

*SEGURIDAD
DE LOS
SISTEMAS
DISTRIBUIDOS*

Iriana Nadia Strycek

INTRODUCCIÓN

La computación desde sus inicios ha sufrido muchos cambios, desde los grandes ordenadores que permitían realizar tareas en forma limitada y de uso un tanto exclusivo de organizaciones muy selectas, hasta los actuales ordenadores ya sean personales o portátiles que tienen las mismas e incluso mayores capacidades que los primeros y que están cada vez más introducidos en el quehacer cotidiano de una persona.

Los mayores cambios se atribuyen principalmente a dos causas, que se dieron desde las décadas de los setenta:

- El desarrollo de los microprocesadores, que permitieron reducir en tamaño y costo a los ordenadores y aumentar en gran medida las capacidades de los mismos y su acceso a más personas.
- El desarrollo de las redes de área local y de las comunicaciones que permitieron conectar ordenadores con posibilidad de transferencia de datos a alta velocidad.

Es en este contexto que aparece el concepto de "Sistemas Distribuidos" que se ha popularizado tanto en la actualidad y que tiene como ámbito de estudio las redes como por ejemplo: Internet, redes de teléfonos móviles, redes corporativas, redes de empresas, etc.

SISTEMAS **DISTRIBUIDOS**

Desde el inicio de la era de la computadora moderna (1945), hasta cerca de 1985, solo se conocía la computación centralizada.

A partir de la mitad de la década de los ochentas aparecen los sistemas distribuidos, en contraste con los sistemas centralizados. Los sistemas distribuidos necesitan un software distinto al de los sistemas centralizados.

Los S. O. para sistemas distribuidos han tenido importantes desarrollos pero todavía existe un largo camino por recorrer. Los usuarios pueden acceder a una gran variedad de recursos computacionales:

- De hardware y de software.
- Distribuidos entre un gran número de sistemas computacionales conectados.

Que son los Sistemas Distribuidos?:

- Son sistemas cuyos componentes hardware y software, que están en ordenadores conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo.
- Se establece la comunicación mediante un protocolo prefijado por un esquema "cliente-servidor".

Características de los Sistemas Distribuidos:

- Concurrencia.
- Carencia de reloj global.
- Fallos independientes de los componentes.

Evolución de los Sistemas Distribuidos:

- **Procesamiento central (Host).**
- **Grupo de Servidores.**
- **La Computación Cliente – Servidor.**

Cliente-Servidor

Es el sistema donde el cliente es una máquina que solicita un determinado servicio y se denomina servidor a la máquina que lo proporciona.

Los servicios pueden ser:

- Ejecución de un determinado programa.
- Acceso a un determinado banco de información.
- Acceso a un dispositivo de hardware.

Es un elemento primordial, la presencia de un medio físico de comunicación entre las máquinas, y dependerá de la naturaleza de este medio la viabilidad del sistema.

Categorías de Servidores:

- Servidores de archivos.
- Servidores de Base de Datos.
- Servidores de Software de Grupo.
- Servidores WEB.
- Servidores de correo.
- Servidor de objetos.
- Servidores de impresión.
- Servidores de aplicación.

Componentes de Software:

Se distinguen tres componentes básicos de software:

- Presentación: Tiene que ver con la presentación al usuario de un conjunto de objetos visuales y llevar a cabo el procesamiento de los datos producidos por el mismo y los devueltos por el servidor.
- Lógica de aplicación: Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- Base de datos: Esta compuesta de los archivos que contienen los datos de la aplicación.

Arquitecturas Cliente / Servidor:

Arquitectura Cliente-Servidor de Dos Capas: Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

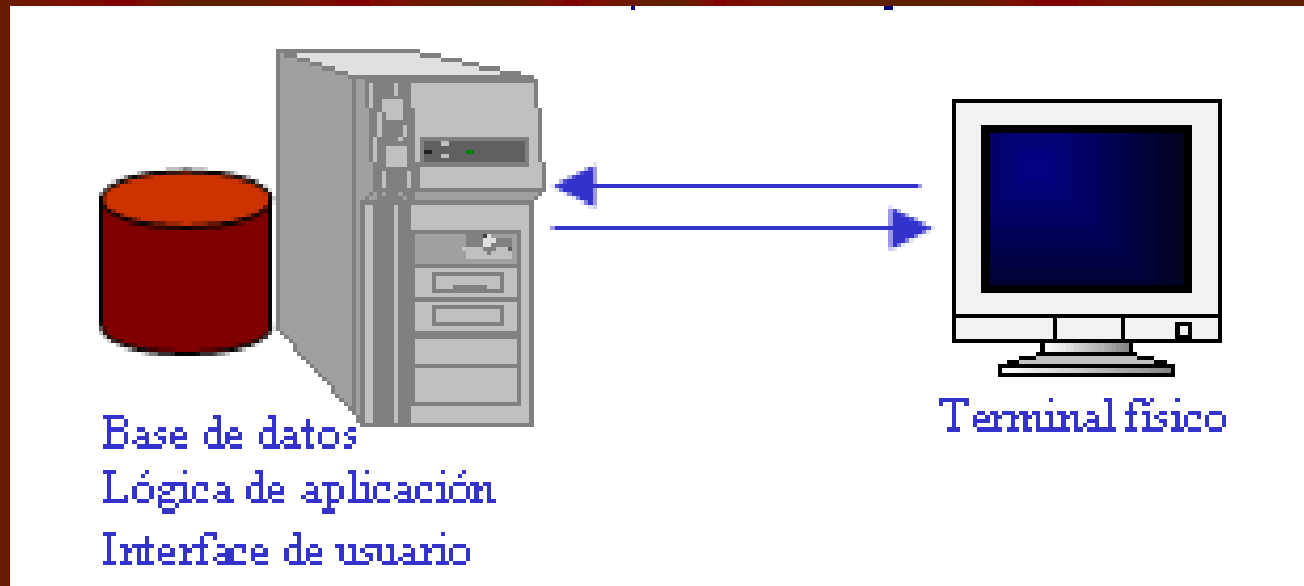
- Cuando se requiera poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

Arquitectura Cliente-Servidor de Tres Capas: Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

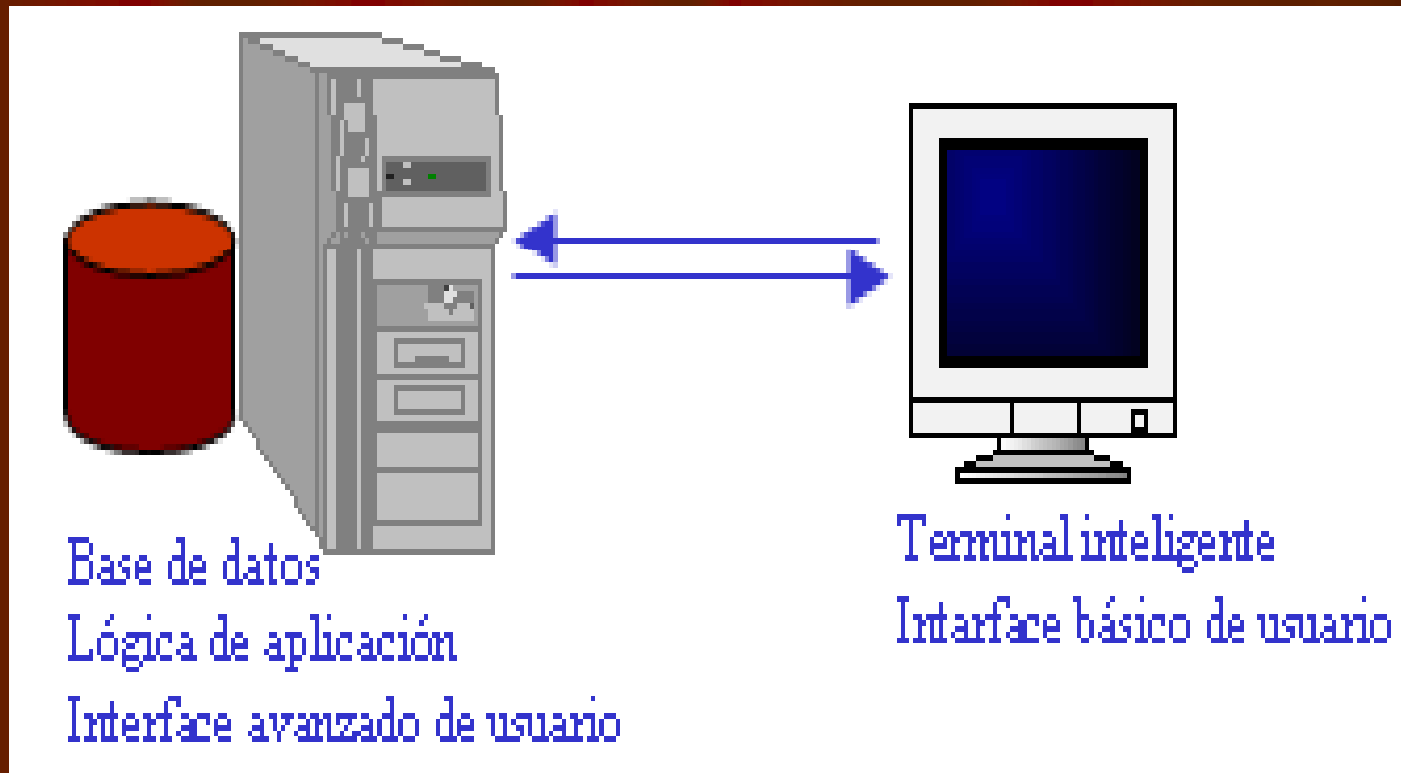
- Cuando se requiera mucho procesamiento de datos en la aplicación.
- En aplicaciones donde la funcionalidad este en constante cambio.
- Cuando los procesos no están relativamente muy relacionados con los datos.
- Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
- Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.

Clasificación de los sistemas cliente servidor:

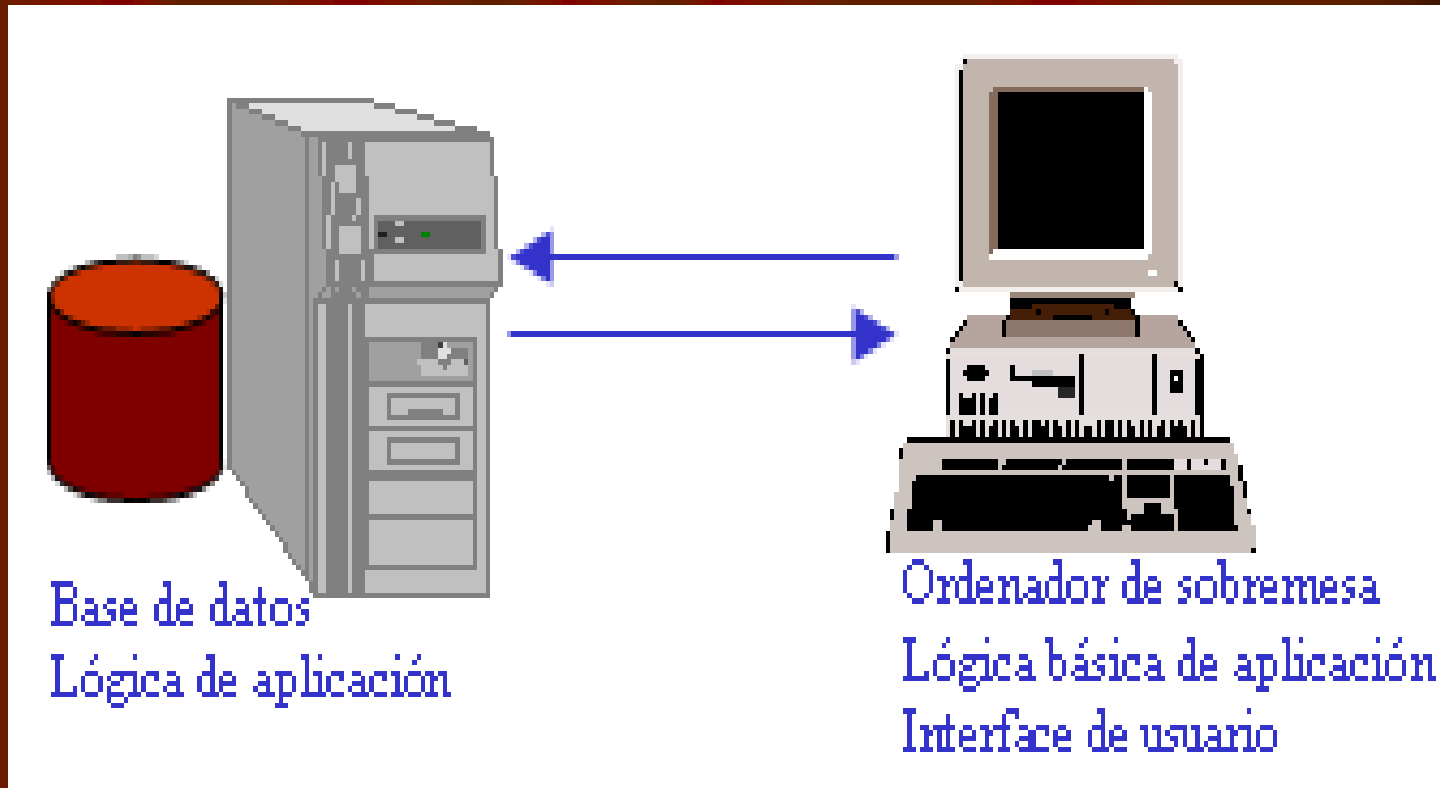
- Representación distribuida:



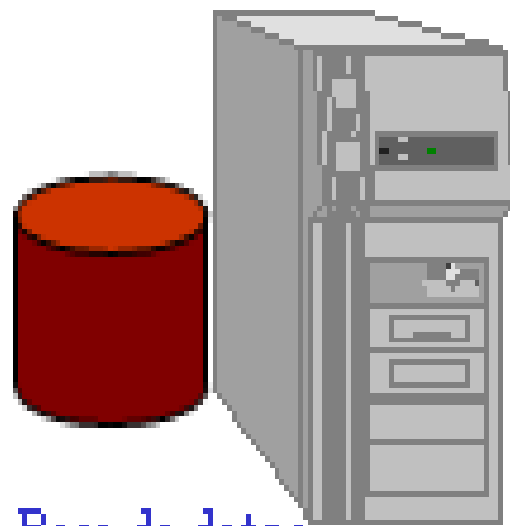
Representación Remota:



Lógica Distribuida:



Gestión Remota de Datos:



Base de datos

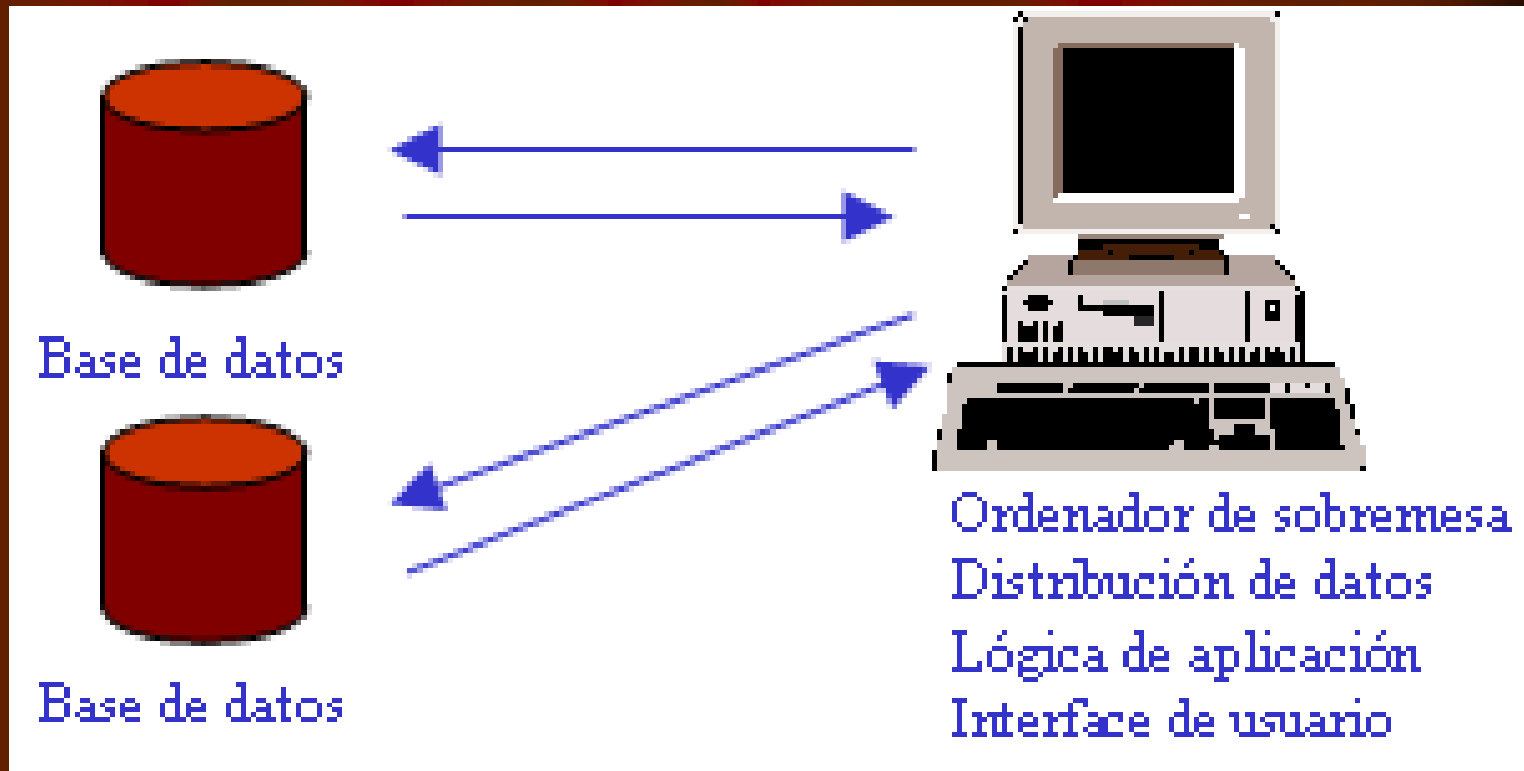


Ordenador de sobremesa

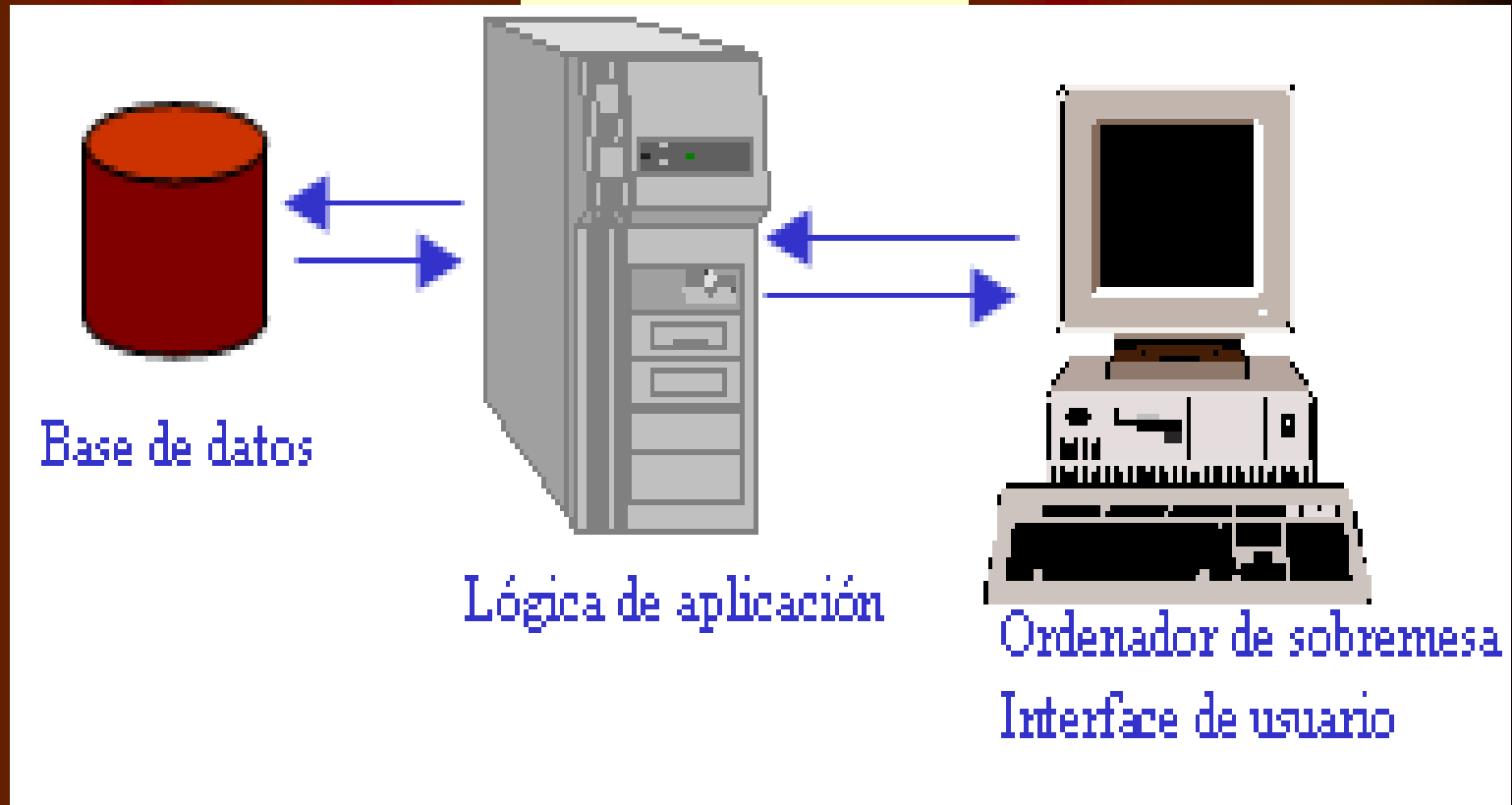
Lógica de aplicación

Interface de usuario

Base de Datos Distribuidas:



Cliente servidor a tres niveles:



Protocolo:

Es un conjunto bien conocido de reglas y formatos que se utilizan para la comunicación entre procesos que realizan una determinada tarea. Se requieren dos partes:

- Especificación de la secuencia de mensajes que se han de intercambiar.
- Especificación del formato de los datos en los mensajes.

Ejemplos de Protocolos Usados en los Sistemas Distribuidos:

- IP: Protocolo de Internet.
- TCP: Protocolo de Control de Transmisión.
- HTTP: Protocolo de Transferencia de Hipertexto.
- SMTP: Protocolo de Transferencia de Correo Simple.
- POP3: Protocolo de Oficina de Correo.

Conceptos de Hardware:

Todos los sistemas distribuidos constan de varias cpu, organizadas de diversas formas, existen diversos esquemas de clasificación para los sistemas de cómputos con varias cpu:

- Uno de los más conocidos es la "Taxonomía de Flynn": Considera como características esenciales el número de flujo de instrucciones y el número de flujos de datos.

La clasificación incluye equipos: **SISD, SIMD, MISD y MIMD.**

- **MIMD** (Multiple Instruction Multiple Data: un grupo de computadoras independientes, cada una con su propio contador del programa, programa y datos): **Todos los sistemas distribuidos son de este tipo**

Un avance sobre la clasificación de Flynn incluye la división de las computadoras MIMD en dos grupos:

- **Multiprocesadores**: poseen memoria compartida: Los distintos procesadores comparten el mismo espacio de direcciones virtuales.
- **Multicomputadoras**: no poseen memoria compartida: Ej.: grupo de PC conectadas mediante una red.

Cada una de las categorías indicadas se puede clasificar según la arquitectura de la red de interconexión en:

- **Esquema de bus:** Existe una sola red, bus, cable u otro medio que conecta todas las máquinas.
- **Esquema con conmutador:** No existe una sola columna vertebral de conexión: Hay múltiples conexiones y varios patrones de conexionado.

Otro aspecto de la clasificación considera el acoplamiento entre los equipos:

- **Sistemas fuertemente acoplados:** El retraso al enviar un mensaje de una computadora a otra es corto y la tasa de transmisión es alta. Generalmente se los utiliza como sistemas paralelos.
- **Sistemas débilmente acoplados:** El retraso de los mensajes entre las máquinas es grande y la tasa de transmisión es baja. Generalmente se los utiliza como sistemas distribuidos.

“Generalmente los multiprocesadores están más fuertemente acoplados que las multicomputadoras”.

Conceptos de los Sistemas Distribuidos:

- Compatición de Recursos: La idea de compartición de recursos no es nueva ni aparece en el marco de los sistemas distribuidos. Los sistemas multiusuario clásicos desde siempre han provisto compartición de recursos entre sus usuarios. Sin embargo, los recursos de una computadora multiusuario se comparten de manera natural entre todos sus usuarios. Por el contrario, los usuarios de estaciones de trabajo monousuario o computadoras personales dentro de un sistema distribuido no obtienen automáticamente los beneficios de la compartición de recursos.
- Apertura (openness): La apertura de los sistemas distribuidos se determina primariamente por el grado hacia el que nuevos servicios de compartición de recursos se pueden añadir sin perjudicar ni duplicar a los ya existentes.
- Concurrencia: Cuando existen varios procesos en una única maquina decimos que se están ejecutando concurrentemente. Si el ordenador está equipado con un único procesador central, la concurrencia tiene lugar entrelazando la ejecución de los distintos procesos. Si la computadora tiene N procesadores, entonces se pueden estar ejecutando estrictamente a la vez hasta N procesos.

- **Escalabilidad:** La demanda de escalabilidad en los sistemas distribuidos ha conducido a una filosofía de diseño en que cualquier recurso simple -hardware o software- puede extenderse para proporcionar servicio a tantos usuarios como se quiera. Esto es, si la demanda de un recurso crece, debería ser posible extender el sistema para darle servicio.
- **Tolerancia a Fallos:** El diseño de sistemas tolerantes a fallos se basa en dos cuestiones, complementarias entre sí: Redundancia hardware (uso de componentes redundantes) y recuperación del software (diseño de programas que sean capaces de recuperarse de los fallos).
- **Transparencia:** La transparencia se define como la ocultación al usuario y al programador de aplicaciones de la separación de los componentes de un sistema distribuido, de manera que el sistema se percibe como un todo, en vez de una colección de componentes independientes. La transparencia ejerce una gran influencia en el diseño del software de sistema.

Middleware:

El software distribuido requerido para facilitar las interacciones cliente-servidor se denomina middleware. El acceso transparente a servicios y recursos no locales distribuidos a través de una red se provee a través del middleware, que sirve como marco para la comunicación entre las porciones cliente y servidor de un sistema.

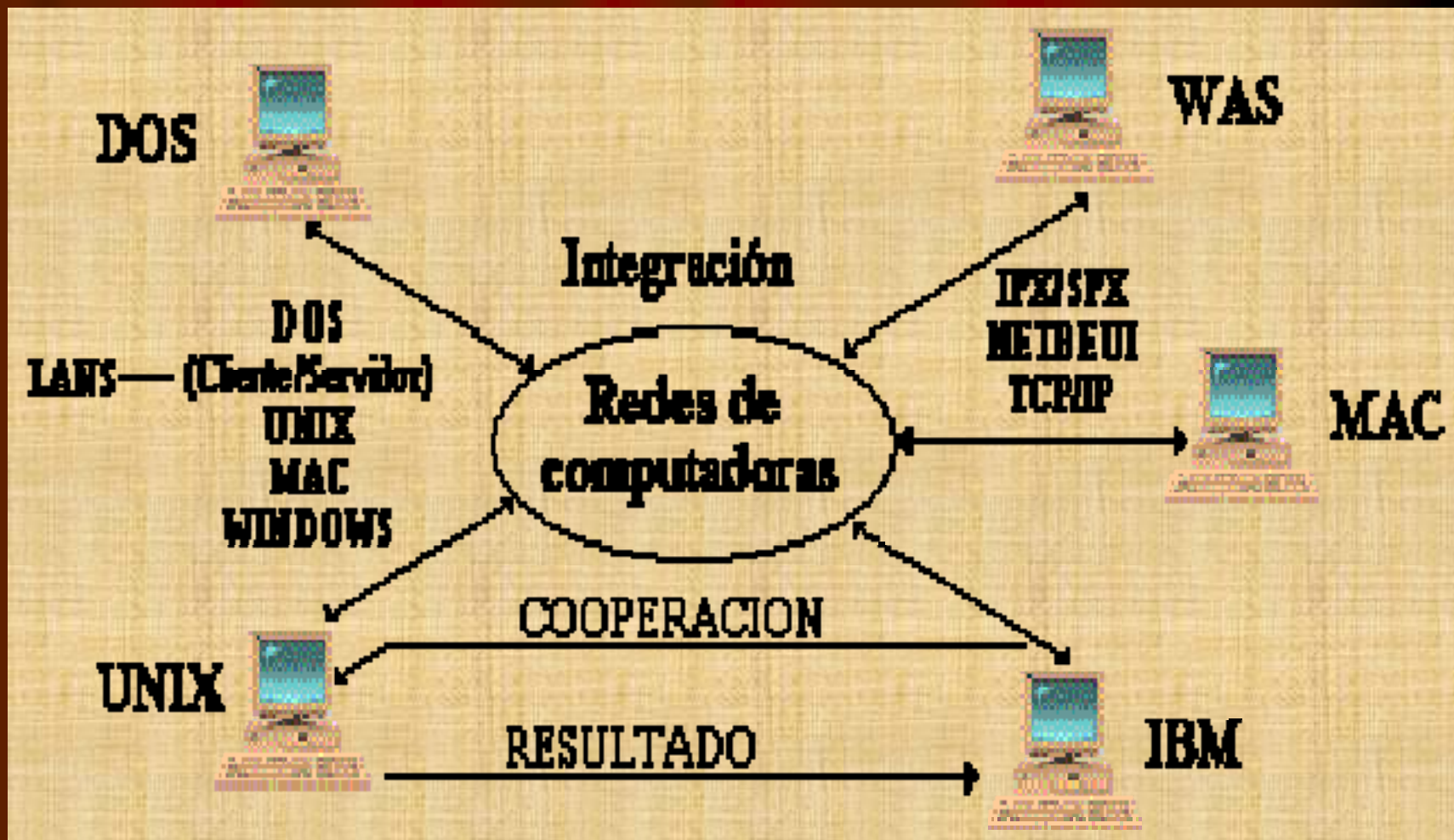
El middleware define: el API que usan los clientes para pedir un servicio a un servidor, la transmisión física de la petición vía red, y la devolución de resultados desde el servidor al cliente.

El middleware fundamental o genérico es la base de los sistemas cliente-servidor.

El protocolo de comunicaciones más usado por el middleware, tanto genérico como específico, es TCP/IP.

Factores que Han Afectado el Desarrollo de los Sistemas Distribuidos:

- Avances Tecnológicos.
- Nuevos requerimientos.
- Globalización.
- Aspectos Externos (Culturales, Políticos, Económicos).
- Integración.



Ventajas y Desventajas de los Sistemas Distribuidos

Ventajas:

- Procesadores más poderosos y a menos costos.
- Desarrollo de Estaciones con más capacidades.
- Avances en la Tecnología de Comunicaciones.
- Disponibilidad de elementos de Comunicación.
- Comparición de Recursos.
- Eficiencia y Flexibilidad.
- Respuesta Rápida.
- Ejecución Concurrente de procesos (En varias computadoras).
- Disponibilidad y Confiabilidad.
- Sistema poco propenso a fallas (Si un componente falla no afecta a la disponibilidad del sistema).
- Mayores servicios que elevan la funcionalidad (Monitoreo, Telecontrol, Correo Eléctrico, Etc.).
- Inclusión rápida de nuevos recursos.

Desventajas:

- **Requerimientos de mayores controles de procesamiento.**
- **Velocidad de propagación de información (Muy lenta a veces).**
- **Servicios de replicación de datos y servicios con posibilidades de fallas.**
- **Mayores controles de acceso y proceso (Commit).**
- **Administración más compleja.**
- **Costos.**

Evolución de las Necesidades de Seguridad:

	1965-75	1975-89	1990-99	Actualmente
<i>Plataformas</i>	Computadores multiusuario de tiempo compartido	Sistemas distribuidos basados en redes locales	Internet, servicios de área extensa	Internet + dispositivos móviles
<i>Recursos compartidos</i>	Memoria, archivos	Servicios locales (p. ej.: NFS), redes locales	e-mail, lugares web, comercio Internet	Objetos distribuidos, código móvil
<i>Requisitos de seguridad</i>	Identificación y autenticación de usuario	Protección de servicios	Seguridad robusta para transacciones comerciales	Control de acceso para objetos individuales, código móvil seguro
<i>Entorno de gestión de la seguridad</i>	Autoridad única, base de datos de autorización única (p. ej.: etc/passwd)	Autoridad única, delegación, bases de datos de autorización replicadas (p. e.): NIS)	Muchas autoridades, sin autoridad en la red, en general	Autoridades por actividad, grupos con responsabilidades compartidas

La Emergencia de la Criptografía *en el Dominio Público:*

- La criptografía proporciona la base para la mayoría de los sistemas de seguridad de los computadores. Hoy en día es un tema de investigación abierta y con una comunidad de investigadores amplia y muy activa. La criptografía de clave pública es fruto de esta apertura.
- Un ejemplo, el algoritmo de encriptación estándar DES fue inicialmente un secreto militar. Su eventual publicación y los esfuerzos exitosos para romperlo han traído consigo el desarrollo de algoritmos de encriptación de clave secreta mucho más resistentes.
- Otro producto secundario útil ha sido el desarrollo de una terminología y aproximación común. Un ejemplo de esto último es la adopción de un conjunto de nombres familiares para los protagonistas (principales) involucrados en las transacciones que hay que asegurar.

- Alice Primer participante.
- Bob Segundo participante.
- Carol Otro participante en los protocolos a tres o cuatro bandas.
- Dave Participante en protocolos a cuatro bandas.
- Eve Fisgón.
- Mallory Atacante malevolente.
- Sara Un servidor.

***Nombres familiares (del mundo anglosajón)
para los protagonistas de los protocolos de
seguridad.***

Amenazas y Ataques

Las amenazas de seguridad se dividen en tres clases:

- Fuga: la adquisición de información por receptores no autorizados.
- Alteración: la modificación no autorizada de información.
- Vandalismo: interferencia en el modo de operación adecuado de un sistema, sin ganancia para el responsable.

Los métodos de ataque pueden clasificarse en función del modo en que se abusa del canal:

- Fisgar: obtener copias sin autorización.
- Suplantar: enviar o recibir mensajes utilizando la identidad de otro sin su autorización.
- Alterar mensajes: interceptar mensajes y alterar sus contenidos antes de pasarlos al receptor.
- Reenviar: almacenar mensajes interceptados y enviarlos más tarde.
- Denegación de servicio: desbordar un canal o recurso para impedir que otros accedan a él.

Diseño de Sistemas Seguros:

Debemos diferenciar las tareas específicas de un diseñador de sistemas seguros y de un programador. El objetivo del diseñador es excluir todos los posibles ataques y agujeros. La situación es análoga a la del programador cuyo principal objetivo es excluir todos los errores de su programa.

Para demostrar la validez de los mecanismos de seguridad, empleados en un sistema, los diseñadores deben construir, en primer lugar, una lista de amenazas y probar que cada una de ellas se puede prevenir mediante los mecanismos empleados como por ej. Un histórico de seguridad contendrá una secuencia de registros fechados de las acciones de los usuarios. Como mínimo, los registros incluirán la identidad del principal, la operación realizada), la identidad del objeto sobre el que se opera y la fecha y hora.

El diseño de sistemas seguros es un ejercicio de balance entre los costos y las amenazas ya que:

- Su uso acarrea un costo (en esfuerzo computacional y uso de la red). Los costos deben compensar la amenaza.
- Unas especificaciones de medidas de seguridad inapropiadas podrían impedir a los usuarios legítimos el realizar ciertas acciones necesarias.

Criptografía:

La encriptación es el proceso de codificación de un mensaje de forma que queden ocultos sus contenidos.

La criptografía moderna incluye algunos algoritmos seguros de encriptación y desencriptación de mensajes.

Todos ellos se basan en el uso de ciertos secretos llamados claves.

Una clave criptográfica es un parámetro empleado en un algoritmo de encriptación de manera que no sea reversible sin el conocimiento de una clave.

Hay dos clases principales de algoritmos de encriptación de uso general:

- *La primera emplea claves secretas compartidas.*
- *La segunda emplea pares de claves pública / privada.*

Notación Criptográfica

- KA: Clave secreta de Alice.
- KB: Clave secreta de Bob.
- KAB: Clave secreta compartida por Alice y Bob.
- KApriv: Clave privada de Alice (solo conocida por Alice).
- KApub: Clave pública de Alice (publicada por Alice para lectura de cualquiera).
- $\{M\}_k$: Mensaje M encriptado con la clave K.
- $\{M\}^k$: Mensaje M firmado con la clave K.

Usos de la Criptografía

La criptografía juega tres papeles principales en la implementación de los sistemas seguros:

- Secreto e integridad: se emplea para mantener el secreto y la integridad de la información dondequiera que pueda estar expuesta a ataques potenciales.

- Autenticación: La criptografía se emplea como base para los mecanismos para autenticar la comunicación entre pares de principales.

Un principal que descifra un mensaje con éxito empleando una clave particular puede presuponer que el mensaje es auténtico si contiene una suma de chequeo correcta o, si se emplea el modo de encriptación de encadenamiento de bloques.

- Firmas digitales: Ésta emula el papel de las firmas convencionales, verificando a una tercera parte que un mensaje o un documento es una copia inalterada producida por el firmante.

Otros usos

- Certificados: es un documento que contiene una sentencia (generalmente corta) firmada por un principal. Estos pueden emplearse para establecer la autenticidad de muchos tipos de enunciados.
- Control de Acceso: Los servidores reciben mensajes con peticiones de la forma <op, principal, recurso>, donde op es la operación solicitada, principal es una identidad o un conjunto de credenciales del principal que realiza la petición y recurso identifica el recurso sobre el que se aplica la operación. El servidor debe, en primer lugar, comprobar la autenticidad del mensaje de petición y las credenciales del principal y después aplicar el control de acceso, rehusando cualquier petición para la cual el principal solicitante no tenga los derechos de acceso pertinentes para realizar la operación requerida sobre el recurso especificado.

- **Credenciales:** Las credenciales son un conjunto de evidencias presentadas por un principal cuando pide acceso a un recurso.
- **Cortafuegos:** Con ellos se protege una intranet, se realizan acciones de filtrado en las comunicaciones entrantes y salientes.

Los cortafuegos producen un entorno de comunicación local en el que se intercepta toda comunicación externa. Los mensajes se reenvían al recipiente local final sólo para las comunicaciones que estén autorizadas explícitamente.

Los cortafuegos no son particularmente útiles contra ataques de denegación de servicios, basado en la suplantación de direcciones IP.

Algoritmos de Clave Secreta *(Simétricos):*

- TEA: *Tiny Encryption Algorithm* (TEA, pequeño algoritmo de encriptación), se ha escogido por la simplicidad de su diseño e implementación:
 - Emplea vueltas de sumas enteras, XOR y desplazamientos lógicos de bits, para obtener la difusión y confusión de los patrones de bits en el texto en claro.

Función de encriptación de TEA:

```
Void encripta (unsigned long k [ ], unsigned long
  texto [ ]){
    unsigned long y = texto [0], z = texto [1];
    unsigned long delta = 0x 9e3779v9, suma = 0;
    int n;
    for ( n = 0 ; n < 32; n ++){
    suma + = delta;
    y + = (( z << 4) + k [0]) ^ ( z + suma) ^ (( z >>
    5) + ( k [1]));
    z + = (( y << 4) + k [2]) ^ ( y + suma ) ^ (( y >>
    5 ) + ( k [3]));
    }
    texto [0] = y; texto [1] = z;
}
```

Función de descriptación de TEA:

```
Void encripta (unsigned long k [ ], unsigned long
  texto [ ]){
    unsigned long y = texto [0], z = texto [1];
    unsigned long delta = 0x 9e3779v9, suma = 0;
    int n;
    for ( n = 0 ; n < 32; n ++){
    suma + = delta;
    y + = (( z << 4) + k [2]) ^ ( z + suma) ^ (( z >>
    5) + ( k [3]));
    z + = (( y << 4) + k [0]) ^ ( y + suma ) ^ (( y >>
    5) + ( k [1]));
    }
    texto [0] = y; texto [1] = z;
}
```

- **DES**: El Estándar de Encriptación de Datos (*Data Encryption Standard*), usado para aplicaciones gubernamentales y de negocios. Ha sido un estándar nacional de los EE.UU. aunque su interés es histórico dado que sus claves de 56 bits son demasiado reducidas para resistir un ataque por fuerza bruta con el hardware actual.
- **IDEA**: El Algoritmo de Encriptación de Datos Internacional (*International Data Encryption Algorithm*, IDEA) se desarrolló a comienzos de los años noventa como sucesor de DES. Emplea una clave de 128 bits y es, probablemente, el algoritmo de encriptación simétrico de bloques más efectivo.

Algoritmos de Clave Pública (Asimétricos):

Hasta la fecha sólo se han desarrollado unos pocos esquemas prácticos de clave pública.

Un principal que desee participar en una comunicación segura con otros confecciona un par de claves K_e , y K_d y guarda en secreto la clave de descryptación K_d . La clave de encriptación K_e puede publicarse para cualquiera que desee comunicar.

$$D (K_e E (K_e, M)) = M$$

El algoritmo RSA es ciertamente el algoritmo de clave pública más conocido que lo describiremos a continuación:

RSA: (*Rivest, Shamir y Adelman*) el diseño para el encriptador de clave pública, se basa en el uso del producto de dos números primos muy grandes, la determinación de los factores primos de números tan grandes es computacionalmente imposible de calcular.

Bosquejo del método RSA:

Para encontrar las claves e, d :

1. Elíjanse dos números primos grandes, P y Q (mayores que 10^{100}), y fórmese

$$N = P \times Q$$

$$Z = (P - 1) \times (Q - 1)$$

2. Para d elíjase cualquier número primo con relación a Z (esto es, que d no tenga factores en común con Z).

Ilústreme los cálculos correspondientes empleando valores pequeños de P y Q

$$P = 13 . Q = 17 \rightarrow N = 221 . Z = 192$$

$$d = 5$$

3. Para encontrar e resuélvase la ecuación: $e \times d = 1 \pmod{Z}$

Esto es, $e \times d$ es el número más pequeño divisible por d en la serie $Z + 1, 2Z + 1, 3Z + 1,$

Para encriptar el texto en claro se divide en bloques iguales de longitud k bits donde $2^k < N$ (el valor numérico de un bloque es siempre menor que N ; en aplicaciones prácticas, k está generalmente en el rango de 512 a 1.024).

La función de encriptación de un bloque de texto en claro M es:

$$E'(e, N, M) = M^e \pmod{N}$$

Para desencriptar un bloque de texto encriptado c para producir el bloque de texto en claro original es:

$$D'(d, N, c) = c^d \pmod{N}$$

Se probó que E' y D' son inversas mutuas (esto es, que $E'(D'(x)) = D'(E'(x)) = x$) para cualquier valor P en el rango $0 \leq P \leq N$.

● Algoritmos de Curvas Elípticas:

- Un algoritmo puede generar pares de claves pública / privada basándose en las propiedades de las curvas elípticas.
- Las claves de derivan de una rama diferente de las matemáticas, y a diferencia de RSA su seguridad no depende de la dificultad de la factorización de números grandes.
- Las claves cortas son seguras, y los requisitos de procesamiento para la encriptación y la descryptación son menores.

Firmas Digitales:

Una firma digital robusta es un requisito esencial para los sistemas seguros.

Las firmas manuscritas necesitan verificar que éste es:

- **Auténtico:** convence al receptor de que el firmante firmó deliberadamente el documento y que la firma no ha sido alterado por nadie.
- **Infalsificable:** aporta la prueba de que el firmante firmó el documento.
- **No repudiable:** el firmante no puede negar de forma creíble que el documento fue por él.

- *Firmas Digitales Con Claves Públicas:*

- La criptografía de clave pública se adapta particularmente bien a la generación de firmas digitales dado que es relativamente simple y no requiere ninguna comunicación entre el destinatario de un documento firmado y el firmante.

- *Firmas Digitales Con Claves Secretas, Mac:*

- No hay ninguna razón técnica por la que un algoritmo de encriptación de clave secreta no pueda usarse para encriptar una firma digital, pero existen algunos problemas:
 1. El firmante debe conseguir que el verificador reciba la clave secreta empleada para firmar de modo seguro.
 2. Debe ser necesario poder verificar una firma en varios contextos diferentes y en momentos diferentes.
 3. El descubrimiento de una clave secreta empleada para una firma es poco deseable puesto que debilita la seguridad de las firmas realizadas con ellas, podrían falsificarse una firma por alguien que posea la clave y que no sea su propietario.

Existe una excepción cuando se utiliza un canal seguro para transmitir los mensajes descriptados pero subsiste la necesidad de verificar la autenticidad de los mensajes. Estas firmas se denominan **códigos de autenticidad de mensajes (MAC)** para reflejar su objetivo más limitado: autentican la comunicación entre pares de principales basándose en un secreto compartido.

Kerberos:

Kerberos es una forma eliminar la necesidad de aquellos protocolos que permiten métodos de autenticación inseguros, y de esta forma mejorar la seguridad general de la red.

Kerberos es un protocolo de seguridad creado por MIT que usa una criptografía de claves simétricas para validar usuarios con los servicios de red evitando así tener que enviar contraseñas a través de la red:

Al validar los usuarios para los servicios de la red por medio de Kerberos, se frustran los intentos de usuarios no autorizados que intentan interceptar contraseñas en la red.

Conclusiones:

Los ataques a la seguridad son partes de la realidad de los sistemas distribuidos. Es esencial proteger los canales e interfaces de comunicaciones de cualquier sistema que trate con información que sea susceptible de ser atacada.

- Los sistemas distribuidos abarcan una cantidad de aspectos considerables, por lo cual su desarrollo implica mucha complejidad.
- Existen ciertos aspectos que requieren extremo cuidado al desarrollarse e implantarse como el manejo de fallos, el control de la concurrencia, etc.
- Existen muchos temas de investigación relacionados con los sistemas distribuidos.
- Se nota también que muchas tecnologías están en constante desarrollo y maduración, lo cual implica un minucioso estudio previo de muchos factores antes de apostar por alguna tecnología en especial.

MUCHAS GRACIAS

