

# Arquitectura

## Datacenters y Servidores

1. Datacenters.
2. Servidores. Concepto y uso.
3. Protocolos de comunicación.
4. Tipos de servidores y protocolos soportados (mails, bases de datos, web's, FTP).
5. Que trabajo realiza un servidor
6. Como funciona la lógica de un servidor
7. Protocolos Cliente Servidor
8. Sistemas operativos y conectividad.
9. Modelo Cliente Servidor
10. Tipos de servidores.
11. Desde el procesamiento: FTP – Correo – Web – IMAP - Streaming
12. Desde la estructura: Hosting -VPS - Servidor dedicado Cloud - Co-Location.
13. Comparativas: Amazon – Google Cloud – Microsoft Azure.

## ¿Que es un Datacenter?

El término Data Center es un término habitual para muchos. Sin embargo entrar en la definición del mismo puede ser interesante para otros muchos y, como una imagen vale más que mil palabras, los más interesados pueden ver en este video

<https://youtu.be/XZmGGAbHqa0>



algunas de las instalaciones del Data Center de la empresa Google y como es un el personal que trabaja en esos centros de información.

Un Data Center es, tal y como su nombre indica, un “centro de datos” o “Centro de Proceso de Datos” (CPD). Esta definición engloba las dependencias y los sistemas asociados gracias a los cuales:

- Los datos son almacenados, tratados y distribuidos al personal o procesos autorizados para consultarlos y/o modificarlos.
- Los servidores en los que se albergan estos datos se mantienen en un entorno de funcionamiento óptimo.

Los primeros Data Centers se diseñaron siguiendo las arquitecturas clásicas de informática de red, en las que los equipos eran “apilables” en mesas, armarios o racks como la siguiente imagen:



La necesidad de fácil gestión y de optimización del espacio han hecho que se evolucione hacia sistemas basados en equipos cuyas dimensiones permiten aprovechar al máximo el volumen

disponible en los racks (equipos “enracables”), logrando una alta densidad de equipos por unidad de espacio.

Un Datacenter físico puede alojar Datacenters virtuales, cuyo costo es menor gracias a la virtualización. Cada Centro de Datos virtual es independiente del resto y dispone de máximas garantías de seguridad, disponibilidad y flexibilidad.

Los Data Center iniciales tampoco estaban diseñados para proporcionar facilidades de red avanzadas, ni los requerimientos mínimos de ancho de banda y velocidad de las arquitecturas actuales. La rápida evolución de Internet y la necesidad de estar conectados en todo momento han obligado a las empresas a requerir un alto nivel de fiabilidad y seguridad, de tal forma que se proteja la información corporativa y esté disponible sin interrupciones o degradación del acceso, con el objetivo de no poner en peligro sus negocios, sean del tamaño que sean. El cumplimiento de estos requisitos, cada día mas demandados, es posible dentro de un Data Center. Igual que un banco es el mejor sitio para guardar y gestionar el dinero, un centro de datos lo es para albergar los equipos y sistemas de información.

Los datos almacenados, no son datos estáticos, están en constante movimiento, se interrelacionan unos con otros y dan como resultado nuevos datos. Su crecimiento es constante y ello implica no solo que deben estar protegidos mediante las medidas de seguridad adecuadas, sino también dotados de estupendos “motores que les permitan moverse ágilmente por las autopistas de la información”.

El crecimiento exponencial del número de usuarios de los servicios online ha llevado a las empresas a subcontratar la gestión, mantenimiento y administración de sus equipos informáticos y de comunicaciones en los Data Center. Esto les permite centrarse en el desarrollo de su propio negocio y olvidarse de complejidades tecnológicas derivadas de las características anteriormente comentadas, así como prestar el servicio sin la necesidad de realizar una inversión elevada en equipamiento dedicado a este fin.

Que busca una empresa contratando infraestructura externa?

El modelo de Datacenter se ha replicado y mejorado con el tiempo para que las empresas clientes puedan solicitar:

- **Ancho de banda:** Es necesario contar con una gran capacidad de transferencia de datos, de tal forma que no sea apreciable ningún tipo de “retardo” provocado por la utilización de la red.

- **Fiabilidad y Alta disponibilidad:** Los sistemas deben responder a cualquier situación crítica, haciendo posible la prestación del servicio sin pérdida apreciable de calidad incluso cuando es necesario atender gran cantidad de peticiones de forma puntual (“pico”) o continuada. Resulta imprescindible contar con sistemas de alta disponibilidad y redundancia a través de modernas arquitecturas de red y servicios.

- **Seguridad:** Cubriendo fundamentalmente tres aspectos:

*1 – Seguridad física:* Comprendiendo la seguridad de los sistemas hardware, soportes, dependencias y demás entidades «tangibles» del entorno del Data Center.

*2 – Seguridad lógica:* Incluyendo los aspectos de protección aportados por aplicaciones, protocolos y procesos que intervienen en el sistema, y complementado por elementos de seguridad de red (cortafuegos), detección de intrusos (IDS), y análisis a nivel de aplicación (antivirus).

*3 – Seguridad político-corporativa:* Formada por los aspectos de seguridad relativos a política general de la organización, normas, procedimientos y convenciones internas aplicables. En este aspecto se debe tener en cuenta el cumplimiento de la legislación aplicable.

Dichas áreas están interrelacionadas, y la existencia coherente de medidas de seguridad en cada una ellas garantiza el nivel de protección óptimo frente a las posibles amenazas de seguridad.

- **Escalabilidad, flexibilidad y rapidez a la hora de implementar sus proyectos (time-to-market):** Los equipos y los web sites del Data Center tienen que funcionar con la misma agilidad y rapidez que Internet para ser competitivos. La infraestructura existente tiene que permitir poner en marcha un proyecto en el mínimo tiempo posible, así como ampliar el número de elementos o la capacidad de los existentes de forma rápida y sin impacto en el servicio. Cada proyecto requerirá una solución específica cuyos requisitos variarán en el tiempo, si la infraestructura y el Data Center no permiten efectuar estos cambios con la celeridad necesaria nunca se logrará estar a la altura de la demanda del usuario o el servicio prestado por los posibles competidores.

- **Arquitecturas sofisticadas de comunicaciones:** Para cubrir la demanda de los usuarios es necesario contar con los últimos desarrollos que la industria Internet lanza al mercado (firewalls, balanceadores de carga, sistemas de replicación de contenidos, etc).

- **La gestión y administración por personal especializado:** Los sistemas implicados en la prestación del servicio requieren un alto nivel de especialización. Así mismo, la monitorización continuada 24×7 y detección precoz de errores se han convertido en requisitos imprescindibles para garantizar la calidad del servicio prestado, y lograr detectar los “cuellos de botella” antes de que representen un problema.

# Tipos de Servidores Web

## VPS + Cloud + Hosting Compartido + Dedicado + Hosting Wordpress

### VPS

Un servidor virtual privado (VPS, por sus siglas en inglés) es una partición virtual dentro de un servidor físico que le asigna recursos exclusivos a cada partición. Éstas otorgan acceso raíz que permite instalar un sistema operativo y trabajar con alto grado de libertad.

El hosting VPS es uno de los servicios de alojamiento más populares que puedes elegir para tu sitio web. Utiliza tecnología de virtualización para proporcionarte recursos dedicados (privados) en un servidor con múltiples usuarios.

Es una solución más segura y estable que el hosting compartido donde no se obtiene espacio de servidor dedicado. Sin embargo, es de menor escala y más barato que alquilar un servidor completo.

El hosting VPS generalmente es elegido por los propietarios de sitios web que tienen un tráfico de nivel medio que excede los límites de los planes de hosting compartido pero que aún no necesitan los recursos de un servidor dedicado.

Las soluciones de VPS generalmente ofrecen varios planes de alojamiento. Por ejemplo, una empresa de alojamiento, tiene seis planes de hosting VPS que se adaptan a las diferentes necesidades y te permiten escalar tu sitio sin problemas cuando necesites más recursos.

*¿Cuándo usar VPS y cómo decidir si es hora de una mejora?*

*¿Hay alguna alternativa al VPS? ¿Qué es un VPS administrado?*

*En resumen, ¿qué es un VPS?*

*¿Cómo funciona el hosting VPS?*

Un servidor es una computadora en la que tu proveedor de alojamiento web almacena los archivos y las bases de datos necesarios para tu sitio web. Cada vez que un visitante en línea quiere acceder a tu sitio web, su navegador le envía una solicitud a tu servidor y transfiere los archivos necesarios a través de Internet. El hosting VPS te proporciona un servidor virtual que simula un servidor físico; sin embargo, en realidad, la máquina se comparte entre varios usuarios.

Al usar la tecnología de virtualización, tu proveedor de alojamiento web instala una capa virtual sobre el sistema operativo del servidor. Esta capa divide el servidor en particiones y le permite a cada usuario instalar su propio sistema operativo y software.

Por lo tanto, un servidor privado virtual (VPS) es tanto virtual como privado porque tienes control absoluto. Está separado de otros usuarios del servidor a nivel del sistema operativo. De hecho, la tecnología VPS es similar a la creación de particiones en tu computadora cuando quieres ejecutar más de un sistema operativo (por ejemplo, Windows y Linux) sin tener que reiniciar.

Un VPS te permite configurar tu sitio web dentro de un contenedor seguro con recursos garantizados (memoria, espacio en disco, núcleos de CPU, etc.) que no tienes que compartir con otros usuarios. Con el hosting VPS, tienes el mismo acceso de nivel raíz que si alquilaras un servidor dedicado, pero a un costo mucho más bajo.

### Comparación de VPS con otros tipos de alojamiento web

Los diferentes tipos de alojamiento web te permiten realizar diferentes niveles de personalización en tu servidor. Varían en precio, rendimiento (por ejemplo, tiempo de carga de la página) y la disponibilidad del servicio (por ejemplo, tiempo de actividad). A continuación, cómo el hosting VPS se compara con otras soluciones de hosting.

### **Hosting Compartido**

El hosting compartido es la solución para los propietarios de sitios web con menos tráfico. Es el punto de partida de la mayoría de las pequeñas empresas y bloggers. Con el hosting compartido, se divide el mismo servidor físico entre varios clientes de la empresa de hosting. No obtienes recursos dedicados para ti, ya que tu sitio se ejecuta en el mismo sistema operativo que el de los demás.

Por lo tanto, la memoria y el poder de computación que tu sitio puede usar se ven afectados por las necesidades de los otros usuarios del servicio. Por ejemplo, si hay un pico de tráfico repentino en un sitio web alojado en el mismo servidor, puede aumentar el tiempo de carga de tu página. Tampoco puedes elegir tu sistema operativo ni otro software de servidor, ya que todos los usuarios usan la misma configuración. En definitiva, es tu proveedor de hosting el que se encarga de cada aspecto de tu entorno de hosting compartido.

Puedes pensar en hosting compartido como un alquiler en el que compartes el mismo apartamento con varios compañeros. El hosting VPS sigue siendo una solución compartida, sin embargo, cada quien tiene su propio cuarto donde pueden personalizar el espacio de acuerdo a sus necesidades. Por ejemplo, pueden elegir la pintura, los muebles, la decoración, etc.

### **Cloud Hosting**

Con el cloud hosting o hosting en la nube no usas un solo servidor, sino un clúster (varias computadoras interconectadas) que se ejecuta en la nube. Cada servidor en el clúster almacena una copia actualizada de tu sitio web. Cuando uno de los servidores está demasiado ocupado, el clúster redirige automáticamente el tráfico a uno que no lo esté tanto. Como resultado, el hosting en la nube no tiene tiempo de inactividad (downtime), ya que siempre hay un servidor en el clúster que puede atender las solicitudes de los visitantes de tu sitio web.

El alojamiento en la nube y el alojamiento VPS no son mutuamente excluyentes. En cambio, muchas empresas de hosting ofrecen alojamiento VPS en una infraestructura en la nube. Esta es también una solución que hemos elegido para comentar, ya que descubrimos que la combinación de VPS con las tecnologías en la nube conduce al rendimiento y la confiabilidad más altos posibles en el entorno del servidor virtual.

### **Hosting WordPress**

El hosting WordPress es un servicio específicamente ofrecido para los propietarios de sitios de WordPress. Viene con varias características relacionadas con WordPress que solo puedes usar si tienes un sitio de WordPress, como instalación con un solo clic, plugins preinstalados o una interfaz de línea de comandos de WP. Los servidores están configurados para las necesidades de WordPress. Por lo tanto, los proveedores de hosting ofrecen hosting para WordPress como parte de su servicio de hosting compartido.

Aunque también es posible configurar un sitio de WordPress en un servidor privado virtual, no puedes obtener acceso a los servidores personalizados que se han configurado con WordPress en mente. Sin embargo, si aún eliges VPS para tu sitio de WordPress, puedes configurar tu entorno de alojamiento según las necesidades de tu negocio.

### **Hosting dedicado**

Con hosting dedicado, alquilas un servidor físico completo para tu negocio. Si tienes un sitio web de alto tráfico, el hosting dedicado puede ser la mejor solución para ti, ya que los servidores dedicados son rápidos, flexibles y totalmente personalizables. Sin embargo, el servicio también viene con un precio, por eso no es la mejor solución para todos, especialmente si tienes un sitio web pequeño o mediano.

Si bien el hosting VPS te permite elegir y configurar tu sistema operativo y aplicaciones del servidor, el hosting dedicado va un paso más allá. No solo te permite configurar el software sino también el hardware, ya que todo el servidor es tuyo y nadie tiene voz en cuanto a la configuración. También puedes ejecutar un servidor dedicado en el sitio (por ejemplo, en tu oficina), sin embargo, no obtendrás el apoyo de un equipo de hosting profesional en ese caso.

### **Pros y contras del VPS**

El hosting VPS puede ser una solución ideal para ti si es el servicio que realmente necesitas. A continuación, puedes leer los pros y contras de tener un servidor privado virtual.

#### **Pros**

- Es más rápido y más confiable que un servidor de hosting compartido.
- Dado que los recursos del servidor, como la memoria o la potencia de procesamiento, están garantizados, existe una fluctuación de cero a mínima en los recursos disponibles.
- Los problemas y aumentos de tráfico de otros usuarios del servidor no afectan a tu sitio.
- Obtienes acceso de superusuario (raíz) a tu servidor.
- Obtienes mayor privacidad, ya que tus archivos y bases de datos están bloqueados para otros usuarios del servidor.
- Es un servicio fácil de escalar. A medida que tu sitio web crece, puedes mejorar fácilmente los recursos de tu servidor (RAM, CPU, espacio en disco, ancho de banda, etc.).

#### **Contras**

- Es más caro que el hosting compartido.
- Se requiere más conocimiento técnico para administrar tu servidor.
- Los servidores mal configurados pueden generar vulnerabilidades de seguridad.

### **¿Cuándo usar VPS y cómo decidir si es hora de una mejora?**

El hosting VPS generalmente se ve como el paso siguiente después de que tu sitio web crezca fuera de los límites de recursos del hosting compartido. Si el plan de hosting web compartido más avanzado ya no es suficiente para ejecutar tu sitio web sin problemas, vale la pena pasarse a un plan de VPS. En tales casos, el hosting VPS puede proporcionarte lo mejor de dos mundos: hosting compartido y dedicado.

Sin embargo, hay otros casos de uso en los que puede ser una buena idea comenzar con un plan de VPS desde el principio. Por ejemplo, los sitios web de comercio electrónico donde se necesita garantizar pagos seguros con un entorno rápido y estable pueden beneficiarse mucho de un servidor privado virtual. De hecho, si almacenas cualquier tipo de información confidencial o tienes que procesar pagos en línea, un VPS puede ayudarte a reducir el riesgo de violaciones de seguridad y robo de identidad.

Si esperas recibir picos de tráfico en determinados momentos en tu sitio, por ejemplo, en el caso de un sitio web de planificación de eventos o de venta de boletos, un plan de hosting VPS puede ser un regalo del cielo. Los servidores de juegos y otros sitios con muchos recursos también pueden ejecutarse con un rendimiento mucho mejor en un servidor privado virtual.

### **¿Hay alguna alternativa al VPS? ¿Qué es un VPS administrado?**

Uno de los mayores escollos del hosting VPS es tener que gestionar todo el entorno del servidor por tu cuenta. Si tu servidor virtual no está bien configurado y mantenido, podrías enfrentarte con serios riesgos de seguridad y de pérdida de rendimiento. Los servicios administrados de hosting VPS han aparecido en el mercado como una solución viable a este problema. Puedes considerarlo como una alternativa mejorada al hosting VPS, que viene con soporte técnico completo.

El VPS administrado sigue siendo una novedad en el mercado, sin embargo cada vez más proveedores de hosting lo ofrecen a sus clientes. Por ejemplo, proporcionar un entorno de hosting totalmente administrado con planes de cloud hosting.

### **En resumen, ¿qué es un VPS?**

El hosting VPS te brinda acceso a recursos garantizados y un control total sin tener que manejar tu propio servidor. Es una excelente solución para sitios web de mediano a alto tráfico, de recursos pesados y de comercio electrónico. Sin embargo, las empresas que esperan un crecimiento rápido también pueden beneficiarse de la estabilidad del servicio. Si quieres un entorno de hospedaje confiable y de escala empresarial a un precio amigable, definitivamente vale la pena considerar el hosting VPS para tu sitio web.



## ¿Qué es un servidor?

El término servidor tiene dos significados en el ámbito informático. El primero hace referencia al ordenador que pone recursos a disposición a través de una red, y el segundo se refiere al programa que funciona en dicho ordenador. En consecuencia aparecen dos definiciones de servidor:

*Definición Servidor (hardware):* un servidor basado en hardware es una máquina física integrada en una red informática en la que, además del sistema operativo, funcionan uno o varios servidores basados en software. Una denominación alternativa para un servidor basado en hardware es "host" (término inglés para "anfitrión"). En principio, todo ordenador puede usarse como "host" con el correspondiente software para servidores.

*Definición Servidor (software):* un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes (clients) pueden usar a nivel local o a través de una red. El tipo de servicio depende del tipo de software del servidor. La base de la comunicación es el modelo cliente-servidor y, en lo que concierne al intercambio de datos, entran en acción los protocolos de transmisión específicos del servicio.

## ¿Cómo funciona un servidor?

La puesta a disposición de los servicios del servidor a través de una red informática se basa en el modelo cliente-servidor, concepto que hace posible distribuir las tareas entre los diferentes ordenadores y hacerlas accesibles para más de un usuario final de manera independiente. Cada servicio disponible a través de una red será ofrecido por un servidor (software) que está permanentemente en espera. Este es el único modo de asegurar que los clientes como el navegador web o los clientes de correo electrónico siempre tengan la posibilidad de acceder al servidor activamente y de usar el servicio en función de sus necesidades.

## Modelo cliente-servidor

### Tipos de servidores

La comunicación entre cliente y servidor depende de cada servicio y se define por medio de un protocolo de transmisión. Este principio puede aclararse partiendo de los siguientes tipos de servidores:

**Servidor web:** la tarea principal de un servidor web es la de guardar y organizar páginas web y entregarlas a clientes como navegadores web o crawlers. La comunicación entre servidor (software) y cliente se basa en HTTP, es decir, en el protocolo de transferencia de hipertexto o en HTTPS, la variante codificada. Por regla general, se transmiten documentos HTML y los elementos integrados en ellos, tales como imágenes, hojas de estilo o scripts. Los servidores web más populares son el servidor HTTP Apache, los servicios de Internet Information Server de Microsoft (ISS) o el servidor Nginx.

**Servidor de archivos:** un servidor de archivos se encarga de almacenar los datos a los que acceden los diferentes clientes a través de una red. Las empresas apuestan por dicha gestión de archivos para que sea mayor el número de grupos de trabajo que tengan acceso a los mismos datos. Un servidor de archivos contrarresta los conflictos originados por las diferentes versiones de archivos locales y hace posible tanto la creación automática de las diferentes versiones de datos como la realización de una copia de seguridad central de la totalidad de datos de la empresa. En el acceso al servidor de archivos por medio de Internet entran en juego protocolos de transmisión como FTP (File Transfer Protocol), SFTP (Secure File Transfer Protocol), FTPS (FTP over SSL) o SCP (Secure Copy). Los protocolos SMB (Server Message Block) y NFS (Network File System) se encuentran habitualmente en las redes de área locales (LAN).

**Servidor de correo electrónico:** un servidor de correo electrónico consta de varios módulos de software cuya interacción hace posible la recepción, el envío y el reenvío de correos electrónicos, así como su puesta a punto para que estén disponibles. Por regla general funciona mediante el protocolo de transferencia simple de correo (SMTP). Los usuarios que quieran acceder a un servidor de correo electrónico necesitan un cliente de correo electrónico que recoja los mensajes del servidor y los entregue en la bandeja de entrada, proceso que tiene lugar a través de los protocolos IMAP (Internet Message Access Protocol) o POP (Post Office Protocol).

**Servidor de base de datos:** un servidor de base de datos es un programa informático que posibilita que otros programas puedan acceder a uno o varios sistemas de bases de datos a través de una red. Las soluciones de software con una elevada cuota de mercado son Oracle, MySQL, Microsoft SQL Server, PostgreSQL y DB2. Los servidores de bases de datos ayudan a los servidores web, por regla general, a la hora de almacenar y entregar datos.

**Servidor de juegos:** los servidores de juegos son servidores (software) creados específicamente para juegos multijugador online. Estos servidores gestionan los datos del juego online y permiten la interacción sincrónica con el mundo virtual. La base de hardware de un servidor de juegos se encuentra en el centro de datos de los proveedores especializados o está disponible en una red doméstica local.

**Servidor proxy:** el servidor proxy sirve como interfaz de comunicación en las redes informáticas. En su papel de intermediario, el servidor proxy recibe las solicitudes de red y las transmite a través de su propia dirección IP. Los servidores proxy se usan para filtrar la comunicación, para controlar el ancho de banda, para aumentar la disponibilidad a través del reparto de cargas, así como para guardar datos temporalmente (caching). Además, los servidores proxy permiten una amplia anonimización, ya que la dirección IP del cliente queda oculta en el proxy.

**Servidor DNS:** el servidor DNS o servidor de nombres permite la resolución de nombres en una red. Los servidores DNS son de vital importancia para la red informática mundial (WWW), ya que traducen los nombres de host como [www.example.com](http://www.example.com) en la correspondiente dirección IP. Si quieres saber más sobre los servidores de nombres y sobre el sistema de nombres de dominio (DNS), visita nuestra guía digital.

En teoría, un único dispositivo físico puede alojar diferentes tipos de servidores. Sin embargo, es habitual alojar cada uno de los servidores en un ordenador independiente o que estos se repartan en más de un ordenador. De esta manera, se evita que la utilización del hardware de un servicio repercuta en el rendimiento de otros servicios.

### **¿En qué consiste el alojamiento de servidores?**

Mientras que a las grandes empresas les sale rentable la adquisición de hardware de servidores, los autónomos y los particulares que quieren desarrollar proyectos en un servidor propio recurren normalmente al alquiler. Los proveedores especializados ofrecen diferentes modelos de servidores de alquiler en los que los usuarios no tienen que preocuparse por el funcionamiento de la máquina física. La gama de productos abarca desde servidores dedicados cuyos componentes de hardware se ponen a disposición de los usuarios de manera exclusiva, hasta servicios de hosting compartido para alojar a varios clientes virtuales en una base de hardware común. Para obtener más información, visita nuestra guía sobre las ventajas y los inconvenientes de los diferentes modelos de alojamiento.

## **Funcionamiento de un servidor Web**

Un servidor es un dispositivo virtual que le brinda espacio y estructura a los sitios web para que almacenen sus datos y manejen sus páginas.

¿Alguna vez te has preguntado dónde están localizados todos los contenidos de tu estrategia de marketing digital que subes a tu página web?

Es, pues, en los servidores web. Estos son fundamentales para Internet y si ellos no existieran, navegar por la web sería muy diferente a lo que conocemos hoy en día.

Con la transformación digital y el creciente uso de Internet era más que necesario crear servidores que fueran capaces de almacenar y emitir la biblioteca de información (casi infinita) que encontramos en la web.

### **¿Qué es un servidor?**

Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes (otros ordenadores, dispositivos móviles, impresoras, personas, etc.)

Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hypertext Transfer Protocol).

### **¿Para qué sirve un servidor web en Internet?**

El rol principal de un servidor web es almacenar y transmitir el contenido solicitado de un sitio web al navegador del usuario.

Este proceso, para los internautas no dura más que un segundo, sin embargo, a nivel del web server es una secuencia más complicada de lo que parece.

Para cumplir con sus funciones el servidor deberá tener la capacidad de estar siempre encendido para evitar interrumpir el servicio que le ofrece a sus clientes. Si dicho servidor falla o se apaga, los internautas tendrán problemas al ingresar al sitio web.

### **¿Cómo funciona un servidor web?**

La comunicación entre un servidor y sus clientes se basa en HTTP, es decir, en el protocolo de transferencia de hipertexto o en su variante codificada HTTPS.

Para saber cómo funciona, primero es necesario conocer que el web server está permanentemente en espera de una solicitud de información.

Además, ten en cuenta que toda computadora, smartphone o tablet tiene una dirección IP única e irrepetible que lo identifica de otro dispositivo en la red, así es como el servidor web envía la información exacta que el internauta está esperando.

Ahora bien, para que el web server pueda cumplir con su función es necesario que reciba la petición por parte de un navegador, en otras palabras, se envía un pedido desde una dirección IP hacia la dirección IP del servidor que aloja los archivos del sitio en cuestión.

A continuación, el servidor web busca en sus archivos la información que se le está solicitando, procede a interpretar las líneas de código y a enviar el resultado al navegador cuya dirección IP fue la solicitante.

Este resultado se le muestra a los internautas y es lo que siempre sucede cuando se navega en sitios de Internet. Cuando este proceso se completa podemos decir que el web server ha cumplido con su función.

### **¿Cuáles son las características de un servidor web?**

Como características necesarias de un servidor web a nivel de software y hardware, podemos encontrar:

#### **A nivel de software**

Sistema Operativo

Se encarga de que el hardware funcione y logre interactuar con los servicios que corre el sistema.

Algunos ejemplos son:

Unix,  
Linux,  
o Windows.

#### **Sistemas de archivos**

Es una guía lógica que permite que el sistema pueda ubicar, ordenar y filtrar datos en el disco duro, con el fin de que podamos leerlos, modificarlos o eliminarlos.

Software servidor HTTP

Son los diferentes tipos de servidores web especializados en transmitir el contenido vía web (Apache, Nginx, IIS, Caddy, etc.).

#### **Virtual Hosting**

Permite que bajo el mismo web server e IP se alojen en varios sitios web distinto.

#### **Despacho de ficheros estáticos y dinámicos**

Los ficheros estáticos brindan soporte para alojar y despachar archivos como:

JPG,  
GIF,  
PNG,  
BMP,  
CSS,  
TXT,  
HTML,  
Javascript,  
MP3  
y MP4.

Los ficheros dinámicos funcionan para información y se basan en archivos que serán procesados por el servidor dependiendo generalmente de un lenguaje de desarrollo, por ejemplo: en PHP, ASP, Python, Ruby y GO.

#### **Monitoreo de Red y Límites**

Permite monitorear el tránsito de red, paquetes que entran y salen, así como servicios de sistema y uso de hardware como:

1. el uso del Almacenamiento;
2. consumo de RAM;
3. porcentaje de ocupación del CPU;
4. velocidad de la red;
5. rendimiento de escritura/lectura en disco.

### **Sistema de seguridad**

El sistema de seguridad de un servidor debe:

- imponer límites de acceso por dirección IP;
- denegar o permitirle acceso a ciertos archivos o URLs;
- solicitar usuario y contraseña para autenticación básica HTTP;
- realizar un filtrado de peticiones inseguras;
- dar soporte para despachar información cifrada con certificados de seguridad SSL vía HTTPS.

### **A nivel de hardware**

#### *Rack y gabinete*

El rack se refiere al lugar donde se alojan los servidores físicamente y el gabinete es el armazón que sostiene los componentes de hardware de una computadora.

#### *CPU*

Es el centro de procesamiento de datos del servidor desde donde se realizan todos los cálculos lógicos y matemáticos para que el usuario pueda manipular y acceder a los datos como necesita.

#### *Memoria RAM*

Se utiliza para almacenar información y datos de forma temporal dependiendo de la demanda del usuario a través del sistema operativo.

#### *Unidades de almacenamiento*

El almacenamiento de servidores web se hace en discos duros, los cuales permiten almacenar la información del sistema operativo, los servicios de sistema, y en última instancia los datos cargados por el usuario.

#### *Puerto de red*

El ancho de banda es el que te permite tener un volumen suficiente para transmitir información de ida y vuelta hacia y desde tu servidor web.

#### *¿Qué tipos de servidores web existen?*

Existen muchos tipos de servidores web, conoce cuáles son los servidores web más usados en la actualidad:

### **Apache**

Es el más común y utilizado en el mundo, sin embargo, ha perdido popularidad frente a Microsoft IIS y Nginx.

Entre las ventajas de Apache está que es un código abierto, con software gratuito y multiplataforma, y entre sus desventajas su bajo rendimiento cuando recibe miles de requests (peticiones) simultáneas en procesamiento de contenido dinámico o archivos estáticos.

## Nginx

Conocido y popularizado como una de las mejores alternativas de Apache. Nginx es un servidor web de código abierto y gratuito (aunque también existe una versión comercial) que se destaca por su alto rendimiento.

Entre sus beneficios resalta una configuración simple, ligera, rápida y excelente en cuanto a seguridad y rendimiento, además permite ser configurado para integrarse nativamente con casi cualquier tecnología y lenguaje de programación moderno.

Como desventajas podemos encontrar que no soporta los archivos .htaccess (de Apache), aunque incluye su propio lenguaje de rewrites.

## LiteSpeed

Es un software de despacho HTTP desarrollado por LiteSpeedTech, existe una versión de código abierto (open source) y una versión comercial que incluye diferentes tipos de licencia.

Algunos de los beneficios de LiteSpeed es que soporta grandes cantidades de conexiones simultáneas con un consumo de recursos realmente bajo (incluso con aplicaciones demandantes como las que utilizan PHP) y a nivel de archivos estáticos está a la altura de Nginx.

## Microsoft IIS

Internet Information Services o IIS se ha popularizado para ofrecer servicios en la nube, principalmente en Azure (la plataforma de Cloud Hosting de Microsoft).

Además, su perfecta integración con Windows (claro está), Visual Studio y sus herramientas hicieron que este web server se posicionara como el servidor líder en el mundo empresarial.

## *Otros servidores web conocidos*

Lighttpd;

Caddy;

Cherokee;

NodeJS;

Sun Java System Web Server;

Google Web Server (GWS), es el servidor privado de Google por lo que no puede ser descargado.

Cada servidor web es indicado para ciertas funciones, por lo que escoger cuál vas a usar va a depender de lo que pretendas hacer con ellos.

En resumen:

No olvides que al considerar elegir un web server debes tener en cuenta:

- que tan bien funciona con el sistema operativo y otros servidores;
- su capacidad para manejar la programación del servidor;
- las características de seguridad;
- las herramientas particulares de publicación;
- motor de búsqueda;
- la creación de sitios que vienen con él.

Los servidores Web no solo permiten publicar tu sitio en Internet, sino que también entregan el contenido a tus lectores. Conocer cuál se adapta más a ti y a tus lectores es fundamental para que el web server que elijas cumpla adecuadamente sus funciones.

# Arquitectura de servidores: Cliente-Servidor y Multicapa

Ya sabemos lo que es un servidor. Se trata de una computadora, con alta capacidad que se encarga de compartir toda clase de información con un número casi infinito de otras computadoras en todo el mundo, siempre y cuando estén conectados a él.

Esta conexión usualmente es inalámbrica pero puede ser alámbrica también. Sin embargo, muchos aún desconocen cómo funcionan estas arquitecturas. Hablaremos ahora sobre la arquitectura de servidores: Cliente-Servidor y Multicapa.

## Qué es la arquitectura de servidores

La arquitectura de servidores, se refiere a la arquitectura de una red informática en la que muchos procesadores remotos, conocidos como “clientes” solicitan y reciben servicio de un sistema centralizado, es decir, de una computadora host a la que se le llama servidor. De allí el término “modelo cliente servidor” del cual hablaremos a continuación.

## Modelo Cliente Servidor

Cliente-servidor es una relación en la cual un programa (el cliente) solicita un servicio o recurso de otro programa (el servidor).

A principios del siglo pasado, la etiqueta cliente-servidor se usó para distinguir la computación distribuida por computadoras personales (PC) del modelo de computación monolítico y centralizado utilizado por los mainframes. Hoy en día, las transacciones informáticas en las que el servidor cumple una solicitud realizada por un cliente son muy comunes y el modelo cliente-servidor se ha convertido en una de las ideas centrales de la informática en red.

En este contexto, el cliente establece una conexión con el servidor a través de una red de área local (LAN) o una red de área amplia (WAN), como Internet. Una vez que el servidor ha cumplido la solicitud del cliente, la conexión finaliza. Debido a que varios programas de clientes comparten los servicios del mismo programa de servidor, un servidor especial llamado “daemon” puede activarse solo para esperar las solicitudes de los clientes.

En los primeros días de Internet, la mayoría del tráfico de red era entre clientes remotos que solicitaban contenido web y los servidores del centro de datos que proporcionaban el contenido. Este patrón de tráfico se conoce como tráfico norte-sur. Hoy, con la madurez de la virtualización y la computación en la nube, es más probable que el tráfico de red sea de servidor a servidor, un patrón conocido como tráfico este-oeste.

Esto, a su vez, ha cambiado el enfoque del administrador de un modelo de seguridad centralizado diseñado para proteger el perímetro de la red a un modelo de seguridad descentralizado que se centra más en controlar el acceso de los usuarios individuales a los servicios y datos, y auditar su comportamiento para garantizar el cumplimiento de las políticas y regulaciones.

## Ventajas y desventajas del modelo cliente-servidor

Una ventaja importante del modelo cliente-servidor es que su arquitectura centralizada ayuda a facilitar la protección de los datos con controles de acceso impuestos por las políticas de seguridad. Además, no importa si los clientes y el servidor se basan en el mismo sistema operativo porque los datos se transfieren a través de protocolos cliente-servidor que son independientes de la plataforma.



Una desventaja importante del modelo cliente-servidor es que si demasiados clientes solicitan simultáneamente datos del servidor, pueden sobrecargarse. Además de causar congestión en la red, demasiadas solicitudes pueden resultar en una denegación de servicio.

### **Protocolos cliente-servidor**

Los clientes generalmente se comunican con los servidores mediante el conjunto de protocolos TCP / IP. TCP es un protocolo orientado a la conexión, lo que significa que se establece y mantiene una conexión hasta que los programas de aplicación en cada extremo hayan terminado de intercambiar mensajes.

Determina cómo dividir los datos de la aplicación en paquetes que las redes pueden entregar, envía paquetes y acepta paquetes de la capa de red, gestiona el control de flujo y maneja la retransmisión de paquetes descartados o confusos, así como el reconocimiento de todos los paquetes que llegan. En el modelo de comunicación de interconexión de sistemas abiertos (OSI), TCP cubre partes de la capa 4, la capa de transporte y partes de la capa 5, la capa de sesión.

Por el contrario, IP es un protocolo sin conexión, lo que significa que no hay una conexión continua entre los puntos finales que se están comunicando.

Cada paquete que viaja a través de Internet se trata como una unidad de datos independiente sin ninguna relación con ninguna otra unidad de datos. (La razón por la que los paquetes se colocan en el orden correcto es debido a TCP). En el modelo de comunicación de Interconexión de Sistemas Abiertos (OSI), IP está en la capa 3, la Capa de Redes.

### **Modelo Multicapa**

Cuando hablamos del modelo multicapa, hablamos del modelo OSI que significa interconexión de sistemas abiertos. Ha sido desarrollado por ISO (Organización Internacional de Normalización), en el año 1974. Es una arquitectura de 7 capas, cada una tiene una funcionalidad específica para realizar. Todas estas 7 capas trabajan en colaboración para transmitir los datos de una persona a otra en todo el mundo. Te las explicamos a continuación:

#### **Capa 1 o Capa Física**

La capa más baja del modelo de referencia OSI es la capa física. Es responsable de la conexión física real entre los dispositivos. La capa física contiene información en forma de bits. Es responsable de la conexión física real entre los dispositivos. Al recibir datos, esta capa obtendrá la señal recibida y la convertirá en 0 y 1 y los enviará a la capa de enlace de datos, que volverá a unir el marco.

Las funciones de la capa física son:

**Sincronización de bits:** la capa física proporciona la sincronización de los bits al proporcionar un reloj. Este reloj controla tanto el emisor como el receptor, proporcionando así la sincronización a nivel de bit.

**Control de velocidad de bits:** la capa física también define la velocidad de transmisión, es decir, el número de bits enviados por segundo.

**Topologías físicas:** la capa física especifica la forma en que los diferentes dispositivos / nodos se organizan en una red, es decir, topología de bus, estrella o malla.

Modo de transmisión: la capa física también define la forma en que los datos fluyen entre los dos dispositivos conectados. Los diversos modos de transmisión posibles son: Simplex, half-duplex y full-duplex.

## **Capa 2 o Capa de enlace de datos**

La capa de enlace de datos es responsable de la entrega nodo a nodo del mensaje. La función principal de esta capa es asegurarse de que la transferencia de datos esté libre de errores de un nodo a otro, sobre la capa física. Cuando un paquete llega a una red, es responsabilidad de DLL transmitirlo al Host utilizando tu dirección MAC.

La capa de enlace de datos se divide en dos subcapas:

Control de enlace lógico (LLC)  
Control de acceso a medios (MAC)

El paquete recibido de la capa de red se divide en tramas dependiendo del tamaño de trama de la NIC (tarjeta de interfaz de red). DLL también encapsula la dirección MAC del remitente y del receptor en el encabezado. La dirección MAC del receptor se obtiene colocando una solicitud ARP (Protocolo de resolución de direcciones) en el cable preguntando "¿Quién tiene esa dirección IP?" Y el host de destino responderá con su dirección MAC.

Las funciones de la capa de enlace de datos son:

Enmarcado: el enmarcado es una función de la capa de enlace de datos. Proporciona una manera para que un remitente transmita un conjunto de bits que son significativos para el receptor. Esto se puede lograr adjuntando patrones de bits especiales al principio y al final del cuadro.

Direccionamiento físico: después de crear tramas, la capa de enlace de datos agrega direcciones físicas (dirección MAC) del remitente y / o receptor en el encabezado de cada trama.

Control de errores: la capa de enlace de datos proporciona el mecanismo de control de errores en el que detecta y retransmite tramas dañadas o perdidas.

Control de flujo: la velocidad de datos debe ser constante en ambos lados; de lo contrario, los datos pueden corromperse, por lo tanto, el control de flujo coordina esa cantidad de datos que se pueden enviar antes de recibir el acuse de recibo.

Control de acceso: cuando varios dispositivos comparten un único canal de comunicación, la subcapa MAC de la capa de enlace de datos ayuda a determinar qué dispositivo tiene control sobre el canal en un momento dado.

## **Capa 3 o Capa de Red**

La capa de red funciona para la transmisión de datos de un host a otro ubicado en diferentes redes. También se encarga del enrutamiento de paquetes, es decir, la selección de la ruta más corta para transmitir el paquete, a partir del número de rutas disponibles. La dirección IP del remitente y el receptor se colocan en el encabezado por capa de red.

Las funciones de la capa de red son:

Enrutamiento: los protocolos de la capa de red determinan qué ruta es adecuada desde el origen hasta el destino. Esta función de la capa de red se conoce como enrutamiento.

Direccionamiento lógico: para identificar cada dispositivo en la red interna de manera única, la capa de red define un esquema de direccionamiento. La dirección IP del remitente y el receptor se

colocan en el encabezado por capa de red. Tal dirección distingue cada dispositivo de manera única y universal.

#### **Capa 4 o Capa Transporte**

La capa de transporte proporciona servicios a la capa de aplicación y toma ayuda de la capa de red. Los datos en la capa de transporte se denominan segmentos. Es responsable de la entrega de extremo a extremo del mensaje completo. La capa de transporte también proporciona el reconocimiento de la transmisión exitosa de datos y retransmite los datos si se encuentra un error.

En el lado del remitente: la capa de transporte recibe los datos formateados de las capas superiores, realiza la segmentación y también implementa el control de flujo y error para garantizar una transmisión de datos adecuada. También agrega el número de puerto de origen y destino en su encabezado y reenvía los datos segmentados a la capa de red.

En el lado del receptor: la capa de transporte lee el número de puerto de su encabezado y reenvía los datos que ha recibido a la aplicación correspondiente. También realiza la secuenciación y el reensamblaje de los datos segmentados.

Las funciones de la capa de transporte son:

Segmentación y reensamblaje: esta capa acepta el mensaje de la capa (sesión), divide el mensaje en unidades más pequeñas. Cada uno de los segmentos producidos tiene un encabezado asociado. La capa de transporte en la estación de destino vuelve a ensamblar el mensaje.

Direccionamiento de punto de servicio: para entregar el mensaje al proceso correcto, el encabezado de la capa de transporte incluye un tipo de dirección llamada dirección de punto de servicio o dirección de puerto. Por lo tanto, al especificar esta dirección, la capa de transporte se asegura de que el mensaje se entregue al proceso correcto.

#### **Capa 5 o Capa de sesión**

Esta capa es responsable del establecimiento de la conexión, el mantenimiento de las sesiones, la autenticación y también garantiza la seguridad.

Las funciones de la capa de sesión son:

Establecimiento, mantenimiento y finalización de la sesión: la capa permite que los dos procesos establezcan, usen y finalicen una conexión.

Sincronización: esta capa permite que un proceso agregue puntos de verificación que se consideran puntos de sincronización en los datos. Estos puntos de sincronización ayudan a identificar el error para que los datos se vuelvan a sincronizar correctamente y los extremos de los mensajes no se corten prematuramente y se evite la pérdida de datos.

Controlador de diálogo: la capa de sesión permite que dos sistemas inicien la comunicación entre sí en half-duplex o full-duplex.

#### **Capa 6 o Capa de presentación**

La capa de presentación también se llama capa de traducción. Los datos de la capa de aplicación se extraen aquí y se manipulan según el formato requerido para transmitir a través de la red.

Las funciones de la capa de presentación son:

Traducción: por ejemplo, ASCII a EBCDIC.

Cifrado / descifrado: el cifrado de datos traduce los datos a otra forma o código. Los datos cifrados se conocen como texto cifrado y los datos descifrados se conocen como texto sin formato. Se utiliza un valor clave para cifrar y descifrar datos.

Compresión: reduce la cantidad de bits que deben transmitirse en la red.

### **Capa 7 o Capa de Aplicación**

En la parte superior de la pila de capas del modelo de referencia OSI, encontramos la capa de aplicación que implementan las aplicaciones de red.

Estas aplicaciones producen los datos, que deben transferirse a través de la red. Esta capa también sirve como una ventana para que los servicios de la aplicación accedan a la red y para mostrar la información recibida al usuario.

Las funciones de la capa de aplicación son:

Terminal virtual de red

Acceso y gestión de transferencia de archivos FTAM

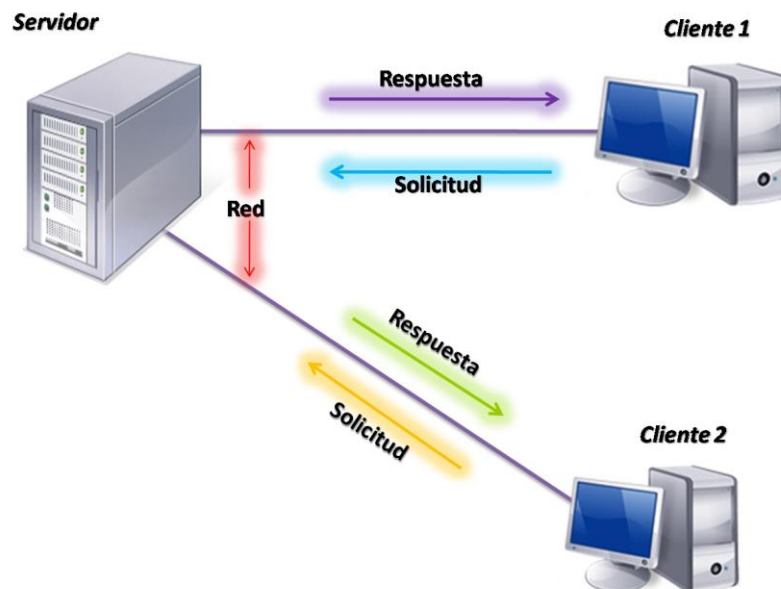
Servicios de correo

Directorio de Servicios

## PROTOCOLO TCP/IP

### Arquitectura cliente/servidor

Según TIC, (2016) La estructura cliente - servidor es una arquitectura de computación en la que se consigue un procesamiento cooperativo de la información por medio de un conjunto de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores.



De esta forma, y gracias a esta arquitectura, la totalidad de los procesadores, clientes y servidores, trabajan de forma cooperativa para realizar un determinado tratamiento de la información.

Atendiendo a esta visión descentralizada, la arquitectura cliente - servidor consiste en una arquitectura distribuida de computación, en la que las tareas de cómputo se reparten entre distintos procesadores, obteniendo los usuarios finales el resultado final de forma transparente, con independencia del número de equipos (servidores) que han intervenido en el tratamiento. Se puede decir por tanto que la arquitectura cliente - servidor es un tipo de arquitectura distribuida, posiblemente la más extendida.

### Elementos que forman parte de una arquitectura cliente - servidor

Un sistema Cliente/Servidor es un Sistema de Información distribuido basado en las siguientes características:

**Servicio:** unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.

**Recursos compartidos:** Muchos clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.

**Protocolos asimétricos:** Los clientes inician "conversaciones". Los servidores esperan su establecimiento pasivamente.

**Transparencia de localización física de los servidores y clientes:** El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.

**Independencia de la plataforma** hardware y/o software que se emplee.

**Sistemas débilmente acoplados.** Interacción basada en envío de mensajes.

**Encapsulamiento de servicios.** Los detalles de la implementación de un servicio son transparentes al cliente.

**Escalabilidad** horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).

**Integridad:** Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

En el modelo usual Cliente/Servidor, un servidor, (daemon en la terminología sajona basada en sistemas UNIX/LINUX, traducido como "demonio") se activa y espera las solicitudes de los clientes.

Lo normal es que los servicios de un mismo servidor puedan ser utilizados por múltiples clientes distintos. Tanto los programas cliente como los servidores son con frecuencia parte de un programa o aplicación mayores.

### **ESQUEMA DE FUNCIONAMIENTO DE UN SISTEMA SEGÚN LA ARQUITECTURA CLIENTE - SERVIDOR.**

El Esquema de funcionamiento de un Sistema Cliente/Servidor sería:

- 1) El cliente solicita una información al servidor.
- 2) El servidor recibe la petición del cliente.
- 3) El servidor procesa dicha solicitud.
- 4) El servidor envía el resultado obtenido al cliente.
- 5) El cliente recibe el resultado y lo procesa.

### **COMPONENTES DE LA ARQUITECTURA CLIENTE - SERVIDOR**

El modelo Cliente/Servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos.

Esta descomposición principalmente consiste en separar los elementos estructurales de esta tecnología en función de aspectos más funcionales de la misma:

Nivel de Presentación: Agrupa a todos los elementos asociados al componente Cliente.

Nivel de Aplicación: Agrupa a todos los elementos asociados al componente Servidor.

Nivel de comunicación: Agrupa a todos los elementos que hacen posible la comunicación entre los componentes Cliente y servidor.

Nivel de base de datos: Agrupa a todas las actividades asociadas al acceso de los datos.

## Elementos principales

### CLIENTE

Un cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término front-end.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

De este modo el cliente se puede clasificar en:

**Cliente basado en aplicación de usuario.** Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.

**Cliente basado en lógica de negocio.** Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

### SERVIDOR

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

### MIDDLEWARE

El middleware es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red. En el caso que nos concierne, es el intermediario entre el cliente y el servidor y se ejecuta en ambas partes.

El middleware se estructura en tres niveles:

- Protocolo de transporte.
- Network Operating System (NOS).

- Protocolo específico del servicio.

Las principales características de un middleware son:

- Simplifica el proceso de desarrollo de aplicaciones al independizar los entornos propietarios.
- Permite la interconectividad de los Sistemas de Información del Organismo.
- Proporciona mayor control del negocio al poder contar con información procedente de distintas plataformas sobre el mismo soporte.
- Facilita el desarrollo de sistemas complejos con diferentes tecnologías y arquitecturas.

## COMUNICACIÓN ENTRE LOS ELEMENTOS (NOS)

Como se ha comentado en el apartado anterior, el middleware es un conjunto de aplicaciones encargadas de enlazar al cliente con el servidor. Para ello se estructura en tres capas diferentes:

Protocolo de transporte: comunes a otras aplicaciones.

Network Operating System (NOS).

Protocolo específico del servicio: especiales para distintos tipos de sistemas Cliente/Servidor.

El NOS es el encargado de proporcionar una apariencia de sistema único a un sistema Cliente/Servidor. Se trata pues, de una extensión del Sistema Operativo:

El cliente realiza una llamada a un servicio como si fuera local.

El NOS:

- Intercepta la llamada.
- Redirige la llamada al servidor apropiado.
- Devuelve la contestación.

El NOS debe proporcionar transparencia a los procesos Cliente/Servidor con respecto a:

- **Localización:** Los recursos sólo se conocen por su nombre. El sistema en el que se ejecutan es irrelevante.
- **Espacio de nombres:** Las convenciones de los nombres de los recursos deben ser iguales, independientemente del sistema que los soporte.
- **Conexión:** Un único usuario y contraseña para todo el sistema.
- **Replicación:** No se debe diferenciar entre copias de un mismo recurso.
- **Acceso local / remoto:** El acceso a un recurso se debe realizar como si estuviera localizado en el mismo sistema que el programa cliente.
- **Tiempo:** Los relojes de todos los elementos del sistema deben estar sincronizados.
- **Fallos:** El sistema debe proporcionar servicios de detección de fallos, redundancia y reconexión tras un fallo.
- **Administración:** Un único sistema de gestión de todos los recursos.
- **Protocolos:** Idéntica interfaz de programación para todos los protocolos de transporte.



## **Tipos de arquitectura cliente / servidor**

Uno de los aspectos claves para entender la tecnología Cliente/Servidor, y por tanto contar con la capacidad de proponer y llevar a cabo soluciones de este tipo, es llegar a conocer la arquitectura de este modelo y los conceptos o ideas asociados al mismo.

Un esquema de clasificación basado en los conceptos de Fat Client/Thin Client, Fat Server/Thin Server, es decir, basado en el tamaño de los componentes. En segundo lugar tenemos una clasificación según la naturaleza del servicio que nos ofrecen.

### **TIPOS DE ARQUITECTURA CLIENTE - SERVIDOR POR TAMAÑO DE COMPONENTES.**

Este tipo de clasificación se basa en los grados de libertad que brinda el modelo Cliente/Servidor para balancear la carga de proceso entre los niveles de presentación, aplicación y base de datos.

Dependiendo de qué segmento de las capas de software tenga que soportar la mayor o menor carga de procesamiento, se habla de Fat Client (Thin Server) o Fat server (Thin Client).

Consideraciones de este tipo son importantes en el momento de decidir una plataforma de desarrollo, al mismo tiempo que pueden definir la viabilidad o no de las mismas para enfrentar un cierto número de restricciones impuestas por una problemática a resolver.

#### **FAT CLIENT (THIN SERVER)**

En este esquema de arquitectura el peso de la aplicación es ejecutada en el cliente, es decir, el nivel de presentación y el nivel de aplicación corren en un único proceso cliente, y el servidor es relegado a realizar las funciones que provee un administrador de base de datos.

En general este tipo de arquitectura tiene mejor aplicación en sistemas de apoyo de decisiones (DSS: Decision Support System) y sistemas de información ejecutiva (EIS: Executive Information System), y como se concluirá más adelante, tiene pocas posibilidades de aplicarse en sistemas de misión crítica.

#### **FAT SERVER (THIN CLIENT)**

Este es el caso opuesto al anterior, el proceso cliente es restringido a la presentación de la interfaz de usuario, mientras que el peso de la aplicación corre por el lado del servidor de aplicación.

En general este tipo de arquitectura presenta una flexibilidad mayor para desarrollar una gran variedad de aplicaciones, incluyendo los sistemas de misión crítica a través de servidores de transacciones.

### **TIPOS DE ARQUITECTURA CLIENTE - SERVIDOR SEGÚN**

#### **LA NATURALEZA DE SERVICIO PROPORCIONADO.**

##### **SERVIDORES DE FICHEROS**

Con un servidor de archivos, un cliente lo que hace es requerimientos de los mismos sobre una red. Esta es una forma muy primitiva de servicios de datos, la cual necesita intercambio de muchos mensajes sobre una red para hallar el dato requerido.

Los servidores de archivos usan recursos compartidos sobre la red y son necesarios para crear repositorios de documentos, imágenes y archivos grandes sobre la red.

## **SERVIDORES DE BASES DE DATOS**

Este análisis está elaborado desde el punto de vista del modelo Cliente/Servidor, y está directamente relacionado con la arquitectura en dos planos, que se describirá en el apartado siguiente.

Obviamente la creación de aplicaciones Cliente/Servidor está asociada a la utilización de servidores de bases de datos relacionales SQL, y dependiendo de los requerimientos y restricciones se debe elegir entre una arquitectura dos o tres planos.

## **SERVIDORES DE TRANSACCIONES**

Estos tipos de sistemas se pueden implementar con cualquiera de las modalidades Cliente/Servidor en dos o tres planos, pero incorporan un elemento principal sobre el cual se elabora y basa toda la fortaleza de este modelo, el concepto de transacción.

Con un servidor de transacciones el proceso cliente llama a funciones, procedimientos o métodos que residen en el servidor, ya sea que se trate de un servidor de bases de datos o un servidor de aplicaciones.

## **SERVIDORES DE OBJETOS**

Con un servidor de objetos, las aplicaciones Cliente/Servidor son escritas como un conjunto de objetos que se comunican. Los objetos cliente se comunican con los objetos servidores usando un Object Request Broker (ORB). El cliente invoca un método de un objeto remoto. El ORB localiza el método del objeto en el servidor, y lo ejecuta para devolver el resultado al objeto cliente.

Los servidores de objetos deben soportar concurrencia. La parte central de la comunicación en los servidores de objetos es el ORB:

- Elemento central y principal de esta arquitectura.
- Bus de objetos. Permite la comunicación entre ellos.
- Middleware avanzado: Permite llamadas estáticas y dinámicas a objetos.
- Lenguaje de descripción de interfaces independiente del lenguaje de programación.

## **SERVIDORES WEB**

La primera aplicación cliente servidor que cubre todo el planeta es el World Wide Web. Este nuevo modelo consiste en clientes simples que hablan con servidores Web. Un servidor Web devuelve documentos cuando el cliente pregunta por el nombre de los mismos. Los clientes y los servidores se comunican usando un protocolo basado en RPC, llamado HTTP. Este protocolo define un conjunto simple de comandos, los parámetros son pasados como cadenas y no provee tipos de datos. La Web y los objetos distribuidos están comenzando a crear un conjunto muy interactivo de computación Cliente/Servidor.

### **Modelos cliente/servidor**

Una de las clasificaciones mejor conocidas de las arquitecturas Cliente/Servidor se basa en la idea de planos (tier), la cual es una variación sobre la división o clasificación por tamaño de componentes.

## **A nivel de software**

Este enfoque o clasificación es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura Cliente/Servidor y en la madurez y popularidad de la computación distribuida

## **MODELO CLIENTE/SERVIDOR 2 CAPAS**

Esta estructura se caracteriza por la conexión directa entre el proceso cliente y un administrador de bases de datos. Dependiendo de donde se localice el grupo de tareas correspondientes a la lógica de negocios se pueden tener a su vez dos tipos distintos dentro de esta misma categoría:

### **IMPLEMENTADO CON SQL REMOTO**

En este esquema el cliente envía mensajes con solicitudes SQL al servidor de bases de datos y el resultado de cada instrucción SQL es devuelto por la red, no importando si son uno, diez, cien o mil registros. Es el mismo cliente quien debe procesar todos los registros que le fueron devueltos por el servidor de base de datos, según el requerimiento que él mismo hizo.

Ventajas:

Presenta una estructura de desarrollo bastante simple ya que el programador maneja un único ambiente de desarrollo (es más simple respecto al Cliente/Servidor en tres planos, puesto que reduce una capa de programación, como se verá más adelante).

Inconvenientes:

La gran cantidad de información que viaja al cliente congestiona demasiado el tráfico de red, lo que se traduce en bajo rendimiento.

Por su bajo rendimiento esta estructura tiene un bajo espectro de aplicación, limitándose a la construcción de sistemas no críticos.

### **IMPLEMENTADO CON PROCEDIMIENTOS ALMACENADOS**

En este esquema el cliente envía llamadas a funciones que residen en la base de datos, y es ésta quien resuelve y procesa la totalidad de las instrucciones SQL agrupadas en la mencionada función.

Ventajas: Presenta las mismas ventajas de una arquitectura dos planos con procedimientos almacenados, pero mejora considerablemente el rendimiento sobre ésta, dado que reduce el tráfico por la red al procesar los datos en la misma base de datos, haciendo viajar sólo el resultado final de un conjunto de instrucciones SQL.

Inconvenientes: Si bien la complejidad de desarrollo se ve disminuida, se pierde flexibilidad y escalabilidad en las soluciones implantadas. Obliga a basar el peso de la aplicación en SQL extendido, propios del proveedor de la base de datos que se elija. Debiera considerarse que si bien los procedimientos almacenados (stored procedures), los desencadenantes (triggers) y las reglas (constraint) son útiles, en rigor son ajenos al estándar de SQL

## **MODELO CLIENTE/SERVIDOR 3 CAPAS**

Esta estructura se caracteriza por elaborar la aplicación en base a dos capas principales de software, más la capa correspondiente al servidor de base de datos. Al igual que en la arquitectura dos capas, y según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el nuevo proceso correspondiente al servidor de aplicación.

#### Ventajas:

Reduce el tráfico de información en la red por lo que mejora el rendimiento de los sistemas (especialmente respecto a la estructura en dos planos).

Brinda una mayor flexibilidad de desarrollo y de elección de plataformas sobre la cual montar las aplicaciones. Provee escalabilidad horizontal y vertical.

Se mantiene la independencia entre el código de la aplicación (reglas y conocimiento del negocio) y los datos, mejorando la portabilidad de las aplicaciones.

#### Inconvenientes:

Dependiendo de la elección de los lenguajes de desarrollo, puede presentar mayor complejidad en comparación con Cliente/Servidor dos planos.

Existen pocos proveedores de herramientas integradas de desarrollo con relación al modelo Cliente/Servidor dos planos, y normalmente son de alto costo.

#### A nivel de hardware

Esta clasificación del modelo Cliente/Servidor se basa igualmente en la distribución de los procesos y elementos entre sus componentes, pero centrándose en la parte física del mismo, en el que la administración de la interfaz gráfica se asocia a los clientes PC y la seguridad e integridad de los datos quedan asociados a ambientes mainframe o por lo menos a servidores locales y/o centrales.

### **MODELO CLIENTE / SERVIDOR 2 CAPAS**

Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual, dependiendo de la aplicación puede dar acceso a los datos administrados por él.

### **MODELO CLIENTE / SERVIDOR 3 CAPAS**

Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación.

## **Tipos de Servidores según el servicio realizado**

### **FTP – Correo – Web – IMAP - Streaming**

Un servidor o server, en el lenguaje informático, es un ordenador y sus programas, que están al servicio de otros ordenadores.

El servidor atiende y responde a las peticiones que le hacen los otros ordenadores. Los otros ordenadores, que le hacen peticiones, serán los "clientes" del servidor.

Precisamente se llaman servidores porque sirven cosas y están al servicio de otros ordenadores.

Por ejemplo si tienes un correo electrónico, lo recibes de un servidor de correo electrónico, si deseas ver una página web, la recibes de un servidor web, si trabajas en una red de ordenadores todos los servicios compartidos de la red estarán en un servidor de red y así otros muchos servicios y tipos de servidores que veremos.

El modelo o arquitectura que siguen los servidores es el de cliente-servidor, es decir el cliente/s pide y el servidor proporciona los recursos o servicios.

Los servidores se utilizan para gestionar los recursos de una red.

Un servidor deberá estar siempre encendido, ya que si se apaga dejará de dar servicio a los demás. Cuando un servidor falla (se apaga o tiene errores) hace que los demás usuarios de la red tengan problemas, porque no disponen de los servicios que proporciona ese servidor.

Por ejemplo, un usuario puede configurar un servidor para controlar el acceso a una red, enviar/recibir correo electrónico, gestionar los trabajos de impresión, o alojar un sitio web.

La red más conocida y más grande es Internet, y está llena de servidores. Pero ojo hay servidores dentro de redes pequeñas y particulares, incluso tu puedes hacer que tu propio ordenador sea un servidor.

Dependiendo del servicio que de el servidor, tiene que disponer de software (programas) específicos capaces de ofrecer esos servicios. El hardware es simplemente un ordenador, aunque es recomendable que sea de gama alta, para dar respuesta a las peticiones lo más rápido posible.

En la siguiente imagen vemos el apilamiento de los servidores de una empresa que se dedica a proporcionar almacenamiento de información, también llamado Cloud Computing.

#### **servidores de internet**

Normalmente, la mayoría de los servidores están diseñadas para operar sin ninguna intervención manual durante su funcionamiento. Eso sí, antes se deberán configurar correctamente.

#### **Tipos de Servidores**

Vamos a ver los principales tipos de servidores y explicar para qué sirve cada uno:

- **Servidor de Correo Electrónico o Mail Server:** Es un ordenador dentro de una red que funciona como una oficina de correo virtual. Transfiere y almacena los mensajes de correo electrónico a través de una red.

Estos servidores tienen programas capaces de almacenar correos para los usuarios locales y con un conjunto de reglas definidas por el usuario que determinan cómo el servidor de correo debe reaccionar ante el destino de un mensaje específico.

Normalmente estos servidores se dividen en otros 2 diferentes, una para el correo entrante (llamados POP3) y otro para el correo saliente (llamados SMTP).

Los servidores POP3 retienen los mensajes de correo electrónico entrantes hasta que el usuario compruebe su correo y entonces los transfieren al equipo cuando el usuario lo pide.

Los servidores SMTP administran el envío de los mensajes de correo electrónico a Internet. El servidor SMTP administra el correo electrónico saliente y se utiliza en combinación con un servidor POP3 o IMAP de correo electrónico entrante. Cuando el usuario da la orden de enviar, el servidor lo envía.

Otro tipo de servidores de correo son los IMAP que permiten trabajar con los mensajes de correo electrónico sin necesidad de descargarlos antes al equipo. Puedes obtener una vista previa, eliminar y organizar los mensajes directamente en el servidor de correo sin descargarlos en tu equipo. Ejemplos son los correos de yahoo, Hotmail, etc.

También están los servidores Fax que hacen lo mismo que los de correo, pero para la recepción y transmisión de faxes.

- **Servidor FTP:** Se trata de uno de los más antiguos en Internet, "file transfer protocol" o en Español Protocolo Para la Transferencia de Archivos. Se utilizan para realizar una transferencia segura de archivos entre ordenadores (envío de archivos de un sitio a otro). Los FTP garantiza la seguridad de los archivos y control de su transferencia.

En este caso el cliente 1 envía una petición al servidor FTP para que le envíe un archivo al cliente 2. El servidor se lo envía y el cliente 2 lo recibe. Todo este proceso se realiza mediante un programa llamado FTP instalado en el cliente 1 y en el 2. El servidor dispondrá de otro programa (software) que se encargará de la recepción y el envío.

Este tipo de servidores se utilizan para subir archivos de páginas web a los servidores web, archivos de imágenes, videos, para hacer backup (copias de seguridad), etc.

- **Web Server o Servidor Web:** Todas las páginas web que puedes ver por internet están almacenadas en servidores, llamados servidores web.

Un servidor web almacena los archivos de una web y los proporciona a los clientes que los solicitan haciendo la transferencia de los archivos a través de la red mediante los navegadores. El cliente lo pide a través de su navegador y el servidor web lo envía al mismo navegador del cliente para que este lo pueda visualizar.

Los archivos web incluyen texto, imágenes, videos, etc.. y que solo los navegadores pueden visualizar.

El servidor "sirve" (envía) el archivo web (por ejemplo una web en formato html) al navegador del cliente para que lo pueda visualizar. El servidor, el navegador y la comunicación a través de la red seguirán unas normas llamadas "protocolo HTTP".

El espacio que te dejan estos servidores para alojar tu web se llama Hosting. Hay dos tipos principales de hosting:

**Hosting Compartido:** en el servidor web hay varias páginas alojadas de distintos clientes.

**Hosting Dedicado:** tienes un servidor para ti solito donde puedes alojar tus webs. Lógicamente son más caros.

Muchas veces se dice servidor web compartido o dedicado para hacer referencia a este tipo de hosting.

Los servidores web utilizan programas específicos para administrar sus servicios. En función del programa que utiliza el servidor web para administrar y servir las páginas web pueden ser de varios tipos. Todos los tipos que vamos a ver a continuación son realmente programas de gestión del servidor web (software).

### **Tipos de Servidores Web**

- **Servidor Apache HTTP:** Este es el servidor web más popular del mundo desarrollado por la Apache Software Foundation. El servidor web Apache es un software de código abierto y se puede instalar en casi todos los sistemas operativos incluyendo Linux, Unix, Windows, FreeBSD, Mac OS X y más. Alrededor del 60% de los ordenadores usados como servidor web ejecuta el servidor Web Apache.

- **Microsoft IIS** es un Servidor Web de alto rendimiento de Microsoft. Este servidor Web se ejecuta en plataformas Windows NT / 2000 y 2003 (y en la próximas nuevas versiones de Windows también). IIS viene incluido con Windows NT / 2000 y 2003; Dado que IIS está estrechamente integrado con el sistema operativo, es relativamente fácil administrarlo.

- **El Lighttpd**, pronunciado lighty es también un servidor web gratuito que se distribuye con el sistema operativo FreeBSD. Este servidor web de código abierto es rápido, seguro y consume mucha menos energía de la CPU. Lighttpd también se puede ejecutar en los sistemas operativos Windows, Mac OS X, Linux y Solaris.

- **Sun Java System Web Server** es un servidor web adecuado para grandes sitios web de medianas y grandes empresas. Aunque el servidor es libre no es de código abierto. Sin embargo, se ejecuta en plataformas Windows, Linux y Unix. El servidor web de Sun Java System soporta varios idiomas, guiones y tecnologías necesarias para la Web 2.0, tales como JSP, servlets Java, PHP, Perl, Python, Ruby on Rails, ASP y ColdFusion, etc.

- **Jigsaw (Servidor de W3C)** proviene del World Wide Web Consortium. Es de código abierto y libre y puede ejecutarse en varias plataformas como Linux, Unix, Windows, Mac OS X Free BSD, etc. Jigsaw ha sido escrito en Java y se puede ejecutar scripts CGI y programas PHP.

- **El servidor Nginx** es un servidor Web muy ligero y trabaja sobre sistemas Unix y Windows. Se ha convertido en el 4º servidor HTTP más popular de la red y también se distribuye bajo licencia BSD. Se utiliza en el 19% de los servidores web.

Dentro de la red de internet hay unos servidores que se llaman DNS que son los que se encargan de gestionar los nombres de los dominios de las páginas web (las direcciones de las webs). Estos servidores se llaman Servidores DNS. Para saber más sobre esto visita el siguiente enlace: [¿Qué es el DNS y servidores DNS?](#).

- **Servidores Proxy o Servidores de Red:** Se utilizan para administrar una red de ordenadores, permitiendo el acceso o no a la red de los clientes. Suelen incluir protección de la red como por ejemplo un firewall (cortafuegos).

- **Servidores de Bases de Datos:** Son ordenadores preparados para alojar bases de datos para ser utilizadas por uno o más clientes. Además estos servidores realizan tareas como el análisis de los datos, el almacenamiento, la manipulación de datos, y otras tareas específicas.

- **Servidores de Audio/Video:** Permiten transmitir contenido multimedia en streaming. El streaming es una técnica de envío continuo de información, que permite por ejemplo, ir viendo una película según se va descargando, sin necesidad de descargarla por completo para visualizarla.

- **Chat Server o Servidor Chat:** Es un equipo dedicado a manejar y mantener un chat y sus usuarios. Los más famosos son los IRC. Ahora también se les conoce como servidores en tiempo real, porque permiten intercambiar información de forma instantánea.

- **Servidores Groupware:** Son servidores que facilitan el trabajo en grupo de varios ordenadores, con un objetivo común (por ejemplo un proyecto).

Estos servidores disponen de software que permite colaborar a los usuarios del servidor independientemente de donde están ubicados, permitiéndoles así hacer un trabajo colaborativo.

Los archivos y datos almacenados en un servidor groupware pueden ser alterados, acceder y recuperados por los miembros del grupo de trabajo. Groupware también se conoce como software de colaboración.

- **Servidor Telnet:** Son servidores que nos permiten iniciar sesión en cualquier ordenador y realizar tareas en otro ordenador. Podemos trabajar con nuestro ordenador de forma remota, es decir desde otro ordenador.

- **Servidor SIP:** Se encargan de gestionar el establecimiento de las llamadas telefónicas por internet. Los SIP almacenan la dirección IP donde deben acceder para realizar la comunicación con un usuario. No transmite ni audio ni video, solo establece la comunicación.

- **List Server o Servidores Lista:** Permiten gestionar listas de correos.

- **Servidores Cloud:** Realmente estos servidores lo único que hacen es dejarte o alquilarte un espacio del servidor. La mayoría se utilizan para almacenar grandes cantidades de información en el servidor y tenerla protegida fuera de nuestro ordenador. Muchas empresas alquilan servidores cloud (en la nube) para tener en ellos toda la valiosa información de la empresa, utilizándola cuando quieran y realizando el propio servidor copias de seguridad.

- **Cluster de Servidores:** Un clúster de servidores es la agrupación de varios servidores dedicados a la misma tarea, Hay veces que un solo servidor se queda pequeño para toda la demanda de los clientes y es necesario más. En estos casos se agrupan en lo que se conoce como Cluster de Servidores.



### 3-PROTOS Y FUNCIONALIDAD DE LA CAPA DE APLICACIÓN

### 3.0 INTRODUCCIÓN DEL CAPITULO

### 3.0.1 Introducción del capítulo

La mayoría de nosotros experimentamos Internet a través de World Wide Web, servicios de e-mail y programas para compartir archivos. Éstas y muchas otras aplicaciones proporcionan la interfaz humana a la red subyacente, lo que nos permite enviar y recibir información con relativa facilidad. Generalmente, las aplicaciones que utilizamos son intuitivas; es decir, podemos acceder a ellas y usarlas sin saber cómo funcionan. Sin embargo, para los profesionales de redes es importante conocer cómo una aplicación puede formatear, transmitir e interpretar mensajes que se envían y reciben a través de la red.

La visualización de los mecanismos que permiten la comunicación a través de la red se hace más sencilla si utilizamos el marco en capas del modelo Interconexión de sistema abierto (OSI). En este capítulo, enfatizaremos el rol de una capa, la capa de Aplicación, y sus componentes: aplicaciones, servicios y protocolos. Exploraremos cómo esos tres elementos hacen posible la comunicación sólida a través de la red de información.

En este capítulo aprenderá a:

- Describir cómo las funciones de las tres capas superiores del modelo OSI proporcionan servicios de red a las aplicaciones de usuario final.
- Describir cómo los protocolos de la capa de aplicación TCP/IP proporcionan los servicios especificados por las capas superiores del modelo OSI.
- Definir cómo la gente utiliza la capa de aplicación para comunicarse a través de la red de información.
- Describir la función de las conocidas aplicaciones TCP/IP, como la World Wide Web y el correo electrónico, y sus servicios relacionados (HTTP, DNS, SMB, DHCP, SMTP/POP y Telnet).
- Describir los procesos para compartir archivos que utilizan las aplicaciones entre pares y el protocolo Gnutella.
- Explicar cómo los protocolos garantizan que los servicios que se ejecutan en una clase de dispositivo puedan enviar y recibir datos desde y hacia muchos dispositivos de red diferentes.
- Utilizar herramientas de análisis de red para examinar y explicar cómo funcionan las aplicaciones comunes de usuarios.



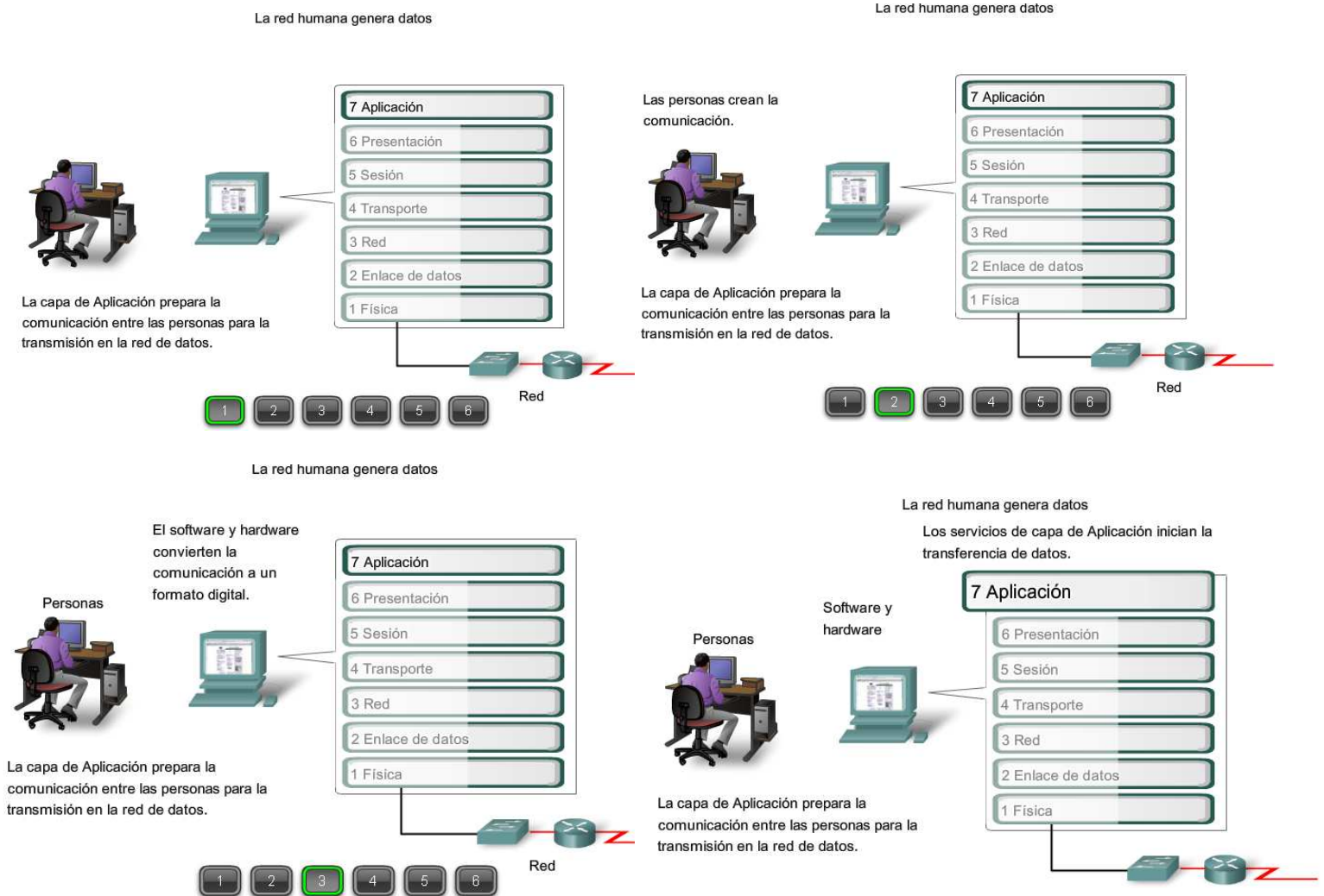
## 3.1 APLICACIONES: LA INTERFACE ENTRE REDES

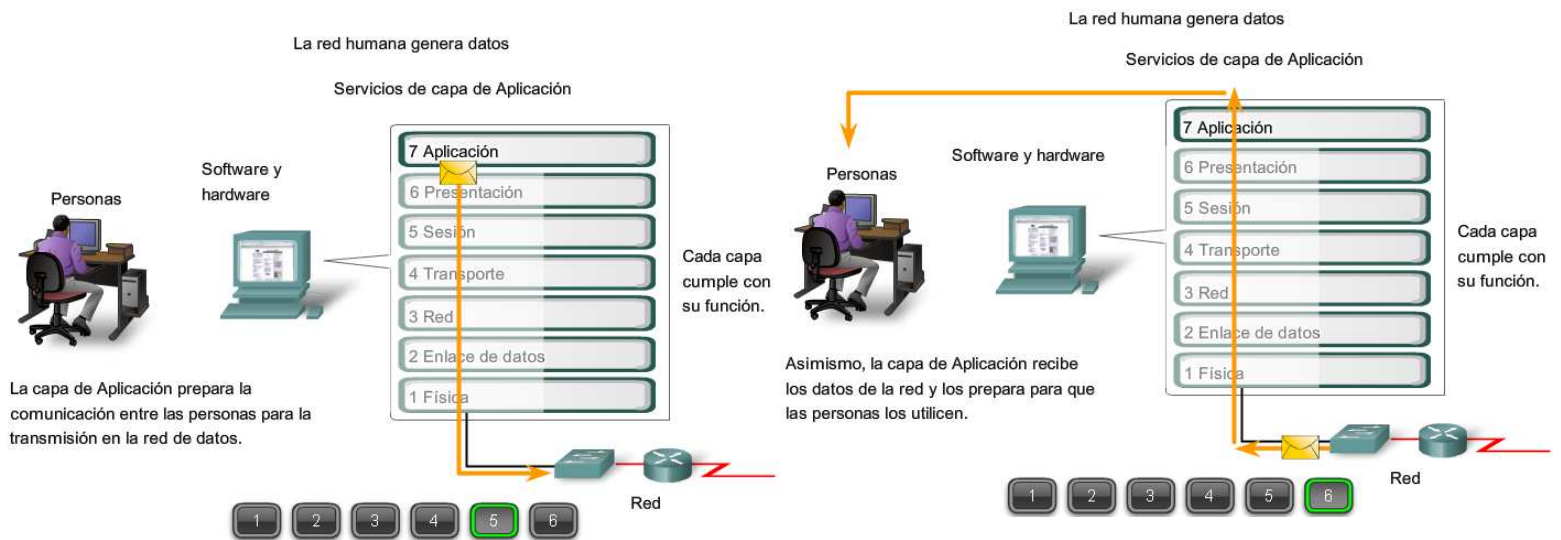
### 3.1.1 Modelo OSI y Modelo TCP/IP

El modelo de referencia de interconexión de sistemas abiertos es una representación abstracta en capas, creada como guía para el diseño del protocolo de red. El modelo OSI divide el proceso de networking en diferentes capas lógicas, cada una de las cuales tiene una única funcionalidad y a la cual se le asignan protocolos y servicios específicos.

En este modelo, la información se pasa de una capa a otra, comenzando en la capa de Aplicación en el host de transmisión, siguiendo por la jerarquía hacia la capa Física, pasando por el canal de comunicaciones al host de destino, donde la información vuelve a la jerarquía y termina en la capa de Aplicación. La figura ilustra los pasos en este proceso.

La capa de Aplicación, Capa siete, es la capa superior de los modelos OSI y TCP/IP. Es la capa que proporciona la interfaz entre las aplicaciones que utilizamos para comunicarnos y la red subyacente en la cual se transmiten los mensajes. Los protocolos de capa de aplicación se utilizan para intercambiar los datos entre los programas que se ejecutan en los hosts de origen y destino. Existen muchos protocolos de capa de aplicación y siempre se desarrollan protocolos nuevos.





Aunque el grupo de protocolos TCP/IP se desarrolló antes de la definición del modelo OSI, la funcionalidad de los protocolos de capa de aplicación de TCP/IP se adaptan aproximadamente a la estructura de las tres capas superiores del modelo OSI: Capas de Aplicación, Presentación y Sesión.

La mayoría de los protocolos de capa de aplicación de TCP/IP se desarrollaron antes de la aparición de computadoras personales, interfaces del usuario gráficas y objetos multimedia. Como resultado, estos protocolos implementan muy poco de la funcionalidad que se especifica en las capas de Sesión y Presentación del modelo OSI.

### Capa de Presentación

La capa de Presentación tiene tres funciones primarias:

- Codificación y conversión de datos de la capa de aplicación para garantizar que los datos del dispositivo de origen puedan ser interpretados por la aplicación adecuada en el dispositivo de destino.
- Compresión de los datos de forma que puedan ser descomprimidos por el dispositivo de destino.
- Encriptación de los datos para transmisión y descifre de los datos cuando se reciben en el destino.

Las implementaciones de la capa de presentación generalmente no se vinculan con una stack de protocolos determinada. Los estándares para vídeos y gráficos son algunos ejemplos. Dentro de los estándares más conocidos para vídeo encontramos QuickTime y el Grupo de expertos en películas (MPEG). QuickTime es una especificación de Apple Computer para audio y vídeo, y MPEG es un estándar para la codificación y compresión de vídeos.

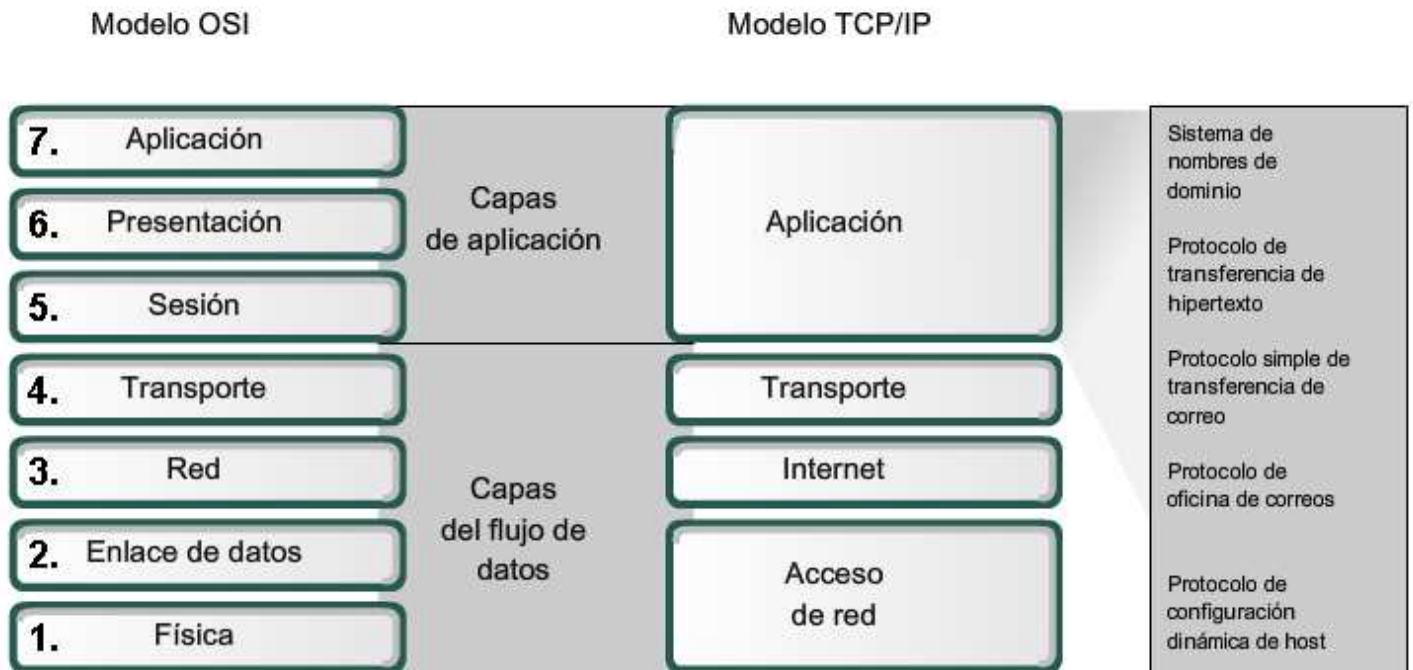
Dentro de los formatos de imagen gráfica más conocidos encontramos Formato de intercambio gráfico (GIF), Grupo de expertos en fotografía (JPEG) y Formato de archivo de imagen etiquetada (TIFF). GIF y JPEG son estándares de compresión y codificación para imágenes gráficas, y TIFF es una formato de codificación estándar para imágenes gráficas.

### Capa de Sesión

Como lo indica el nombre de la capa de Sesión, las funciones en esta capa crean y mantienen diálogos entre las aplicaciones de origen y destino. La capa de sesión maneja el intercambio de información para iniciar los diálogos y

mantenerlos activos, y para reiniciar sesiones que se interrumpieron o desactivaron durante un periodo de tiempo prolongado.

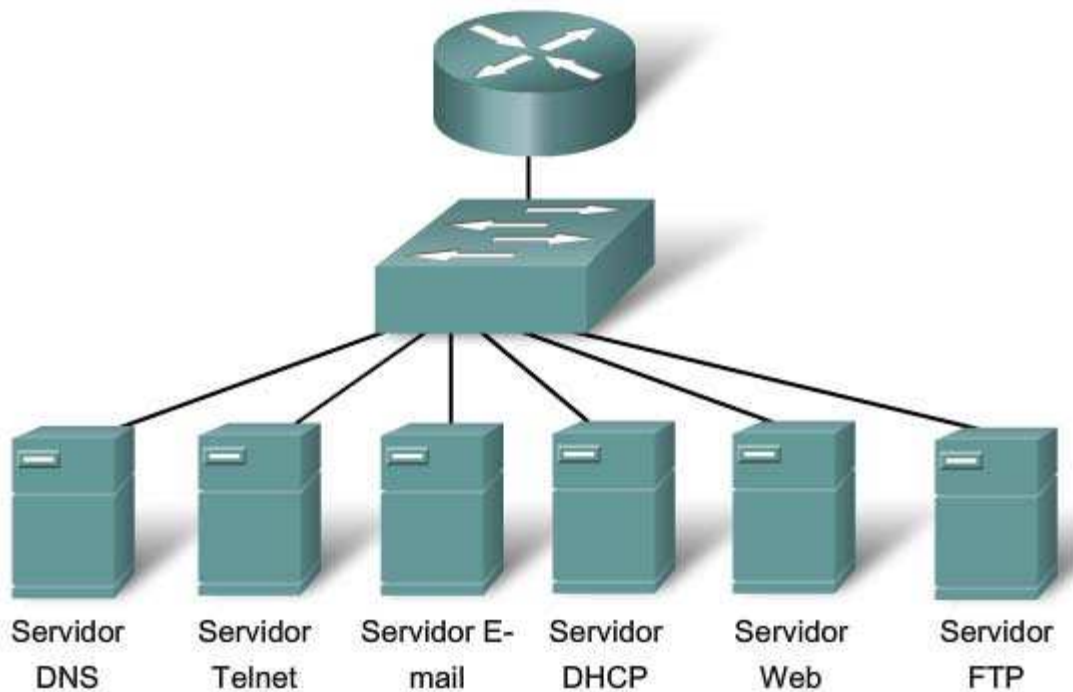
La mayoría de las aplicaciones, como los exploradores Web o los clientes de correo electrónico, incorporan la funcionalidad de las capas 5, 6 y 7 del modelo OSI.



Los protocolos de capa de aplicación de TCP/IP más conocidos son aquellos que proporcionan intercambio de la información del usuario. Estos protocolos especifican la información de control y formato necesaria para muchas de las funciones de comunicación de Internet más comunes. Algunos de los protocolos TCP/IP son:

- El protocolo Servicio de nombres de dominio (DNS, Domain Name Service) se utiliza para resolver nombres de Internet en direcciones IP.
- El protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) se utiliza para transferir archivos que forman las páginas Web de la World Wide Web.
- El Protocolo simple de transferencia de correo (SMTP) se utiliza para la transferencia de mensajes de correo y adjuntos.
- Telnet, un protocolo de emulación de terminal, se utiliza para proporcionar acceso remoto a servidores y a dispositivos de red.
- El Protocolo de transferencia de archivos (FTP, File Transfer Protocol) se utiliza para la transferencia interactiva de archivos entre sistemas.

Los protocolos de la suite TCP/IP generalmente son definidos por Solicitudes de comentarios (RFCs). El Grupo de trabajo de ingeniería de Internet mantiene las RFCs como los estándares para el conjunto TCP/IP.



Granja de servidores

#### Servidor de nombres de dominios (DNS)

- Servicio que ofrece la dirección IP de un sitio Web o nombre de dominio para que un host pueda conectarse a éste

#### Servidor Telnet

- Servicio que permite a los administradores conectarse a un host desde una ubicación remota y controlar el host como si estuvieran registrados en forma local

#### Servidor E-mail

- Utiliza el Simple Mail Transfer Protocol (SMTP) y Post Office Protocol (POP3) o Internet Message Access Protocol (IMAP)
- Se utiliza para enviar mensajes de e-mail de clientes a servidores a través de Internet
- Los destinatarios se especifican a través del formato usuario@xyz

#### Servidor Dynamic Host Configuration Protocol (DHCP)

- Servicio que asigna el gateway por defecto de la máscara de subred de dirección IP y demás información a los clientes



#### Servidor Web

- Hypertext Transfer Protocol (HTTP)
- Se utiliza para transferir información entre clientes Web y servidores Web
- Se accede a la mayoría de las páginas Web a través de HTTP

#### Servidor File Transfer Protocol (FTP)

- Servicio que permite descargar y subir archivos entre un cliente y un servidor

### 3.1.2 Software de la capa de aplicación

Las funciones asociadas con los protocolos de capa de Aplicación permiten a la red humana comunicarse con la red de datos subyacente. Cuando abrimos un explorador Web o una ventana de mensajería instantánea, se inicia una aplicación, y el programa se coloca en la memoria del dispositivo donde se ejecuta. Cada programa ejecutable cargado a un dispositivo se denomina proceso.

Dentro de la capa de Aplicación, existen dos formas de procesos o programas de software que proporcionan acceso a la red: aplicaciones y servicios.

#### Aplicaciones reconocidas por la red

Aplicaciones son los programas de software que utiliza la gente para comunicarse a través de la red. Algunas aplicaciones de usuario final son compatibles con la red, lo cual significa que implementan los protocolos de la capa de aplicación y pueden comunicarse directamente con las capas inferiores del stack de protocolos. Los clientes de correo electrónico y los exploradores Web son ejemplos de este tipo de aplicaciones.

#### Servicios de la capa de Aplicación

Otros programas pueden necesitar la ayuda de los servicios de la capa de Aplicación para utilizar los recursos de la red, como transferencia de archivos o cola de impresión en red. Aunque son transparentes para el usuario, estos servicios son los programas que se comunican con la red y preparan los datos para la transferencia. Diferentes tipos de datos, ya sea texto, gráfico o vídeo, requieren de diversos servicios de red para asegurarse de que estén bien preparados para procesar las funciones de las capas inferiores del modelo OSI.

Cada servicio de red o aplicación utiliza protocolos que definen los estándares y formatos de datos a utilizarse. Sin protocolos, la red de datos no tendría una manera común de formatear y direccionar los datos. Para comprender la función de los distintos servicios de red, es necesario familiarizarse con los protocolos subyacentes que rigen su operación.

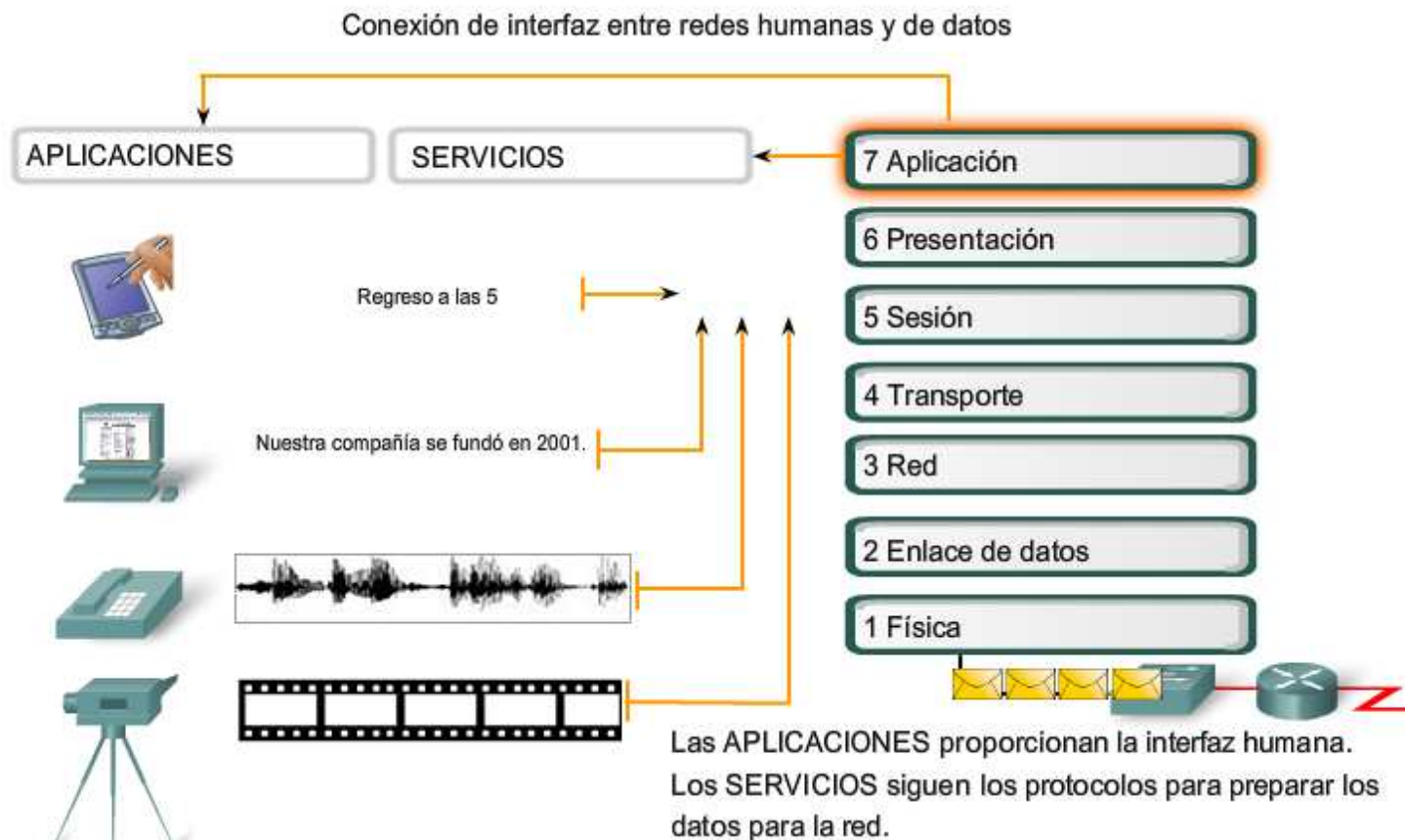
Los procesos pueden ser

- 1 Aplicaciones
- 2 Servicios
- 3 Operaciones del sistema
- 4 Un programa puede estar en ejecución varias veces, cada vez dentro de su propio proceso.  
Coloque el cursor sobre un elemento.

## Ejemplos de procesos en ejecución en el sistema operativo Windows

Como se mencionó anteriormente, la capa de Aplicación utiliza los protocolos implementados dentro de las aplicaciones y servicios. Mientras que las aplicaciones proporcionan a las personas una forma de crear mensajes y los servicios de la capa de aplicación establecen una interfaz con la red, los protocolos proporcionan las reglas y los formatos que regulan el tratamiento de los datos. Un único programa ejecutable debe utilizar los tres componentes e inclusive el mismo nombre. Por ejemplo: cuando analizamos “Telnet” nos podemos referir a la aplicación, el servicio o el protocolo.

En el modelo OSI, se considera que las aplicaciones que interactúan directamente con las personas se encuentran en la parte superior del stack, al igual que las personas. Al igual que todas las personas dentro del modelo OSI, la capa de Aplicación se basa en las funciones de las capas inferiores para completar el proceso de comunicación. Dentro de la capa de aplicación, los protocolos especifican qué mensajes se intercambian entre los host de origen y de destino, la sintaxis de los comandos de control, el tipo y formato de los datos que se transmiten y los métodos adecuados para notificación y recuperación de errores.



### 3.1.4 Funciones del protocolo de la Capa de Aplicación

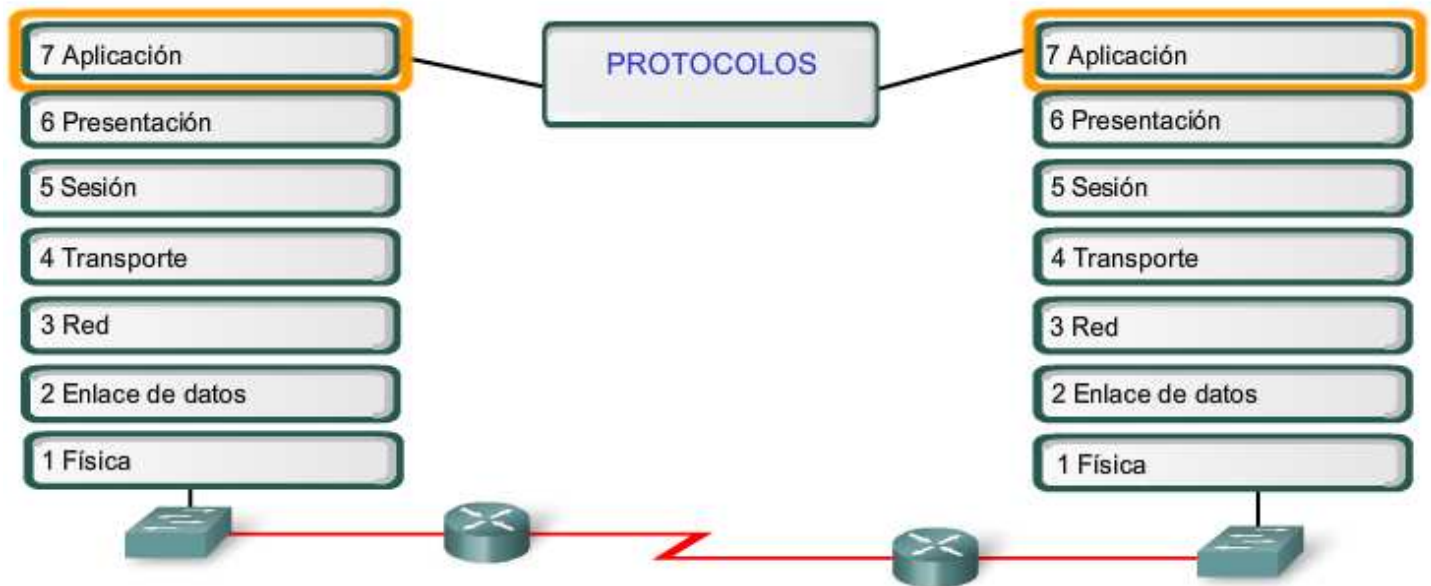
Los protocolos de la capa de aplicación son utilizados tanto por los dispositivos de origen como de destino durante una sesión de comunicación. Para que las comunicaciones sean exitosas, deben coincidir los protocolos de capa de aplicación implementados en el host de origen y destino.

Los protocolos establecen reglas consistentes para intercambiar datos entre las aplicaciones y los servicios cargados en los dispositivos participantes. Los protocolos especifican cómo se estructuran los datos dentro de los mensajes y los tipos de mensajes que se envían entre origen y destino. Estos mensajes pueden ser solicitudes de servicios, acuses de recibo, mensajes de datos, mensajes de estado o mensajes de error. Los protocolos también definen los diálogos de mensajes, asegurando que un mensaje enviado encuentre la respuesta esperada y se invoquen los servicios correspondientes cuando se realiza la transferencia de datos.

Muchos y diversos tipos de aplicaciones se comunican a través de las redes de datos. Por lo tanto, los servicios de la capa de Aplicación deben implementar protocolos múltiples para proporcionar la variedad deseada de experiencias de comunicación. Cada protocolo tiene un fin específico y contiene las características requeridas para cumplir con dicho propósito. Deben seguirse los detalles del protocolo correspondiente a cada capa, así las funciones en una capa se comunican correctamente con los servicios en la capa inferior.

Las aplicaciones y los servicios también pueden utilizar protocolos múltiples durante el curso de una comunicación simple. Un protocolo puede especificar cómo se establece la conexión de redes y otro describir el proceso para la transferencia de datos cuando el mensaje se pasa a la siguiente capa inferior.





Los protocolos de capa de Aplicación proporcionan las reglas para la comunicación entre las aplicaciones.

#### Protocolos:

- Define los procesos en cada uno de los extremos de la comunicación
- Define los tipos de mensajes
- Define la sintaxis de los mensajes
- Define el significado de los campos de información
- Define la forma en que se envían los mensajes y la respuesta esperada
- Define la interacción con la próxima capa inferior

## 3.2 TOMA DE MEDIDAS PARA LAS APLICACIONES Y SERVICIOS

### 3.2.1 El modelo cliente-servidor

Cuando la gente intenta acceder a información en sus dispositivos, ya sean éstos una computadora personal o portátil, un PDA, teléfono celular o cualquier otro dispositivo conectado a la red, los datos pueden no estar físicamente almacenados en sus dispositivos. Si así fuere, se debe solicitar al dispositivo que contiene los datos, permiso para acceder a esa información.

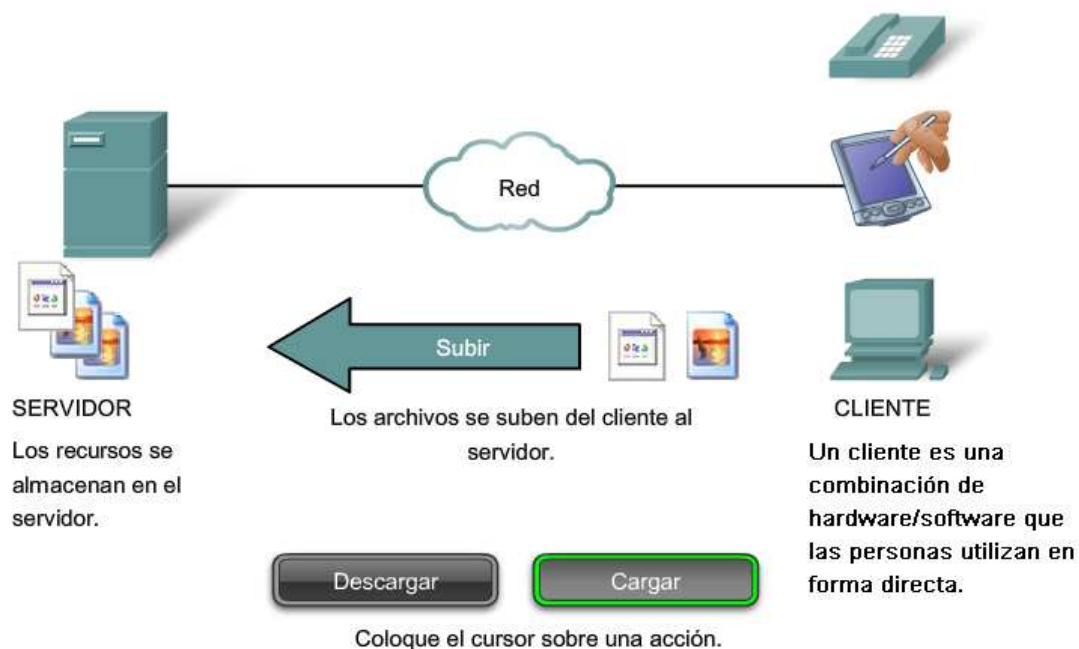
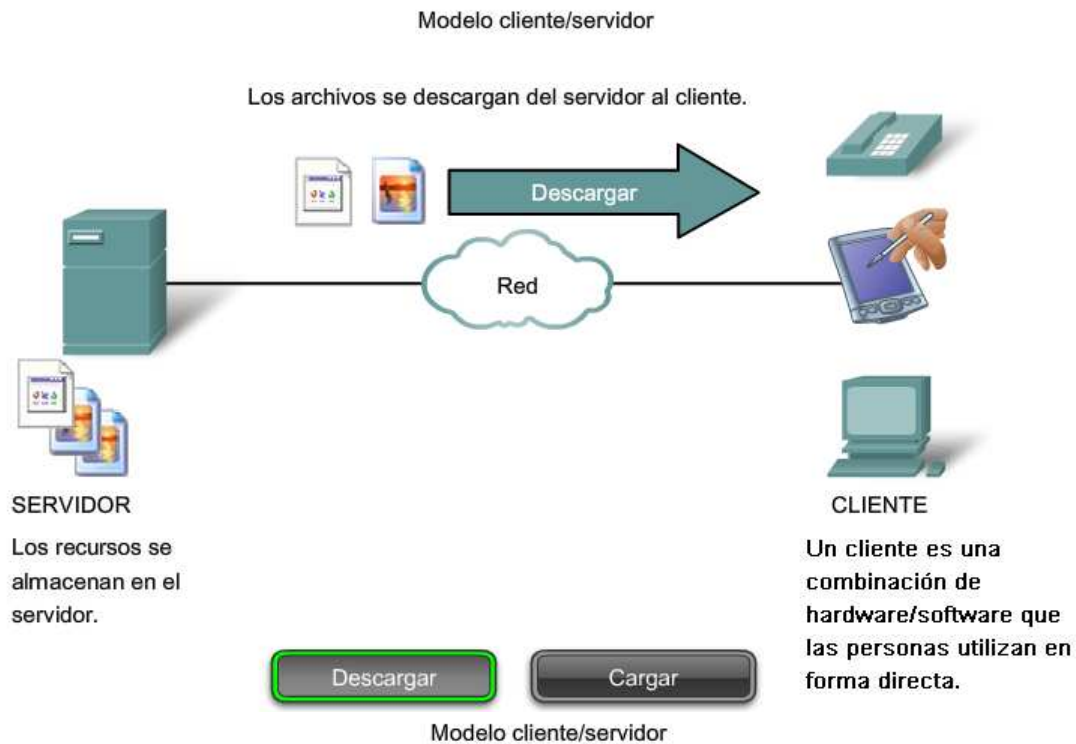
#### Modelo cliente-servidor

En el modelo cliente-servidor, el dispositivo que solicita información se denomina cliente y el dispositivo que responde a la solicitud se denomina servidor. Los procesos de cliente y servidor se consideran una parte de la capa de Aplicación. El cliente comienza el intercambio solicitando los datos al servidor, que responde enviando uno o más streams de datos al cliente. Los protocolos de capa de Aplicación describen el formato de las solicitudes y respuestas entre clientes y servidores. Además de la transferencia real de datos, este intercambio puede requerir de información adicional, como la autenticación del usuario y la identificación de un archivo de datos a transferir.

Un ejemplo de una red cliente/servidor es un entorno corporativo donde los empleados utilizan un servidor de e-mail de la empresa para enviar, recibir y almacenar e-mails. El cliente de correo electrónico en la computadora de un empleado

emite una solicitud al servidor de e-mail para un mensaje no leído. El servidor responde enviando el e-mail solicitado al cliente.

Aunque los datos generalmente se describen como un flujo del servidor al cliente, algunos datos siempre fluyen del cliente al servidor. El flujo de datos puede ser el mismo en ambas direcciones o inclusive ser mayor en la dirección que va del cliente al servidor. Por ejemplo, un cliente puede transferir un archivo al servidor con fines de almacenamiento. La transferencia de datos de un cliente a un servidor se conoce como subida y la de los datos de un servidor a un cliente, descarga.

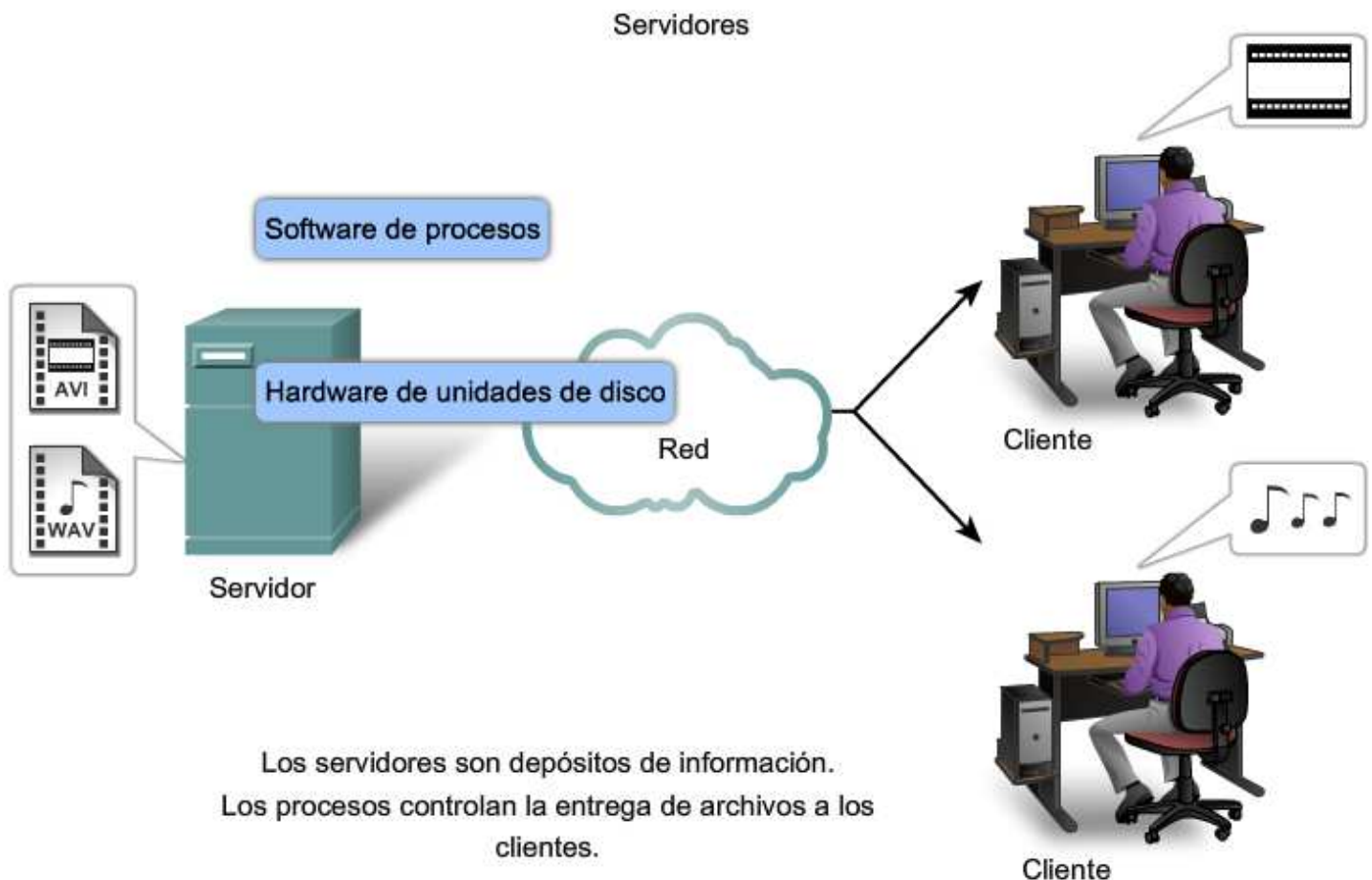


### 3.2.2 Servidores

**En un contexto general de redes, cualquier dispositivo que responde a una solicitud de aplicaciones de cliente funciona como un servidor.** Un servidor generalmente es una computadora que contiene información para ser compartida con muchos sistemas de cliente. Por ejemplo, páginas Web, documentos, bases de datos, imágenes, archivos de audio y vídeo pueden almacenarse en un servidor y enviarse a los clientes que lo solicitan. En otros casos, como una impresora de red, el servidor de impresión envía las solicitudes de impresión del cliente a la impresora específica.

Diferentes tipos de aplicaciones del servidor tienen diferentes requerimientos para el acceso de clientes. Algunos servidores pueden requerir de autenticación de la información de cuenta del usuario para verificar si el usuario tiene permiso para acceder a los datos solicitados o para utilizar una operación en particular. Dichos servidores deben contar con una lista central de cuentas de usuarios y autorizaciones, o permisos (para operaciones y acceso a datos) otorgados a cada usuario. Cuando se utiliza un cliente FTP, por ejemplo, si usted solicita subir datos al servidor FTP, se le puede dar permiso para escribir su carpeta personal pero no para leer otros archivos del sitio.

En una red cliente-servidor, el servidor ejecuta un servicio o proceso, a veces denominado daemon de servidor. Al igual que la mayoría de los servicios, los daemons generalmente se ejecutan en segundo plano y no se encuentran bajo control directo del usuario. Los daemons se describen como servidores que “escuchan” una solicitud del cliente, porque están programados para responder cada vez que el servidor recibe una solicitud para el servicio proporcionado por el daemon. Cuando un daemon “escucha” una solicitud de un cliente, intercambia los mensajes adecuados con el cliente, según lo requerido por su protocolo, y procede a enviar los datos solicitados al cliente en el formato correspondiente.

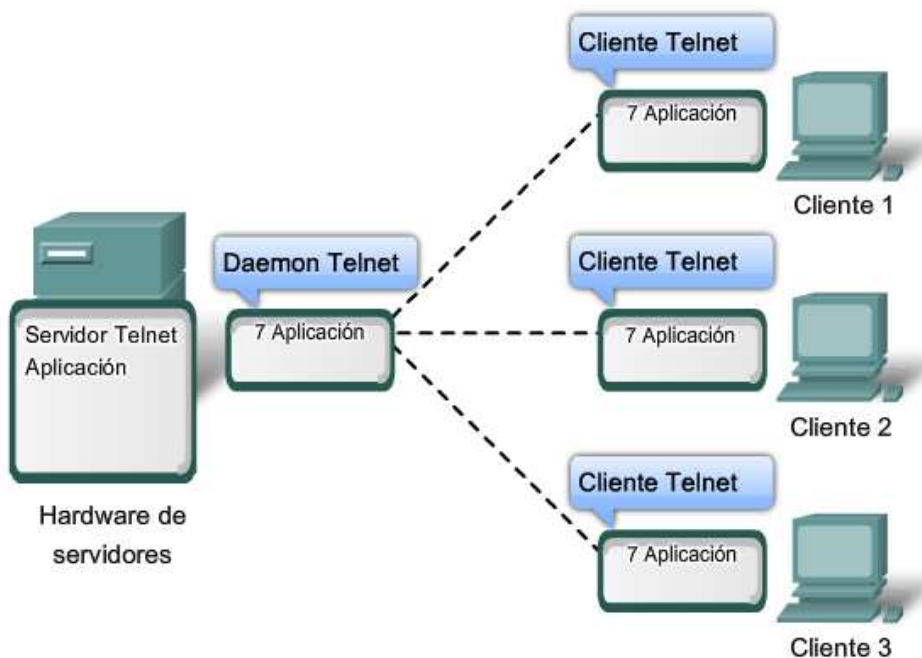


### 3.2.3 Protocolos y servicios de la capa de Aplicación

Una única aplicación puede emplear diferentes servicios de la capa de Aplicación, así lo que aparece para el usuario como una solicitud para una página Web puede, de hecho, ascender a docenas de solicitudes individuales. Y, para cada solicitud, pueden ejecutarse múltiples procesos. Por ejemplo, un cliente puede necesitar de diversos procesos individuales para formular sólo una solicitud al servidor.

Además, los servidores generalmente tienen múltiples clientes que solicitan información al mismo tiempo. Por ejemplo, un servidor Telnet puede tener varios clientes que requieren conectarse a él. Estas solicitudes individuales del cliente pueden manejarse en forma simultánea y separada para que la red sea exitosa. Los servicios y procesos de capa de Aplicación dependen del soporte de las funciones de la capa inferior para administrar en forma exitosa las múltiples conversaciones.

Los procesos de servidores pueden admitir múltiples clientes.



### 3.2.4 Redes y aplicaciones entre pares (P2P, Peer-to-Peer)

#### Modelo Punto a Punto

Además del modelo cliente/servidor para redes, existe también un modelo punto a punto. Las redes punto a punto tienen dos formas distintivas: diseño de redes punto a punto y aplicaciones punto a punto (P2P). Ambas formas tienen características similares pero en la práctica funcionan en forma muy distinta.

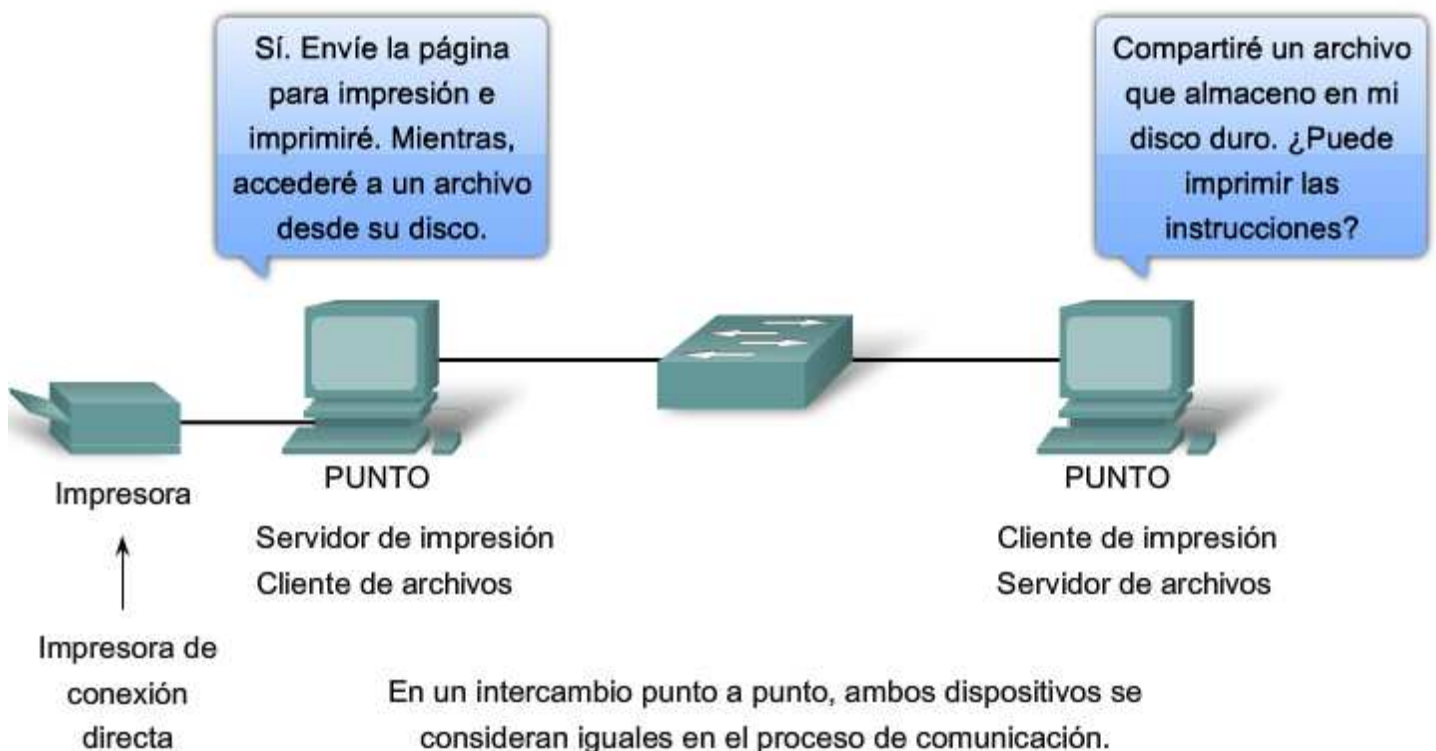
#### Redes entre pares

En una red entre pares, dos o más computadoras están conectadas a través de una red y pueden compartir recursos (por ejemplo, impresora y archivos) sin tener un servidor dedicado. Cada dispositivo final conectado (conocido como punto) puede funcionar como un servidor o como un cliente. Una computadora puede asumir el rol de servidor para una transacción mientras funciona en forma simultánea como cliente para otra transacción. Los roles del cliente y el servidor se configuran según las solicitudes.

Un ejemplo de una red entre pares es una simple red doméstica con dos computadoras conectadas que comparten una impresora. Cada persona puede configurar su computadora para compartir archivos, habilitar juegos en red o compartir una conexión de Internet. Otro ejemplo sobre la funcionalidad de la red punto a punto son dos computadoras conectadas a una gran red que utilizan aplicaciones de software para compartir recursos entre ellas a través de la red.

A diferencia del modelo cliente/servidor, que utiliza servidores dedicados, las redes punto a punto descentralizan los recursos en una red. En lugar de ubicar información para compartir en los servidores dedicados, la información puede colocarse en cualquier parte de un dispositivo conectado. La mayoría de los sistemas operativos actuales admiten compartir archivos e impresoras sin requerir software del servidor adicional. Debido a que las redes punto a punto generalmente no utilizan cuentas de usuarios centralizadas, permisos ni monitores, es difícil implementar las políticas de acceso y seguridad en las redes que contienen mayor cantidad de computadoras. Se deben establecer cuentas de usuario y derechos de acceso en forma individual para cada dispositivo.

### Redes punto a punto



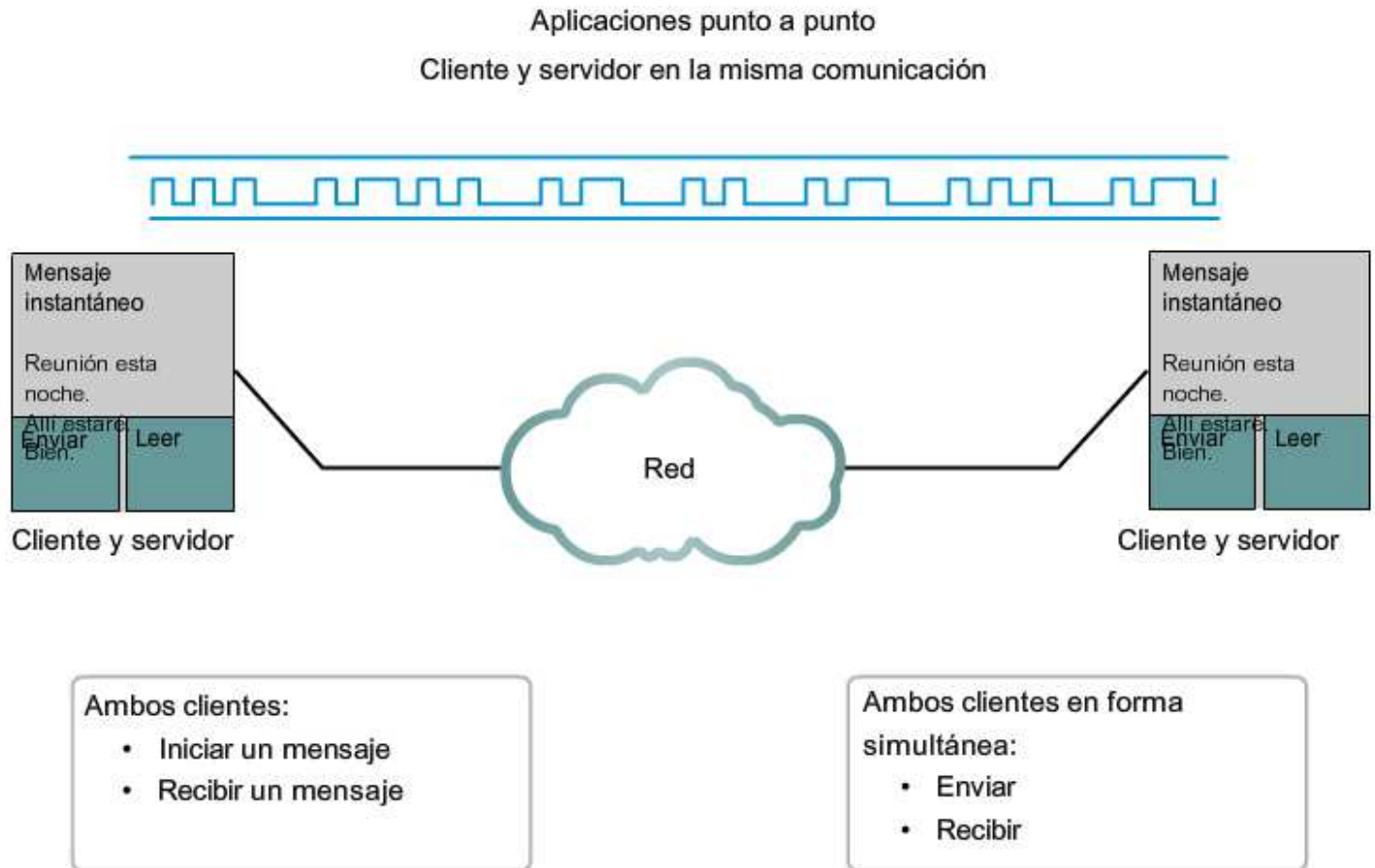
### Aplicaciones punto a punto

Una aplicación punto a punto (P2P), a diferencia de una red punto a punto, permite a un dispositivo actuar como cliente o como servidor dentro de la misma comunicación. En este modelo, cada cliente es un servidor y cada servidor es un cliente. Ambos pueden iniciar una comunicación y se consideran iguales en el proceso de comunicación. Sin embargo, las aplicaciones punto a punto requieren que cada dispositivo final proporcione una interfaz de usuario y ejecute un servicio en segundo plano. Cuando inicia una aplicación punto a punto específica, ésta invoca la interfaz de usuario requerida y los servicios en segundo plano. Luego, los dispositivos pueden comunicarse directamente.



Algunas aplicaciones P2P utilizan un sistema híbrido donde se descentraliza el acceso a los recursos pero los índices que apuntan a las ubicaciones de los recursos están almacenados en un directorio centralizado. En un sistema híbrido, cada punto accede a un servidor de índice para alcanzar la ubicación de un recurso almacenado en otro punto. El servidor de índice también puede ayudar a conectar dos puntos, pero una vez conectados, la comunicación se lleva a cabo entre los dos puntos, sin comunicación adicional al servidor de índice.

Las aplicaciones punto a punto pueden utilizarse en las redes punto a punto, en redes cliente/servidor y en Internet.



### 3.3 EJEMPLOS DE SERVICIOS Y PROTOCOLOS DE LA CAPA DE APLICACIÓN

#### 3.3.1 Protocolo y servicios DNS

Ahora que comprendemos mejor cómo las aplicaciones proporcionan una interfaz para el usuario y acceso a la red, veremos algunos protocolos específicos que se utilizan comúnmente.

Como veremos más adelante, la capa de transporte utiliza un esquema de direccionamiento que se llama número de puerto. Los números de puerto identifican las aplicaciones y los servicios de la capa de Aplicación que son los datos de origen y destino. Los programas del servidor generalmente utilizan números de puerto predefinidos comúnmente conocidos por los clientes. Mientras examinamos los diferentes servicios y protocolos de la capa de Aplicación de TCP/IP, nos referiremos a los números de puerto TCP y UDP normalmente asociados con estos servicios. Algunos de estos servicios son:

- Sistema de nombres de dominio (DNS): puerto TCP/UDP 53.
- Protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol): puerto TCP 80.
- Protocolo simple de transferencia de correo (SMTP, Simple Mail Transfer Protocol): puerto TCP 25.

- Protocolo de oficina de correos (POP): puerto UDP 110.
- Telnet: puerto TCP 23.
- Protocolo de configuración dinámica de host: puerto UDP 67.
- Protocolo de transferencia de archivos (FTP, File Transfer Protocol): puertos TCP 20 y 21.

## DNS

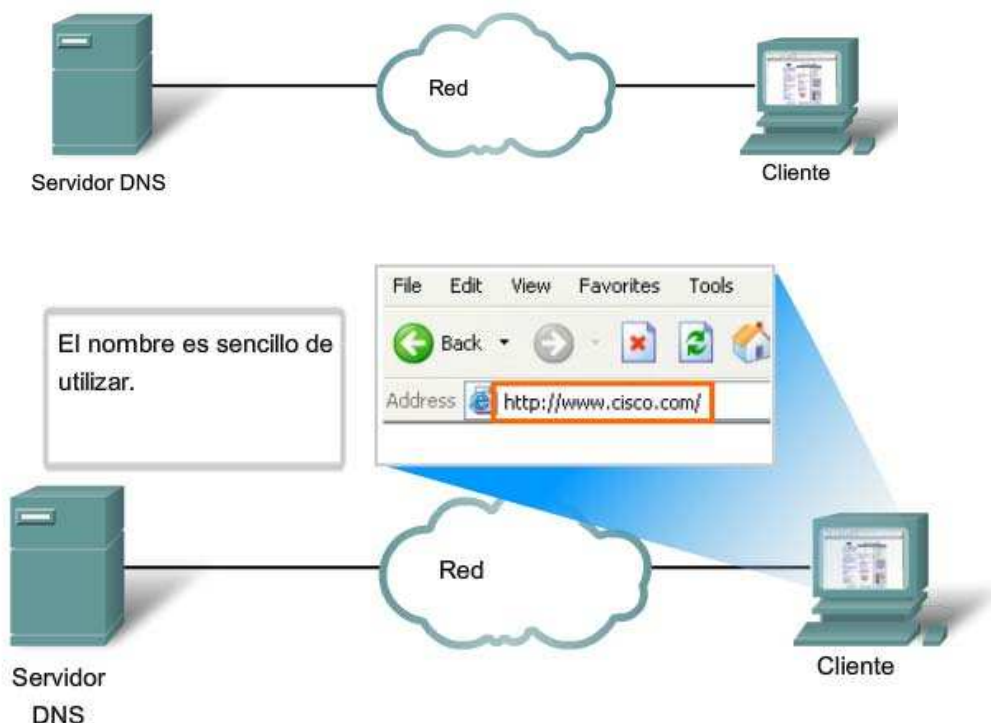
En redes de datos, los dispositivos son rotulados con direcciones IP numéricas para que puedan participar en el envío y recepción de mensajes a través de la red. Sin embargo, la mayoría de las personas pasan mucho tiempo tratando de recordar estas direcciones numéricas. Por lo tanto, los nombres de dominio fueron creados para convertir las direcciones numéricas en nombres simples y reconocibles.

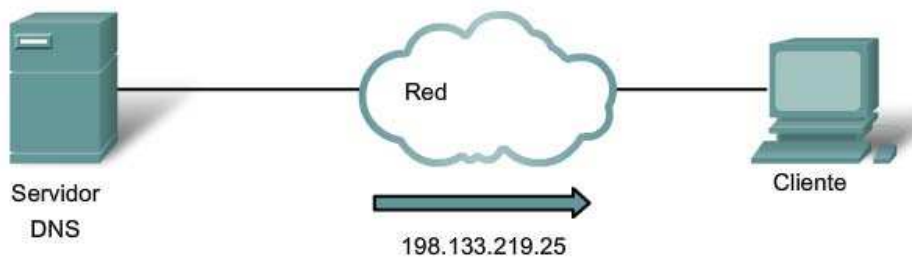
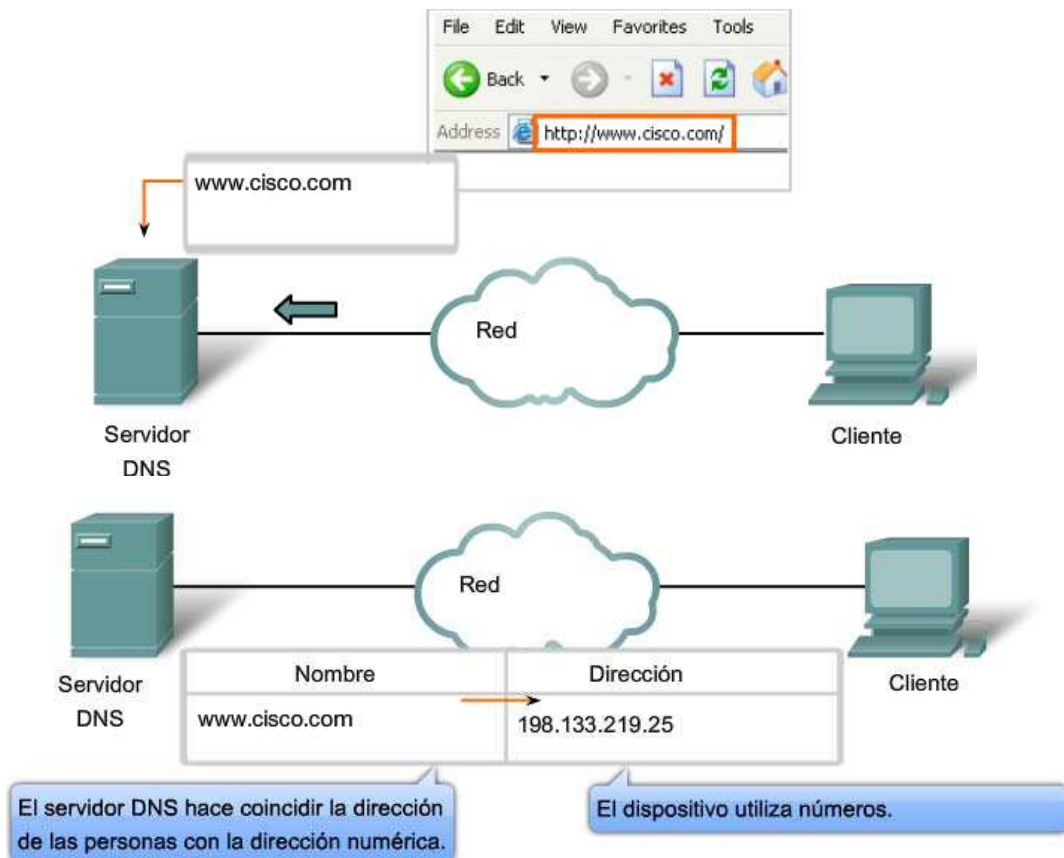
En Internet, esos nombres de dominio, como [www.cisco.com](http://www.cisco.com), son mucho más sencillos de recordar que 198.133.219.25, que es la dirección numérica real para este servidor. Además, si Cisco decide cambiar la dirección numérica, para el usuario es transparente ya que el nombre de dominio seguirá siendo [www.cisco.com](http://www.cisco.com). La nueva dirección simplemente estará enlazada con el nombre de dominio existente y la conectividad se mantendrá. Cuando las redes eran pequeñas, resultaba fácil mantener la asignación entre los nombres de dominios y las direcciones que representaban. Sin embargo, a medida que las redes y el número de dispositivos comenzó a crecer, el sistema manual dejó de ser práctico.

El Sistema de nombres de dominio (DNS) se creó para que el nombre del dominio busque soluciones para estas redes. DNS utiliza un conjunto distribuido de servidores para resolver los nombres asociados con estas direcciones numéricas.

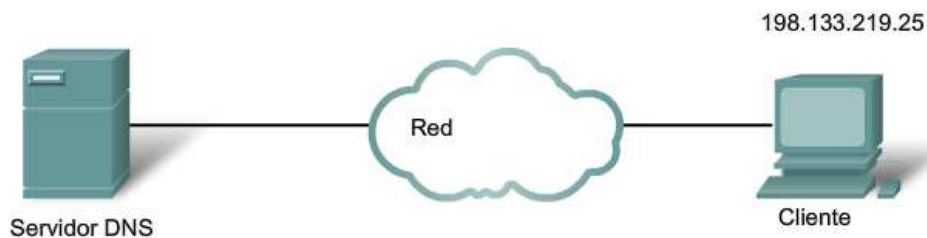
El protocolo DNS define un servicio automatizado que coincide con nombres de recursos que tienen la dirección de red numérica solicitada. Incluye las consultas sobre formato, las respuestas y los formatos de datos. Las comunicaciones del protocolo DNS utilizan un formato simple llamado mensaje. Este formato de mensaje se utiliza para todos los tipos de solicitudes de clientes y respuestas del servidor, mensajes de error y para la transferencia de información de registro de recursos entre servidores.

### RESOLUCION DE NOMBRES DNS





El número se envía de regreso al cliente para su utilización en la realización de solicitudes del servidor.



Se resuelve un nombre de persona legible para la dirección del dispositivo de red numérico por parte del protocolo DNS.



DNS es un servicio cliente/servidor; sin embargo, difiere de los otros servicios cliente/servidor que estamos examinando. Mientras otros servicios utilizan un cliente que es una aplicación (como un explorador Web o un cliente de correo electrónico), el cliente DNS ejecuta un servicio por sí mismo. El cliente DNS, a veces denominado resolución DNS, admite resolución de nombre para otras aplicaciones de red y servicios que lo necesiten.

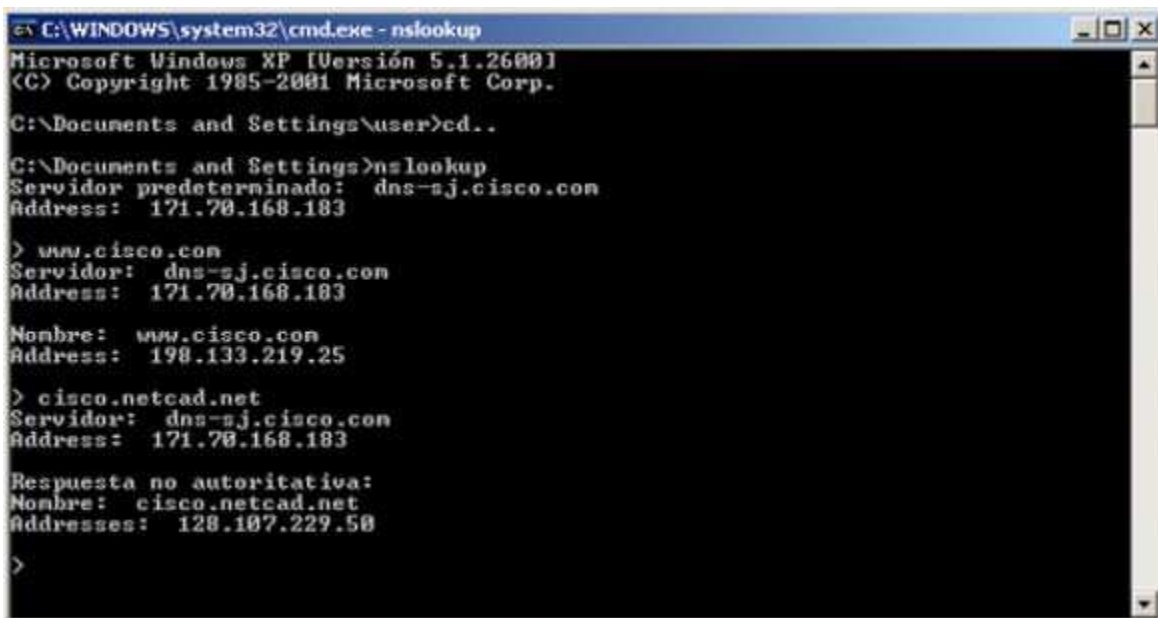
Al configurar un dispositivo de red, generalmente proporcionamos una o más direcciones del servidor DNS que el cliente DNS puede utilizar para la resolución de nombres. En general, el proveedor de servicios de Internet provee las direcciones para utilizar con los servidores DNS. Cuando una aplicación de usuario solicita conectarse con un dispositivo remoto por nombre, el cliente DNS solicitante envía una petición a uno de esos servidores de nombre para resolver el nombre en una dirección numérica.

Los sistemas operativos informáticos también tienen una utilidad denominada nslookup que permite al usuario consultar manualmente los servidores de nombre para resolver un determinado nombre de host. Esta utilidad también puede utilizarse para resolver los problemas de resolución de nombres y verificar el estado actual de los servidores de nombres.

En la figura, cuando se ejecuta nslookup, se muestra el servidor DNS por defecto configurado para su host. En este ejemplo, el servidor DNS es dns-sjk.cisco.com que tiene una dirección de 171.68.226.120.

Luego podemos escribir el nombre de un host o dominio para el cual deseamos obtener la dirección. En la primer consulta de la figura, se hace una consulta para www.cisco.com. El servidor de nombre que responde proporciona la dirección 198.133.219.25.

Las consultas mostradas en la figura son sólo pruebas simples. La utilidad nslookup tiene muchas opciones disponibles para lograr una extensa verificación y prueba del proceso DNS.



```
C:\WINDOWS\system32\cmd.exe - nslookup
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\user>cd..

C:\Documents and Settings>nslookup
Servidor predeterminado:  dns-sjk.cisco.com
Address:  171.70.168.183

> www.cisco.com
Servidor:  dns-sjk.cisco.com
Address:  171.70.168.183

Nombre:   www.cisco.com
Address:  198.133.219.25

> cisco.netcad.net
Servidor:  dns-sjk.cisco.com
Address:  171.70.168.183

Respuesta no autoritativa:
Nombre:   cisco.netcad.net
Addresses: 128.107.229.50

>
```

Un servidor DNS proporciona la resolución de nombres utilizando el daemon de nombre que generalmente se llama named (se pronuncia name-dee).

El servidor DNS almacena diferentes tipos de registros de recursos utilizados para resolver nombres. Estos registros contienen el nombre, la dirección y el tipo de registro.

Algunos de estos tipos de registro son:

- A: una dirección de un dispositivo final.
- NS: un servidor de nombre autoritativo.
- CNAME: el nombre ideal (o Nombre de dominio completamente calificado) para un alias, que se utiliza cuando varios servicios tienen una única dirección de red pero cada servicio tiene su propia entrada en DNS.
- MX: registro de intercambio de correos, asigna un nombre de dominio a una lista de servidores de intercambio de correos para ese dominio.

Cuando un cliente realiza una consulta, el proceso “nombrado” del servidor primero observa en sus propios registros para ver si puede resolver el nombre. Si no puede resolver el nombre utilizando los registros almacenados, contacta a otros servidores para hacerlo.

La solicitud puede pasar por un número de servidores, lo cual lleva tiempo adicional y consume ancho de banda. Una vez que se encuentra una coincidencia y se devuelve al servidor solicitante original, el servidor almacena temporalmente en la caché la dirección numerada que coincide con el nombre.

Si vuelve a solicitarse ese mismo nombre, el primer servidor puede regresar la dirección utilizando el valor almacenado en el caché de nombres. El almacenamiento en caché reduce el tráfico de la red de datos de consultas DNS y las cargas de trabajo de los servidores más altos de la jerarquía. El servicio del cliente DNS en las PC de Windows optimiza el rendimiento de la resolución de nombres DNS almacenando previamente los nombres resueltos en la memoria. El comando `ipconfig /displaydns` muestra todas las entradas DNS en caché en un sistema informático con Windows XP o 2000.

## Formato del mensaje DNS

### DNS utiliza el mismo formato de mensaje para:

- todos los tipos de consultas de clientes y respuestas de servidor
- mensajes de error
- la transferencia de información de registros de recursos entre servidores

Encabezado	
Pregunta	La pregunta para el servidor de nombres
Respuesta	Registros de recursos que responden la pregunta
Autoridad	Registros de recursos que apuntan a una autoridad
Adicional	Registros de recursos que poseen información adicional

El sistema de nombres de dominio utiliza un sistema jerárquico para crear una base de datos para proporcionar una resolución de nombres. La jerarquía es similar a un árbol invertido con la raíz en la parte superior y las ramas por debajo.

En la parte superior de la jerarquía, los servidores raíz mantienen registros sobre cómo alcanzar los servidores de dominio de nivel superior, los cuales a su vez tienen registros que apuntan a los servidores de dominio de nivel secundario y así sucesivamente.

Los diferentes dominios de primer nivel representan el tipo de organización o el país de origen. Algunos ejemplos de dominios de primer nivel son:

- .au: Australia
- .co: Colombia
- .com: una empresa o industria
- .jp: Japón
- .org: una organización sin fines de lucro

Después de los dominios de primer nivel se encuentran los dominios de segundo nivel y, debajo de estos, hay otros dominios de nivel inferior.

Cada nombre de dominio es una ruta a través de este árbol invertido que comienza desde la raíz.

Por ejemplo: como se muestra en la figura, el servidor DNS raíz puede no saber exactamente dónde se encuentra el servidor de correo electrónico mail.cisco, pero lleva un registro de los dominios “com” dentro de los dominios de primer nivel. Asimismo, los servidores dentro del dominio “com” pueden no tener un registro de mail.cisco.com, pero sí tienen un registro para el dominio “cisco.com”. Los servidores dentro del dominio cisco.com tienen un registro (un registro MX para ser exactos) para mail.cisco.com.

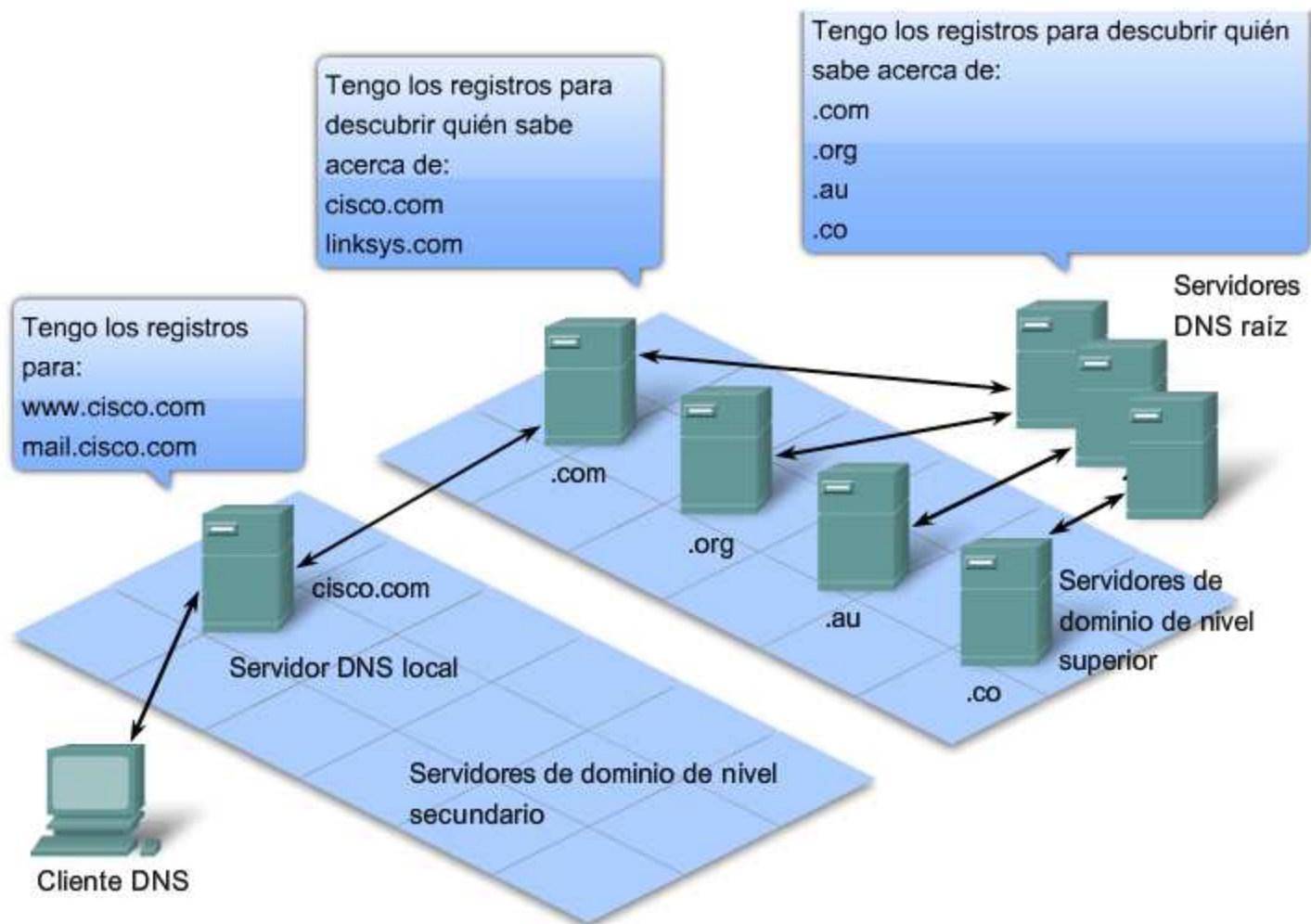
El sistema de nombres de dominio depende de esta jerarquía de servidores descentralizados y mantiene estos registros de recursos. Los registros de recursos enumeran nombres de dominios que el servidor puede resolver y servidores alternativos que también pueden procesar solicitudes. Si un determinado servidor tiene registros de recursos que corresponden a su nivel en la jerarquía de dominios, se dice que es autoritativo para esos registros.

Por ejemplo: un servidor de nombres en el dominio cisco.netacad.net no sería autoritativo para el registro mail.cisco.com porque ese registro se mantiene en un servidor de nivel de dominio superior, específicamente el servidor de nombres en el dominio cisco.com .

Enlaces

<http://www.ietf.org/rfc/rfc1034.txt>

<http://www.ietf.org/rfc/rfc1035.txt>



Una jerarquía de servidores DNS contiene los registros de recursos que coordinan los nombres con las direcciones.

### 3.3.2 Servicio WWW y HTTP

Cuando se escribe una dirección Web (o URL) en un explorador de Internet, el explorador establece una conexión con el servicio Web del servidor que utiliza el protocolo HTTP. URL (o Localizador uniforme de recursos) y URI (Identificador uniforme de recursos) son los nombres que la mayoría de las personas asocian con las direcciones Web.

El URL <http://www.cisco.com/index.html> es un ejemplo de un URL que se refiere a un recurso específico: una página Web denominada index.html en un servidor identificado como cisco.com (haga clic en las fichas de la figura para ver los pasos utilizados por HTTP).

Los exploradores Web son las aplicaciones de cliente que utilizan nuestras computadoras para conectarse con la World Wide Web y para acceder a los recursos almacenados en un servidor Web. Al igual que con la mayoría de los procesos de servidores, el servidor Web funciona como un servicio básico y genera diferentes tipos de archivos disponibles.

Para acceder al contenido, los clientes Web realizan conexiones al servidor y solicitan los recursos deseados. El servidor responde con los recursos y, una vez recibidos, el explorador interpreta los datos y los presenta al usuario.

Los exploradores pueden interpretar y presentar muchos tipos de datos, como texto sin formato o Lenguaje de marcado de hipertexto (HTML, el lenguaje que se utiliza para construir una página Web). Otros tipos de datos, sin embargo, requieren de otro servicio o programa. Generalmente se los conoce como plug-ins o complementos. Para ayudar al

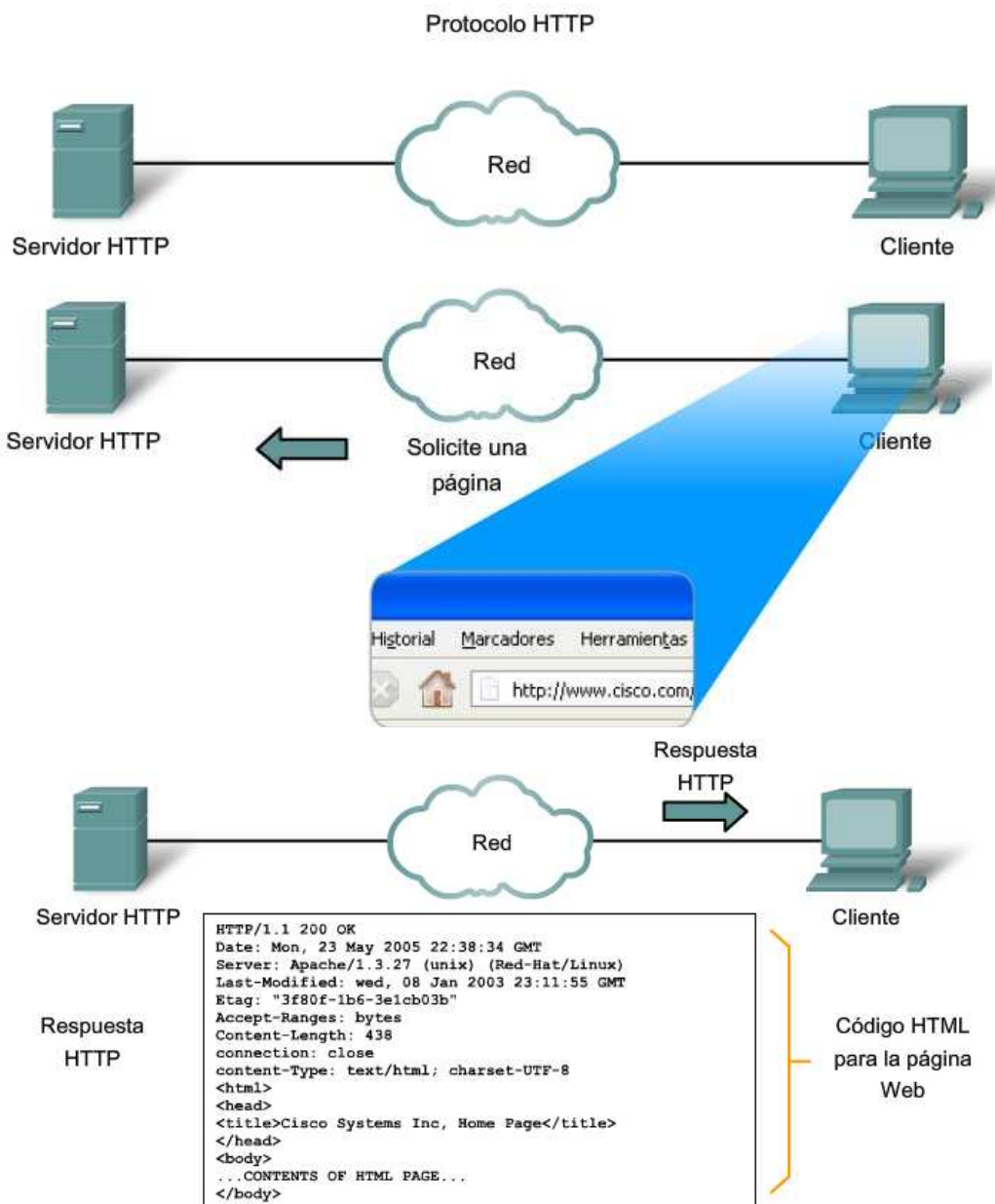
explorador a determinar qué tipo de archivo está recibiendo, el servidor especifica qué clase de datos contiene el archivo.

Para comprender mejor cómo interactúan el explorador Web con el cliente Web, podemos analizar cómo se abre una página Web en un explorador. Para este ejemplo, utilizaremos la dirección URL: <http://www.cisco.com/web-server.htm>.

Primero, el explorador interpreta las tres partes de la URL:

1. http (el protocolo o esquema),
2. www.cisco.com (el nombre del servidor), y
3. web-server.htm (el nombre específico del archivo solicitado).

El explorador luego verifica con un servidor de nombres para convertir a [www.cisco.com](http://www.cisco.com) en una dirección numérica que utilizará para conectarse con el servidor. Al utilizar los requerimientos del protocolo HTTP, el explorador envía una solicitud GET al servidor y pide el archivo web-server.htm. El servidor, a su vez, envía al explorador el código HTML de esta página Web. Finalmente, el explorador descifra el código HTML y da formato a la página para la ventana del



En respuesta a la solicitud, el servidor HTTP envía el código para una página Web.



El protocolo de transferencia de hipertexto (HTTP), uno de los protocolos del grupo TCP/IP, se desarrolló en sus comienzos para publicar y recuperar las páginas HTML, y en la actualidad se utiliza para sistemas de información distribuidos y de colaboración. HTTP se utiliza a través de la World Wide Web para transferencia de datos y es uno de los protocolos de aplicación más utilizados.

HTTP especifica un protocolo de solicitud/respuesta. Cuando un cliente, generalmente un explorador Web, envía un mensaje de solicitud a un servidor, el protocolo HTTP define los tipos de mensajes que el cliente utiliza para solicitar la página Web y envía los tipos de mensajes que el servidor utiliza para responder. Los tres tipos de mensajes más comunes son GET, POST y PUT.

GET es una solicitud de datos del cliente. Un explorador Web envía el mensaje GET para solicitar las páginas desde un servidor Web. Como se muestra en la figura, una vez que el servidor recibe la solicitud GET, responde con una línea de estado, como HTTP/1.1 200 OK, y un mensaje solo, cuyo cuerpo puede ser el archivo solicitado, un mensaje de error o alguna otra información.

POST y PUT se utilizan para enviar mensajes que cargan los datos al servidor Web. Por ejemplo, cuando el usuario ingresa datos en un formulario incorporado en una página Web, POST incluye los datos en el mensaje enviado al servidor.

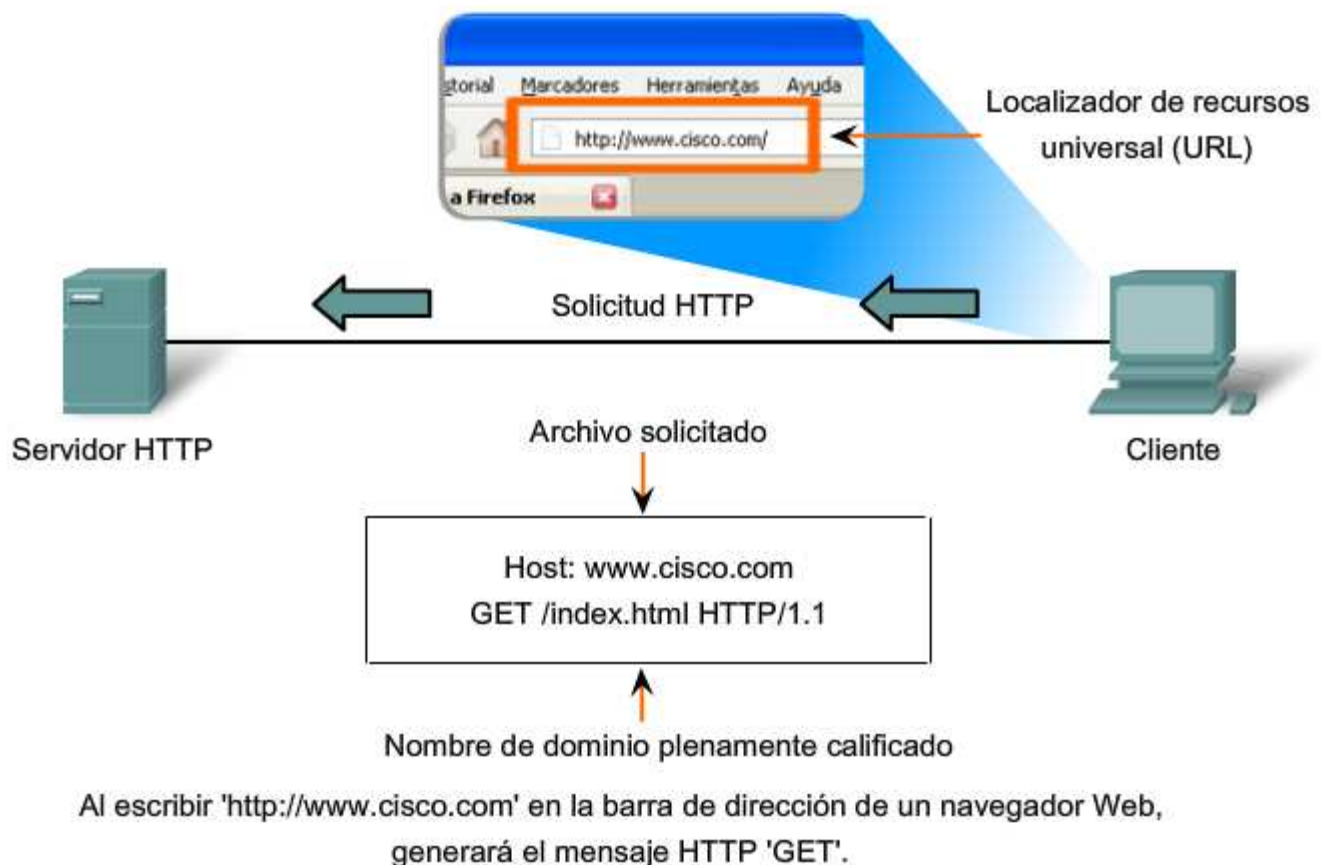
PUT carga los recursos o el contenido al servidor Web.

Aunque es muy flexible, HTTP no es un protocolo seguro. Los mensajes POST cargan información al servidor en un texto sin formato que puede ser interceptado y leído. De forma similar, las respuestas del servidor, generalmente páginas HTML, también son descifradas.

Para una comunicación segura a través de Internet, se utiliza el protocolo HTTP seguro (HTTPS) para acceder o subir información al servidor Web. HTTPS puede utilizar autenticación y encriptación para asegurar los datos cuando viajan entre el cliente y el servidor. HTTPS especifica reglas adicionales para pasar los datos entre la capa de Aplicación y la capa de Transporte.



### Protocolo HTTP mediante GET

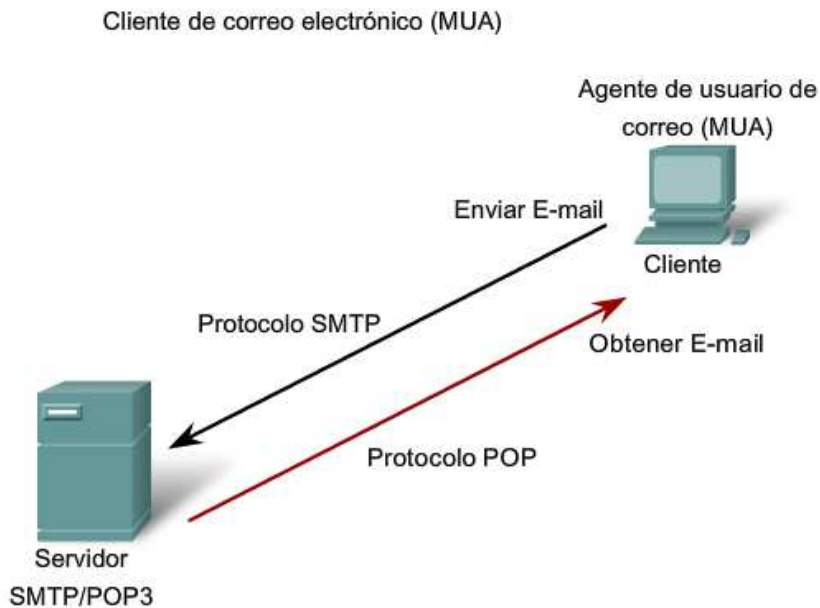


### 3.3.3 Servicios de email y protocolos SMTP/POP

E-mail, el servidor de red más conocido, ha revolucionado la manera en que nos comunicamos, por su simpleza y velocidad. Inclusive para ejecutarse en una computadora o en otro dispositivo, los e-mails requieren de diversos servicios y aplicaciones. Dos ejemplos de protocolos de capa de aplicación son Protocolo de oficina de correos (POP) y Protocolo simple de transferencia de correo (SMTP), que aparecen en la figura. Como con HTTP, estos protocolos definen procesos cliente-servidor.

Cuando una persona escribe mensajes de correo electrónico, generalmente utiliza una aplicación denominada Agente de usuario de correo (MUA) o cliente de correo electrónico. MUA permite enviar los mensajes y colocar los mensajes recibidos en el buzón del cliente; ambos procesos son diferentes.

Para recibir e-mails desde un servidor de e-mail, el cliente de correo electrónico puede utilizar un POP. Al enviar un e-mail desde un cliente o un servidor, se utilizan formatos de mensajes y cadenas de comando definidas por el protocolo SMTP. En general, un cliente de correo electrónico proporciona la funcionalidad de ambos protocolos dentro de una aplicación.

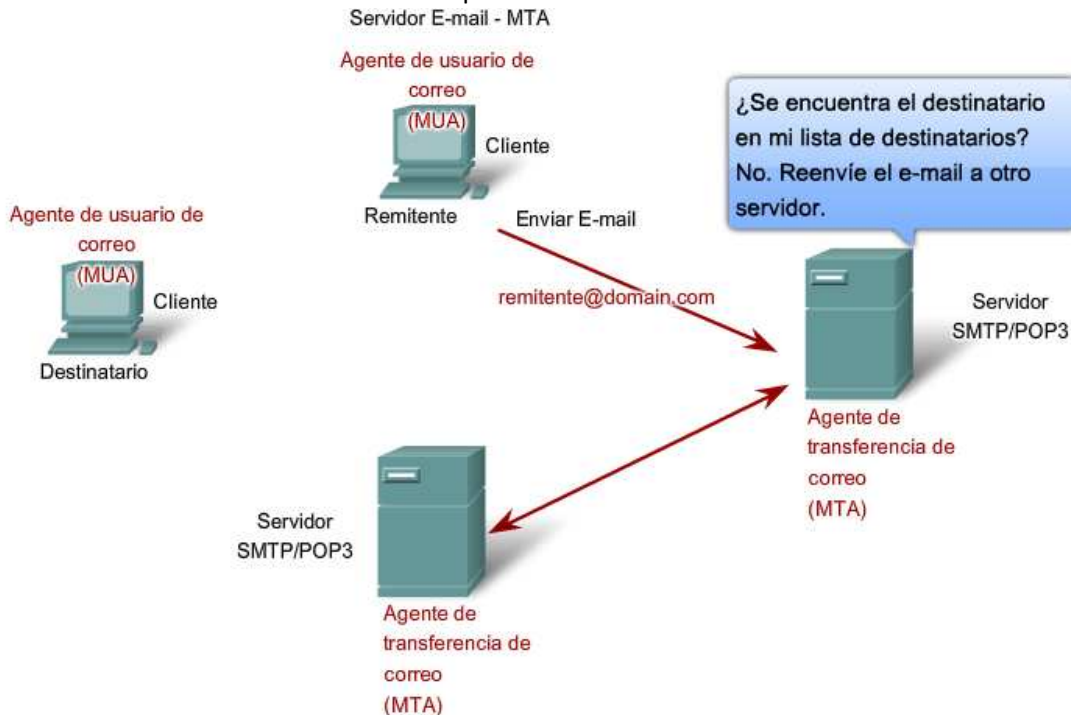


Los clientes envían e-mails a un servidor mediante SMTP y reciben e-mails mediante POP3.

Procesos del servidor de e-mail: MTA y MDA

- El servidor de e-mail ejecuta dos procesos individuales:
- Agente de transferencia de correo (MTA, Mail Transfer Agent).
- Agente de entrega de correo (MDA, Mail Delivery Agent).

El proceso Agente de transferencia de correo (MTA) se utiliza para enviar correos electrónicos. Como se muestra en la figura, el MTA recibe mensajes desde el MUA u otro MTA en otro servidor de e-mail. Según el encabezado del mensaje, determina cómo debe reenviarse un mensaje para llegar a destino. Si el correo está dirigido a un usuario cuyo buzón está en el servidor local, el correo se pasa al MDA. Si el correo es para un usuario que no está en el servidor local, el MTA enruta el e-mail al MTA en el servidor correspondiente.



El proceso de agente de transferencia de correo rige el manejo de e-mails entre servidores.

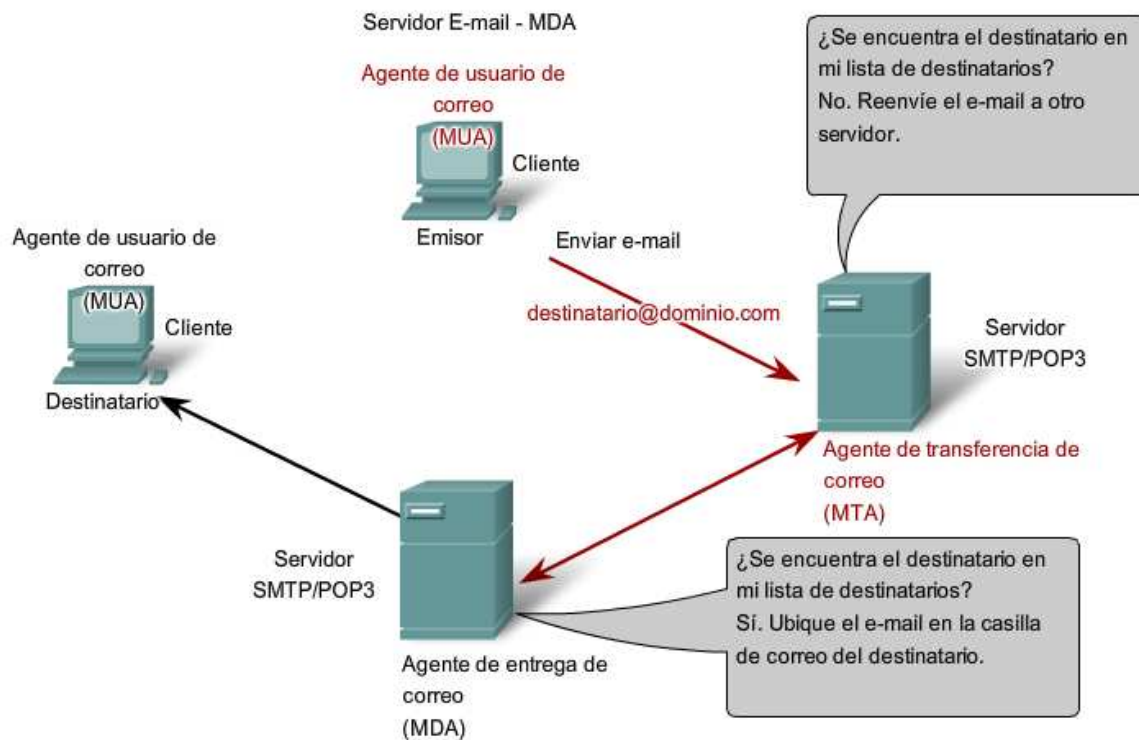


En la figura, vemos que el Agente de envío de correo (MDA) acepta una parte del e-mail desde un Agente de transferencia de correo (MTA) y realiza el envío real. El MDA recibe todo el correo entrante desde el MTA y lo coloca en los buzones de los usuarios correspondientes. El MDA también puede resolver temas de entrega final, como análisis de virus, correo no deseado filtrado y manejo de acuses de recibo. La mayoría de las comunicaciones de e-mail utilizan las aplicaciones MUA, MTA y MDA. Sin embargo, existen otras alternativas para enviar e-mails.

El cliente puede estar conectado a un sistema de e-mails corporativo, como Lotus Notes de IBM, Groupwise de Novell o Microsoft Exchange. Estos sistemas a veces tienen su propio formato interno de correo electrónico y sus clientes generalmente se comunican con el servidor de correo electrónico a través de un protocolo propietario.

El servidor envía o recibe correos electrónicos por Internet a través de la 95ersión de correo de internet del producto, que realiza el reformato que sea necesario. Si, por ejemplo, dos personas que trabajan para la misma empresa intercambian e-mails entre ellos utilizando un protocolo propietario, los mensajes pueden permanecer completamente dentro del sistema de e-mails corporativo de la empresa.

Como segunda alternativa, las computadoras que no tienen un MUA pueden conectarse a un servicio de correo en un explorador Web para así recuperar y enviar mensajes. Algunas computadoras pueden ejecutar su propio MTA y administrar e-mails de dominio interno.



El proceso de agente de entrega de correo rige la entrega de e-mails entre servidores y clientes.

Como se mencionó anteriormente, los e-mails pueden utilizar los protocolos POP y SMTP (vea la figura para saber cómo funcionan). POP y POP3 (Protocolo de oficina de correos v.3) son protocolos de envío de correo entrante y protocolos cliente/servidor típicos. Envían e-mails desde el servidor de e-mail al cliente (MUA). El MDA escucha cuando un cliente se conecta a un servidor. Una vez establecida la conexión, el servidor puede enviar el e-mail al cliente.

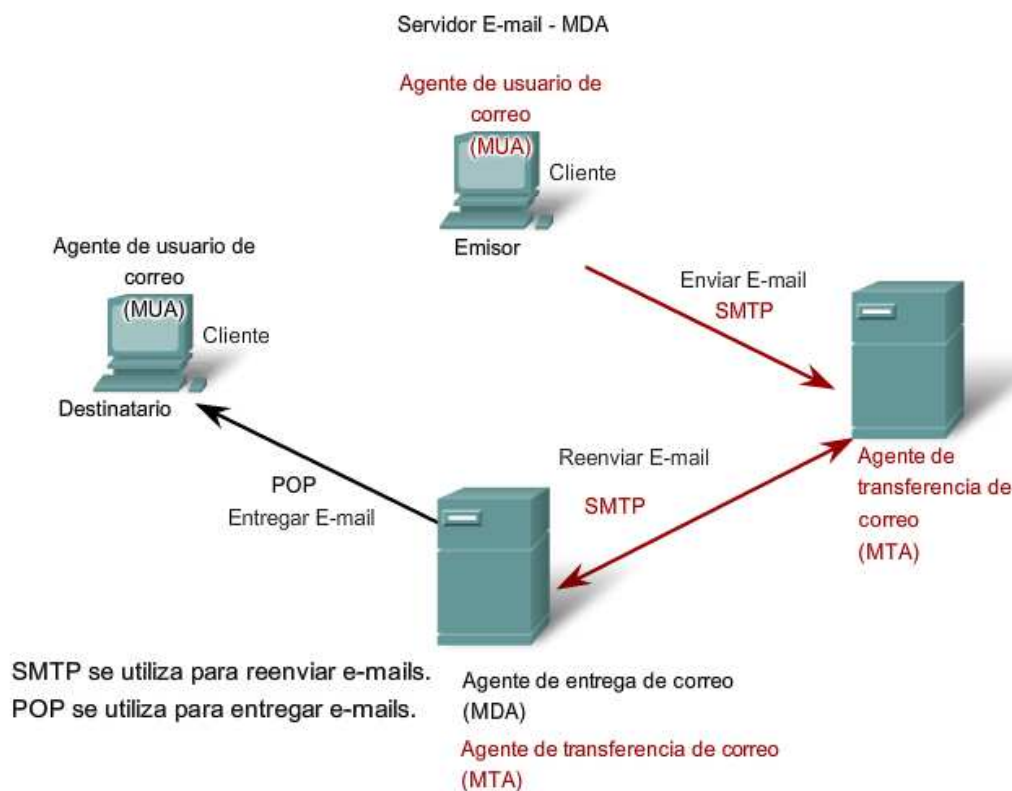
El protocolo simple de transferencia de correo (SMTP), por el contrario, rige la transferencia de e-mails salientes desde el cliente emisor al servidor de e-mail (MDA), como así también el transporte de e-mails entre servidores de e-mail

(MTA). SMTP permite transportar e-mails por las redes de datos entre diferentes tipos de software de cliente y servidor, y hace posible el intercambio de e-mails en Internet.

El formato de mensajes del protocolo SMTP utiliza un conjunto rígido de comandos y respuestas. Estos comandos admiten los procedimientos utilizados en el SMTP, como inicio de sesión, transacción de correo, reenvío de correo, verificación de nombres de buzones, expansión de listas de correo y apertura y cierre de intercambios.

Algunos de los comandos especificados en el protocolo SMTP son:

- HELO: identifica el proceso de cliente SMTP para el proceso de servidor SMTP.
- EHLO: es la versión más nueva de HELO, que incluye extensiones de servicios, y
- MAIL FROM: identifica al emisor.
- RCPT TO: identifica al receptor, y
- DATA: identifica el cuerpo del mensaje.



### 3.3.4 FTP

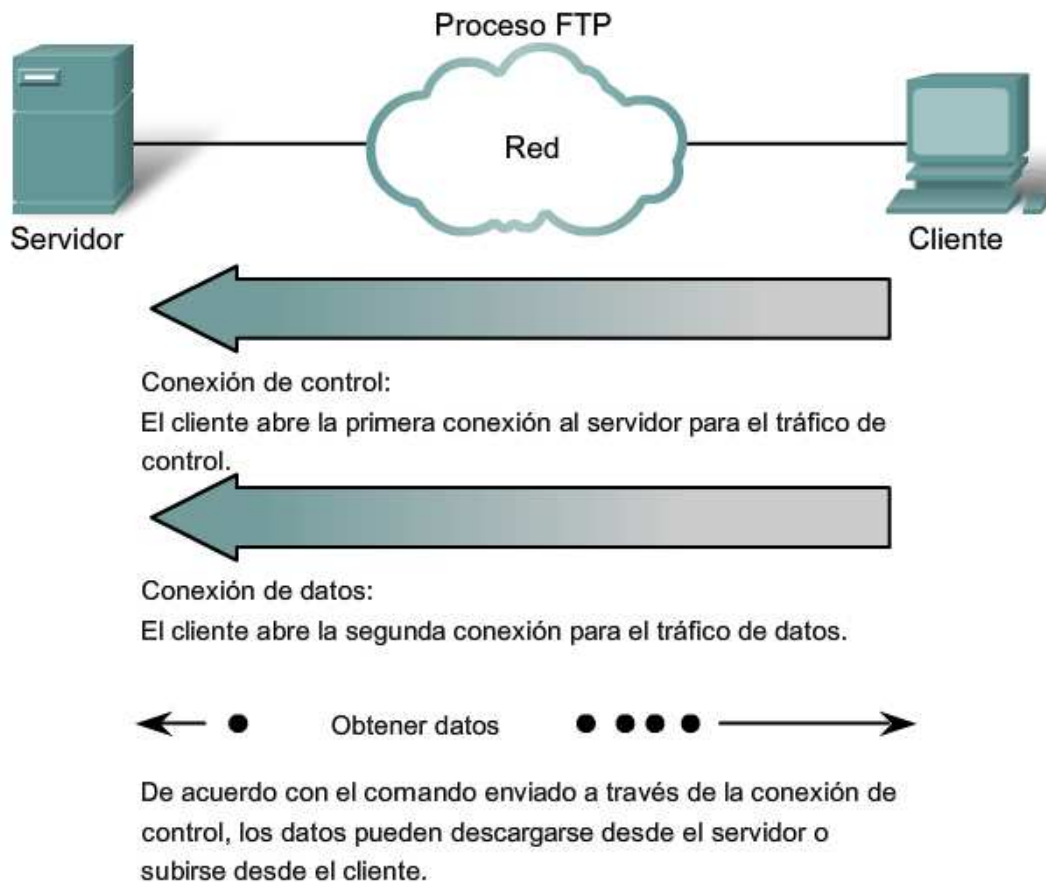
El protocolo de transferencia de archivos (FTP) es otro protocolo de la capa de aplicación comúnmente utilizado. El FTP se desarrolló para permitir las transferencias de archivos entre un cliente y un servidor. Un cliente FTP es una aplicación que se ejecuta en una computadora y se utiliza para cargar y descargar archivos desde un servidor que ejecuta el daemon FTP (FTPd).

Para transferir los archivos en forma exitosa, el FTP requiere de dos conexiones entre cliente y servidor: una para comandos y respuestas, otra para la transferencia real de archivos.

El cliente establece la primera conexión con el servidor en TCP puerto 21. Esta conexión se utiliza para controlar el tráfico, que consiste en comandos del cliente y respuestas del servidor.

El cliente establece la segunda conexión con el servidor en TCP puerto 20. Esta conexión es para la transferencia real de archivos y se crea cada vez que se transfiere un archivo.

La transferencia de archivos puede producirse en ambas direcciones. El cliente puede descargar (bajar) un archivo desde el servidor o el cliente puede cargar (subir) un archivo en el servidor.



### 3.3.5 DHCP

El servicio Protocolo de configuración dinámica de host (DHCP) permite a los dispositivos de una red obtener direcciones IP y demás información de un servidor DHCP. Este servicio automatiza la asignación de direcciones IP, máscaras de subred, gateways y otros parámetros de redes IP.

DHCP permite a un host obtener una dirección IP en forma dinámica cuando se conecta a la red. Se realiza el contacto con el servidor de DHCP y se solicita una dirección. El servidor DHCP elige una dirección de un rango configurado de direcciones denominado "pool" y se la asigna ("alquila") al host por un período establecido.

En redes locales más grandes o donde cambia frecuentemente la población usuaria, es preferible el DHCP. Los nuevos usuarios llegan con computadoras portátiles y necesitan una conexión. Otros tienen nuevas estaciones de trabajo que necesitan conexión. En lugar de tener direcciones IP asignadas por el administrador de red en cada estación de trabajo, resulta más eficiente tener direcciones IP asignadas en forma automática utilizando un DHCP.

Las direcciones de DHCP distribuidas no se asignan a los hosts en forma permanente, sólo se alquilan durante un período de tiempo. Si el host se apaga o se desconecta de la red, la dirección regresa al pool para volver a utilizarse. Esto es muy útil para los usuarios móviles que entran y salen de la red. Los usuarios pueden moverse libremente desde una ubicación a otra y volver a establecer las conexiones de red. El host puede obtener una dirección IP una vez que se realice la conexión del hardware, ya sea mediante una LAN inalámbrica o conectada por cable.

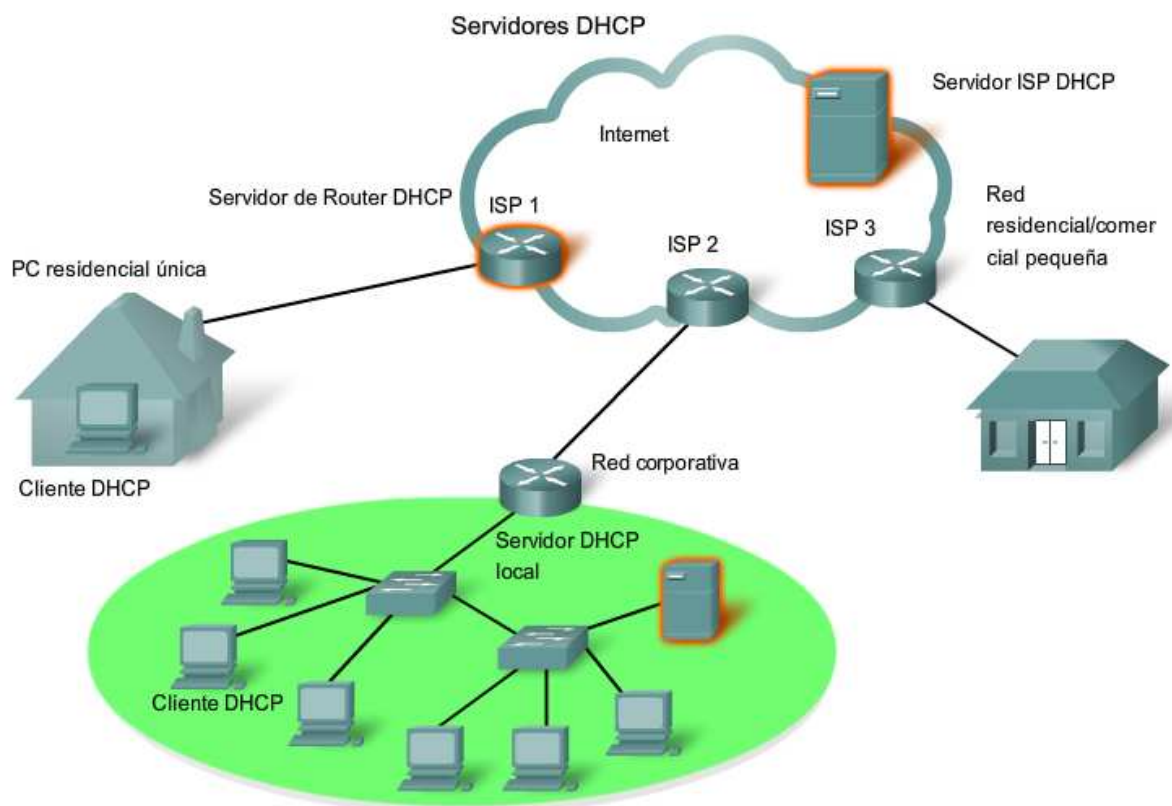
DHCP hace posible el acceso a Internet utilizando zonas activas inalámbricas en aeropuertos o cafés. Una vez que ingresa al área, el cliente de DHCP de la computadora portátil contacta al servidor de DHCP mediante una conexión inalámbrica. El servidor de DHCP asigna una dirección IP a la computadora portátil.

Como muestra la figura, diferentes tipos de dispositivos pueden ser servidores de DHCP al ejecutar el software de servicio de DHCP. El servidor de DHCP en la mayoría de las redes medianas y grandes está generalmente ubicado en un servidor dedicado local basado en PC.

Con las redes domésticas, el servidor de DHCP se ubica en el ISP y un host de la red doméstica recibe la configuración IP directamente desde el ISP.

DHCP puede representar un riesgo a la seguridad porque cualquier dispositivo conectado a la red puede recibir una dirección. Este riesgo hace de la seguridad física un factor importante a la hora de determinar si se utiliza direccionamiento manual o dinámico.

Los direccionamientos dinámico y estático tienen su lugar en los diseños de red. Muchas redes utilizan tanto el direccionamiento estático como el DHCP. DHCP se utiliza para hosts de propósitos generales, como los dispositivos de usuario final, y las direcciones fijas se utilizan para dispositivos de red como gateways, switches, servidores e impresoras.



Sin DHCP los usuarios tienen que ingresar manualmente la dirección IP, la máscara de subred y otras configuraciones para poder unirse a la red. El servidor de DHCP mantiene un pool de las direcciones IP y alquila una dirección a cualquier cliente habilitado por DHCP cuando el cliente está activado. Debido a que las direcciones IP son dinámicas (alquiladas) en lugar de estáticas (asignadas en forma permanente), las direcciones en desuso regresan automáticamente al pool para volver a asignarse. Cuando un dispositivo configurado por DHCP se inicia o conecta a la red, el cliente envía un paquete DESCUBRIMIENTO de DHCP para identificar cualquier servidor de DHCP disponible en la red. Un servidor DHCP

contesta con una oferta de DHCP, que es un mensaje de oferta de alquiler con información asignada de dirección IP, máscara de subred, servidor DNS y 99ersión por defecto, como también la duración del alquiler.

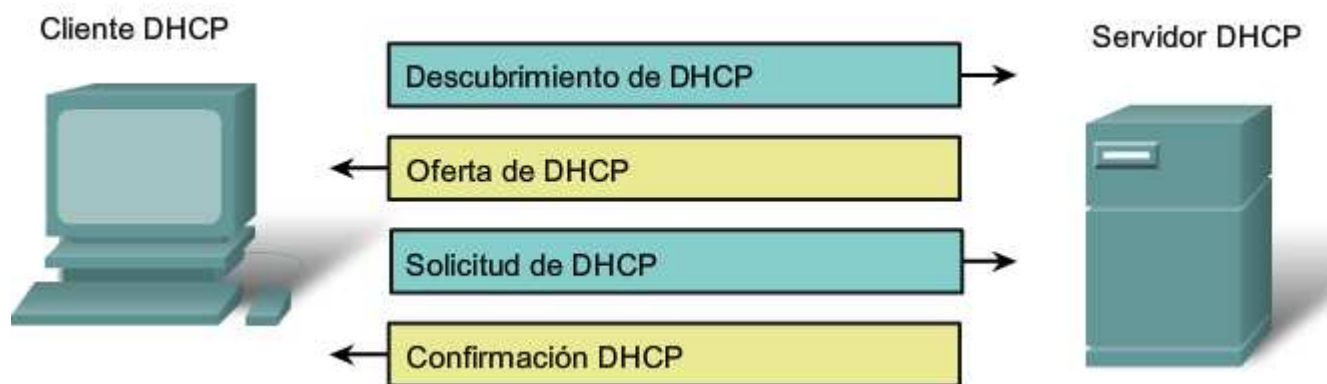
El cliente puede recibir varios paquetes de oferta de DHCP si hay más de un servidor DHCP en la red local, por lo tanto debe escoger entre ellos y enviar un broadcast de paquete con una solicitud de DHCP que identifique el servidor y la oferta de alquiler específicos que el cliente está aceptando. Un cliente puede elegir solicitar una dirección previamente asignada por el servidor.

Teniendo en cuenta que la dirección IP solicitada por el cliente u ofrecida por el servidor, aún es válida, el servidor devolverá un mensaje ACK DHCP que le informa al cliente que finalizó el alquiler. Si la oferta ya no es válida, quizás debido al tiempo o o que a otro cliente se le asign el alquiler, el servidor seleccionado responderá con un mensaje NAK DHCP (acuse de recibo negativo). Si se envía un mensaje NAK DHCP, el proceso de selección debe comenzar nuevamente con la transmisión de un nuevo mensaje DHCP DISCOVER.

Una vez que el cliente tenga el alquiler, debe renovarse antes de la expiración del alquiler por medio de otro mensaje DHCP REQUEST.

El servidor de DHCP asegura que todas las direcciones son únicas (una dirección IP no puede asignarse a dos dispositivos de red diferentes en forma simultánea). Usar DHCP permite a los administradores de red volver a configurar fácilmente las direcciones IP del cliente sin tener que realizar cambios a los clientes en forma manual. La mayoría de los proveedores de Internet utilizan DHCP para asignar las direcciones a sus clientes que no solicitan direcciones estáticas.

El cuarto curso de Exploración de CCNA cubrirá el funcionamiento de DHCP con más detalle.



### 3.3.6 Protocolo SMB y servicios para compartir archivos

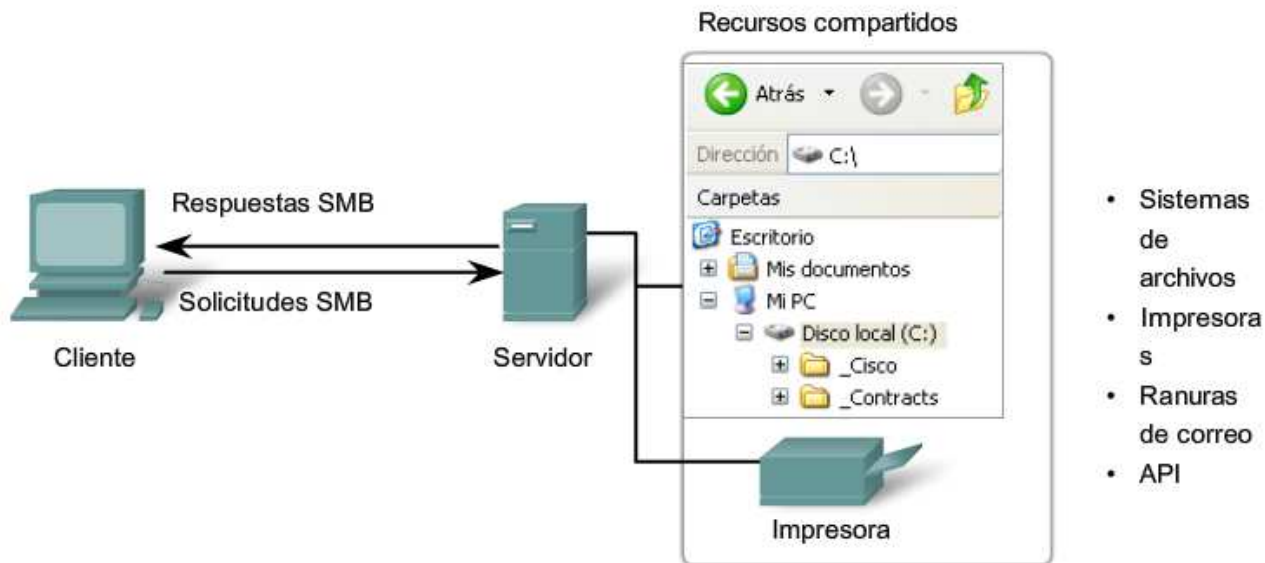
El Bloque de mensajes del servidor (SMB) es un protocolo cliente-servidor para compartir archivos. IBM desarrolló el Bloque de mensajes del servidor (SMB) a fines de la década del '80 para describir la estructura de recursos de red compartidos, como directorios, archivos, impresoras y puertos seriales. Es un protocolo de solicitud-respuesta. A diferencia del protocolo para compartir archivos respaldado por FTP, los clientes establecen una conexión a largo plazo con los servidores. Una vez establecida la conexión, el usuario del cliente puede acceder a los recursos en el servidor como si el recurso fuera local para el host del cliente.

Los servicios de impresión y el SMB para compartir archivos se han transformado en el pilar de las redes de Microsoft. Con la presentación de la serie Windows 2000 del software, Microsoft cambió la estructura subyacente para el uso del SMB. En versiones anteriores de los productos de Microsoft, los servicios de SMB utilizaron un protocolo que no es TCP/IP para implementar la resolución de nombres. Comenzando con Windows 2000, todos los productos subsiguientes

de Microsoft utilizan denominación DNS. Esto permite a los protocolos TCP/IP admitir directamente el compartir recursos SMB, como se muestra en la figura.

Los sistemas operativos LINUX y UNIX también proporcionan un método para compartir recursos con las redes Microsoft a través de una versión de SMB denominada SAMBA. Los sistemas operativos Macintosh de Apple también admiten recursos compartidos utilizando el protocolo SMB.

#### Compartir archivos mediante el protocolo SMB



**SMB es un protocolo de solicitud-respuesta y cliente-servidor. Los servidores pueden poner sus recursos a disposición de los clientes en la red.**

El protocolo SMB describe el acceso al sistema de archivos y la manera en que los clientes hacen solicitudes de archivos. Además describe la comunicación entre procesos del protocolo SMB. Todos los mensajes SMB comparten un mismo formato. Este formato utiliza un encabezado de tamaño fijo seguido por un parámetro de tamaño variable y un componente de datos.

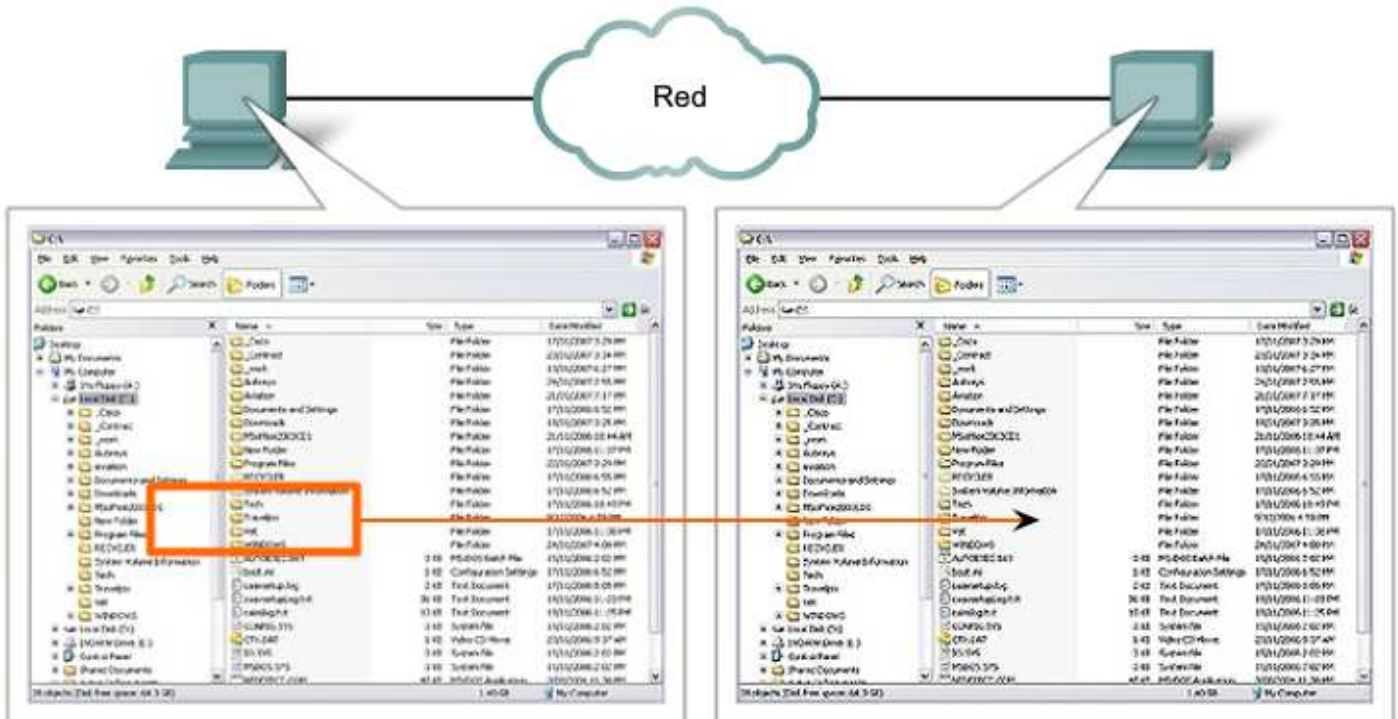
Los mensajes SMB pueden:

- Iniciar, autenticar y terminar sesiones
- Controlar el acceso a archivos e impresoras
- Permitir a una aplicación enviar o recibir mensajes hacia o desde otro dispositivo

El proceso de intercambio de archivos SMB se muestra en la figura.



## Protocolo SMB



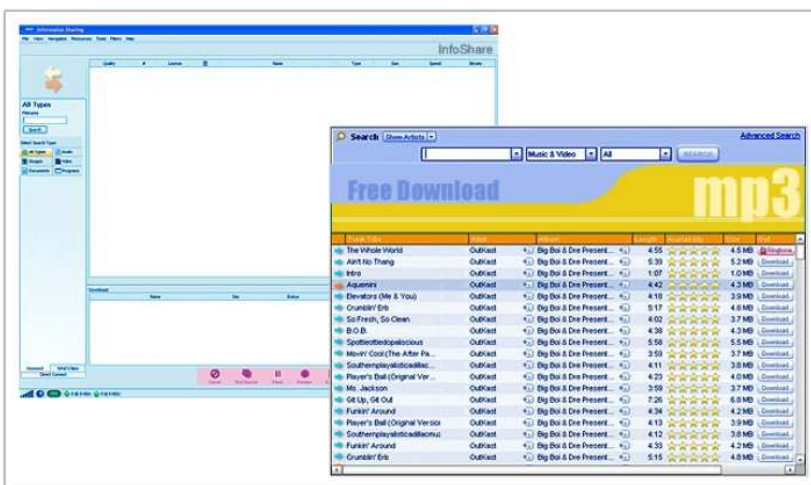
Puede copiarse un archivo desde una PC a otra con Windows Explorer mediante el protocolo SMB.

### 3.3.7 Protocolo GNUTELLA y servicios P2P

Aprendimos acerca de FTP y SMB como formas de obtener archivos; aquí presentamos otro protocolo de aplicación. Compartir archivos en Internet se ha transformado en algo muy popular. Con las aplicaciones P2P basadas en el protocolo Gnutella, las personas pueden colocar archivos en sus discos rígidos para que otros los descarguen. El software del cliente compatible con Gnutella permite a los usuarios conectarse con los servicios Gnutella en Internet, ubicarlos y acceder a los recursos compartidos por otros pares Gnutella.

Muchas aplicaciones del cliente están disponibles para acceder en la red Gnutella, entre ellas: BearShare, Gnucleus, LimeWire, Morpheus, WinMX y XoloX (consulte una captura de pantalla de LimeWire en la figura). Mientras que el Foro de desarrolladores de Gnutella mantiene el protocolo básico, los proveedores de las aplicaciones generalmente desarrollan extensiones para lograr que el protocolo funcione mejor en las aplicaciones.

Aplicaciones punto a punto

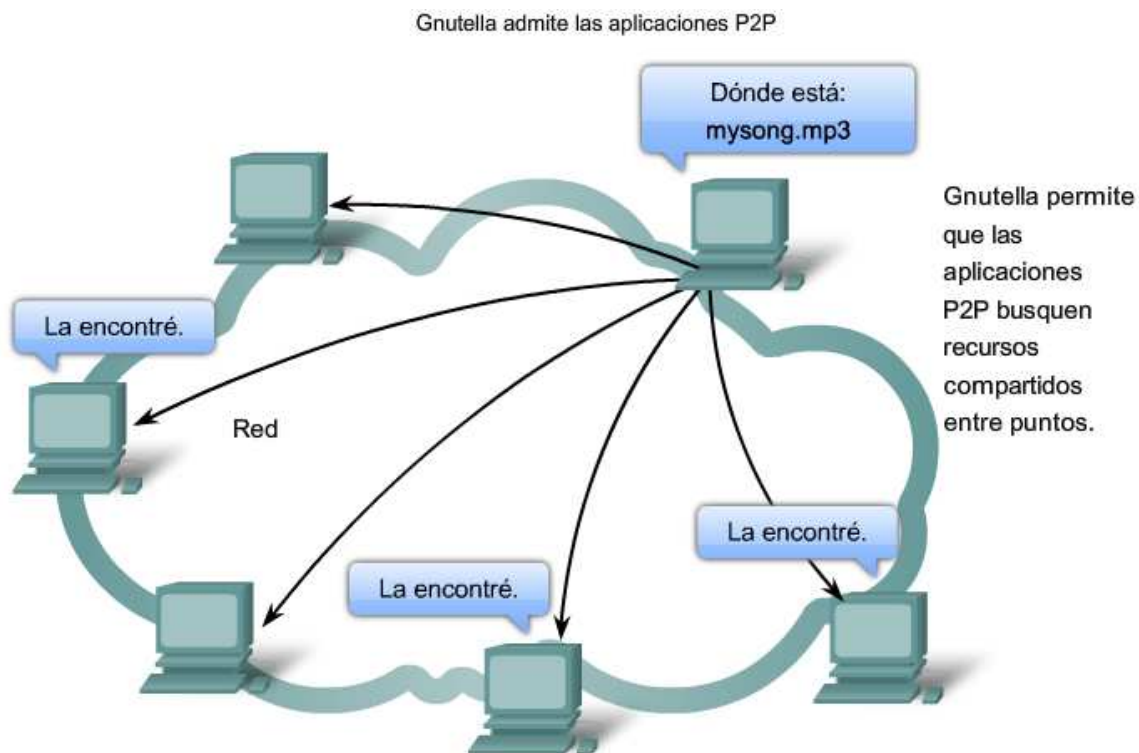


Muchas de las aplicaciones P2P no utilizan una base de datos central para registrar todos los archivos disponibles en los puntos. Por el contrario, los dispositivos en la red se indican entre ellos qué archivos están disponibles cuando hay una consulta, y utilizan el protocolo Gnutella y los servicios para respaldar los recursos ubicados. Consulte la figura.

Cuando un usuario se conecta a un servicio Gnutella, las aplicaciones del cliente buscarán otros nodos Gnutella para conectarse. Estos nodos manejan las consultas para las ubicaciones de los recursos y responden a dichas solicitudes. Además, gobiernan los mensajes de control que ayudan al servicio a descubrir otros nodos. Las verdaderas transferencias de archivos generalmente dependen de los servicios HTTP.

El protocolo Gnutella define cinco tipos de paquetes diferentes:

- ping: para descubrir un dispositivo,
- pong: como respuesta a un ping,
- consulta: para ubicar un archivo,
- query hit: como respuesta a una consulta, y
- push: como una solicitud de descarga.



### 3.3.8 Protocolo y servicios Telnet

Mucho antes de que existieran las computadoras de escritorio con interfaces gráficas sofisticadas, las personas utilizaban sistemas basados en textos que eran simplemente terminales conectadas físicamente a una computadora central. Una vez que las redes estuvieran disponibles, las personas necesitaban acceder en forma remota a los sistemas informáticos de la misma manera en que lo hacían con las terminales conectadas en forma directa.

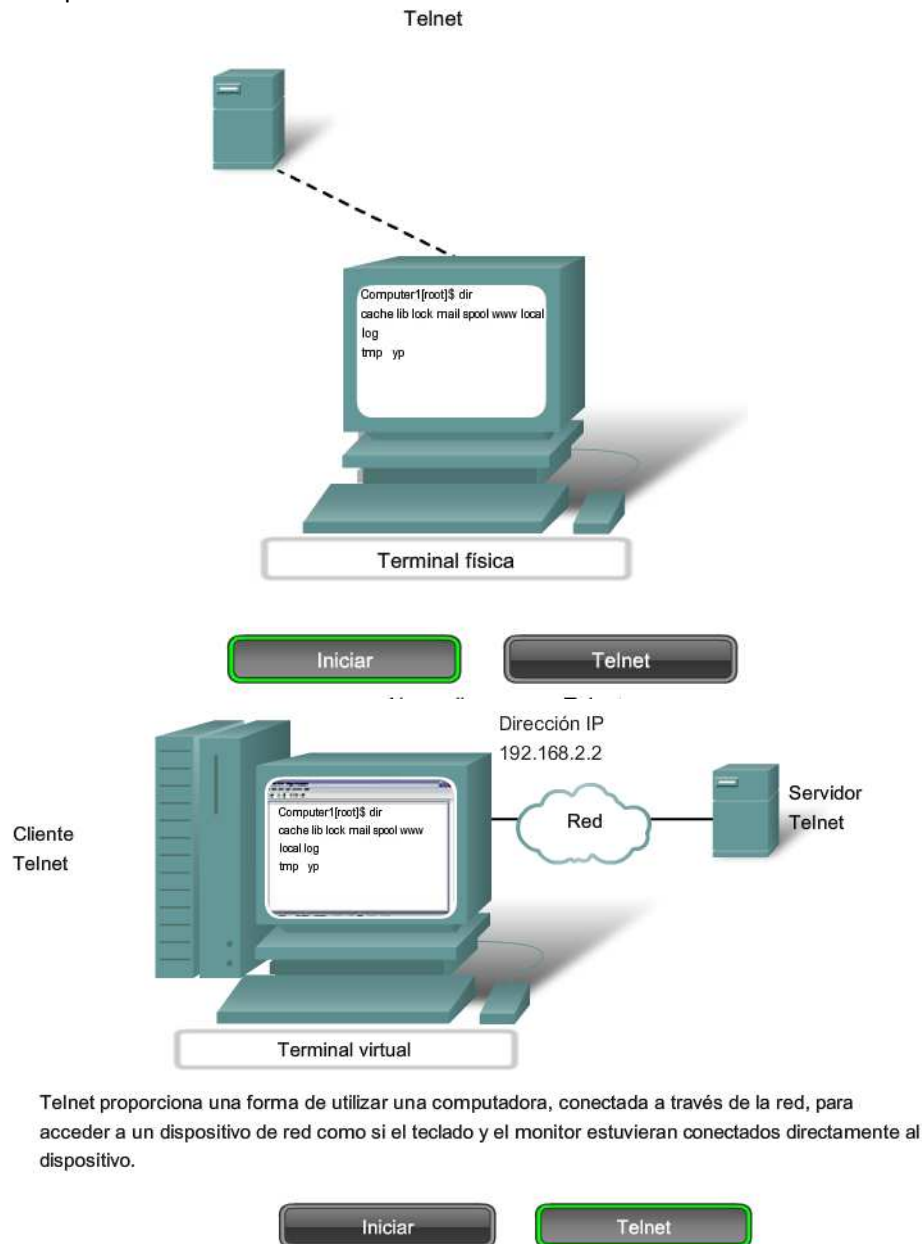
Telnet se desarrolló para satisfacer esta necesidad. Telnet se remonta a principios de la década de los setenta y se encuentra entre los servicios y protocolos de capa de aplicación más antiguo dentro del grupo TCP/IP. Telnet proporciona un método estándar de emulación de dispositivos de terminal basados en texto en la red de datos. El protocolo y el software del cliente que implementa el protocolo comúnmente se definen como Telnet.



Y como consecuencia, una conexión que utiliza Telnet se llama Sesión o conexión de terminal virtual (VTY). En lugar de utilizar un dispositivo físico para conectar al servidor, Telnet utiliza software para crear un dispositivo virtual que proporciona las mismas funciones que una sesión terminal con acceso a la Interfaz de línea de comandos (CLI) del servidor.

Para admitir conexiones al cliente Telnet, el servidor ejecuta un servicio llamado daemon de Telnet. Se establece una conexión de terminal virtual desde un dispositivo final utilizando una aplicación del cliente Telnet. La mayoría de los sistemas operativos incluye un cliente de Telnet de la capa de aplicación. En una PC de Microsoft Windows, Telnet puede ejecutarse desde la entrada del comando. Otras aplicaciones de terminal comunes que ejecutan clientes de Telnet son HyperTerminal, Minicom y TeraTerm.

Una vez establecida una conexión Telnet, los usuarios pueden realizar cualquier función autorizada en el servidor, como si utilizaran una sesión de línea de comandos en el servidor mismo. Si están autorizados, pueden iniciar y detener procesos, configurar el dispositivo o inclusive cerrar el sistema.



Telnet es un protocolo cliente-servidor y especifica cómo se establece y se termina una sesión VTY. Además proporciona la sintaxis y el orden de los comandos utilizados para iniciar la sesión Telnet, como así también los comandos de control que pueden ejecutarse durante una sesión. Cada comando Telnet consiste en por lo menos dos bytes. El primer byte es

un carácter especial denominado Interpretar como comando (IAC). Como su nombre lo indica, el IAC define el byte siguiente como un comando en lugar de un texto.

Algunos de los comandos del protocolo Telnet de muestra son:

**Are You There (AYT):** Permite al usuario solicitar que aparezca algo en la pantalla del terminal para indicar que la sesión VTY está activa.

**Erase Line (EL):** Elimina todo el texto de la línea actual.

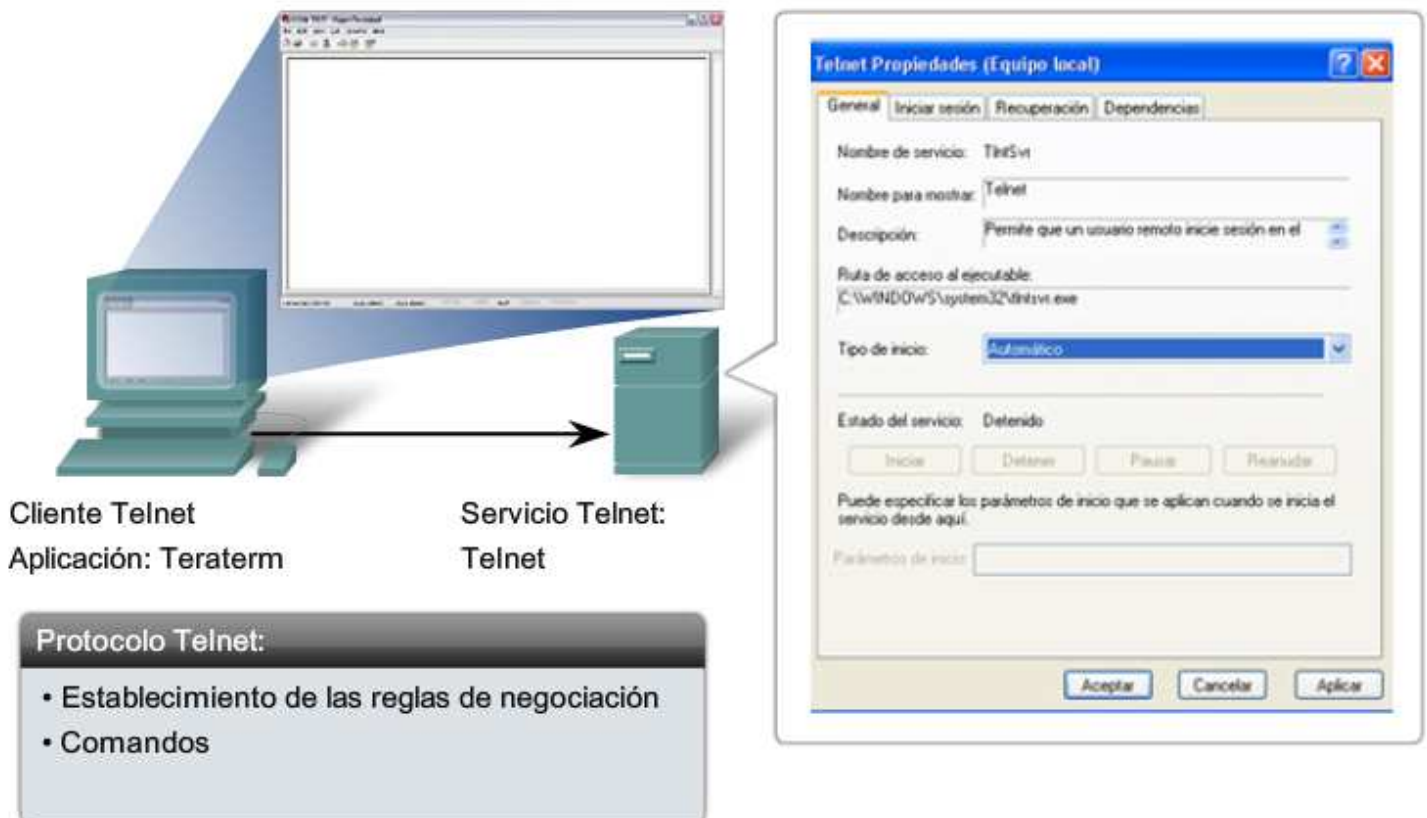
**Interrupt Process (IP):** Suspende, interrumpe, aborta o termina el proceso al cual se conectó la terminal virtual. Por ejemplo, si un usuario inició un programa en el servidor Telnet por medio de VTY, puede enviar un comando IP para detener el programa.

Aunque el protocolo Telnet admite autenticación de usuario, no admite el transporte de datos encriptados. Todos los datos intercambiados durante una sesión Telnet se transportan como texto sin formato por la red. Esto significa que los datos pueden ser interceptados y entendidos fácilmente.

Si la seguridad es un problema, el protocolo Shell seguro (SSH) ofrece un método seguro y alternativo para acceder al servidor. SSH proporciona la estructura para un inicio de sesión remoto seguro y otros servicios de red seguros. Además proporciona mayor autenticación que Telnet y admite el transporte de datos de sesión utilizando cifrado. Como una mejor práctica, los profesionales de red deberían siempre utilizar SSH en lugar de Telnet, cada vez que sea posible.

Más adelante en este curso, utilizaremos Telnet y SSH para acceder y configurar los dispositivos de red en la red de laboratorios.

### Telnet: Aplicación, servicio y protocolo



## 3.5 RESUMEN DEL CAPITULO

### 3.5.1 Resumen y revisión

La capa de Aplicación es responsable del acceso directo a los procesos subyacentes que administran y envían la comunicación a la red humana. Esta capa sirve como origen y destino de las comunicaciones en las redes de datos.

Las aplicaciones, los protocolos y servicios de la capa de Aplicación permiten a los usuarios interactuar con la red de datos de manera significativa y efectiva.

Las aplicaciones son programas informáticos con los cuales el usuario interactúa e inicia el proceso de transferencia de datos a pedido del usuario.

Los servicios son programas básicos que proporcionan la conexión entre la capa de Aplicación y las capas inferiores del modelo de networking.

Los protocolos proporcionan una estructura de reglas y procesos acordados previamente que asegura que los servicios que funcionan en un dispositivo en particular puedan enviar y recibir datos desde una variedad de dispositivos de red diferentes.

El envío de datos en la red puede ser solicitado desde un servidor por un cliente o entre dispositivos que funcionan en una conexión punto a punto, donde la relación cliente/servidor se establece según qué dispositivo es el origen y cuál el destino en ese tiempo. Los mensajes se intercambian entre los servicios de la capa de Aplicación en cada dispositivo final según las especificaciones del protocolo para establecer y utilizar estas relaciones.

Los protocolos como HTTP, por ejemplo, admiten el envío de páginas Web a dispositivos finales. Los protocolos SMTP/POP admiten el envío y la recepción de correos electrónicos. SMB permite a los usuarios compartir archivos. DNS resuelve los nombres utilizados para referirse a los recursos de red en direcciones numéricas utilizables por la red.

#### En este capítulo, aprendió a:

- Describir la manera en que las funciones de las tres capas superiores del modelo OSI proporcionan servicios de red a las aplicaciones de usuario final.
- Describir la manera en que los protocolos de la capa de Aplicación de TCP/IP proporcionan los servicios especificados por las capas superiores del modelo OSI.
- Definir la manera en que las personas utilizan la capa de la Aplicación para comunicarse a través de la red de información.
- Describir la función de las aplicaciones de TCP/IP conocidas, tales como World Wide Web e e-mail, y sus servicios relacionados (HTTP, DNS, SMB, DHCP, SMTP/POP y Telnet).
- Describir los procesos de capacidad para compartir archivos que utilizan las aplicaciones punto a punto y el protocolo Gnutella.
- Explicar la manera en que los protocolos garantizan que los servicios que se ejecutan en un tipo de dispositivo puedan enviar y recibir datos desde varios dispositivos de red diferentes.
- Usar herramientas de análisis de red para examinar y explicar la forma en que funcionan las aplicaciones de usuario.

# Plataformas en la Nube

Al momento de elegir qué plataforma usar para desarrollar nuestros proyectos digitales es necesario estar bien educado sobre todos los esfuerzos necesarios que representan desplegar proyectos que requieren poder de cómputo.

Normalmente nos encontramos con diferentes preguntas como: ¿es mejor una infraestructura física o en la nube? ¿Cual es la mejor opción en la nube? ¿Cual opción es más barata? ¿que beneficios tienen cada una?.

Comencemos por lo tradicional, en la mayoría de empresas prefieren los centros de datos físicos, los cuales requieren una inversión significativa en equipos de hardware, software, instalaciones y un ejército de personal capacitado para la instalación y mantenimiento de todo el Data Center.

Esto representa un impedimento para los proyectos que no cuentan con grandes presupuestos, sin embargo, existen grandes compañías que ya están apostando por las tecnologías de virtualización para el procesamiento de datos a gran escala.

El auge de las tecnologías cloud ha hecho accesible disponer de servicios en poder de cómputo de clase mundial, desde multinacionales hasta dos estudiantes, que desde sus casas desarrollan una nueva app.

En la actualidad existen 3 empresas que reinan en el mundo del Cloud Computing como son: Microsoft Azure, Google Cloud Platform y Amazon Web Service, cada uno con una amplia gama de productos que cumplen diversas funciones.

En esta oportunidad abordaremos los tres al mismo tiempo para explorar cual elegir al momento de desarrollar proyectos propios o cuando vendemos servicios.

## Productos y Soluciones

Vamos a utilizar los términos “productos” y “servicios” indistintamente; una solución, sin embargo, es un concepto más específico que se escucha mucho cuando se trata de servicios en la nube.

En pocas palabras, una solución es un conjunto de productos preconfigurados orientadas a una necesidad muy específica. Con abundante documentación, casos de uso y los testimonios que te guiarán a través del proceso de adopción de la infraestructura en la nube. Amazon Aws vs Microsoft Azure vs Google Cloud.

Uno de los grandes beneficios de los proveedores de servicios Cloud como Amazon Web services, GCP y otros, son sus estrategias de precios competitivos y flexibles.

Estas empresas están en una constante lucha para brindar la mejor oferta al consumidor, en su mayoría apuestan al método de Pago por Uso (PAYG por sus siglas en inglés).

Anteriormente se pagaba por uso mensual, sin embargo las cosas están cambiando y girando hacia la tendencia de cobrar por segundo de potencia en cómputo.

## Precios en AWS.

La novedosa tendencia del PAYG hace que los precios se ajusten a las necesidades del proyecto, no es necesario comprar costosas soluciones de nube privada que se vuelven obsoleta en poco tiempo. Con Amazon Web Services solo necesita ajustar su máquina virtual y la nube de AWS se adaptará según los requerimientos, para solo pagar por el poder de cómputo usado.

Desafortunadamente cuando se trata de almacenar grandes volúmenes de datos (entre 50 TB y 500 TB) AWS pierde rendimiento porcentual.

Amazon puede ser ideal para colocar grandes bases de datos en la nube pero cuando se trata de subir aplicaciones la nube de Microsoft Azure está mejor equipada.

Estos detalles de precios de AWS son útiles porque, como Azure y Google, los niveles de precios de la nube varían mucho y es bueno compararlos con los requisitos que necesite nuestro proyecto.

También tenemos el Costo Total de la Propiedad (TCO por sus siglas en inglés) que es importante cuando se construye un caso de negocios y se obtiene una mejor estimación de lo que se necesita para satisfacer las necesidades de la organización.

## **Precios de Microsoft Azure**

Al igual que AWS, esta herramienta de Microsoft tiene muchos elementos convincentes, como el desglose de precios dependiendo de las necesidades del proyecto, sin embargo sus costos son más agresivos que GCP y AWS.

Esta forma agresiva de marcar costos es con la finalidad de liderar los monopolios Cloud, como es el segmento de mover aplicaciones a la nube donde compete de manera agresiva por esta cuota de mercado.

Para Azure, la calculadora de costo total de la propiedad hace las siguientes preguntas:

¿Desea reducir el costo total de propiedad de su infraestructura local?

¿Cuáles son los ahorros de costos estimados de migrar cargas de trabajo de aplicaciones a Microsoft Azure?

## **Precios de Google Cloud Platform**

Por último la división de negocios del gigante de los buscadores, la cual busca posicionarse cara a cara con sus competidores más cercanos, al tiempo que enfatiza su facturación solo por el costo exacto de uso en potencia de cómputo.

Más allá de comparar el precio de estos tres pesos pesados de la nube, sus características también son un factor muy interesante a la hora de comparar.

En general, estas comparaciones son muy útiles cuando se considera que partner de la nube es el más adecuado para el resultado deseado.

Por ejemplo, aunque todos pueden cubrir análisis de datos y visualización, se puede pensar que AWS es el más progresivo en esta área.

Los tres AWS, Azure y Google tienen su propia forma de categorizar los diferentes elementos, por lo que sugerimos comenzar a evaluar según las necesidades del proyecto y como cada herramienta se ajusta según sus características.

Desafortunadamente, a menudo se ven organizaciones que están tan comprometidas con Azure, por ejemplo, que no reconocen alternativas posiblemente más económicas y eficientes como AWS.

### **AWS: características**

Al igual que los otros dos proveedores de servicios en la nube, AWS tiene diferentes algoritmos con nombres para desglosar sus productos y dividirlos en las siguientes categorías:

- Compute
- Storage
- Database
- Migration
- Networking & Content Delivery
- Developer Tools
- Management Tools
- Security, Identity & Compliance
- Analytics
- Artificial Intelligence
- Mobile Services
- Applications Services
- Messaging
- Business Productivity
- Desktop & App Streaming
- Software
- Internet of Things
- Game Development

Además de esta amplia gama de opciones, en AWS tienen productos específicos con un alto grado de categorización.

Estas soluciones cubren:

- Sitios web
- Copia de seguridad y recuperación
- Archivo
- Recuperación de desastres
- DevOps
- Big Data

### **Microsoft Azure: características**

En la nube de Microsoft tenemos una amplia gama de opciones, muy parecidas a las de AWS con el diferencial de proporcionar ciertas capacidades basadas en usuarios.

Estos beneficios también incluyen la facturación flexible y precios competitivos, cabe destacar que Azure afirma tener un grado de certificación en estándares internacionales mayor a la de sus competidores.

Asegurar esta superioridad es una jugada bastante audaz del gigante de la computación, a pesar de tener un rango de características muy similar a AWS, sin embargo esto lo hacen con la finalidad de buscar que sus clientes depositen la confianza en su nube.

### **Google Cloud Platform: características**

Aunque no es necesariamente el proveedor de computación en la nube más histórico, está lanzándose al mundo del Cloud Computing.

Google tiene tres puntos clave detrás de sus soluciones, destacando:

Infraestructura a prueba de futuro  
Datos y análisis serios y potentes  
Sin servidor, solo código

Con todas estas características de cada una de las herramientas, no es fácil saber cual es la más conveniente, sin embargo con este análisis podrás tener una visión más amplia y ajustar todo según tus requerimientos y recursos.

### **Aws Vs Azure Vs Google Cloud Platform: Implementación**

En lo que respecta al despliegue de proyectos en estas nubes privadas, todas poseen opciones de implementación relativamente sencillas, cabe destacar que esto también depende de la complejidad de lo que necesitemos hacer.

#### **Implementación de AWS <https://aws.amazon.com/>**

La nube de Amazon proporciona una página de implementación muy fácil e intuitiva de usar para comenzar a usar sus herramientas y soluciones.

Proporcionan un IDE donde puedes elegir qué plataforma usar, por lo tanto si estás escribiendo código PHP o implementando una App en iOS, te proporcionan códigos de inicio para hacerlo de forma más sencilla.

#### **Implementación de Microsoft Azure <https://azure.microsoft.com/>**

Microsoft proporciona beneficios en la implementación con guías muy detalladas en profundidad bien sea para los departamentos más relevantes.

- Azure Developer Guide
- Azure Operations Guide

Con estas dos guías y otros recursos como videos instructivos es posible hacer la implementación según nuestros intereses.

#### **Implementación de Google Cloud Platform <https://cloud.google.com/>**

Google siempre se ha caracterizado por tener productos enfocados en el usuario, en este caso llaman “Cloud Launcher” a su proceso de implementación, el cual proporciona documentos de inicio en su nube y diversos beneficios de sus servicios.

Si quieres documentarte mejor con GCP puedes visitar el sitio de <https://cloud.google.com/why-google-cloud/> aquí.

### **¿Cual elegir?**

Ya vimos en detalle los precios, características e implementación, todo con la finalidad de informarte cuando necesites tomar una decisión tan importante.

Si bien no es nada fácil llegar a una decisión concreta por el número de variables involucradas, al menos tendrás el conocimiento necesario para discernir por la opción correcta de soluciones cloud para tu empresa.