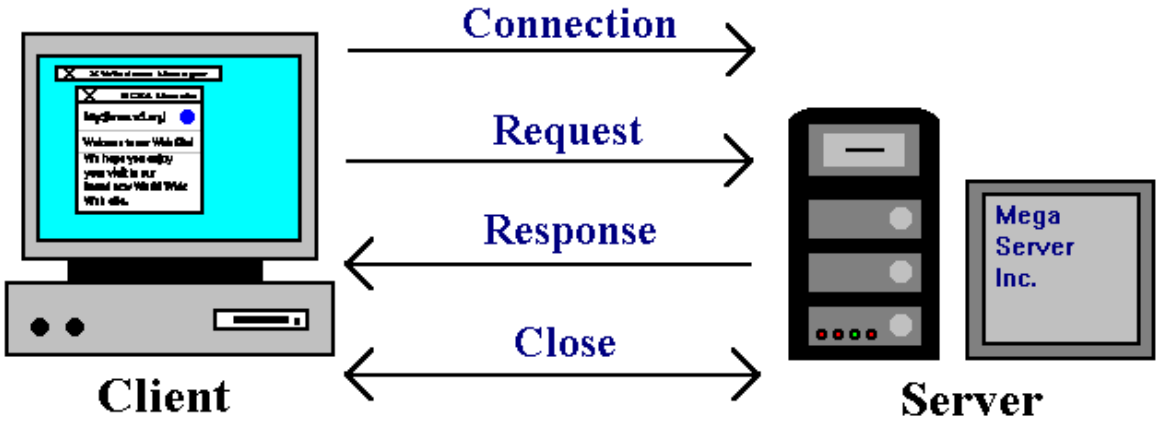

Figura 7. Etapas de una Transacción HTTP.

Protocolo sin estado (Stateless)

HTTP es un protocolo sin estado, lo que significa que trata cada petición como una transacción independiente que no tiene relación con cualquier solicitud anterior, de modo que la comunicación se compone de pares independientes de solicitud y respuesta. Un protocolo Stateless no requiere que el servidor retenga información de la sesión o de estado acerca de cada socio de las comunicaciones durante el tiempo que duren múltiples peticiones. En contraste, un protocolo que requiere el mantenimiento del estado interno en el servidor se conoce como un protocolo **con estado**.

El diseño sin estado simplifica el diseño del servidor porque no hay necesidad de asignar dinámicamente almacenamiento para tratar las conversaciones en curso. Si un cliente desaparece en medio de una transacción, ninguna parte del sistema tiene que ser responsable de limpiar el estado actual del servidor.

Ahora bien, una desventaja de este protocolo es que puede ser necesario incluir información adicional en cada petición, y esta información adicional necesitará ser interpretada por el servidor.

Como se mencionó anteriormente, en HTTP existen dos tipos de mensajes:

- Peticiones: Son mensajes enviados por un cliente, para iniciar una acción en el servidor. Su línea de inicio está formada por tres elementos.
    - Un método HTTP, un verbo como GET, PUT o POST, o bien un nombre como HEAD, o Options, que describan la acción que se pide sea realizada.
    - El objetivo de una petición: normalmente es una URL, o la dirección completa del protocolo, puerto y dominio también suelen ser especificados por el contexto de la petición. El formato del objetivo de la petición varía según los métodos HTTP puede ser:
        - Una dirección absoluta.
        - Una URL completa.
        - El componente de autoridades de una URL.
        - El formato de asterisco.
    - La versión de HTTP, la cual define la estructura de los mensajes, actuando como indicador de la versión que espera se use para la respuesta.
- Respuestas: Una vez que el servidor recibe una solicitud HTTP del cliente, devuelve una respuesta HTTP. Esencialmente sus campos son:
    - La versión del protocolo HTTP utilizada.
    - Un código de estado, indicando si la petición ha sido exitosa o no.
    - Un mensaje de estado describiendo brevemente el código de estado.
    - Cabeceras HTTP.
    - Cuerpo: La última parte del mensaje de respuesta es el cuerpo, aunque no todas las respuestas tienen uno, respuestas con un código de estado como 201 o 204 normalmente prescinden de él. De forma general, los cuerpos se pueden diferenciar en tres categorías
        - Cuerpos con un único dato: Consisten en un simple archivo, de longitud conocida y definido en las cabeceras.
        - Cuerpos con un único dato. Consisten en un simple archivo, de longitud desconocida y codificado en partes.

- Cuerpos con múltiples datos: Consisten en varios archivos, cada uno con una sección distinta de información.

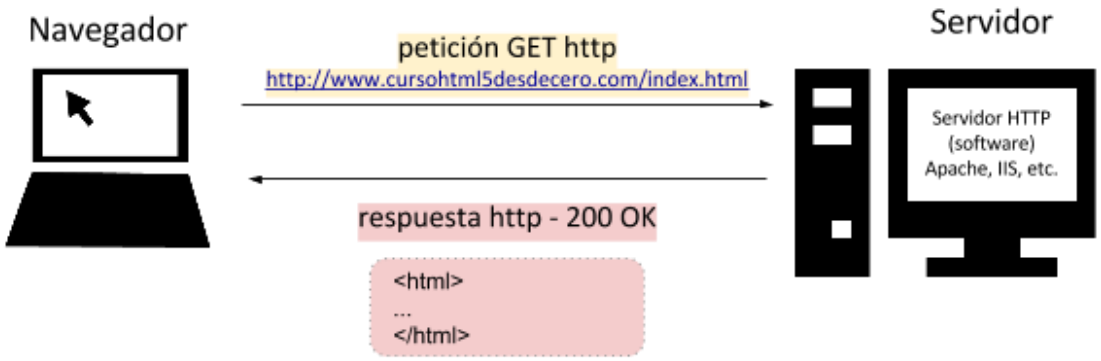

Figura 8. Ejemplo petición y respuesta HTTP.

Métodos

HTTP define un conjunto de **métodos de petición** para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados *HTTP verbs*.

- GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.
- HEAD: El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.
- POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.
- PUT: El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.
- DELETE: El método DELETE borra un recurso en específico.
- CONNECT: El método CONNECT establece un túnel hacia el servidor identificado por el recurso.
- OPTIONS: El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.
- TRACE: El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.
- PATCH: El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

Códigos de respuesta

es un número que indica que ha pasado con la petición. El resto del contenido de la respuesta dependerá del valor de este código. Cada código tiene un significado concreto. Sin embargo, el número de los códigos están elegidos de tal forma que según si pertenece a una centena u otra se pueda identificar el tipo de respuesta que ha dado el servidor:

- Códigos con formato 1xx: Respuestas informativas.
- Códigos con formato 2xx: Respuestas correctas.
- Códigos con formato 3xx: Respuestas de redirección.
- Códigos con formato 4xx: Errores causados por el cliente.
- Códigos con formato 5xx: Errores provocados por el servidor.

Conexiones persistentes y no persistentes

El protocolo HTTP puede utilizar conexiones persistentes y no persistentes. Como ejemplo si pedimos una página web a un servidor y la página consta de un HTML y 5 objetos, en una conexión persistente solo se hará una conexión TCP, mientras que en una conexión no persistente se utilizarán múltiples conexiones TCP, una por cada objeto solicitado.

Estas conexiones pueden ser paralelas para mejorar el rendimiento, por lo que un navegador puede realizar x conexiones al mismo tiempo en vez de ir realizando una conexión tras otra (en serie), que habitualmente alargaría el tiempo de conexión.

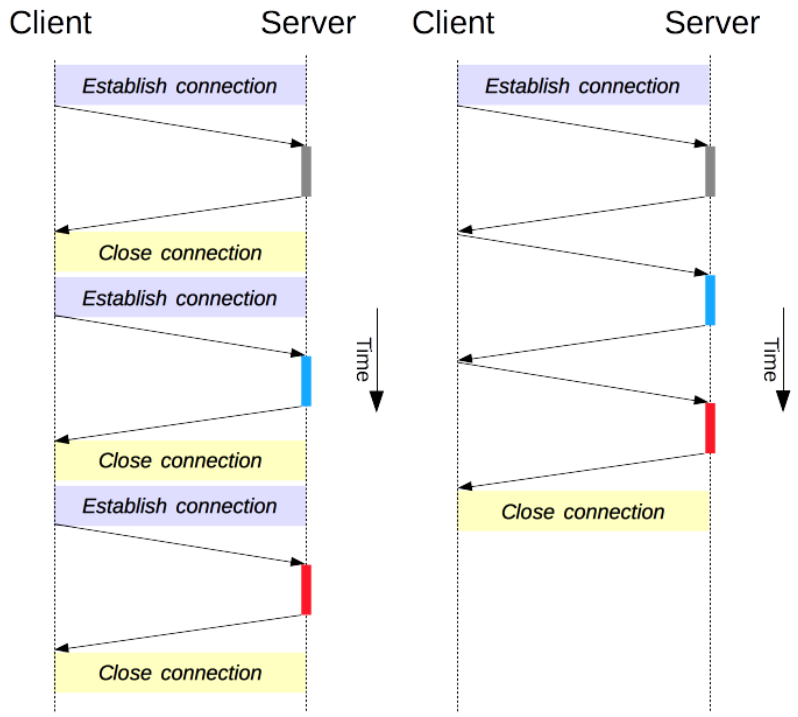

Figura 9. Comparación de conexión persistente y conexión no persistente.

Cookies

HTTP cookies, es una tecnología que en actualmente está definida en el estándar RFC 6265, y que consiste básicamente en información enviada o recibida en las cabeceras HTTP y que queda almacenada localmente client-side durante un determinado tiempo.

En otras palabras, es información que queda almacenada en el dispositivo del usuario y que se envía desde y hacia el servidor web en las cabeceras HTTP.

Cuando un usuario solicita una página web (o cualquier otro recurso), el servidor envía el documento, cierra la conexión y se olvida del usuario. Si el mismo usuario vuelve a solicitar la misma u otra página al servidor, será tratado como si fuera la primera solicitud que realiza. Esta situación puede suponer un problema en muchas situaciones y las cookies son una técnica que permite solucionarlo.

Con las cookies, el servidor puede enviar información al usuario en las cabeceras HTTP de respuesta y esta información queda almacenada en el dispositivo del usuario. En la siguiente solicitud que realice el usuario la cookie es enviada de vuelta al servidor en las cabeceras HTTP de solicitud.

Figura 10. Ejemplo de un envío de cookies HTTP.

Servidor Proxy

El proxy HTTP es un filtro de contenido de alto rendimiento. Examina el tráfico web para identificar contenido sospechoso que puede ser un virus u otro tipo de intrusión. También puede proteger de ataques a su servidor HTTP.

Con un filtro de proxy HTTP, es posible:

- Ajustar los tiempos de espera y los límites de duración de las solicitudes y respuestas HTTP para evitar el mal desempeño de la red, como también varios ataques.
- Personalizar el mensaje de negación que los usuarios ven cuando intentan conectarse a un sitio web bloqueado por el proxy HTTP.
- Filtrar tipos MIME de contenido web.

- Bloquear patrones de ruta y URL especificados.
- Negar cookies de sitios web especificados.

SMTP, POP3 y IMAP4

El correo electrónico en la actualidad es una de las herramientas más utilizadas de comunicación que existen.

Podemos enviar correos electrónicos gracias a los protocolos de la pila TCP/IP llamados SMTP, POP3 e IMAP4.

SMTP (Simple Mail Transfer Protocol)

El objetivo del Protocolo de Transferencia Simple de Correo es transferir correo de manera confiable y eficiente.

SMTP es independiente del subsistema de transmisión particular y requiere solo un canal de flujo de datos ordenado confiable.

Por defecto, este protocolo trabaja con 3 puertos:

- 25: Este es el predeterminado para SMTP no cifrado.
- 2525: Puerto alternativo para SMTP no cifrado.
- 465: Este es el puerto que se debe utilizar si se quiere conectar utilizando SMTP de forma segura (Cifrada).

POP3 (Post Office Protocol)

es un protocolo standard de correo que se usa para la recepción de correo desde un servidor remoto a un cliente de correo local. POP3 te permite descargar los mensajes de correo en tu ordenador local y leerlos cuando estés offline.

De manera predeterminada, POP3 trabaja con 2 puertos:

- 110: Es el puerto predeterminado para POP3 no cifrado.
- 995: Este puerto es el que se utiliza cuando se desea conectar usando POP3 de forma cifrada.

Generalmente POP3 recupera los mensajes de los clientes de la siguiente forma:

- Conecta al servidor de correo con el puerto 110 o 995.
- Recupera el correo electrónico.
- Se deshace de las copias de los mensajes almacenados en el servidor.
- Se desconecta del servidor.

Los clients POP3 pueden estar configurados para permitir al servidor continuar almacenando algunas copias del correo descargado.

Castillo Martínez Leonel Jafet

IMAP4 (Internet Message Access Protocol)

El Protocolo de Acceso a Mensajes de Internet, permite a un cliente acceder y manipular mensajes de correo electrónico en un servidor. IMAPv4 permite la manipulación de buzones (folders de mensajes remotos) en una forma que es funcionalmente equivalente a las carpetas locales. IMAP4 también provee al cliente la capacidad de re-sincronizarse de manera offline con el servidor.

IMAPv4 incluye operaciones para la creación, eliminación, renombrar carpetas de mensajes, acceder a mensajes nuevos, remover mensajes de forma permanente, configurar y borrar banderas, analizar, buscar selectivamente atributos de mensajes, textos y partes de estos. En IMAPv4 se accede a los mensajes por medio de números. Estos números pueden ser secuencias numéricas o identificadores únicos.

De manera predeterminada, el protocolo IMAPv4 trabaja con 2 puertos:

- 143: Es el predeterminado para IMAP no cifrado.
- 993: Es el puerto que se debe utilizar si se desea realizar una conexión usando IMAP de forma cifrada.

El funcionamiento de IMAPv4 se generalmente es:

- Conexión al servidor de correo por medio del puerto 143 o 993.
- Recupera el mensaje de correo electrónico.
- Se mantiene conectado hasta que la aplicación del cliente de correo cierre la conexión y descargue los mensajes deseados.

Los mensajes no son eliminados del servidor, pero esto implica otros elementos.

Para llevar a cabo el establecimiento de sesión, se consideran los siguientes pasos:

- El Servidor SMTP del emisor se comunica con un Servidor DNS (Sistema de Nombre de Dominio) y le pregunta si le puede brindar la dirección IP del SMTP del receptor.
- DNS responde con una o más direcciones IP.
- El Cliente de Correo se conecta con el Servidor SMTP y le comunica la dirección del remitente, la del destinatario y el cuerpo del mensaje.
- El SMTP toma la dirección del receptor y lo divide en dos partes:
    - El nombre
    - El sistema de nombres de dominio.
- El SMTP del receptor verifica en su base de datos si existe una cuenta con el nombre establecido anteriormente
- Si existe, acepta el email y lo deposita en el buzón correspondiente
- Por último, el usuario puede leer o descargar ese mensaje en su PC mediante POP o IMAP.

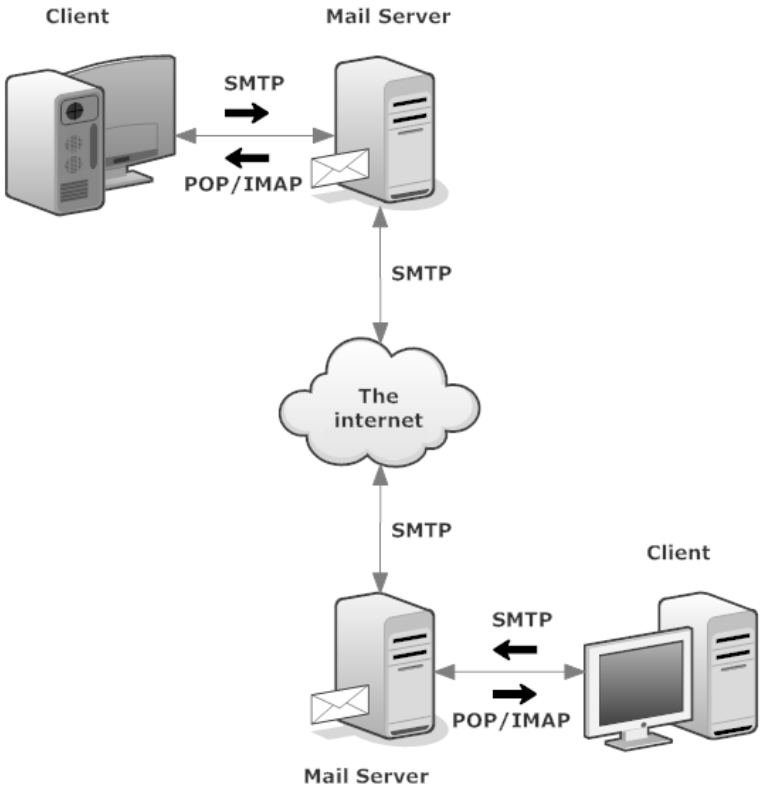

Figura 11. Establecimiento de sesión.

Un sistema de correo electrónico consta de los siguientes Agentes:

- Agentes de Usuario (UA): Responsable de proporcionar los servicios de edición de texto y su presentación al usuario final.
- Agentes de Transferencia de Mensajes (MTA): Está orientado al encaminamiento en concreto del correo electrónico.
- Agentes de Acceso a Mensajes (MAA): Es el sistema que se encarga del acceso al correo almacenado.

Ventajas y desventajas de POP3 e IMAP4.

POP3

- Ventajas:
  - La posibilidad de acceder al correo sin necesidad de estar conectado a la red, puesto que los descarga en el ordenador.
  - No es necesario tener un gran espacio de almacenamiento en el servidor de correo electrónico.
- Desventajas:
  - La acumulación de mensajes puede tener como consecuencia la incapacidad del usuario de recibir nuevos mensajes en el buzón.
  - Dependiendo del mensaje que se pretende enviar, se pueden consumir recursos del sistema.

IMAP4

- Ventajas:
  - Los correos permanecen en todo momento en el servidor.
  - Comunicación bidireccional entre el servidor de correo electrónico y el cliente de correo electrónico.
- Desventajas:
  - No es posible acceder a los correos sin acceso a internet.
  - Es necesaria una gran capacidad de almacenamiento en el servidor.

Formato de un e-mail.

- Cabecera: En ella se usan las palabras clave para definir lo campos del mensaje, tales como asunto, emisor, receptor.
- Cuerpo del mensaje: Es el mensaje propiamente dicho, suele estar compuesto únicamente por texto pero también existe la opción de añadir archivos adjuntos, etc.

Formato de las direcciones de correo.

Las direcciones de correo electrónico están compuestas por:

- Nombre de usuario: Este es determinado por el usuario, y debe ser un nombre de usuario único dentro del dominio especificado.
- @.
- Nombre del dominio: Es la denominación en internet del servidor de correo electrónico, un ejemplo de esto es “hotmail”
- Extensión: Puede adoptar formas muy diversas, tales como .com, .org, .net, etc.

Comandos y respuestas utilizadas por SMTP.

- HELO, para abrir una sesión con el servidor.
- EHLO, para abrir una sesión, en el caso de que el servidor soporte extensiones definidas en el RFC 1651.
- MAIL FROM, para indicar quien envía el mensaje.
- RCPT TO, para indicar el destinatario del mensaje.
- DATA, para indicar el comienzo del mensaje, éste finalizará cuando haya una línea únicamente con un punto.
- QUIT, para cerrar la sesión.
- RSET Aborta la transacción en curso y borra todos los registros.
- SEND Inicia una transacción en la cual el mensaje se entrega a una terminal.
- SOML El mensaje se entrega a un terminal o a un buzón.
- SAML El mensaje se entrega a un terminal y a un buzón.
- VRFY Solicita al servidor la verificación de todo un argumento.
- EXPN Solicita al servidor la confirmación del argumento.
- HELP Permite solicitar información sobre un comando.
- NOOP No decir nada, se emplea para mantener la sesión abierta
- TURN Solicita al servidor que intercambien los papeles.

Una respuesta SMTP está formada por un número de tres cifras seguido de un texto. El número es utilizado por los servidores para determinar cuál va a ser la siguiente etapa. El texto sólo es útil para un usuario humano.

Cada una de las tres cifras de la respuesta tiene un significado: La primera cifra indica si la respuesta es correcta, incorrecta o incompleta. El cliente SMTP determinará su siguiente acción en función de esta primera cifra. La segunda y tercera cifras proporcionan información adicional.

Fases de transferencia de un email

- Cuando un cliente establece una conexión con el servidor SMTP, espera a que éste envíe un mensaje “220 Service ready” o “421 Service non available”.
- Se envía un HELO desde el cliente. Con ello el servidor se identifica. Esto puede usarse para comprobar si se conectó con el servidor SMTP correcto.
- El cliente comienza la transacción del correo con la orden MAIL FROM. Como argumento de esta orden se puede pasar la dirección de correo al que el servidor notificará cualquier fallo en el envío del correo (Por ejemplo, MAIL FROM:<fuente@dominio>). Luego si el servidor comprueba que el origen es válido, el servidor responde “250 OK”.
- Ya le hemos dicho al servidor que queremos mandar un correo, ahora hay que comunicarle a quien. La orden para esto es RCPT TO:<destino@dominio>. Se pueden mandar tantas órdenes RCPT como destinatarios del correo queramos. Por cada destinatario, el servidor contestará “250 OK” o bien “550 No such user here”, si no encuentra al destinatario.
- Una vez enviados todos los RCPT, el cliente envía una orden DATA para indicar que a continuación se envían los contenidos del mensaje. El servidor responde “354 Start mail input, end with <CRLF>.<CRLF>” Esto indica al cliente como ha de notificar el fin del mensaje.
- Ahora el cliente envía el cuerpo del mensaje, línea a línea. Una vez finalizado, se termina con un <CRLF><CRLF>. (la última línea será un punto), a lo que el servidor contestará “250 OK”, o un mensaje de error apropiado.
- Tras el envío, el cliente, si no tiene que enviar más correos, con la orden QUIT corta la conexión. También puede usar la orden TURN, con lo que el cliente pasa a ser el servidor, y el servidor se convierte en cliente. Finalmente, si tiene más mensajes que enviar, repite el proceso hasta completarlos.

Bibliografía

[TELNET]

[1] https://tools.ietf.org/html/rfc854

[2] https://docs.oracle.com/cd/E24842_01/html/820-2981/ipov-6.html

[3] https://www.testdevelocidad.es/test-de-puertos/aplicaciones/protocolo-telnet/

[4] https://www.computerhope.com/unix/utelnet.htm

[5] https://docs.oracle.com/cd/E19957-01/820-2981/ipov-10/

[FTP]

[6] https://tools.ietf.org/html/rfc959

[7] https://www.hostgator.mx/blog/que-es-el-cliente-ftp/

[8] https://www.nsoftware.com/component/?c=FTP

[9] https://sites.google.com/site/mouhlinares/home/modos-de-conexion-del-cliente-ftp

[10] https://docs.oracle.com/cd/E24842_01/html/E22524/remotehowtoaccess-14.html

[11] https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos

[HTTP]

[12] https://tools.ietf.org/html/rfc7231

[13] http://www.cs.cmu.edu/~aist/www_paper/transaction.html

[14] https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto

[15] https://developer.mozilla.org/es/docs/Web/HTTP/Messages

[16] https://developer.mozilla.org/es/docs/Web/HTTP/Methods

[17] https://www.codifica.me/conexiones-persistentes-y-no-persistentes/

[18] https://cybmeta.com/que-son-las-cookies-y-como-funcionan

[19] https://www.watchguard.com/help/docs/help-center/es-419/Content/es-419/Fireware/proxies/http/http_proxy_about_c.html?TocPath=Controlar%20el%20Tr%C3%A1fico%20de%20Red%7CServidores%20Proxy%7CAcerca%20del%20Proxy%20HTTP%7C_____0

[20] http://www.cs.cmu.edu/~aist/www_paper/response.html

[SMTP, POP3 y IMAP4]

[21] https://www.hostinger.mx/tutoriales/smtp-pop-imap

[22] https://tools.ietf.org/html/rfc5321

[23] https://tools.ietf.org/html/rfc3501

[24] https://tecnologia-facil.com/como-usar/diferencia-smtp-pop3-imap/

[25] https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo