

6

APLICACIONES DE RED



Objetivo

● Ofrecer una visión panorámica de las aplicaciones de red, su implementación y el transporte de sus datos a través de Internet.

Manual de clases

Tema 6 de:
PROTOCOLOS DE INTERNET

Edison Coimbra G.

Última modificación:
16 de agosto de 2022

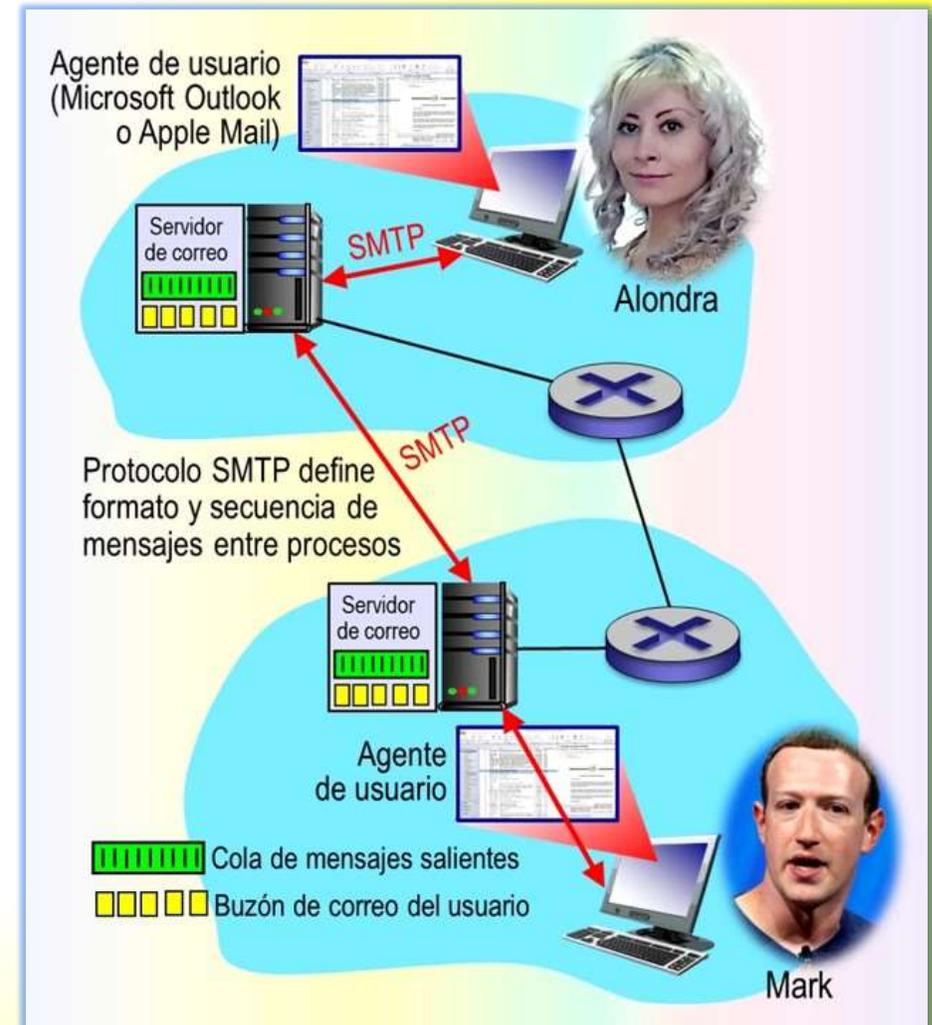
1. LAS APLICACIONES DE RED

APLICACIONES DE RED

Aplicaciones de red en el Siglo XX

(Kurose, 2017)

- **Las aplicaciones de red** son la razón de ser de una red de computadoras; si no se pudiera concebir ninguna aplicación útil, no existiría la necesidad de disponer de una infraestructura de red y de unos protocolos de soporte.
- **Desde la aparición de Internet** se han creado muchas aplicaciones de red útiles y entretenidas, que han sido la fuerza motriz del éxito de Internet, motivando a las personas en sus domicilios, escuelas, Gobierno y empresas a hacer de Internet parte integrante de sus actividades cotidianas.
- **Entre las aplicaciones Internet** se incluyen las clásicas aplicaciones basadas en texto que se hicieron populares en las décadas 1970 y 1980 como son el **correo electrónico** de texto, el **acceso remoto** a computadoras, la **transferencia de archivos** y los **grupos de noticias**.
- **También se incluye** la aplicación por excelencia de mediados de la década de 1990, la **World Wide Web**, que incluye la navegación web, las búsquedas y el comercio electrónico. También se incluye la **mensajería instantánea** y la **compartición de archivos P2P**, las dos aplicaciones estrella introducidas a finales del pasado milenio.



Las aplicaciones de red

APLICACIONES DE RED

Un poco de historia

1971: Se envía el 1º correo electrónico



BBN TECHNOLOGIES

Ray indica que en los primeros correos de prueba, en la red ARPANET, escribió "cualquier cosa". Algo como:

QWERTYUIOP
o
Testing 12345

Ray Tomlinson (1941-2016)

En BBN los trabajadores ya tenían un programa llamado "SNDMSG". Pero, solo permitía intercambiar mensajes dentro de una misma computadora.

Ray estaba trabajando en un protocolo para el intercambio de archivos: CPYNET. Realizó algunas modificaciones al código (Assembler) para poder transferir el correo electrónico.

Ray creó el @ para separar el usuario del servidor. Su primer correo: tomlinson@bbn-tenexa. Roy uso SNDMSG para redactar y recibir el correo, y usó CPYNET para transferirlo.

En 1972 Larry Roberts escribió el programa "RD": Para enviar y recibir correos.

Para 1977 se estandarizó una forma primitiva de correo electrónico, mediante el RFC 733: "Standard for the format of ARPA Network Text Messages".

SMTP (Simple Message Transfer Protocol) llegaría recién en 1981.

Elaborado con información de: "Stanford: The Origins of E-mail", "Raymond Tomlinson: Email Pioneer"-IEEE y "Microsoft Exchange Server 2003".

Hace 50 años

Tim Berners-Lee



En marzo de 1989, mientras trabajaba en el CERN (Organización Europea para la Investigación Nuclear) Tim propuso un sistema de "Gestión de Información".

En 1990 escribió, el primer navegador web, llamado: WorldWideWeb

En agosto de 1991, el navegador se puso a disposición del público en general.

Creó el lenguaje **HTML**: HyperText Markup Language. El cual ahora se encuentra próximo a su versión 6.0

Asimismo creó el protocolo de aplicación **HTTP** (Hypertext Transfer Protocol). Recién en 1996, HTTP sería acogido como un RFC.

Creó los **URL** (Uniform Resource Locators) conocidos como páginas web.

TIM no patentó al WWW, lo cual permitió el crecimiento exponencial de Internet.

Elaborado con información de: "Weaving the web", Tim Berners, 2000.

...el padre del WWW

Así lucía la primera página web

Y aun puedes visitarla: <http://info.cern.ch/hypertext/WWW/TheProject.html>



The World Wide Web project

WORLD WIDE WEB

The WorldWideWeb (WWW) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about WWW is linked directly or indirectly to this document, including an executive summary of the project, mailing lists, Policy, November's WWW news, Frequently Asked Questions.

What's out there? Pointers to the world's online information, subjects, WWW servers, etc.

What's on the browser you are using.

Software: A list of WWW project components and their current state. (e.g. Line Model, X11 Visual, NetStep, Servers, Tools, Mail robot, Library.)

Technical: Details of protocols, formats, program internals, etc.

(ref.number), Back, RETURN for more, or Help: |

Elaborado con información de: <https://line.made.cern.ch/www/hypertext/WWW/TheProject.html>

No tenía colores, gráficos ni video...

Las aplicaciones de red

APLICACIONES DE RED

Las aplicaciones de red en el Siglo XXI

(Kurose, 2017)

- **En este siglo**, continúan apareciendo aplicaciones nuevas y atractivas, como **VoIP** y sistemas de **videoconferencias** como Skype, WhatsApp, Zoom, FaceTime y Google Hangouts; **sistemas de video** generados por el usuario, como YouTube, y de **películas a la carta**, como Netflix, y juegos en línea multijugador como Second Life y World of Warcraft.
- **En este siglo** también se ha producido el nacimiento de una nueva generación de aplicaciones de **redes sociales**, como Facebook, Instagram, Twitter, WhatsApp y WeChat, que han creado adictivas redes de personas por encima de la red de routers y enlaces de comunicaciones de Internet.
- **Junto con la llegada** del teléfono inteligente, ha surgido una profusión de aplicaciones móviles basadas en la ubicación, incluyendo las populares aplicaciones de reservas, de citas y de **previsiones de tráfico**, como Yelp, Tinnder, Waz y Yik Yak, soluciones y diseños con inteligencia artificial.



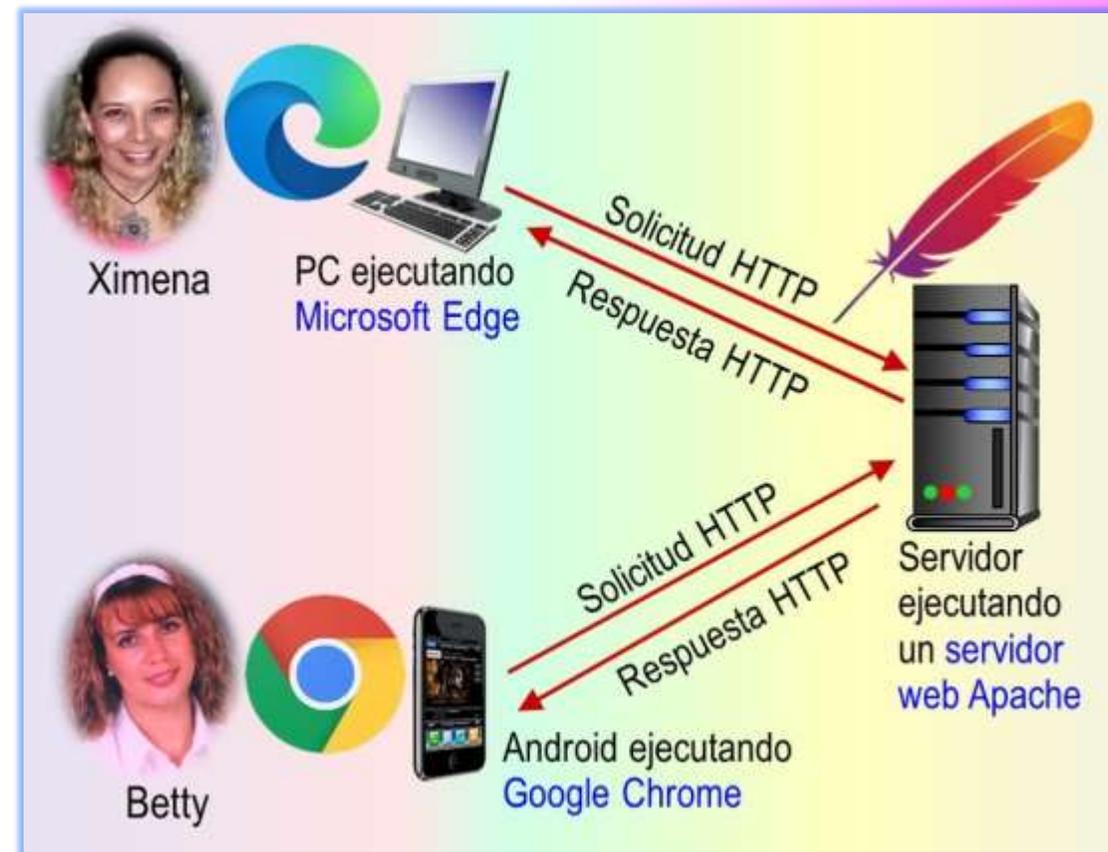
2. DESARROLLO DE APLICACIONES DE RED

APLICACIONES DE RED

¿Qué implica el desarrollo de una aplicación de red?

(Kurose, 2017)

- **Básicamente**, el desarrollo de una aplicación de red implica escribir programas que se ejecuten en distintos sistemas terminales y que se comuniquen entre sí a través de la red.
- **En la aplicación Web**, por ejemplo, se emplean dos programas diferentes que se comunican entre sí:
 - ▶ **El navegador** que se ejecuta en el host del usuario (una PC de escritorio, una portátil, una tablet, un smartphone, etc.) y,
 - ▶ **El programa del servidor web** que se ejecuta en el host que actúa como servidor web.
- **Otro ejemplo** sería el caso de un sistema de compartición de archivos P2P, en el que se emplea un programa en cada host que participa en la comunidad de compartición de archivos. En este caso, los programas instalados en los distintos hosts pueden ser similares o idénticos.
- **Por tanto**, al desarrollar una nueva aplicación, se tendrá que escribir software que se ejecutará en múltiples sistemas terminales. Este software podría escribirse, por ejemplo, en C, Java o Python.

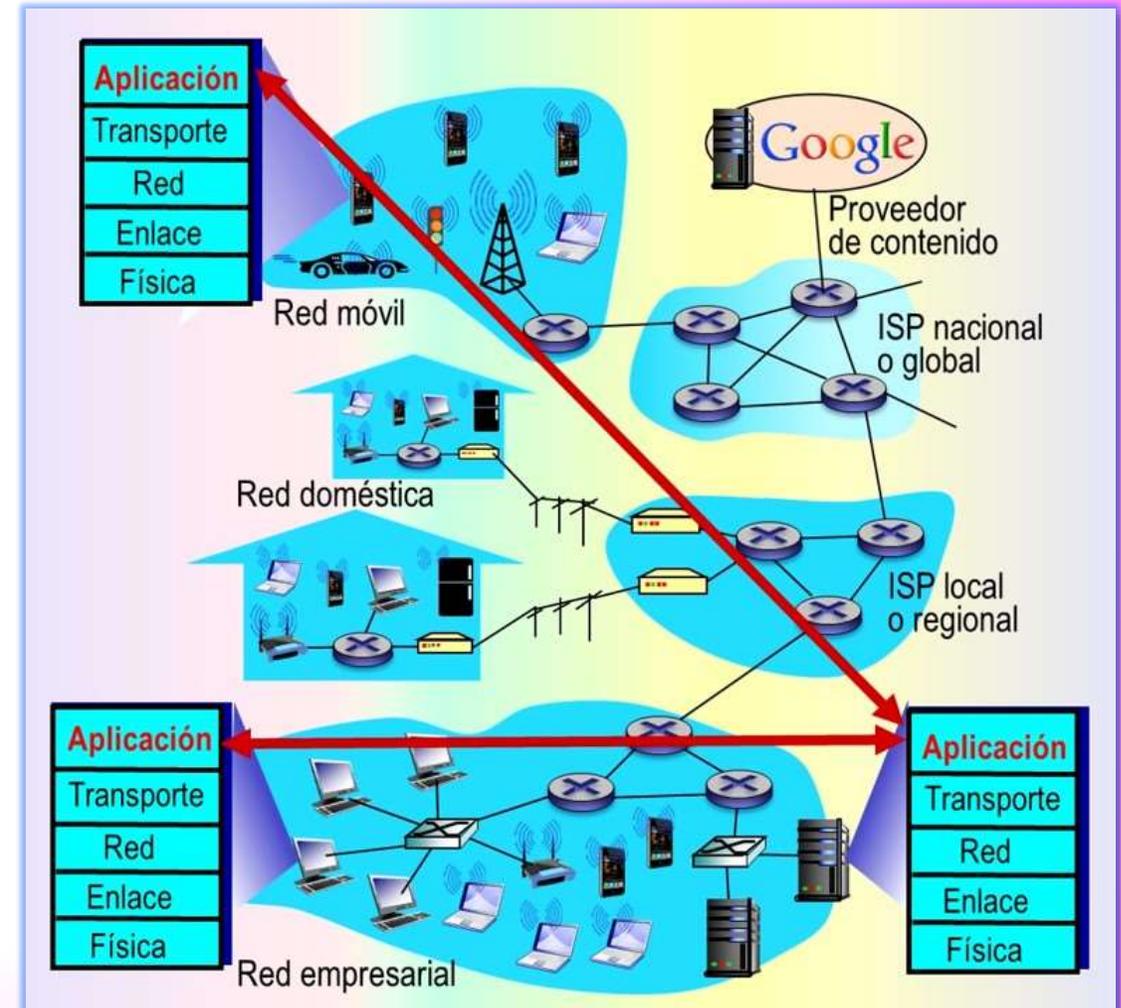


Desarrollo de aplicaciones de red

APLICACIONES DE RED

Software para las aplicaciones de red

- **Una cuestión** importante en el desarrollo de aplicaciones es que no es necesario escribir software que se ejecute en los dispositivos del núcleo de la red, como los routers o los switches, porque ellos no operan en la capa de aplicación.
- **El diseño básico** de las aplicaciones, que confina el software de aplicación a los sistemas terminales, ha facilitado el rápido desarrollo e implementación de una gran variedad de aplicaciones de red.



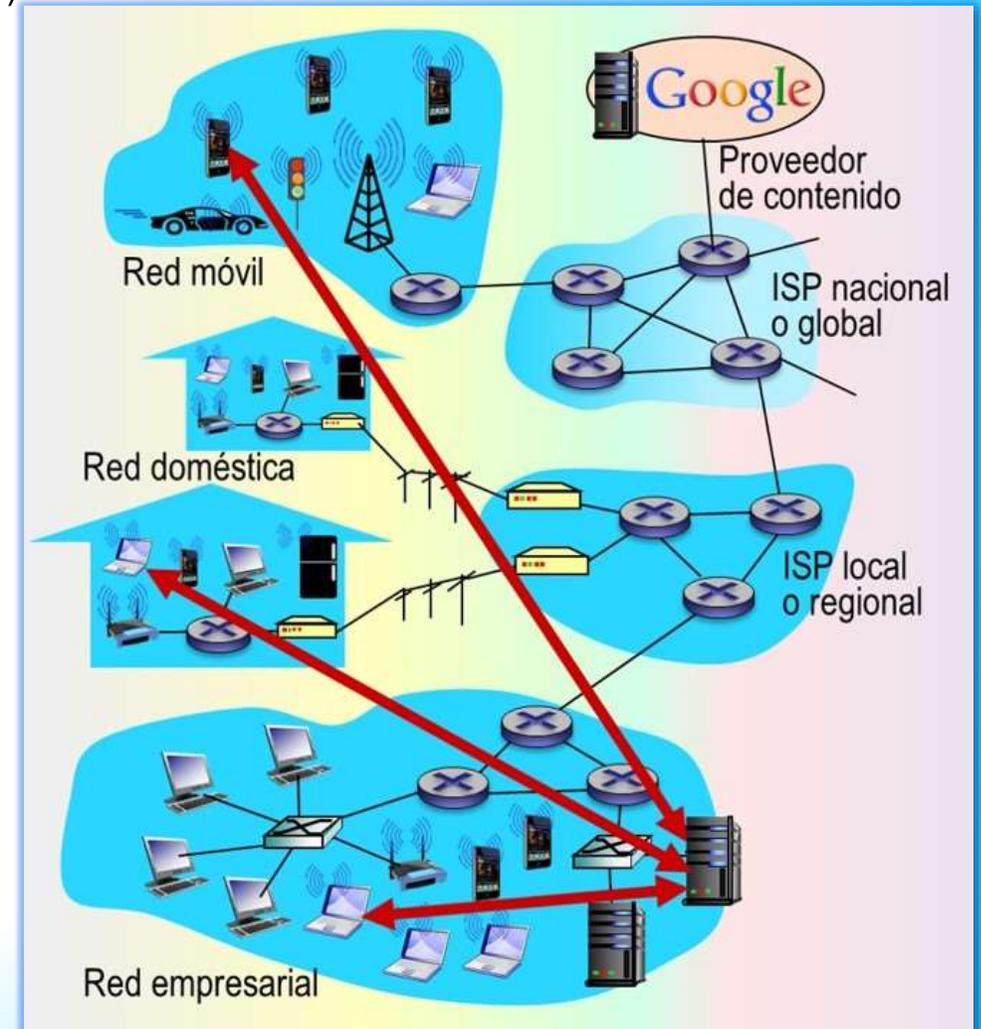
3. ARQUITECTURA DE LAS APLICACIONES DE RED

APLICACIONES DE RED

Tipos de arquitecturas de aplicaciones de red

(Kurose, 2017)

- **La arquitectura de una aplicación** es muy distinta de la arquitectura de la red (que tiene cinco capas).
 - ▶ Desde la perspectiva del desarrollador de aplicaciones, la arquitectura de la red es fija y proporciona un conjunto específico de servicios a las aplicaciones.
 - ▶ Por otro lado, el desarrollador diseña la **arquitectura de la aplicación**, es decir, establece cómo debe estructurarse la aplicación en los distintos sistemas terminales, y utilizará uno de los dos paradigmas arquitectónicos predominantes en las aplicaciones de red modernas: la arquitectura **cliente-servidor** o la arquitectura **P2P**.
- ▶ **1. Arquitectura cliente servidor.** En esta arquitectura, el servidor siempre está encendido y tiene una IP fija. Los clientes se comunican con el servidor de manera intermitente, con direcciones IP dinámicas o fijas, pero no se comunican entre ellos.



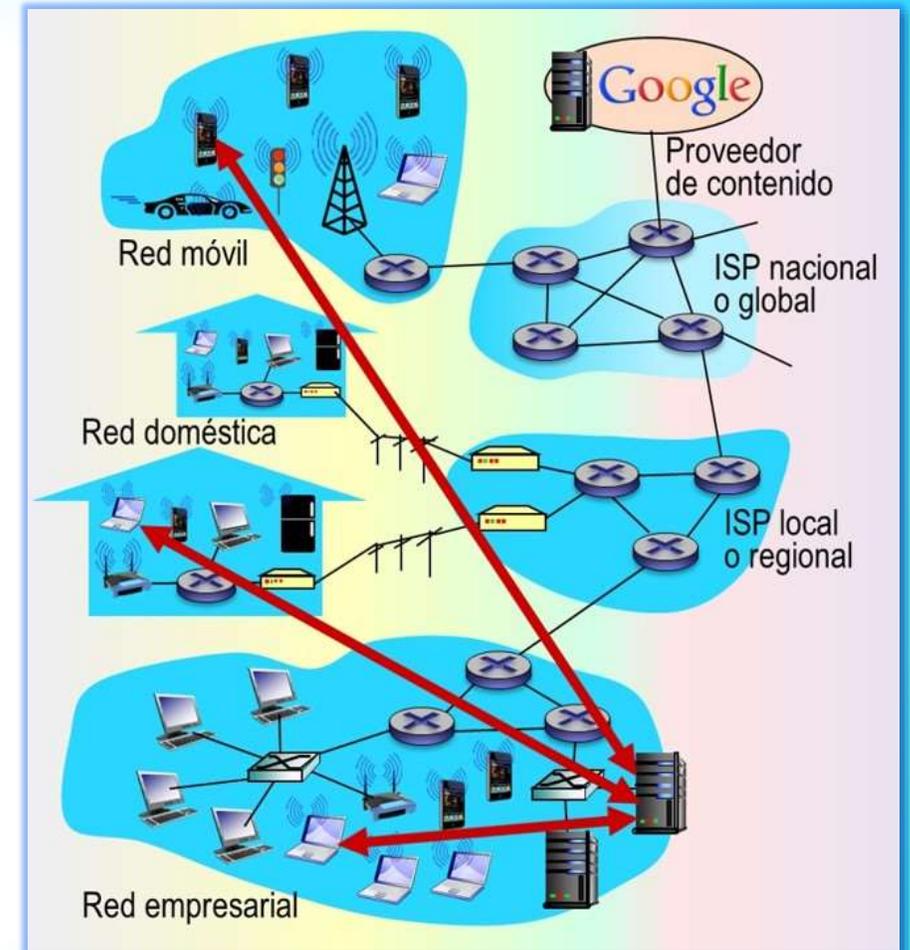
Arquitectura de las aplicaciones de red

APLICACIONES DE RED

Arquitectura cliente servidor

(Kurose, 2017)

- **A menudo**, en una aplicación **cliente-servidor** un único host servidor es incapaz de responder a todas las solicitudes de sus clientes. Por ejemplo, el sitio de una red social popular puede verse rápidamente desbordado si solo dispone de un servidor para gestionar todas las solicitudes.
- **Por esta razón**, en estas arquitecturas suele utilizarse un **Data Center**, que alberga un gran número de host para crear un servidor virtual de gran capacidad.
- **Los servicios Internet** mas populares, como los **motores de búsqueda** (Google, Bing o Baidu), los sitios de **comercio por Internet** (Amazon, e-Bay o Alibaba), el **correo electrónico basado en la Web** (Gmail y Yahoo Mail) o las **redes sociales** (Facebook, Instagram, Twitter, WhatsApp y WeChat), emplean uno o más **Data Centers**.
- **Google** dispone de entre 30 y 50 **Data Centers** distribuidos por todo el mundo, que se encargan de gestionar colectivamente las **búsquedas**, **YouTube**, **Gmail** y otros servicios.
- **Un Data Center** puede tener cientos de miles de servidores, a los que hay que suministrar energía y mantener. Además, los proveedores de servicio deben pagar los costos recurrentes de interconexión y de ancho de banda para poder enviar datos desde sus **Data Centers**.



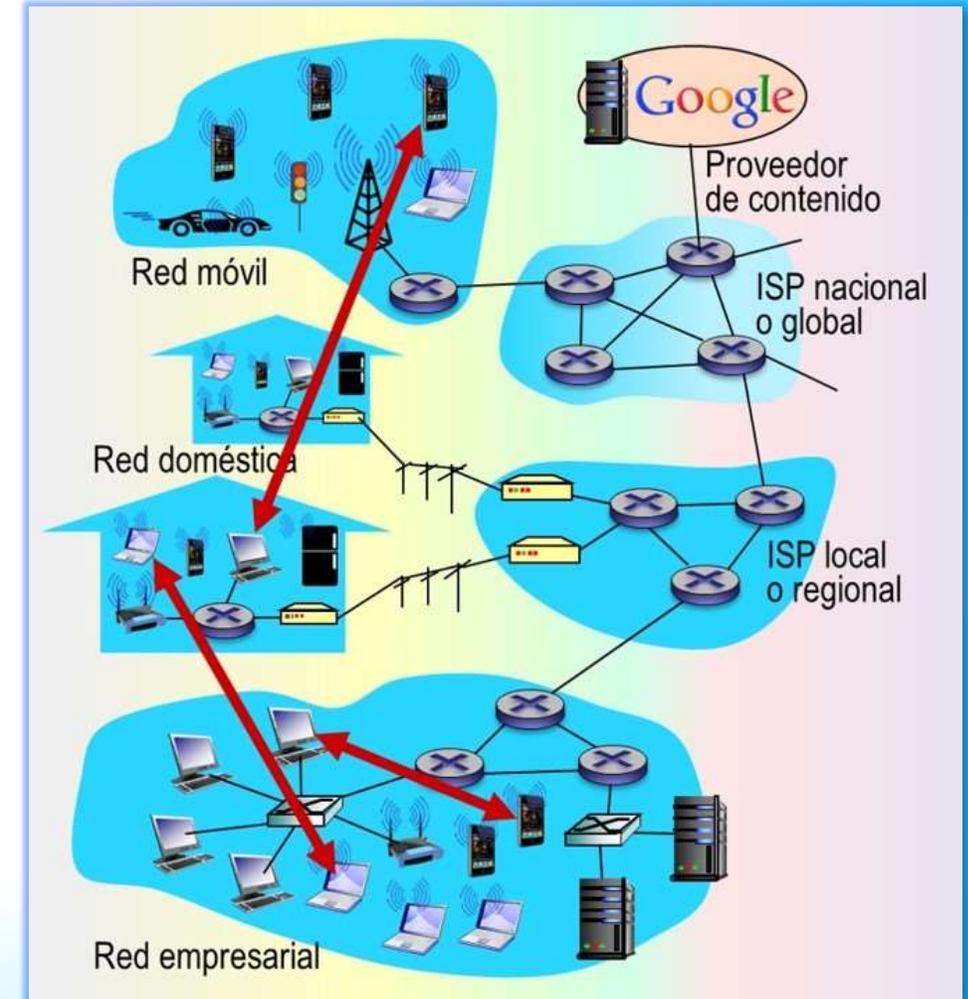
Arquitectura de las aplicaciones de red

APLICACIONES DE RED

Arquitectura peer-to-peer P2P

(Kurose, 2017)

- **En esta arquitectura** no existe ninguna dependencia de una infraestructura de servidores dedicados situados en Data Centers. En su lugar, la aplicación explota la **comunicación directa entre parejas** de hosts conectados, conocidos como pares (peers). Los pares se comunican de manera intermitente y con direcciones IP distintas en cada ocasión.
- **Los peers** no son propiedad del proveedor de servicio, sino que son hosts controlados por los usuarios, encontrándose la mayoría en domicilios, universidades y oficinas. Puesto que los pares se comunican sin pasar por un servidor dedicado, la arquitectura se denomina **peer-to-peer (P2P)**.
- **Muchas de las aplicaciones** actuales más populares y con un mayor nivel de tráfico están basadas en arquitecturas P2P. Entre estas están la **compartición de archivos** (BitTorrent), la **aceleración de descarga** con ayuda de pares (Xunlei) y la **telefonía** y **videoconferencia** por Internet (Skype, WhatsApp, Zoom).



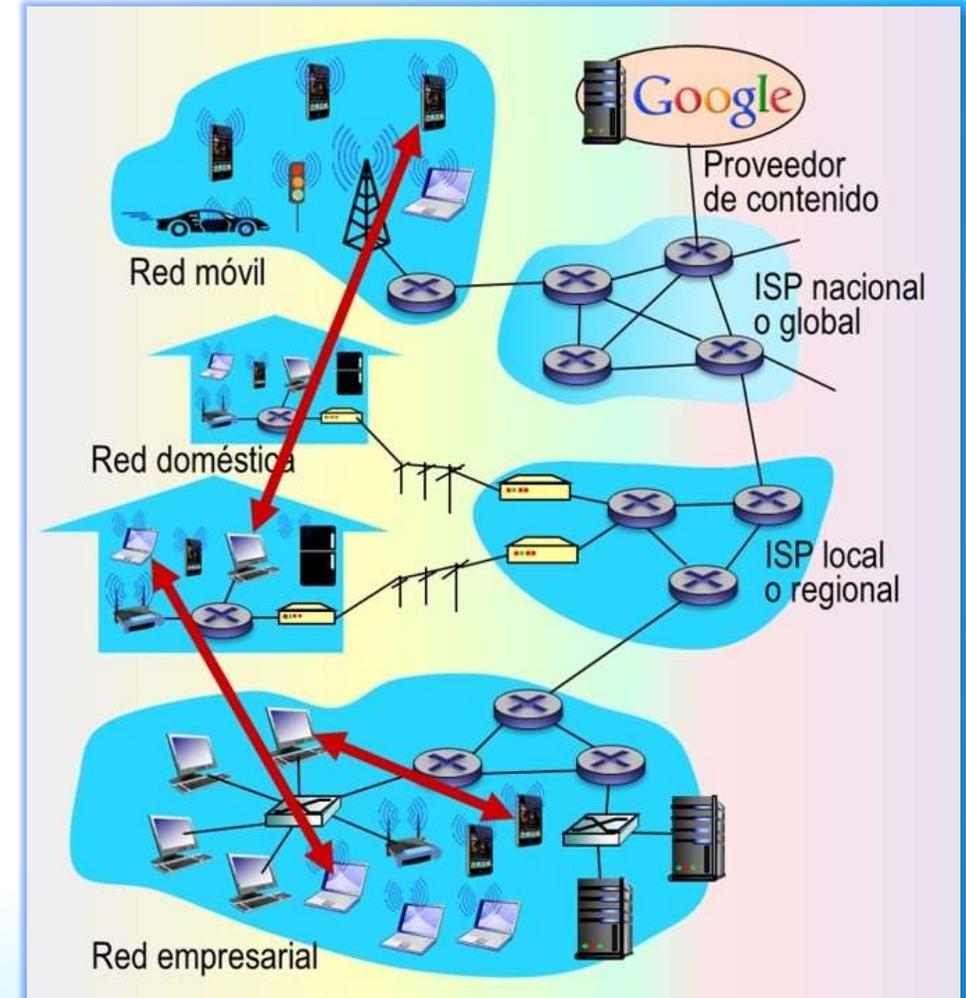
Arquitectura de las aplicaciones de red

APLICACIONES DE RED

Arquitectura híbrida

(Kurose, 2017)

- **Algunas aplicaciones** tienen arquitecturas híbridas, que combinan elementos cliente–servidor y P2P.
- **Ejemplo.** La mensajería instantánea y las aplicaciones de voz sobre IP entre dos usuarios, como Skype y WhatsApp.
 - ► El **servidor centralizado** detecta la presencia y localización de los clientes: los usuarios registran su IP con el **servidor central** al conectarse y dialogan con él en busca de la IP de su contacto.
 - ► Luego de este trámite, la conexión **cliente-cliente** se realiza en forma directa sin pasar por el servidor.
- **Las arquitecturas P2P y las híbridas** presentan una buena relación costo-prestación del servicio, ya que no requieren una infraestructura de servidores significativa ni un gran ancho de banda de servidor (a diferencia de los diseños cliente-servidor con **Data Centers**).
- **Sin embargo**, las aplicaciones P2P y las híbridas plantean problemas de seguridad, rendimiento y fiabilidad, debido a su naturaleza altamente descentralizada.



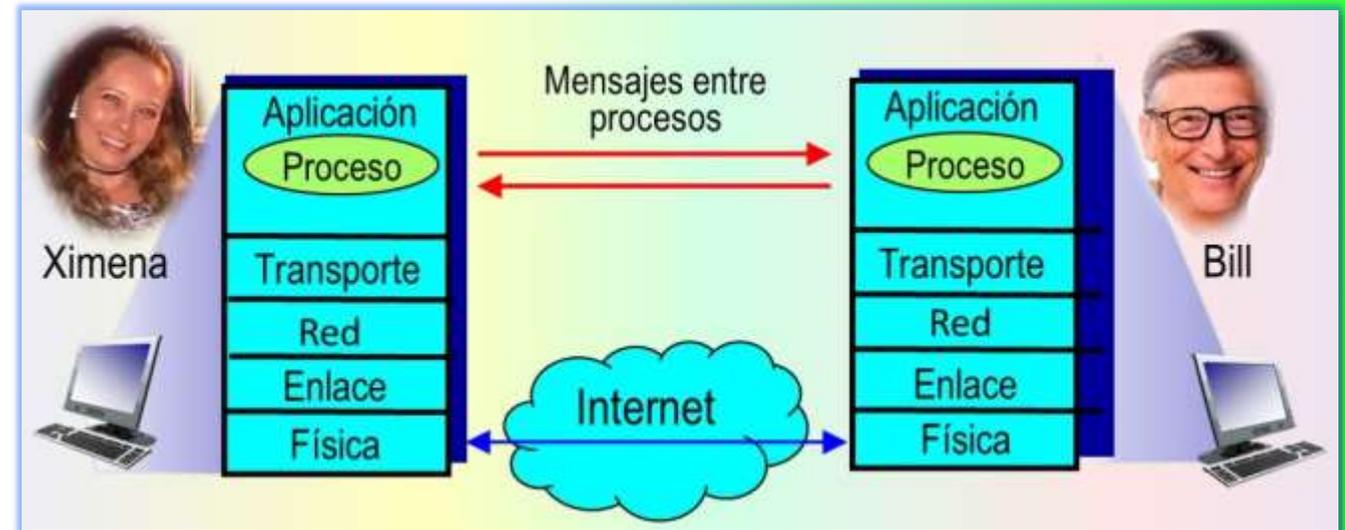
4.- COMUNICACIÓN ENTRE PROCESOS

APLICACIONES DE RED

Concepto de procesos

(Kurose, 2017)

- **Antes de crear** una aplicación de red, se necesita disponer de conocimientos básicos sobre cómo se comunican entre sí los programas que se ejecutan en varios hosts. En la jerga de los sistemas operativos, los que se comunican no son programas, sino **procesos**.
- **Un proceso** puede interpretarse como un programa que se ejecuta dentro de un host.
 - ► **Si se ejecutan** en el mismo host, se comunican entre sí mediante sistemas de comunicación interprocesos, aplicando reglas gobernadas por el sistema operativo del host.
 - ► **Si se ejecutan** en host diferentes (con sistemas operativos potencialmente diferentes), se comunican entre sí intercambiando mensajes a través de la red.
- ► **Un proceso emisor** crea y envía mensajes a la red.
- ► **Un proceso receptor** recibe estos mensajes y responde devolviendo otros.
- **La figura ilustra** que los procesos que se comunican entre sí residen en la capa de aplicación de la pila de protocolos de cinco capas.

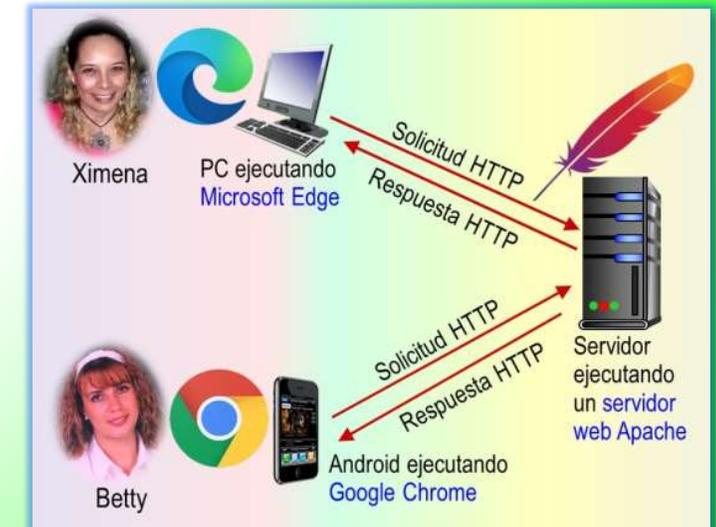
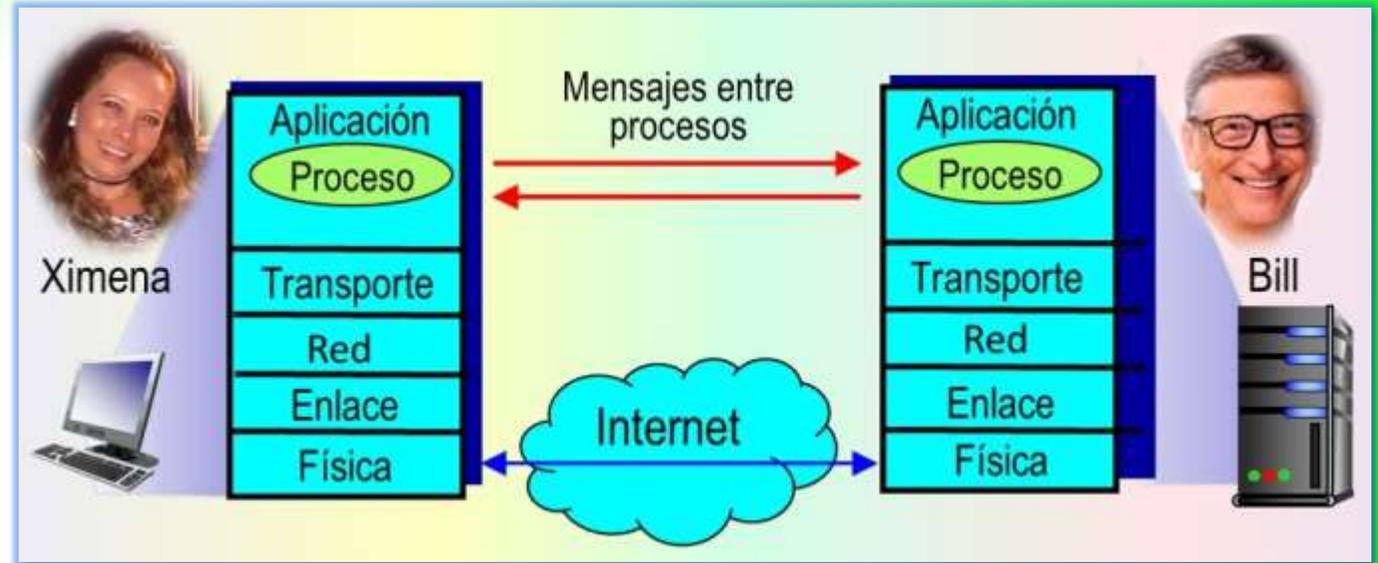


Comunicación entre procesos

APLICACIONES DE RED

Procesos cliente - servidor (Kurose, 2017)

- **Una aplicación de red** consta de parejas de procesos que se envían mensajes entre sí a través de la red.
- **En la aplicación Web** un proceso navegador de un host intercambia mensajes con un proceso de un servidor web. El navegador es el proceso cliente y el servidor web es el proceso servidor.
- **En un sistema** de compartición de archivos P2P, se transfiere un archivo desde un proceso que se ejecuta en un par, a un proceso de otro par. Normalmente, en una pareja de procesos, se designa a uno de los procesos como el cliente y al otro como el servidor. El par que descarga el archivo se designa como el cliente y el host que está cargando el archivo como el servidor.
- **De manera general**, se definen los procesos cliente y servidor como sigue: *en el contexto de una sesión de comunicación entre una pareja de procesos, el proceso que inicia la comunicación se designa como el cliente. El proceso que espera a ser contactado para comenzar la sesión es el servidor.*

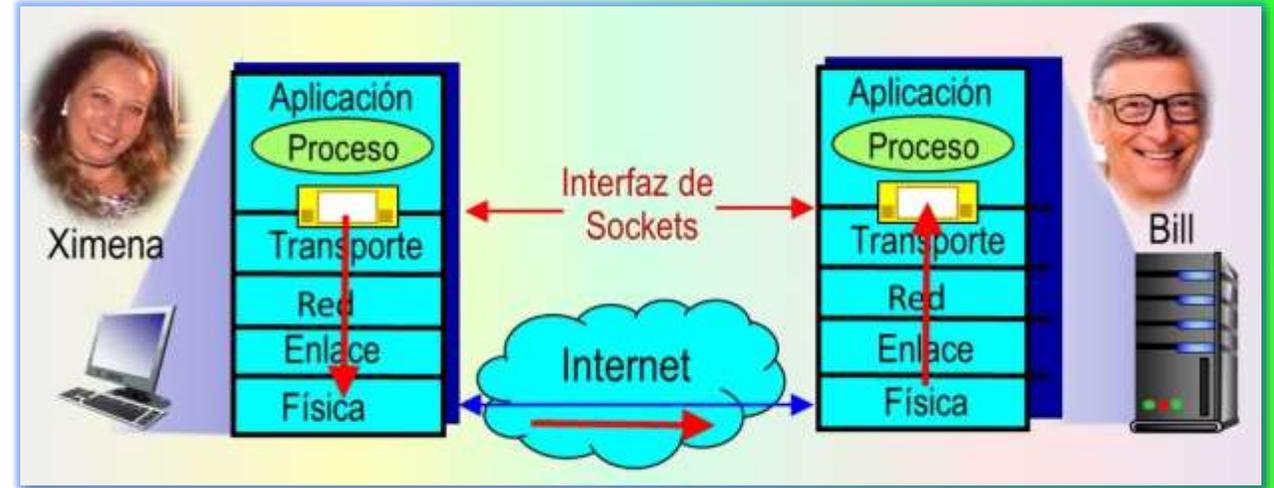


Comunicación entre procesos

APLICACIONES DE RED

Interfaz entre el proceso y la red (Kurose, 2017)

- **La mayoría de las aplicaciones** constan de **parejas de procesos** (cliente – servidor) que se comunican intercambiando mensajes.
- **Cualquier mensaje** enviado de un proceso al otro debe atravesar la red subyacente.
- **Un proceso** envía mensajes a la red y los recibe de la red a través una interfaz software denominada **socket** (enchufe).
- **¿Qué es un socket?** Una analogía para comprender el concepto de socket asociado a un proceso es la siguiente:
 - ► **Un proceso** es análogo a una casa y un **socket** a la puerta de la casa. Cuando un proceso desea enviar un mensaje a otro que se ejecuta en otro host, envía el mensaje a través de su propia puerta (socket).
 - ► **El proceso emisor** supone que existe una infraestructura de transporte al otro lado de la puerta que llevará el mensaje hasta la puerta del proceso de destino.
 - ► **Una vez que el mensaje** llega al host de destino, éste pasa a través de la puerta (**socket**) del proceso receptor, actuando entonces el proceso receptor sobre el mensaje.



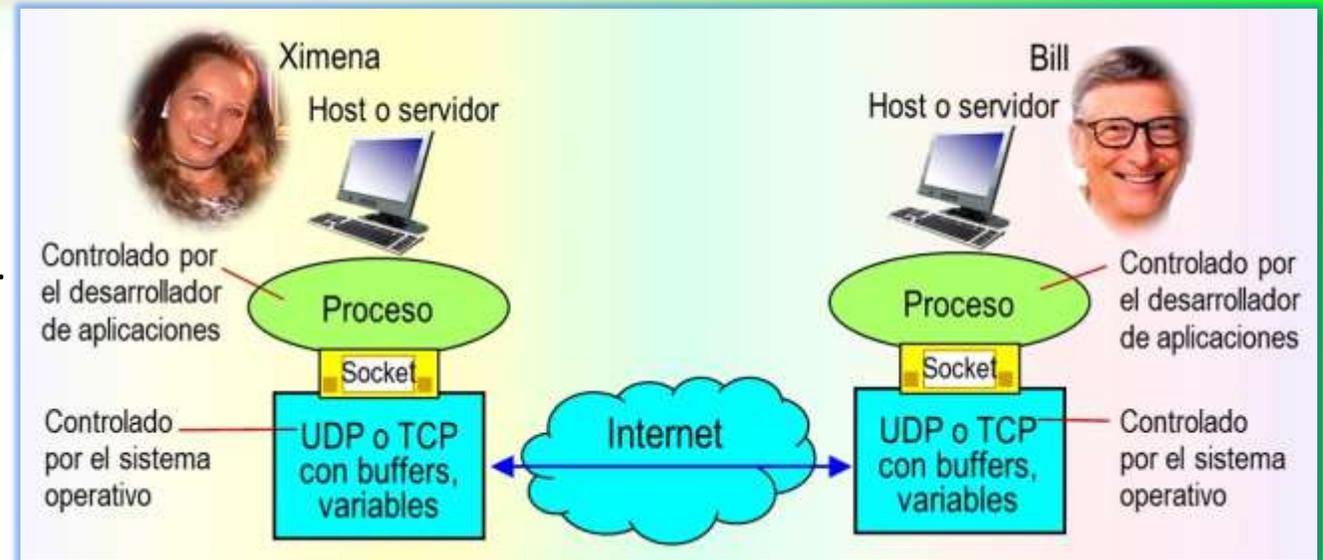
Comunicación entre procesos

APLICACIONES DE RED

Comunicación entre sockets

(Kurose, 2017)

- **En la figura**, se ilustra la comunicación mediante sockets entre dos procesos que se comunican a través de Internet.
- **El protocolo de transporte** subyacente utilizado por los procesos puede ser UDP o TCP.
- **Un socket** es la interfaz entre la capa de aplicación y la de transporte de un host.
- **También se conoce** como **Interfaz de programación de aplicaciones API** entre la aplicación y la red, ya que el socket es la interfaz de programación con la que se construyen las aplicaciones de red.
 - ✉ **Por ejemplo**, un socket UDP se crea mediante la línea de código del programa Python:
`clientsocket=socket(AF_INET,SOCKET_DGRAM)`
- **El desarrollador de la aplicación** tiene el control de todos los elementos del lado de la capa de aplicación del socket, pero no de los del lado de la capa de transporte, a excepción de:
 - ▶(1) **La elección del protocolo** de transporte que puede ser UDP o TCP y
 - ▶(2) **Quizás la capacidad de fijar** unos pocos parámetros, como los tamaños máximos del buffer y de segmento.
- **Una vez** que el desarrollador ha seleccionado un protocolo de transporte, la aplicación se construye utilizando los servicios de la capa de transporte proporcionado por dicho protocolo.



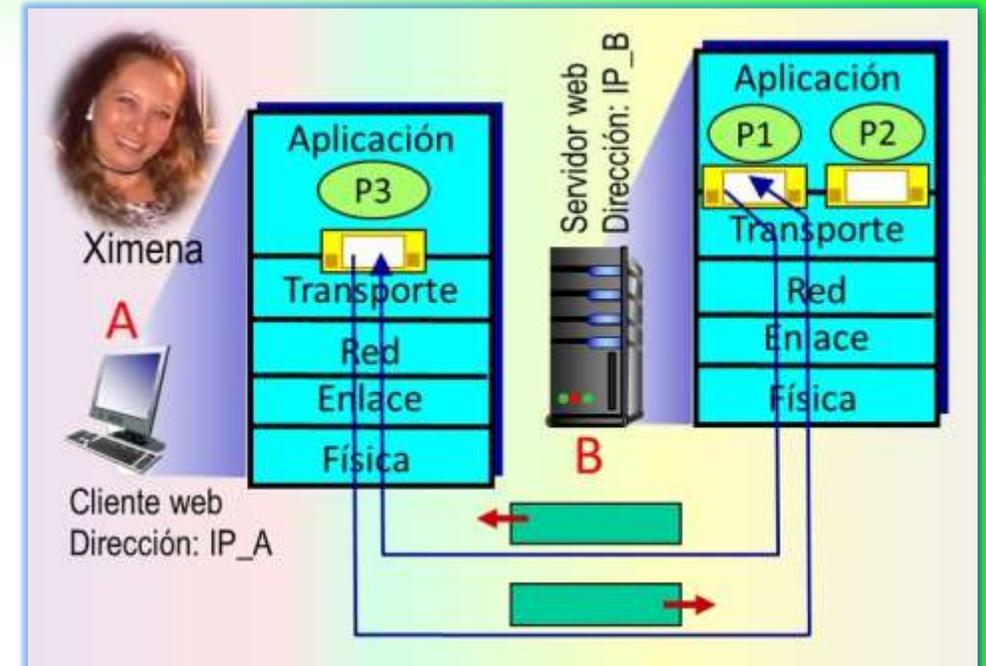
Comunicación entre procesos

APLICACIONES DE RED

Direccionamiento de procesos

(Kurose, 2017)

- **Para enviar una carta** a un destino concreto, se necesita una dirección. De forma similar, para que un proceso que se está ejecutando en un host pueda enviar paquetes a otro proceso que se ejecuta en un host distinto, se necesita de una dirección del proceso receptor.
- **Para identificar** al proceso receptor se necesitan dos elementos de información:
 - (1) La dirección IP del host.
 - (2) Un identificador, llamado número de puerto, que especifique el proceso de recepción en el host de destino.
- **Por tanto**, además de conocer la dirección IP del host receptor, el host emisor también debe identificar al proceso receptor (o, para ser más exactos, al socket receptor) que está ejecutándose en el host. Esta información es necesaria porque, en general, un host podría estar ejecutando varias aplicaciones de red a la vez.
- **Para identificar** a un proceso (o más exactamente a un socket) se utiliza un número de puerto. Cada número de puerto es un número en el rango de 0 a 65535 (16 bits).
- **Los números de puerto** pertenecientes al rango de 0 a 1023 se conocen como números de puertos bien conocidos y son restringidos, lo que significa que están reservados para ser empleados por los protocolos de aplicación bien conocidos, como por ejemplo HTTP que utiliza el puerto 80, FTP que utiliza el número de puerto 21 o el proceso de servidor de correo SMTP que se identifica mediante el puerto 25.



Comunicación entre procesos

APLICACIONES DE RED

Algunos números de puerto

(CISCO, 2016)

- La **tabla** muestra el número de puerto de los protocolos de aplicación bien conocidos y el protocolo de transporte subyacente.
- ✉ Por ejemplo, el protocolo **HTTP** de la **aplicación web** utiliza los servicios del protocolo de transporte **TCP**, y el número de puerto del proceso es **80**.

Número de puerto	Protocolo	Aplicación	Acrónimo
20	TCP	Protocolo de transferencia de archivos (datos)	FTP
21	TCP	Protocolo de transferencia de archivos (control)	FTP
22	TCP	Shell Seguro	SSH
23	TCP	Telnet	-
25	TCP	Protocolo simple de transferencia de correo (Simple Mail Transfer Protocol)	SMTP
53	UDP, TCP	Servicio de nombres de dominios	DNS
67	UDP	Protocolo de configuración dinámica de host (servidor)	DHCP
68	UDP	Protocolo de configuración dinámica de host (cliente)	DHCP
69	UDP	Protocolo de transferencia de archivos trivial	TFTP
80	TCP	Protocolo de transferencia de hipertexto	HTTP
110	TCP	Protocolo de oficina de correos versión 3 (Post Office Protocol version 3)	POP3
143	TCP	Protocolo de acceso a mensajes de Internet (Internet Message Access Protocol)	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Protocolo seguro de transferencia de hipertexto	HTTPS

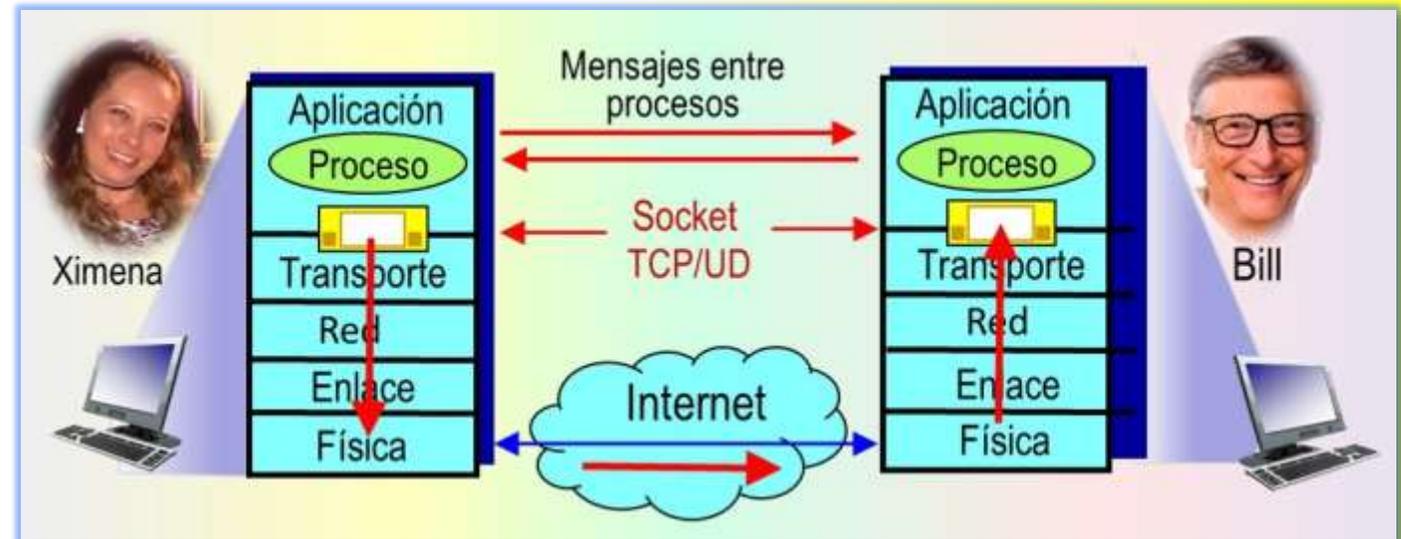
5. PROTOCOLOS DE LA CAPA DE APLICACIÓN

APLICACIONES DE RED

¿Qué define un protocolo de la capa de aplicación?

(Kurose, 2017)

- **Se ha visto que** los procesos de red se comunican entre sí enviando mensajes a sus sockets. Pero, ¿cómo están estructurados estos mensajes? ¿Cuál es el significado de cada uno de los campos de estos mensajes? ¿Cuándo envían los procesos estos mensajes? Estas preguntas llevan al ámbito de los protocolos de la capa de aplicación.
- **Un protocolo de la capa de aplicación** define cómo los protocolos de una aplicación, que se ejecutan en distintos sistemas terminales, se pasan los mensajes entre sí. En particular, un protocolo de la capa de aplicación define:
 - ▶ **Los tipos de mensajes** intercambiados; por ejemplo, mensajes de solicitud y mensajes de respuestas.
 - ▶ **La sintaxis de los diversos** tipos de mensajes, es decir, los campos de los que consta el mensaje y cómo se delimitan estos campos.
 - ▶ **La semántica de los campos**, es decir, el significado de la información contenida en los campos.
 - ▶ **Las reglas** para determinar cuándo y cómo un proceso envía mensajes y responde a los mismos.



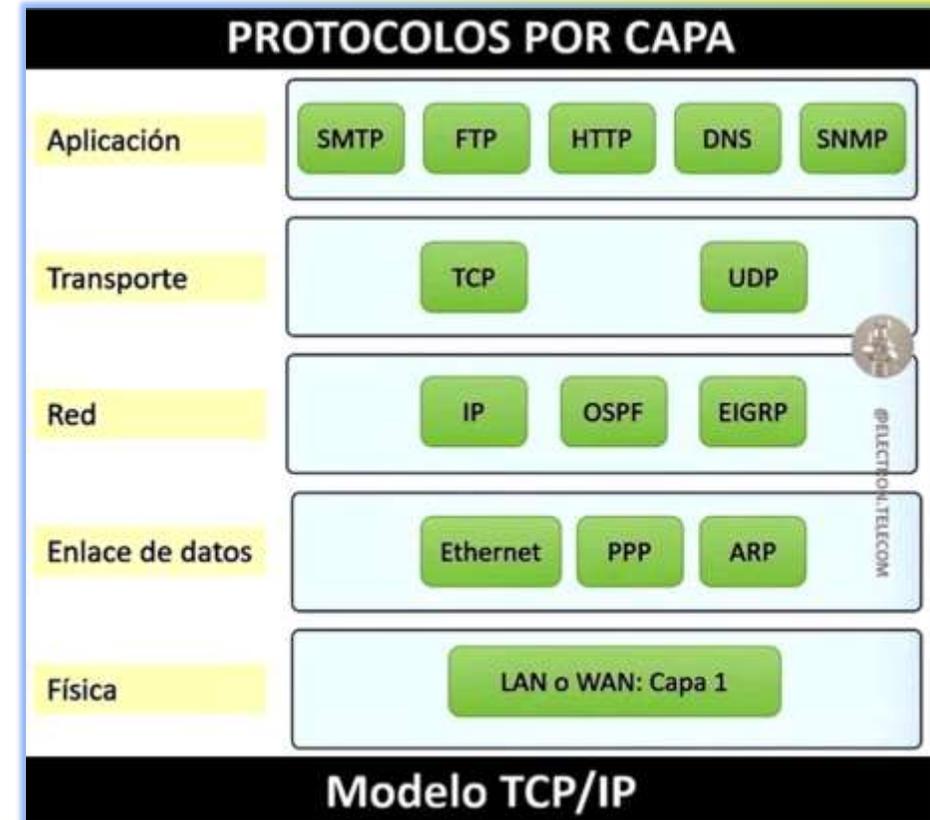
Protocolos de la capa de aplicación

APLICACIONES DE RED

Especificación de los protocolos de la capa de aplicación

(Kurose, 2017)

- **Algunos protocolos** de la capa de aplicación están especificados en documentos RFC (solicitud de Comentario) y, por tanto, son de dominio público.
 - ✉ **Por ejemplo** el protocolo de la capa de aplicación para la Web, HTTP (Protocolo de Transferencia de Hipertexto) está disponible como un RFC (2616).
 - ✉ Si quien desarrolla un navegador web sigue las reglas dadas en el RFC que se ocupa de HTTP, el navegador podrá recuperar páginas web de cualquier servidor web que también se ajuste a las reglas de dicho RFC.
- **Existen muchos otros protocolos** de la capa de aplicación que son propietarios y que intencionadamente no están disponibles para todo el mundo. Por ejemplo, Facebook utiliza protocolos de la capa de aplicación propietarios.
- **Es importante diferenciar** entre aplicaciones de red y protocolos de la capa de aplicación. Un protocolo de la capa de aplicación es únicamente un elemento de una aplicación de red (aunque uno muy importante).



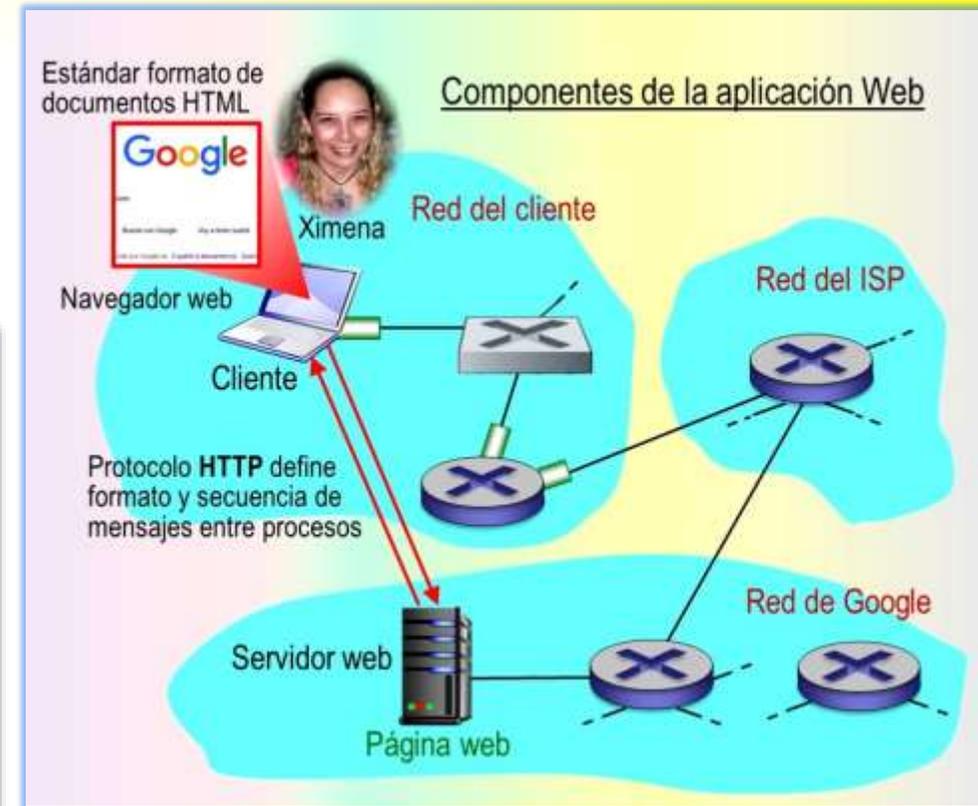
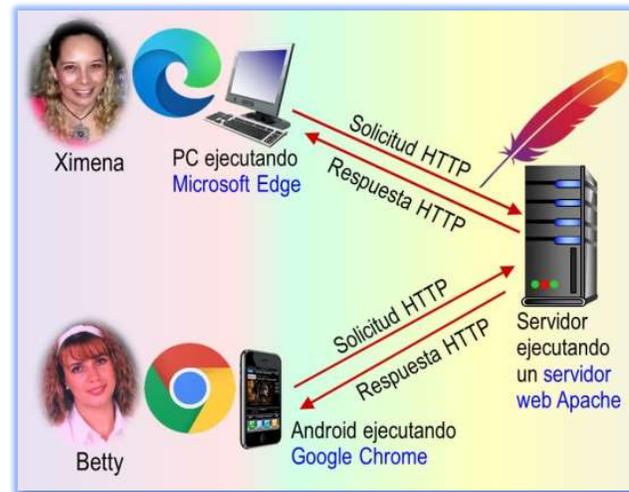
Protocolos de la capa de aplicación

APLICACIONES DE RED

La aplicación Web

(Kurose, 2017)

- **Es una aplicación** cliente-servidor que permite a los usuarios obtener documentos almacenados en servidores web bajo demanda. La aplicación Web consta de muchos componentes, entre los que se incluyen:
- **► Navegadores web**, como Firefox, Microsoft Explorer, Google Chrome.
- **► Un estándar** para el formato de los documentos **HTML** (Lenguaje de marcas de hipertexto que describe la estructura de las páginas web mediante etiquetas que le dicen al navegador como debe mostrarlas).
- **► Servidores web**, como servidores Apache, Microsoft, Amazon.
- **► Un protocolo** de la capa de aplicación. El protocolo de la Web, **HTTP**, define el formato y la secuencia de los mensajes que se pasan entre sí el navegador web y el servidor web. Por tanto, HTTP es solo una pieza (aunque muy importante) de la aplicación Web.



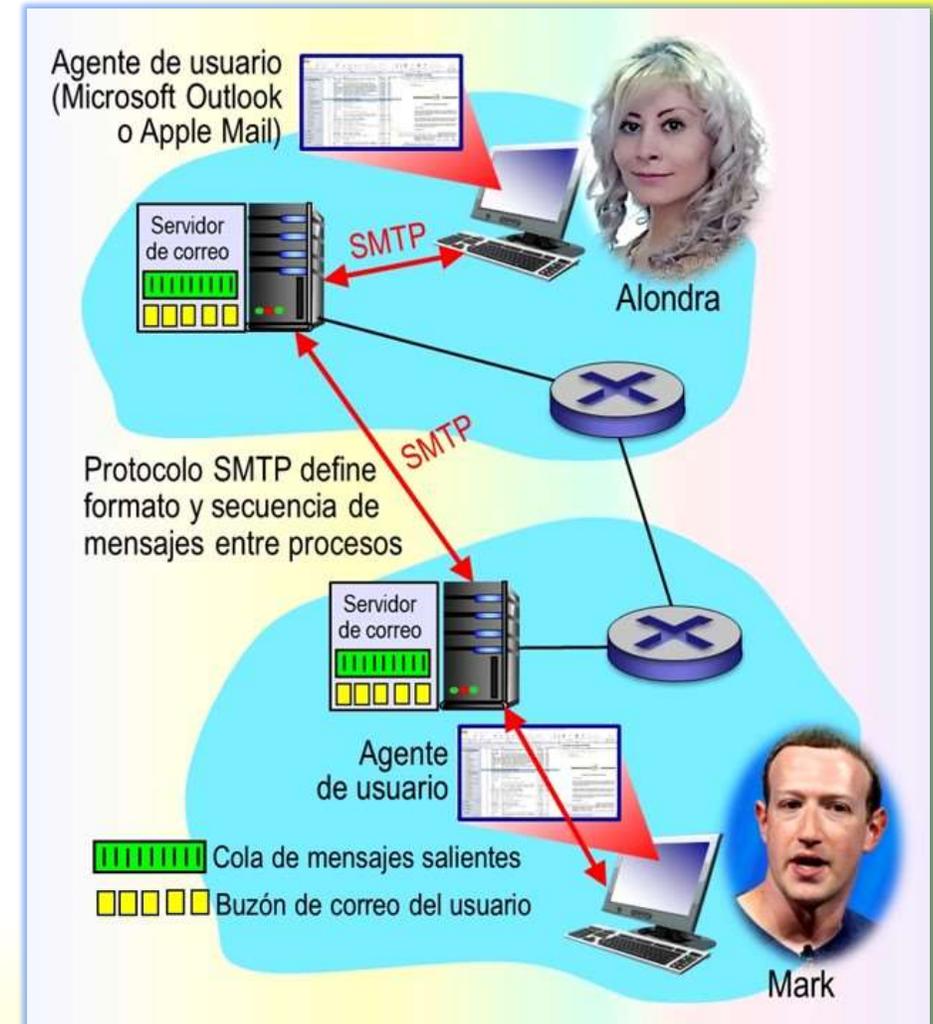
Protocolos de la capa de aplicación

APLICACIONES DE RED

El correo electrónico Internet

(Kurose, 2017)

- **Esta aplicación** está constituida por muchos componentes, entre los que se incluyen:
 - ▶ **Los servidores de correo** que albergan los buzones de los usuarios.
 - ▶ **Los agentes de usuario**, como Microsoft Outlook o Apple Mail, que permiten a los usuarios leer y crear mensajes; es un estándar para definir la estructura de los mensajes de correo electrónico.
 - ▶ **Los protocolos** de la capa de aplicación que definen cómo se pasan los mensajes entre los servidores, cómo se pasan los mensajes entre los servidores y los clientes de correo y cómo se interpretan los contenidos de las cabeceras de los mensajes.
 - ✉ El principal protocolo de la capa de aplicación para el correo electrónico es el protocolo de simple de transferencia de correo **SMTP** (RFC 5321).
- **Por tanto**, el protocolo principal de la capa de aplicación para correo electrónico, SMTP, sólo es un componente (aunque importante) de la aplicación de correo electrónico.



Referencias bibliográficas

APLICACIONES DE RED

Referencias bibliográficas

- CISCO (2015). *CCNA Routing and Switching. Introduction to Networks*. CISCO.
- CISCO (2016). *Introducción a las redes*. Madrid: Pearson Education, S.A.
- Forouzan, B. A. (2020). *Transmisión de datos y redes de comunicaciones*. Madrid: McGraw-Hill.
- Huawei Technologies (2020). *Basics of data communication networks*. Huawei.
- Kurose, J. Keith, R. (2017). *Redes de computadoras: un enfoque descendente*. Madrid: Pearson Education, S.A.

FIN

Tema 6 de:
PROTOCOLOS DE INTERNET
Edison Coimbra G.