

REDES DE COMPUTADORES Y LABORATORIO

Christian Camilo Urcuqui López, MSc



COMPETENCIAS

- Aplicar los Factory methods de la API de Java
- Comparé los modelos de referencia OSI y TCP/IP.
- Describir DNS
- Explicar las unidades métricas de cantidad de información.
- Explicar la aplicación del análisis de red para la investigación de amenazas cibernéticas.

DNS-DOMAIN NAME SYSTEM

Dificultad para que las personas se aprendieran las direcciones IP en notación punteada.

Inicialmente, se utilizó un archivo llamado **host.txt**, donde se guardaba una lista, con la correspondencia entre la dirección IP del equipo y un nombre que permitirá distinguirlo.

Por ejemplo:

- Contabilidad 191.105.6.10
- Finanzas 191.105.6.11
- Gerencia 159.166.99.67

DNS-DOMAIN NAME SYSTEM

host.txt, inconvenientes:

- El archivo host.txt se volvió demasiado grande
- Era necesario actualizar el archivo más de una vez al día
- Era muy difícil controlar que sólo un equipo tuviera el mismo nombre en toda la red.

DNS-DOMAIN NAME SYSTEM

- El sistema de nombres de dominio fue inventado en 1984 y permite a los usuarios de red utilizar **nombres jerárquicos sencillos para comunicarse con otros equipos**, en vez de memorizar y usar sus direcciones IP. Este sistema es muy usado en Internet y en muchas de las redes privadas actuales. Las aplicaciones como: browsers, servidores de Web, FTP y telnet, utilizan DNS.
- **En general DNS es una base de datos distribuida que proporciona un sistema jerárquico de nombres para identificar a los equipos en Internet.**

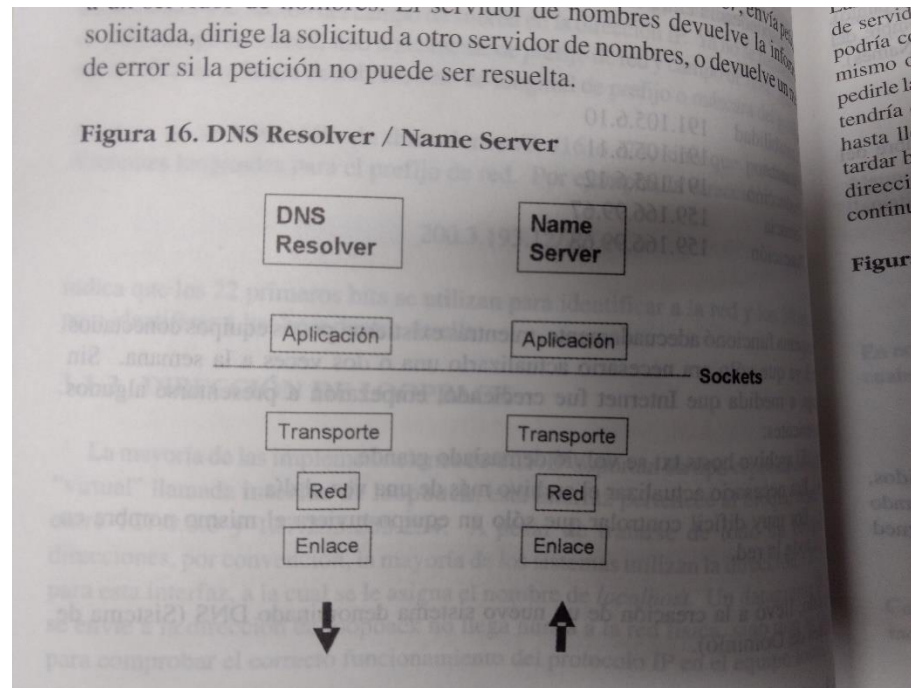
¿CÓMO TRABAJA DNS?

El DNS utiliza tres componentes:

- Resolutores (*resolvers*)
- servidores de nombres (*name servers*)
- El espacio de nombres de dominios (*domain name space*).

RESOLVERS Y SERVIDORES DNS

Con la comunicación básica DNS, un cliente DNS, o *resolver*, envía peticiones a un servidor de nombres. El servidor de nombres devuelve la información solicitada, dirige la solicitud a otro servidor de nombres, o devuelve un mensaje de error si la petición no puede ser resuelta.



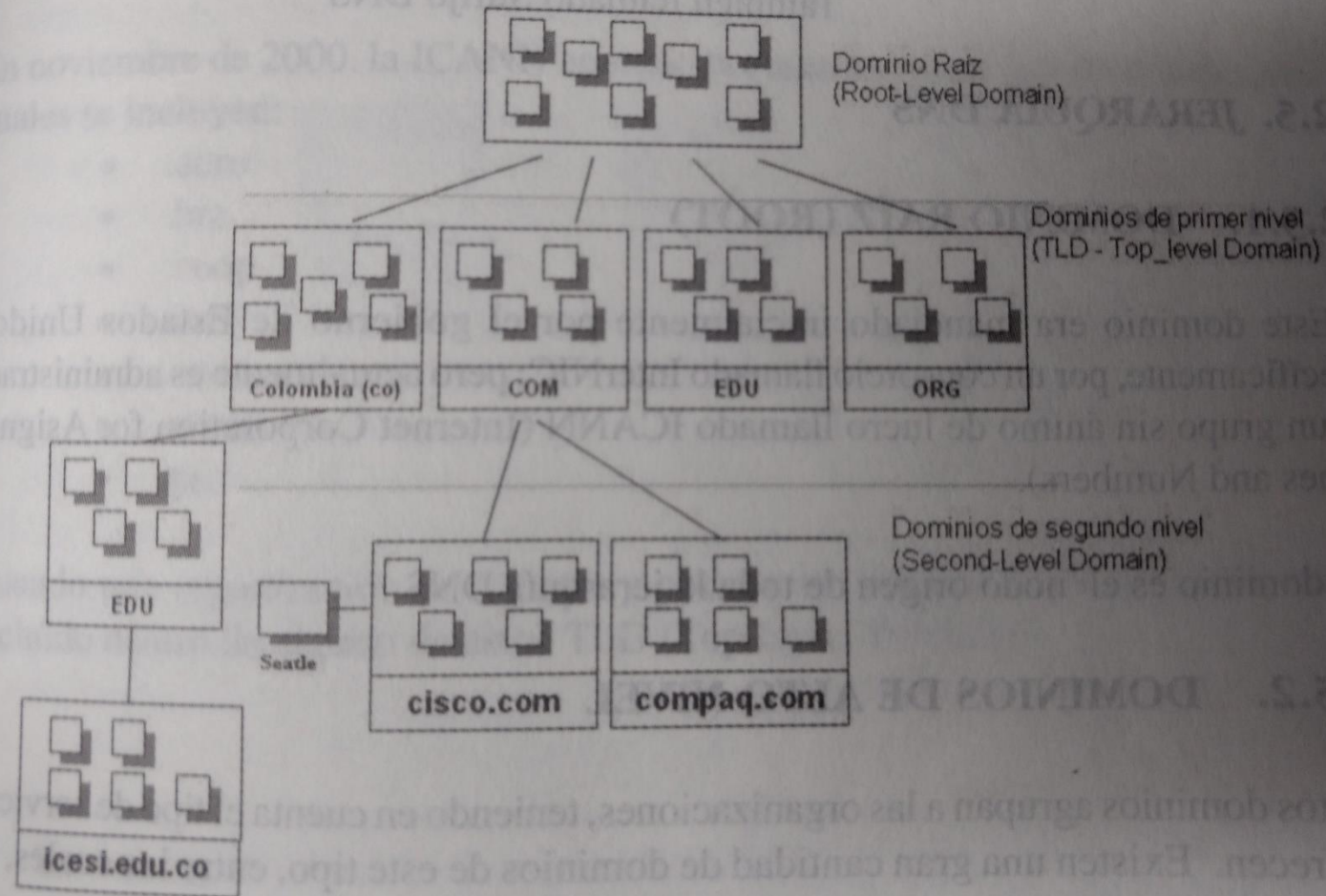
DOMAIN NAME SPACE

La base de datos DNS es una estructura tipo “árbol”, denominada **Espacio de nombres del dominio**. Cada dominio (un nodo/hoja en la estructura) tiene un nombre asignado y puede contener subdominios. EL nombre del dominio, identifica la posición del mismo dentro de la base de datos, en relación con su “Domino padre”. Un punto (.) separa cada segmento del nombre de cada nodo de red del domino DNS.

Por ejemplo, icesi.edu.co especifica el subdominio “icesi”, cuyo dominio padre es “edu”.

direcciones IP localmente en lugar de recorrer toda la jerarquía de servidores continuamente, esto se logra gracias al caché DNS.

Figura 17. Estructura jerárquica del DNS



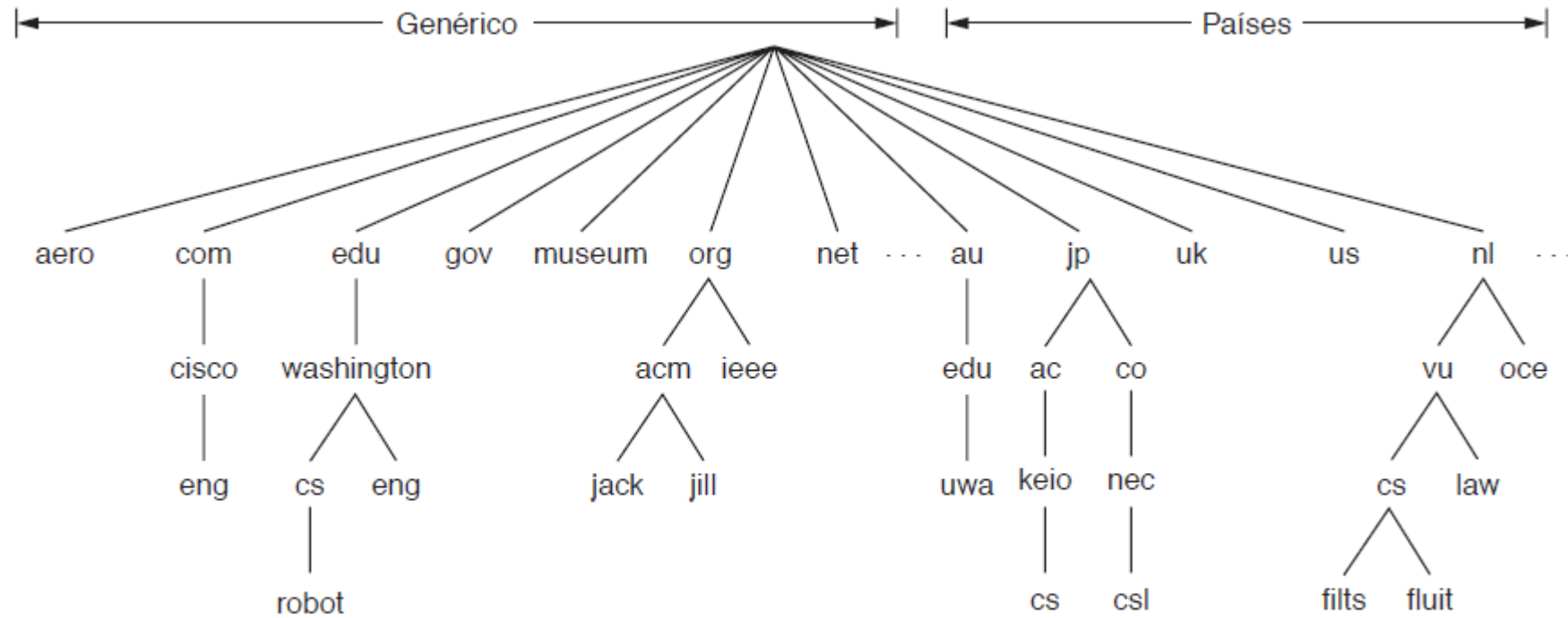


Figura 7-1. Una porción del espacio de nombres de dominio de Internet.

TOP LEVEL DOMAINS

Existen organizaciones sin animo de lucro encargdas de la administración de los dominios de primer nivel (TLD), entre estas se encuentran el Internet Corportation Internet Corporation for Assigned Names and Numbers(ICANN), que opera con el Internet Assigned Numbers Authority (IANA). Existen distintas empresas (por ejemplo, Verisign) que se encargan de proveer la infraestructura y soportar la distribución de los DNS

Existen aproximadamente 250 dominios de nivel superior.



JERARQUÍA DNS

Dominio Raíz (ROOT)

Este dominio era manejado inicialmente por el gobierno de Estados Unidos, específicamente, por un consorcio llamado InterNIC; pero actualmente es administrado por un grupo sin ánimo de lucro llamado ICANN (Internet Corporation for the Assigned Names and Numbers).

JERARQUÍA DNS

Dominios de alto nivel (Top Level Domains, TLD)

Estos dominios agrupan a las organizaciones, teniendo en cuenta el tipo de servicios que ofrecen. Existen una gran cantidad de dominios de este tipo, entre las cuales, más utilizados son:

- .com (organizaciones comerciales)
- .net ()
- .org ()
- .gov (gobierno)
- .edu (educación)
- .mil (militares)

United States set to Hand Over Control of the Internet to ICANN Today

October 01, 2016 Mohit Kumar



Since the foundation of the Internet, a [contract](#) has been handed over to the United States Commerce Department under which the department had given authority to regulate the Internet.

After 47 years, this contract ends tonight at midnight EDT i.e. Saturday, October 1st, 2016.

If you think that the United States owns the Internet, then you're wrong. It doesn't.



Popular News



New FaceTime Bug Lets Callers Hear and See You Without You Picking Up



iCloud Possibly Suffered A Privacy Breach Last Year That Apple Kept a Secret



Facebook Paid Teens \$20 to

Massive DDoS Attack Against Dyn DNS Service Knocks Popular Sites Offline

October 21, 2016 Swati Khandelwal



UPDATE — How an army of million of hacked Internet-connected smart devices almost broke the Internet today.

Cyber attacks are getting evil and worst nightmare for companies day-by-day, and the Distributed Denial of Service (DDoS) attack is one such attacks that cause a massive damage to any service.



Popular News



New FaceTime Bug Lets Callers Hear and See You Without You Picking Up



iCloud Possibly Suffered A Privacy Breach Last Year That Apple Kept a Secret



Facebook Paid Teens \$20 to

Google DNS Service (8.8.8.8) Now Supports DNS-over-TLS Security

📅 January 10, 2019 👤 Swati Khandelwal



Almost every activity on the Internet starts with a DNS query, a key function of the Internet that works as an Internet's directory where your device looks up for the server IP addresses after you enter a human-readable web address (e.g., thehackernews.com).

ManageEngine
Ten **security controls** for effective cybersecurity.

FREE

Download Infographics

Popular News



New FaceTime Bug Lets Callers Hear and See You Without You Picking Up



iCloud Possibly Suffered A Privacy Breach Last Year That Apple Kept a Secret

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

Ambos modelos se basan en ser una pila de protocolos independientes, además, la funcionalidad de las capas es muy similar. Por ejemplo, **en ambos modelos las capas desde transporte a aplicación tienen como finalidad ofrecer el servicio de transporte independiente de la red.** A pesar de contar con algunas similitudes, existen varias diferencias.

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

En el modelo OSI existen tres conceptos:

- Servicios
- Interfaces
- Protocolos

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

OSI hace explícita la distinción entre estos tres conceptos. Cada capa desempeña unos *servicios*, es decir, indica lo que hace para el nivel que está sobre ella, no especifica cómo acceden a ella las entidades ni cómo funciona. Define la semántica de la capa.

La capa es la que debe decidir qué *protocolos* de iguales utilizar. Puede usar los protocolos que quiera, siempre y cuando realice el trabajo (es decir, que provea los servicios ofrecidos). También los puede cambiar a voluntad sin afectar el software de las capas superiores.

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

Al principio, el modelo TCP/IP no tenía una distinción clara entre los servicios, las interfaces y los protocolos, pero a medida que ha pasado las distintas comunidades han venido ajustándolo a algo más parecido al modelo OSI.

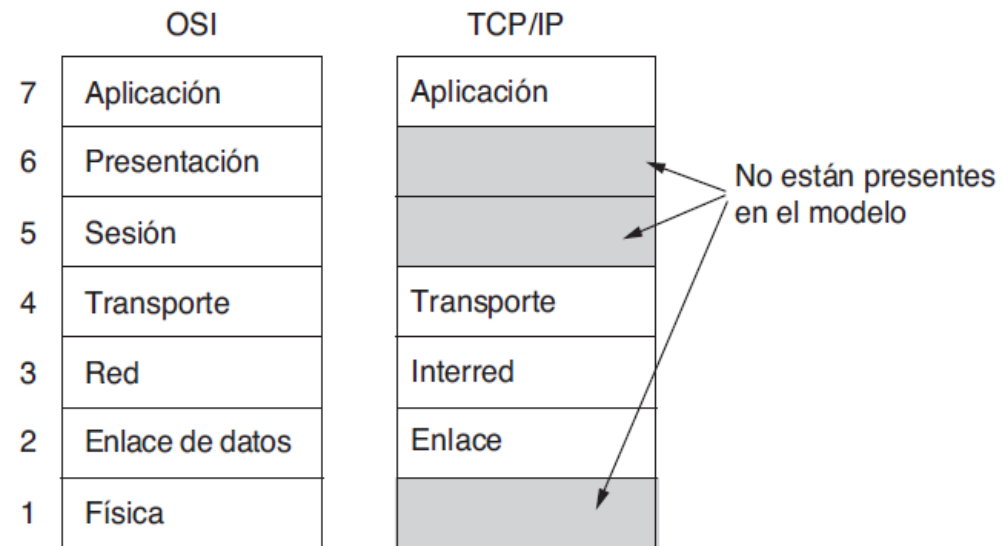


Figura 1-21. El modelo de referencia TCP/IP.

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

El modelo de referencia OSI se idea antes de que se inventaran los protocolos, esto quiere decir que el modelo en un principio no estaba enfocado hacia un conjunto de específico de protocolos, por lo cual lo hizo bastante general. Su desventaja fue que los diseñadores en un principio no tenían mayor experiencia con el tema y no supieron que servicio / funcionalidad agregar a cada una de las capas. *Por ejemplo, en un principio la capa de enlace de datos trabaja sólo con redes de punto a punto.*

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

Por otra parte, en TCP/IP llegaron los protocolos y el modelo era en realidad sólo una descripción sólo una descripción de los protocolos existentes. No hubo problema para que los protocolos se ajustaran al modelo. Encajaron a la perfección. El único problema fue que el *modelo* no encajaba en ninguna otra pila de protocolos. En consecuencia, no era útil para describir otras redes que no fueran TCP/IP.

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

Ni el modelo OSI y sus protocolos, ni el modelo TCP/IP y sus protocolos son perfectos. Ambos pueden recibir bastantes críticas, y así se ha hecho.

Las críticas al modelo OSI se pueden resumir así:

- Mala sincronización
- Mala tecnología
- Malas implementaciones
- Mala política

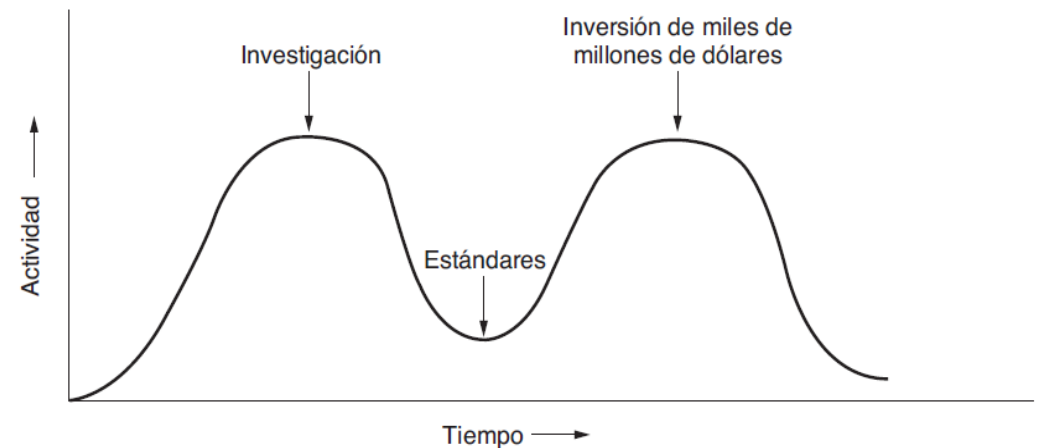
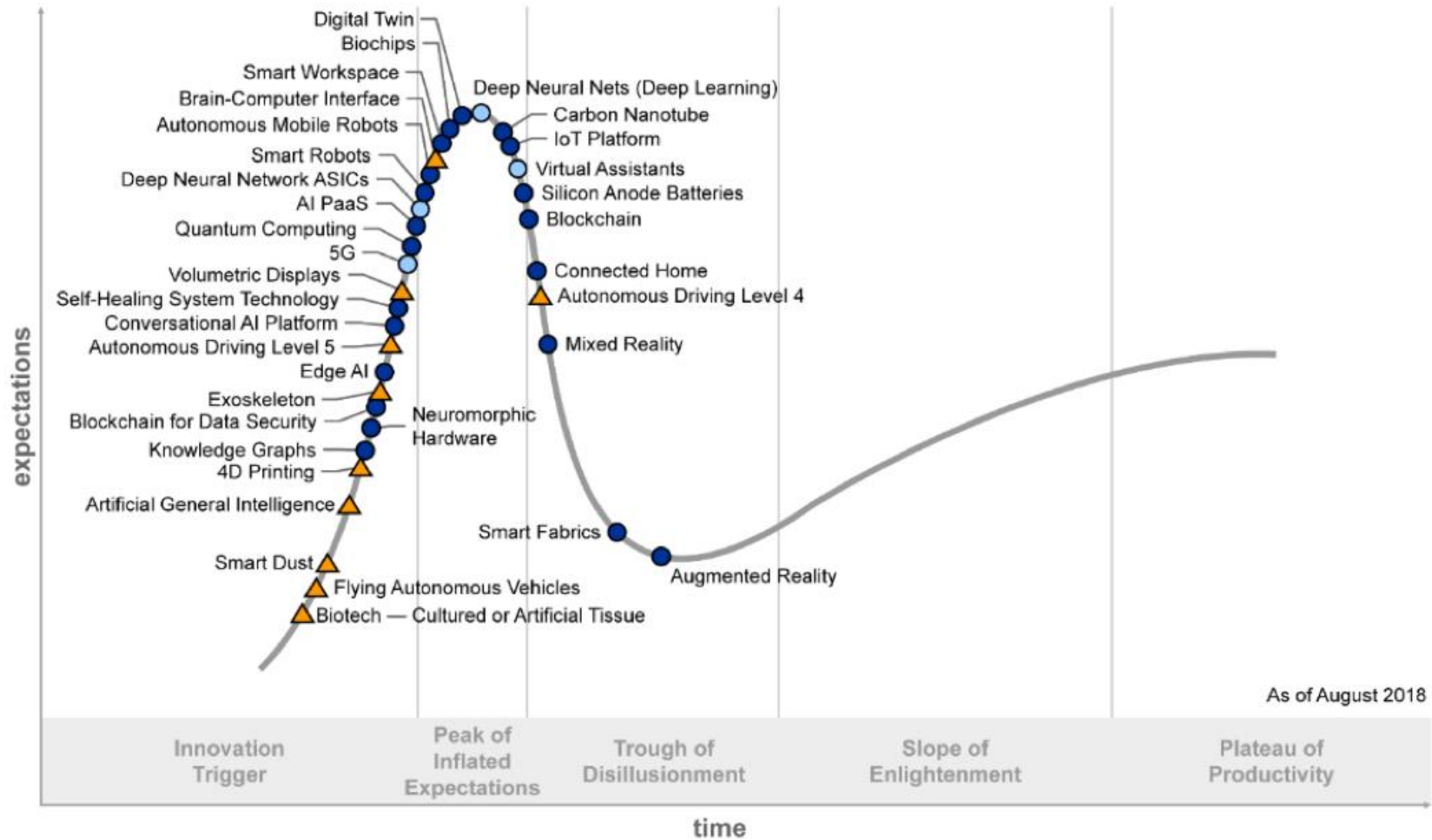


Figura 1-24. El apocalipsis de los dos elefantes.

COMPARACIÓN DE LOS MODELOS DE REFERENCIA OSI Y TCP/IP

Críticas TCP/IP

- Falta de aclaración de los conceptos de servicios, interfaces y protocolos. La buena práctica de la ingeniería de software requiere una distinción entre la especificación y la implementación.
- No es nada general y no es muy apropiado para describir cualquier pila de protocolos aparte de TCP/IP. Por ejemplo, es imposible tratar de usar el modelo TCP/IP para describir Bluetooth.
- La capa de enlace en realidad no es una capa en el sentido normal del término como se utiliza en el contexto de protocolos en capas. Es una interfaz (entre las capas de red y de enlace de datos). La diferencia entre una interfaz y una capa es crucial, y hay que tener mucho cuidado al respecto.
- No distingue entre la capa física y la de enlace de datos.



Plateau will be reached:

○ less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

© 2018 Gartner, Inc.

CLIENTE / SERVIDOR

Es una **arquitectura** que hace en los años noventa como un requerimiento de competitividad en el desarrollo de los sistemas de información. El término cliente/servidor se utiliza frecuentemente en los entornos de red.

Cliente/servidor divide el procesamiento entre:

- Un cliente, generalmente un PC con una interfaz gráfica (GUI).
- Un servidor, en donde se encuentran la base de datos y/o las aplicaciones.
- El cliente y el servidor se conectan a través de una red (el canal de comunicaciones). Lo cual implica que la comunicación a través de ésta debe garantizarse

CLIENTE / SERVIDOR

- En esta arquitectura, los Clientes y Servidores corren en Hardware y Software adecuados a sus funciones. **No obstante, debe tenerse en cuenta que cliente/servidor no es un concepto hardware sino un concepto software**, donde lo que se comunica son procesos, que bien pueden ejecutarse en diferentes máquinas o en una misma.

Aplicación



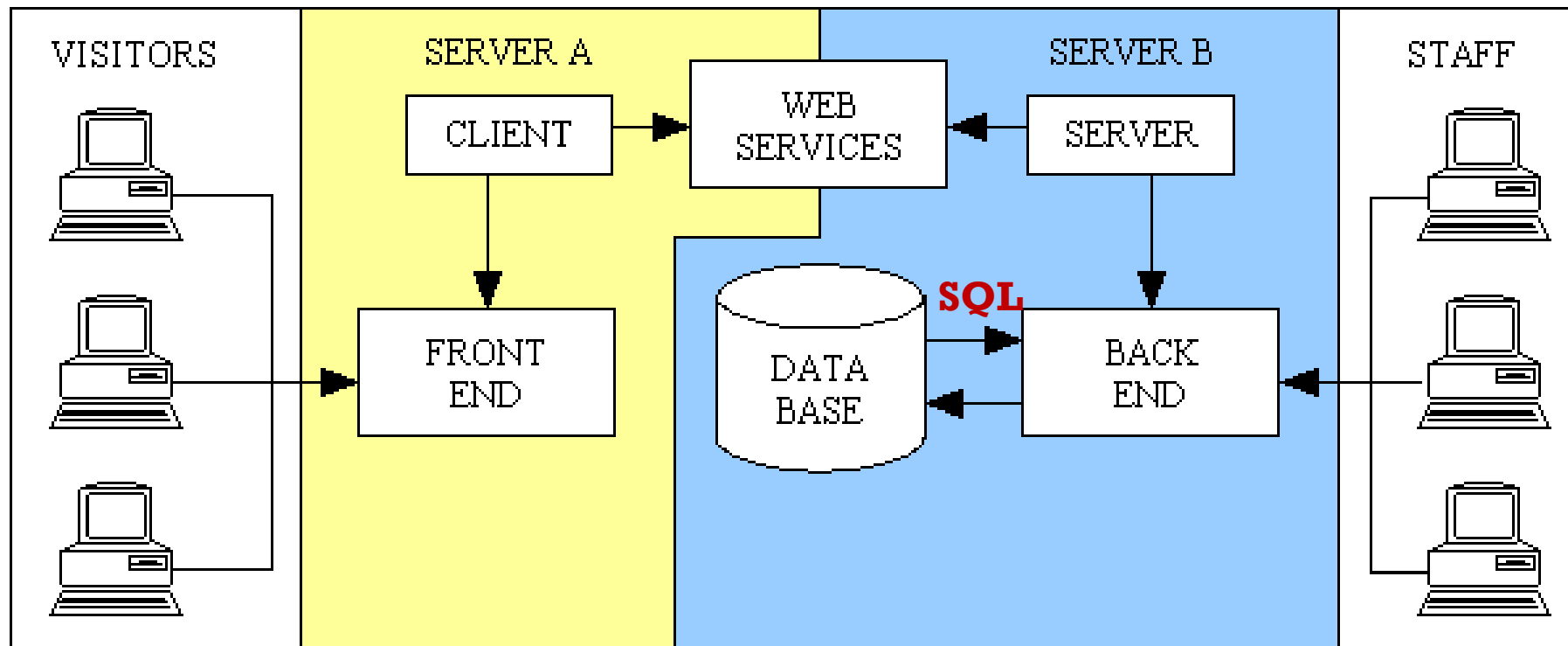
Front end vs. Back end.

■ Frontend

- Imágenes
- Lenguaje para estilos de la página CSS.
- HTML.
- Javascript, es un lenguaje que se ejecuta en el navegador web.

■ Backend

- La lógica del negocio, base de datos.
- Lenguajes más especializados, por ejemplo, C#, Python y Java. Estos se ejecutan en el servidor.
- Servidores e infraestructura que soporta la aplicación web.





Full Stack Software Engineer

Google

Technical Infrastructure

Zürich, Switzerland

[APPLY](#)

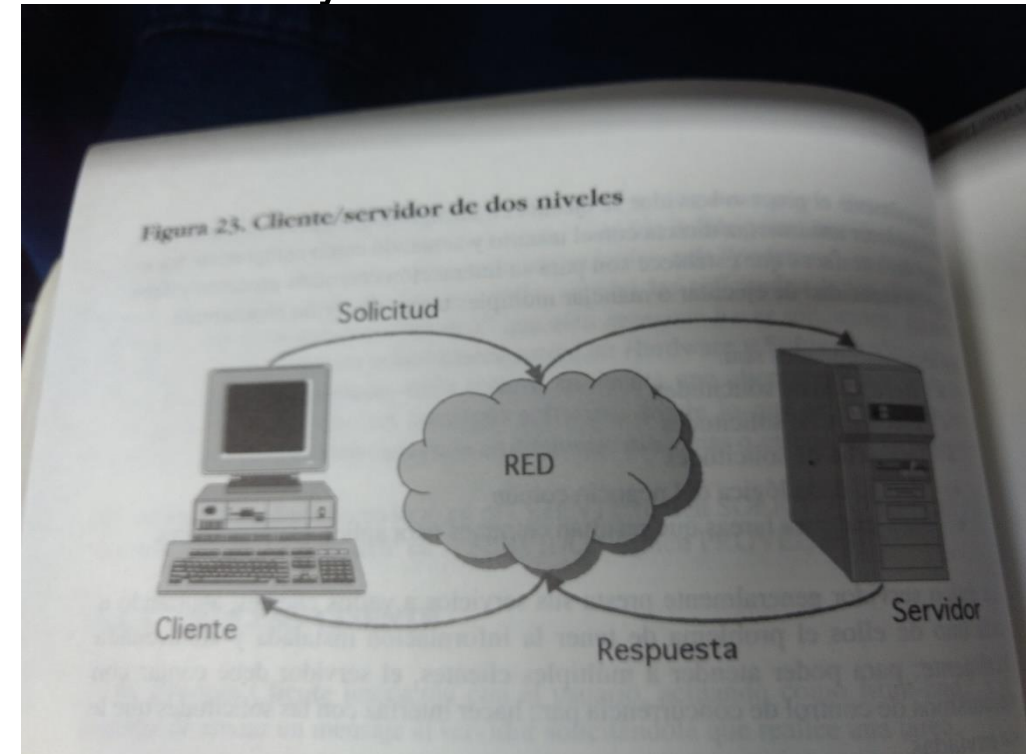
Google's software engineers develop the next-generation technologies that change how billions of users connect, explore, and interact with information and one another. Our products need to handle information at massive scale, and extend well beyond web search. We're looking for engineers who bring fresh ideas from all areas, including information retrieval, distributed computing, large-scale system design, networking and data storage, security, artificial intelligence, natural language processing, UI design and mobile; the list goes on and is growing every day. As a software engineer, you will work on a specific project critical to Google's needs with opportunities to switch teams and projects as you and our fast-paced business grow and evolve. We need our engineers to be versatile, display leadership qualities and be enthusiastic to take on new problems across the full-stack as we continue to push technology forward.

TIPOS DE ARQUITECTURAS

CLIENTE/SERVIDOR

Arquitectura de dos niveles

- Es aquella arquitectura en la cual el proceso cliente se comunica directamente con el proceso servidor y consta de dos computadores: un cliente y un servidor



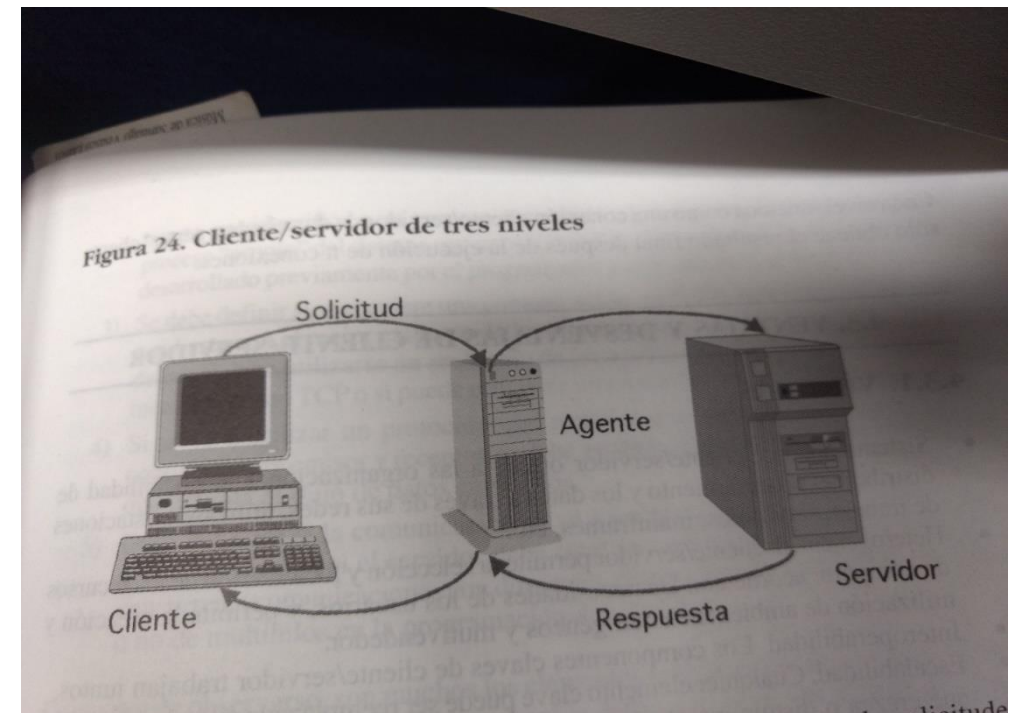
TIPOS DE ARQUITECTURAS

CLIENTE/SERVIDOR

Arquitectura de tres niveles

Esta arquitectura introduce un Agente entre el cliente y el servidor y se divide en tres capas las cuales se encuentran lógicamente separadas en entidades distintas:

- La capa de presentación
- La capa de funcionalidad
- La capa de lógica de los datos



TIPOS DE ARQUITECTURAS CLIENTE/SERVIDOR

Arquitectura de N niveles

- La interacción ya no se realiza entre un único cliente y un único servidor; en este caso, los servicios requeridos por el cliente se encuentran dispersos en varios servidores, los cuales pueden ser contactados directamente por el cliente o a través de otro servidor que funciona como intermediario.
- Cada nivel funciona como una conexión cliente/servidor de dos niveles, pero el cliente sólo obtiene el resultado final después de la ejecución de n conexiones.

PUERTO

- Número entre 0 y 65.536, que permite identificar a un proceso en un computador.
- Por ejemplo, el proceso encargado de atender las solicitudes FTP utiliza el puerto 21, mientras que el proceso telnet utiliza el puerto 23, y ambos pueden estar habilitados de forma simultánea en un mismo computador.
- La cantidad que define el total de puertos viene dada por una palabra (una cadena finita de bits manejados como un conjunto por la máquina) mapeada de un procesador de 16 bits, es decir, 2 bytes, por lo que existen 2^{16} (65536).

PUERTO

Existen dos esquemas de asignación de puertos:

- Asignación universal
- Asignación dinámica

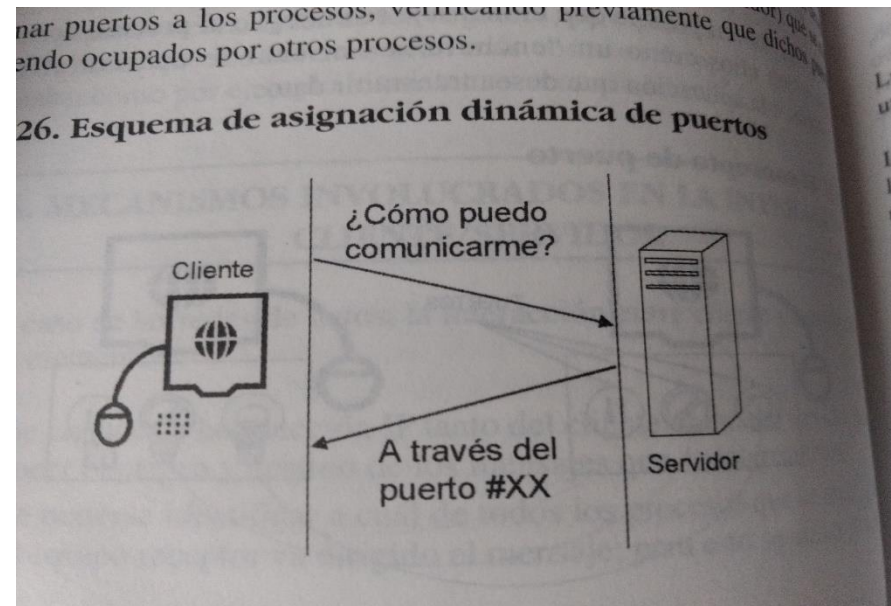
En el esquema de asignación universal, se maneja un rango de números fijos asociados a los procesos más utilizados. Por ejemplo, TCP/IP reserva los primeros 1024 puertos para sus protocolos específicos; estos puertos se conocen como “puertos bien conocidos” (well known ports) y se recomienda no utilizarlos para la ejecución de aplicaciones o procesos diferentes a los que tienen asignados.

Servicio	Puerto	Servicio	Puerto
echo	7	telnet	23
discard	9	smtp	25
daytime	13	finger	79
chargen	19	http	80
ftp	20,21	netnews	119

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

PUERTOS

- En el esquema de asignación dinámica, el emisor solicita al receptor que le asigne un número de puerto por medio del cual se pueda establecer la comunicación deseada y se utiliza una autoridad central (en el sistema operativo del servidor) que se encarga de asignar puertos a los procesos, verificando previamente que dichos puertos no se estén ocupados por otros procesos.



SOCKET

Un socket es un “conector de red”. Es el extremo de un enlace de comunicación de dos-vías entre dos programas que se ejecutan en la red. Se asocia con un número de puerto, para que el protocolo de transporte (TCP o UDP) pueda identificar la aplicación a la cual se están enviando los datos.

Una aplicación crea un socket al especificar tres elementos:

- Dirección IP del equipo.
- El puerto que la aplicación esté utilizando.
- El tipo de servicio (UDP o TCP) utilizado por la aplicación.

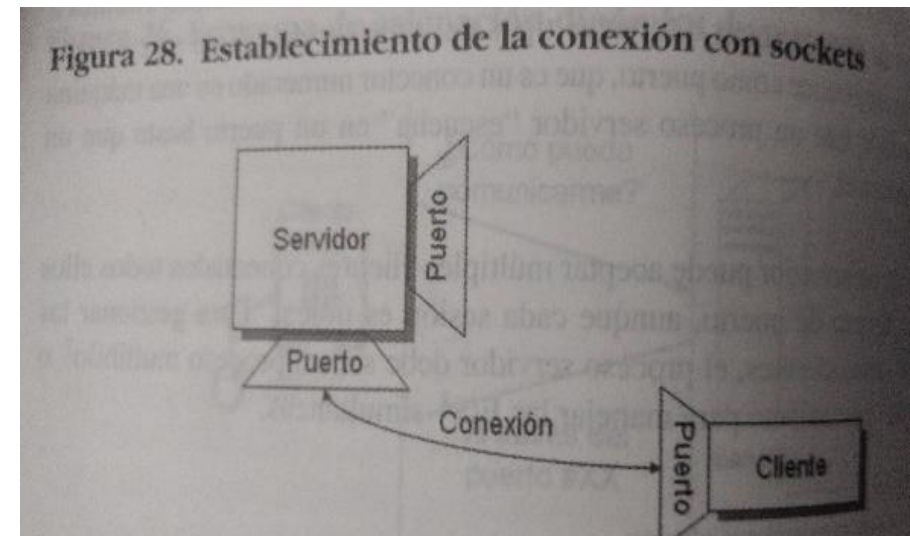
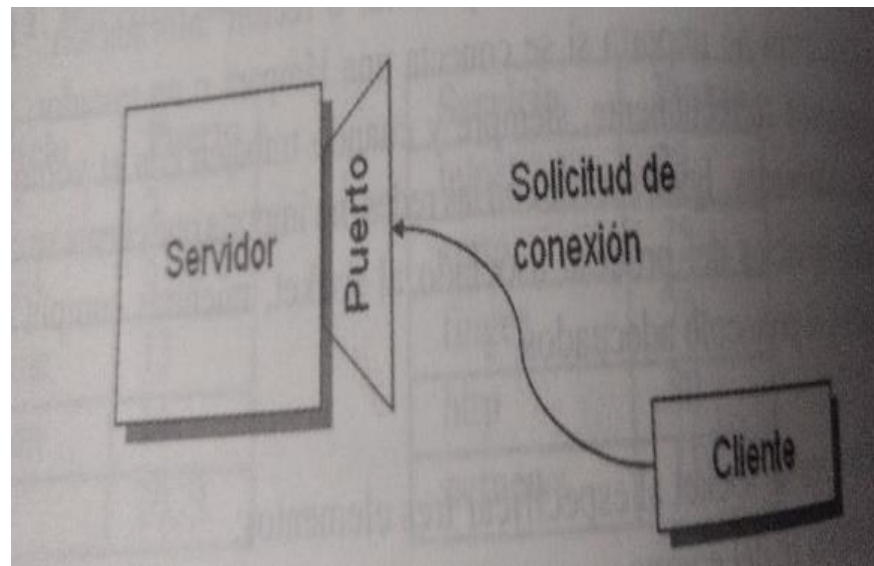
SOCKET

Normalmente, en las aplicaciones cliente/servidor se utilizan sockets a ambos lados de la comunicación:

- En el lado del servidor: el servidor se ejecuta en un computador específico y tiene un socket enlazado a un número específico de puerto. El servidor sólo espera, escuchando en el socket, hasta que el cliente realice una solicitud de conexión.
- En el lado del cliente: el cliente conoce el nombre de host de la máquina en la cual se está ejecutando el servidor, y el número de puerto en el cual se encuentra conectado el servidor. Para realizar la solicitud de conexión, el cliente trata de “reunirse” con el servidor utilizando el nombre del computador donde esté se está ejecutando y el número de puerto.

SOCKET

- Adicionalmente, un servidor puede aceptar múltiples clientes conectados todos ellos por el mismo número de puerto, aunque cada sesión es única. Para gestionar las conexiones de varios clientes, el proceso servidor debe ser un proceso multihilo o disponer de otros mecanismos para manejar las operaciones de entrada y salida simultaneas.



LECTURAS

Material utilizado	<ol style="list-style-type: none">1. Arboleda, L. (2012). Programación en Red con Java.2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.".3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación.
Actividades DESPUÉS clase	A3. Leer del libro 1 la sección 5

COMPETENCIAS, PRÓXIMA CLASE

- Explicar la arquitectura cliente/servidor
- Explicar el uso de sockets y puertos