

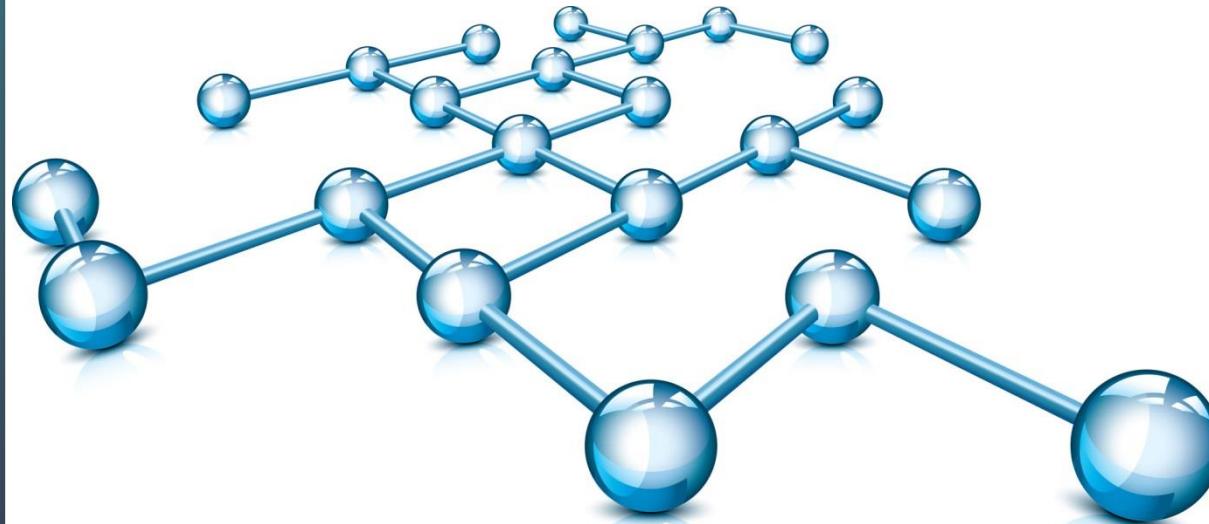


Sistemas distribuidos

Profesor:

Dr. J. Octavio Gutiérrez García

octavio.gutierrez@itam.mx



Objetivos



- Diseñar e **implementar** sistemas distribuidos.
- Entender y **manejar las tecnologías** existentes para implementar sistemas distribuidos.
- Implementar los **métodos de comunicación** entre procesos distribuidos.

Objetivos



- Desarrollar **aplicaciones web**.
- Desarrollar **servicios web**.
- Implementar mecanismos de **conurrencia** entre procesos distribuidos.



Temario

- **Introducción a los sistemas distribuidos**
- **Comunicación entre procesos**
 - Sockets TCP y UDP
 - Serialización de Java, XML, JSON.
- **Invocación remota**
 - RPC
 - RMI
 - Protocolos request-reply

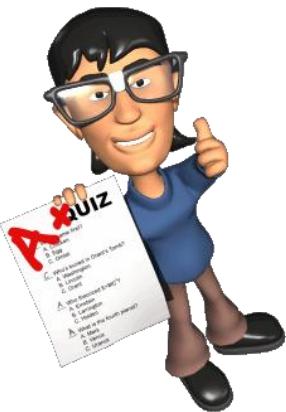


- **Modelos de comunicación indirecta**
 - Servicios de Topics / Queues
 - Java Message Service
- **Sistemas de objetos/componentes distribuidos**
 - Enterprise JavaBeans



Temario

- **Programación Web**
 - HTML5 y CSS (programación web responsiva)
 - JavaScript y comunicación asíncrona
 - Bootstrap
 - Django
- **Servicios web RESTful & SOAP**

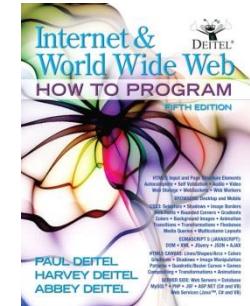
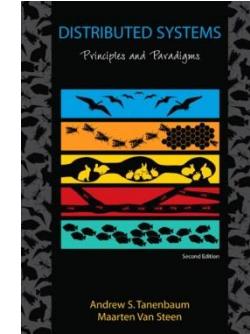
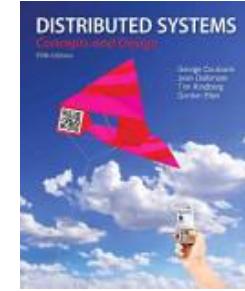


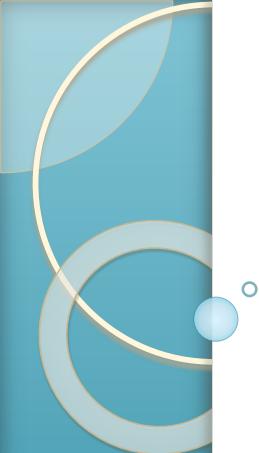
Evaluación

- Ejercicios selectos en clase **5%**
- Examen Parcial
Miércoles 10 de Marzo de 2021
- Proyecto Alpha **20%**
Miércoles 17 de Marzo de 2021
- Proyecto Omega **20%**
Miércoles 12 de Mayo de 2021
- Investigación y mini-exposición en equipo Mayo **10%**
- Examen Final **25%**
A definirse por control escolar

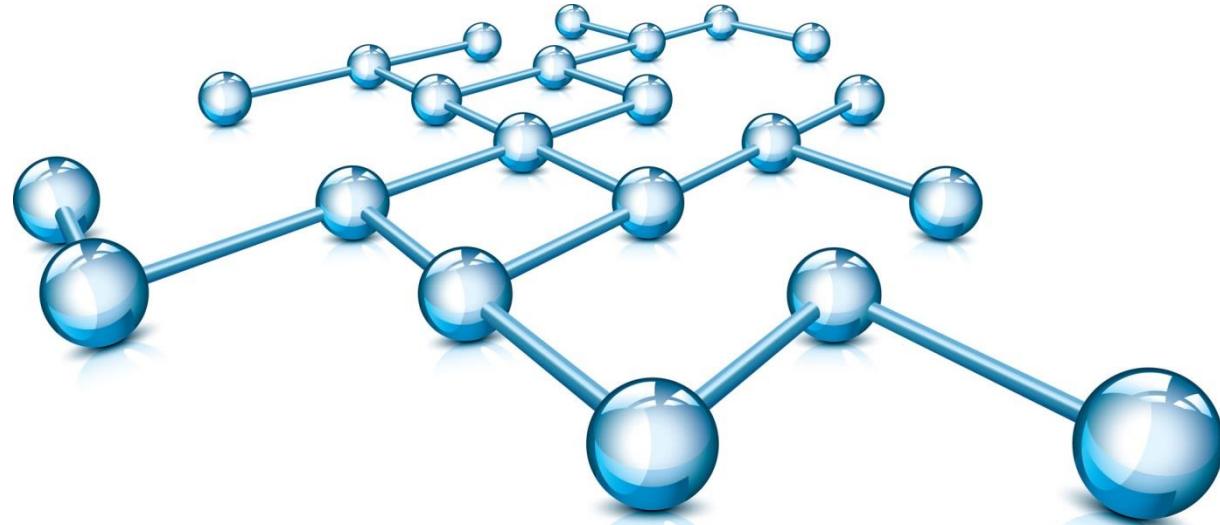
Bibliografía

- Colouris G., Dollimore J., & KindBerg T. (2012). *Distributed Systems Concepts and Designs*. Quinta edición, Addison Wesley.
- Tanenbaum, A., Van Steen M., *Distributed Systems-principles and paradigms*. Segunda edición, Ed. Pearson-Prentice Hall, 2007.
- H.M. Deitel, P.J. Deitel, *Internet & World Wide Web How to Program*. Quinta edición, Ed. Pearson-Prentice Hall, 2012.





Introducción a los sistemas distribuidos



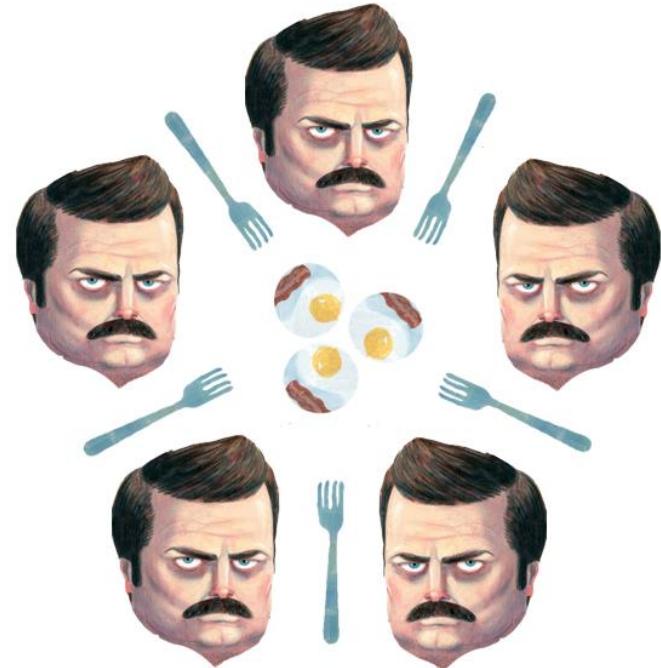
¿Qué es un Sistema Distribuido (SD)?

- Sistema en el cual los componentes hardware o software, localizados en computadoras unidos mediante red, comunican y coordinan sus acciones sólo mediante paso de mensajes.



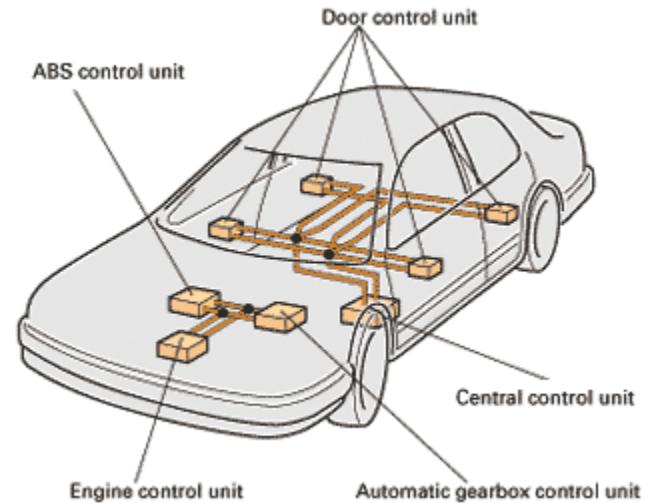
Características de los SDs

- Concurrencia de los componentes
- No hay un reloj global
- Fallos independientes de los componentes.



Ejemplos de SDs

- Internet
- Redes corporativas
- Redes de teléfonos móviles
- Redes de universidades, casas, etc.
- Redes dentro del coche



Ejemplos de SDs

- Búsqueda web



- Centros de datos distribuidos globalmente
- Sistemas de archivos distribuidos
- Estructura distribuida de almacenamiento.
- Sistemas de candados distribuidos
- Modelo de programación (MapReduce)

Ejemplos de SDs

- Juegos online multi-jugador
 - Fortnite



- Arquitecturas distribuidas asignando los usuarios a servidores “cercanos”.
- En investigación ->Punto-a-Punto

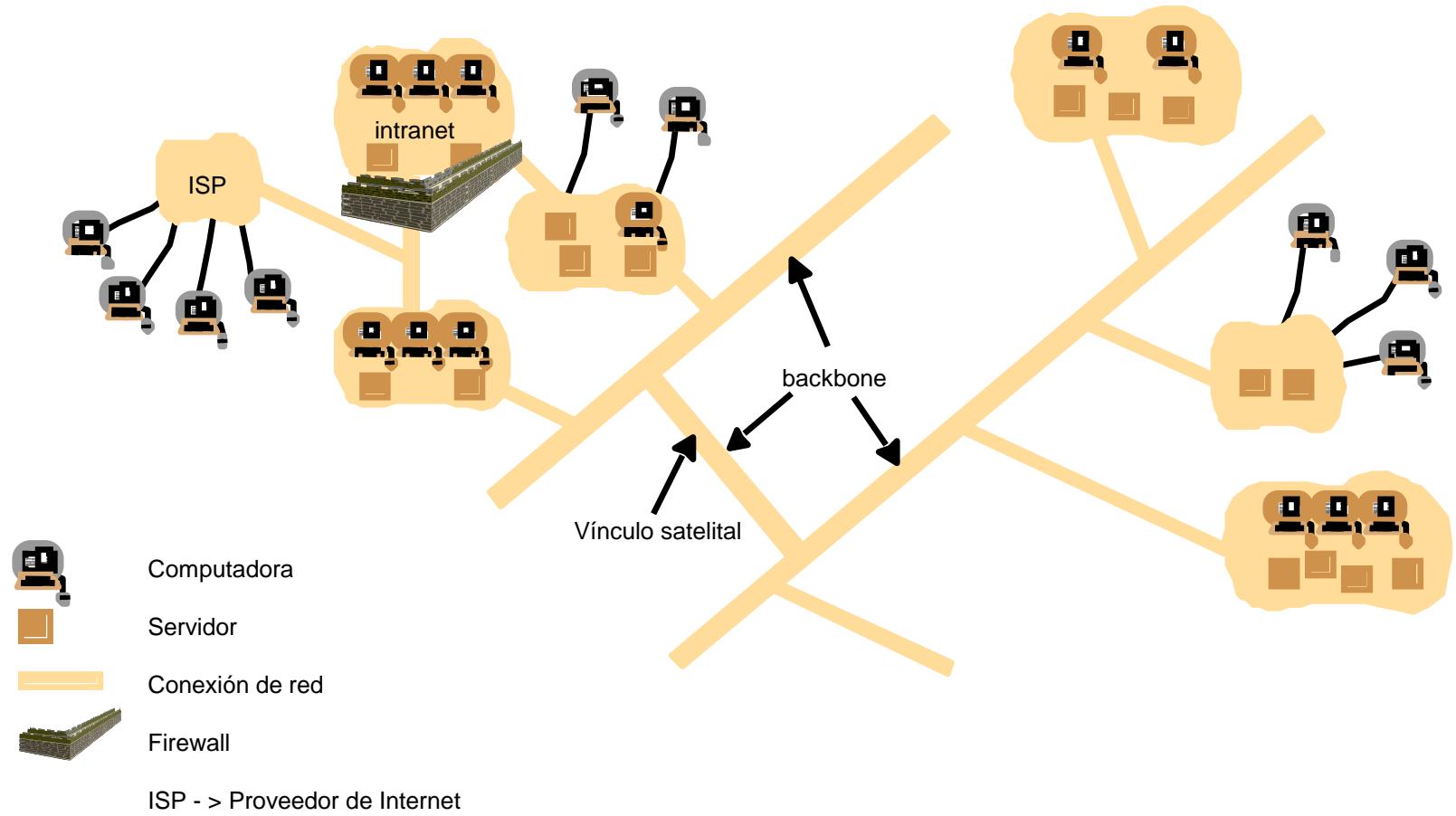
Tendencias en los SDs

- La aparición de la tecnología de **redes omnipresentes**
- Surgimiento de la **computación ubicua**
- **Movilidad del usuario** en los sistemas distribuidos
- Creciente demanda de servicios **multimedia**
- Vista de los SDs como un **servicio básico** más



Redes omnipresentes y la internet moderna

- WiFi, WiMAX, Bluetooth y redes telefónicas inalámbricas nG.



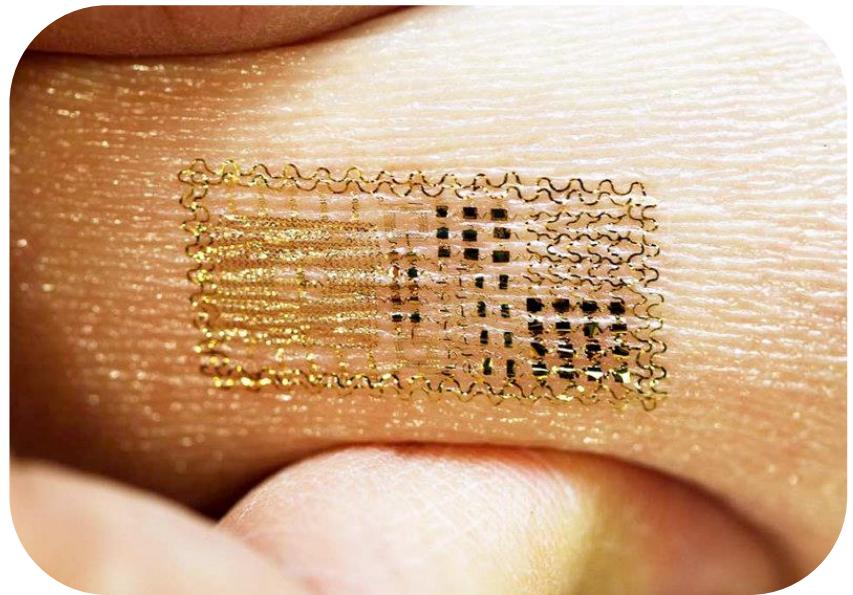
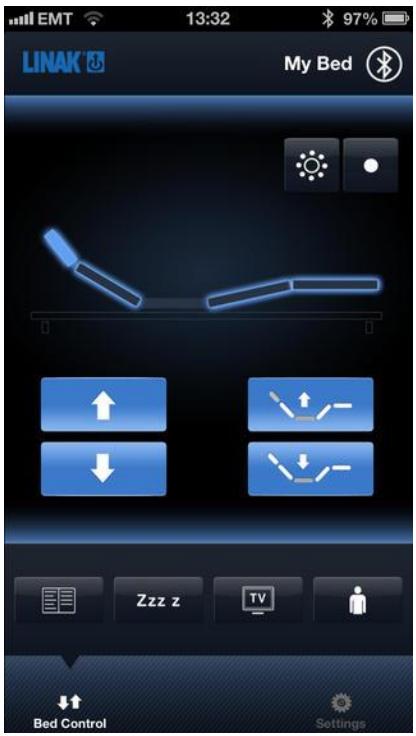
Computación ubicua y móvil

- Redes inalámbricas
- Miniaturización de dispositivos
 - Laptops
 - Celulares, smart phones, GPS, cámaras, smart watches, etc.
 - Dispositivos embebidos: refrigeradores, lavadoras, etc.
- “Location-aware” & “context-aware”



<https://www.microsoft.com/en-us/research/project/smart-tattoos/>

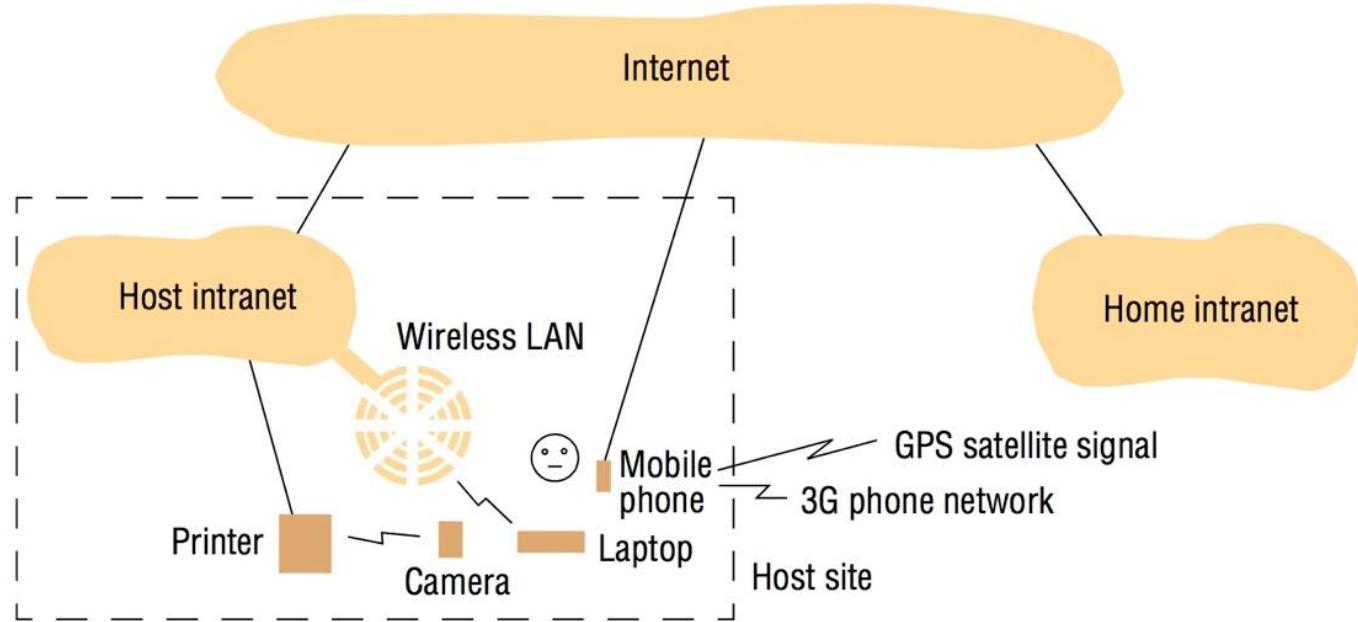
Smart - *



Computación ubicua



- Anywhere & Anytime
- La presencia de dispositivos en todas partes se vuelve (más) útil cuando los dispositivos se comunican entre sí.
- Interoperación espontánea y descubrimiento de servicios.



Sistemas multimedia distribuidos

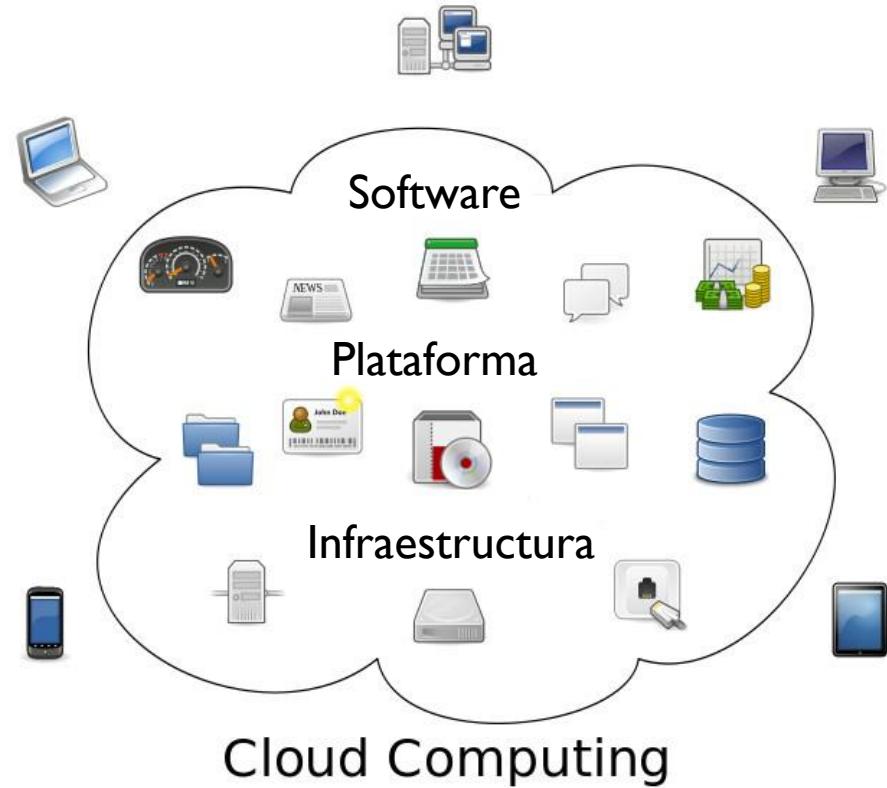
- Webcasting: transmisión de video y audio
- Calidad de servicio



La computación distribuida como un servicio público

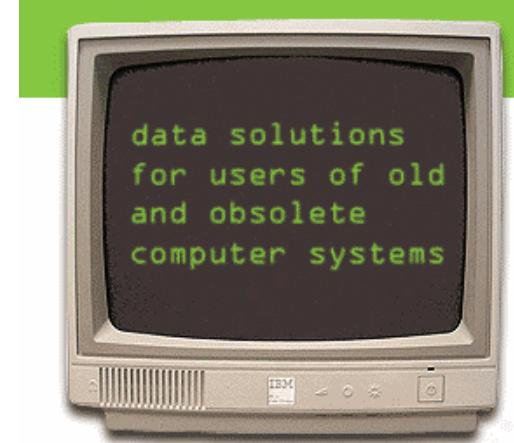
Cloud computing:

Infraestructura computacional virtualizada (hardware y software) accesible vía internet ofrecida bajo contratos de nivel de servicio.



Retos de los sistemas distribuidos

- **Heterogeneidad**
 - Redes
 - Hardware
 - Sistemas operativos
 - Lenguajes de programación
 - Implementaciones por diferentes desarrolladores
- **Middleware** capa intermedia de software que provee una abstracción de programación para solucionar los problemas de heterogeneidad, ejemplo JBoss ESB



Retos de los sistemas distribuidos

- **Código móvil**
 - Código enviado de una computadora a otra para ser ejecutado en la computadora destino.
 - Máquinas virtuales, ejemplo: JVM.
 - Ejemplo de código móvil: Javascript



Retos de los sistemas distribuidos

- **Extensibilidad:** determina si el sistema puede ser extendido y re-implementado en varias maneras.
- Un sistema es **abierto** si se pueden añadir nuevos servicios de compartición de recursos y ponerlos a la disposición de los clientes.



Retos de los sistemas distribuidos

- Sistemas distribuidos abiertos



- Interfaces públicas
- Mecanismos de comunicación estandarizados
- Conformados por hardware y software heterogéneo

Retos de los sistemas distribuidos

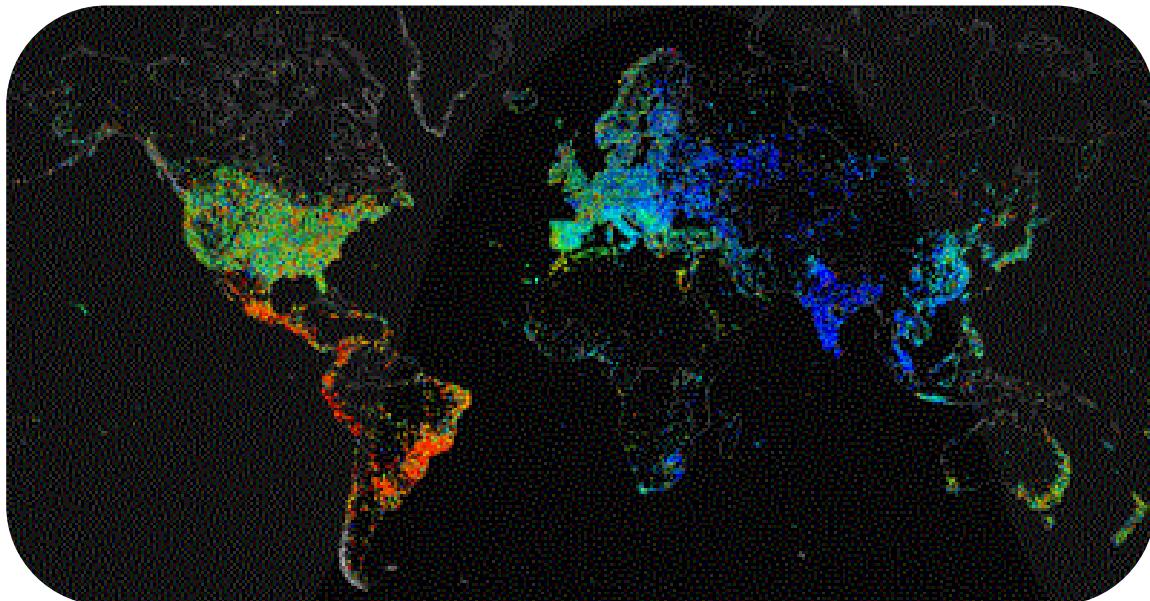
- Seguridad
 - Confidencialidad
 - Integridad
 - Disponibilidad
- Ataques de denegación de servicios: lento y rápido
- Seguridad del código móvil.



Retos de los sistemas distribuidos

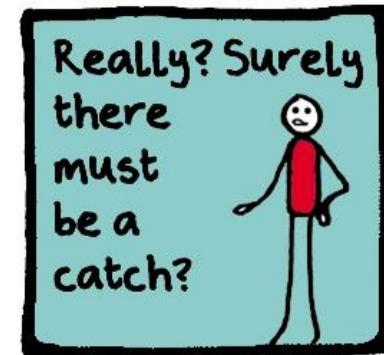
- **Escalabilidad**

Un SD es escalable si es eficiente al aumentar el numero de recursos y el numero de usuarios.



Retos de los sistemas distribuidos

- Manejo de fallos
 - Detección de fallos
 - Enmascaramiento de fallos
 - Tolerancia de fallos
 - Recuperación de fallos
 - Redundancia



Retos de los sistemas distribuidos

- **Concurrencia**

- Tanto servicios como aplicaciones proveen **recursos compartidos** (dentro de los SDs), los cuales pueden ser accedidos al mismo tiempo por múltiples usuarios.



Retos de los sistemas distribuidos

- **Transparencia**
 - Un SD debe verse por el usuario final como una sola entidad.
 - Transparencia de acceso
 - Transparencia de ubicación
 - Transparencia de concurrencia
 - Transparencia de replicación
 - Transparencia de fallas
 - Transparencia de movilidad de componentes



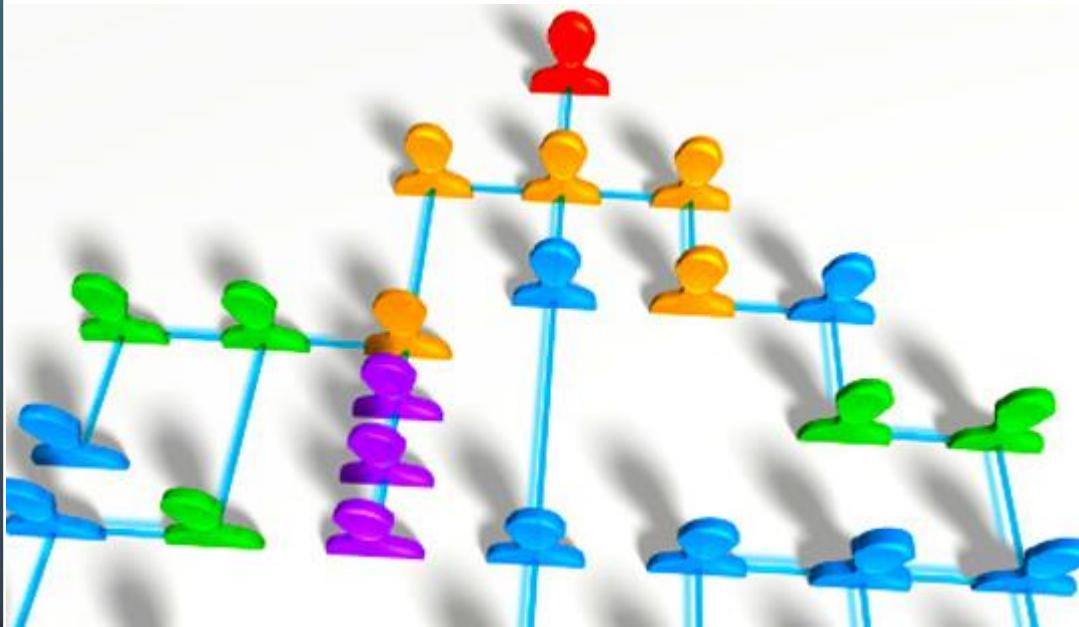
Retos de los sistemas distribuidos

- **Calidad de servicio**
 - Propiedades de los SDs no funcionales:
 - Confiabilidad
 - Seguridad
 - Rendimiento
 - Adaptabilidad



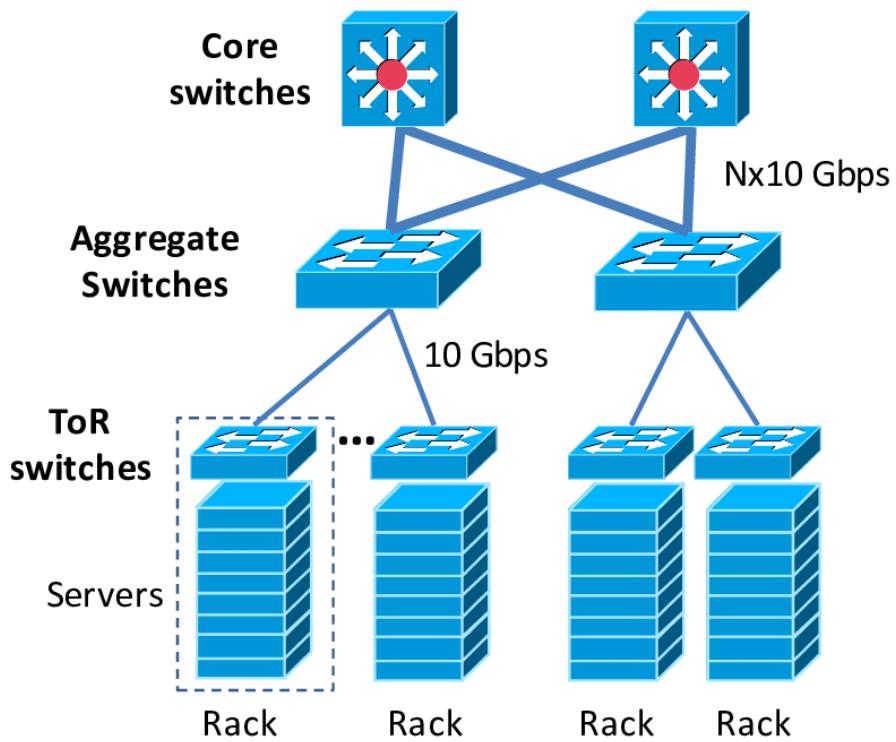
Modelos de sistemas distribuidos

- Modelos físicos
- Modelos arquitectónicos
- Modelos fundamentales



Modelos físicos

- Representación del hardware subyacente de un SD en términos de computadoras (y otros dispositivos) y su red de interconexión.



Modelos de sistemas distribuidos

- Modelos físicos
- Modelos arquitectónicos
- Modelos fundamentales



Modelos Arquitectónicos

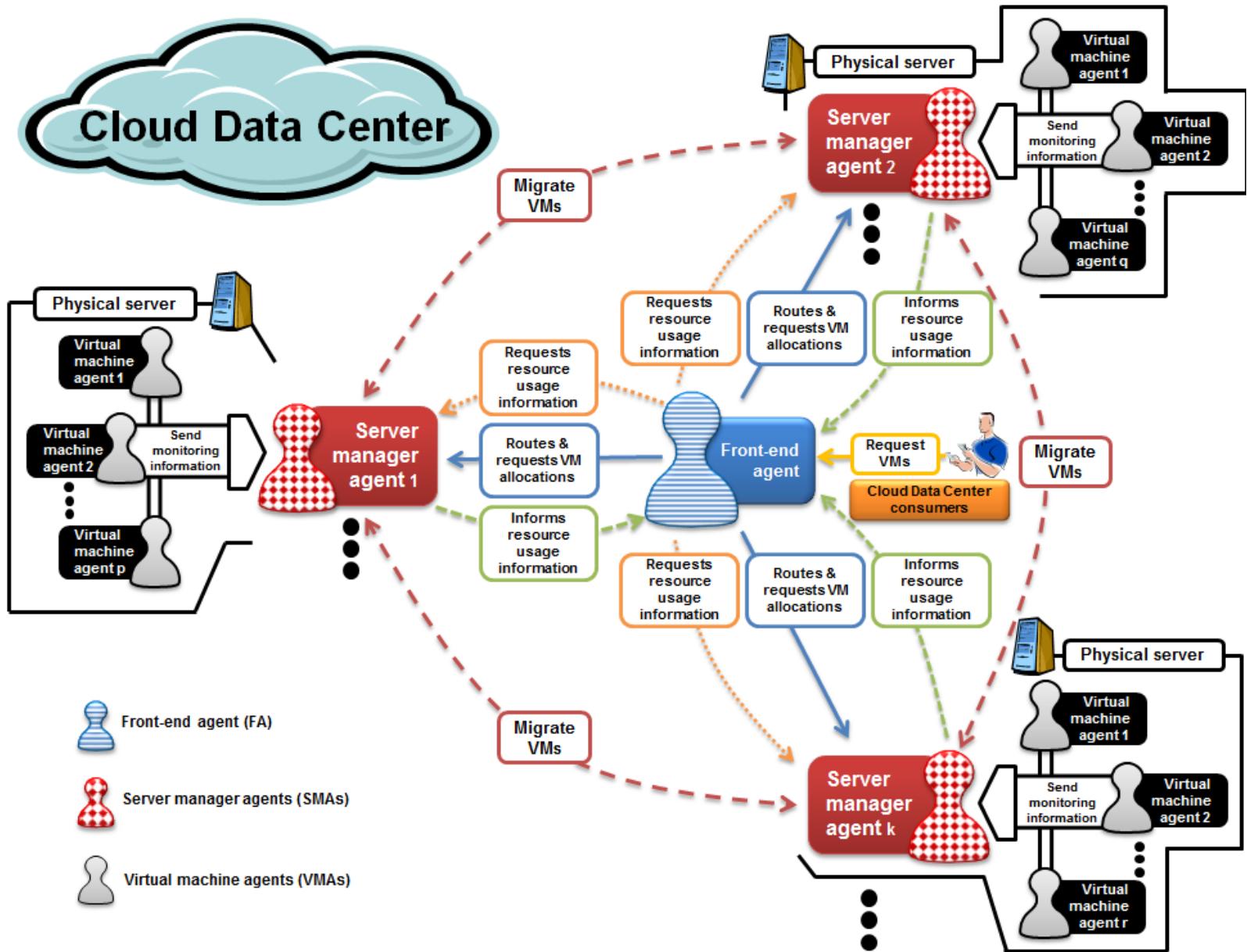
- Describen a los SDs en términos de las tareas computacionales y tareas de comunicación realizadas por sus componentes.
- La arquitectura de un sistema es su estructura en términos de componentes separados y sus interrelaciones.

Elementos arquitectónicos

- ¿Cuáles son las entidades?
- ¿Cómo se comunican?
- ¿Cuáles son sus roles y responsabilidades?
- ¿Dónde se encuentran?



Un modelo arquitectónico



Entidades comunicantes

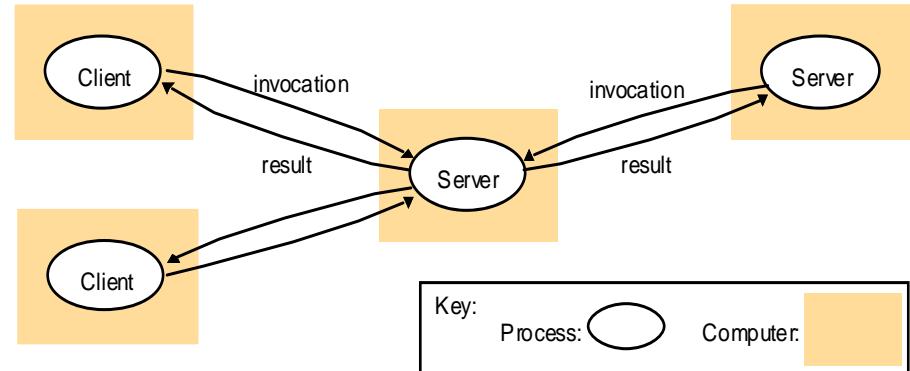
- Desde una perspectiva de sistema
 - Procesos (hilos)
 - Nodos, ejemplo: sensores.
- Desde una perspectiva de programación
 - Objetos
 - Componentes
 - Servicios Web

Paradigmas de comunicación

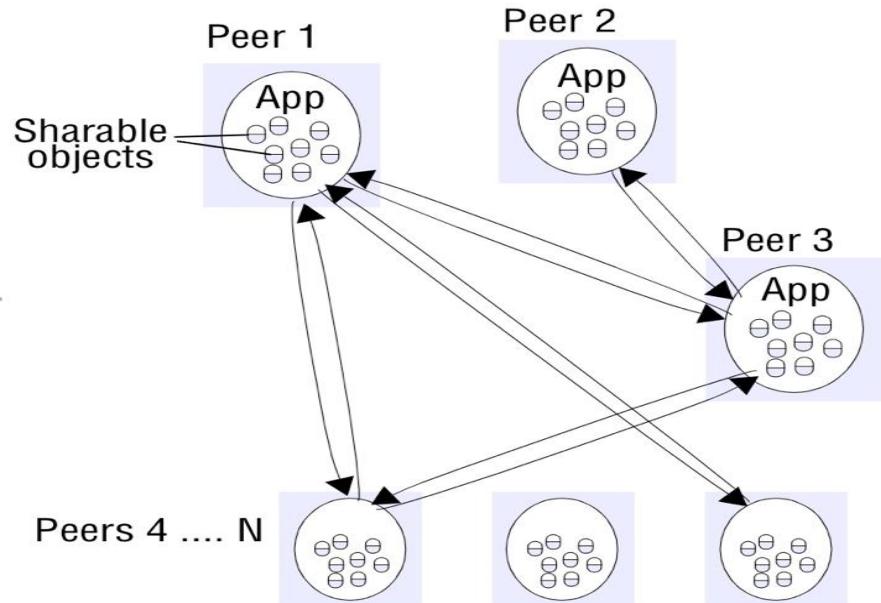
- Comunicación entre procesos (sockets)
- Invocación remota
 - Protocolos de solicitud-respuesta (HTTP)
 - Llamadas a procedimientos remotos (RPC)
 - Invocación de métodos remotos (RMI)
- Comunicación indirecta
 - Comunicación en grupo
 - Sistemas de publicación y subscripción (uno-a-muchos)
 - Colas de mensajes (uno-a-uno)
 - Espacios de tuplas
 - Memoria distribuida compartida

Roles y responsabilidades

- **Cliente-Servidor**



- **Igual-a-Igual
(peer-to-peer)**

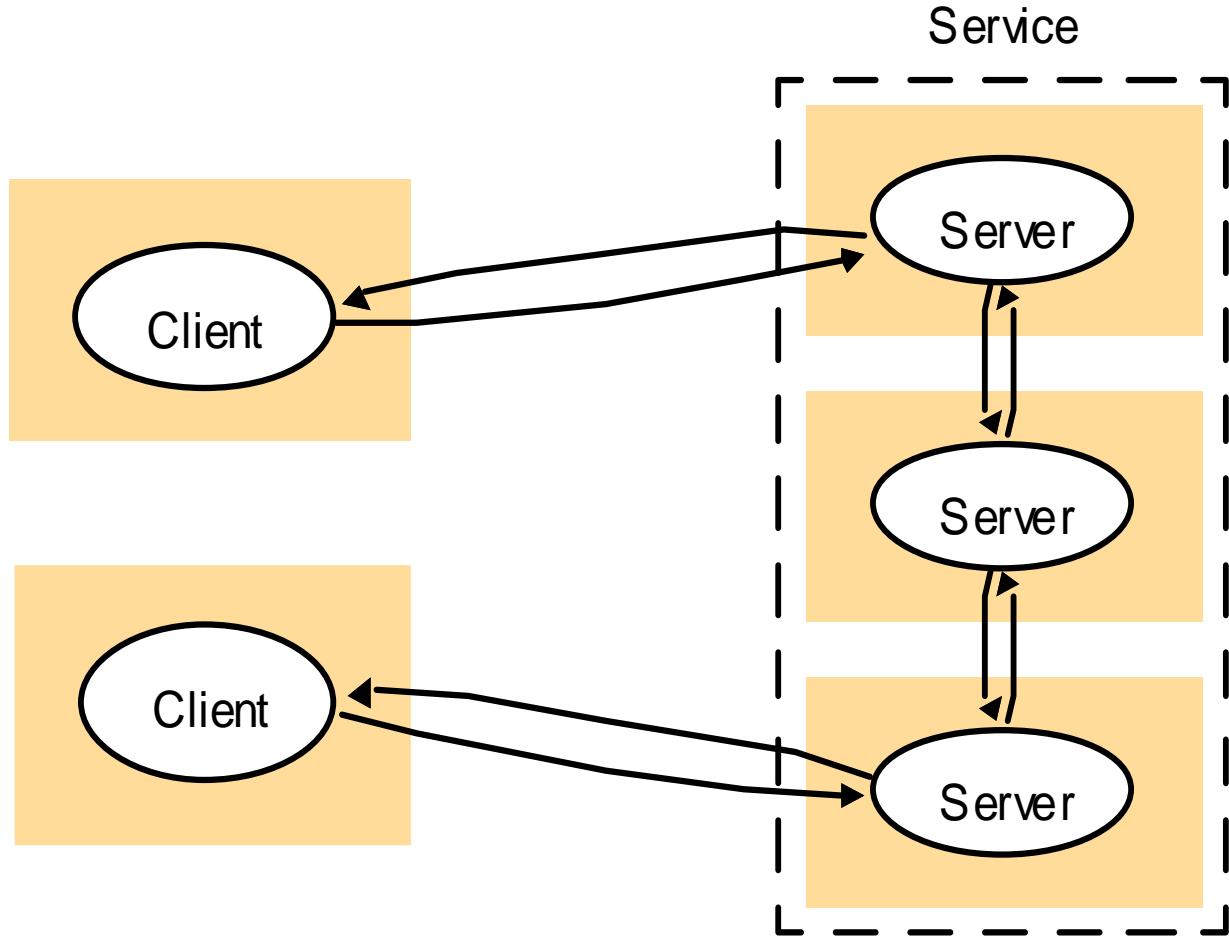


Estrategias de colocación

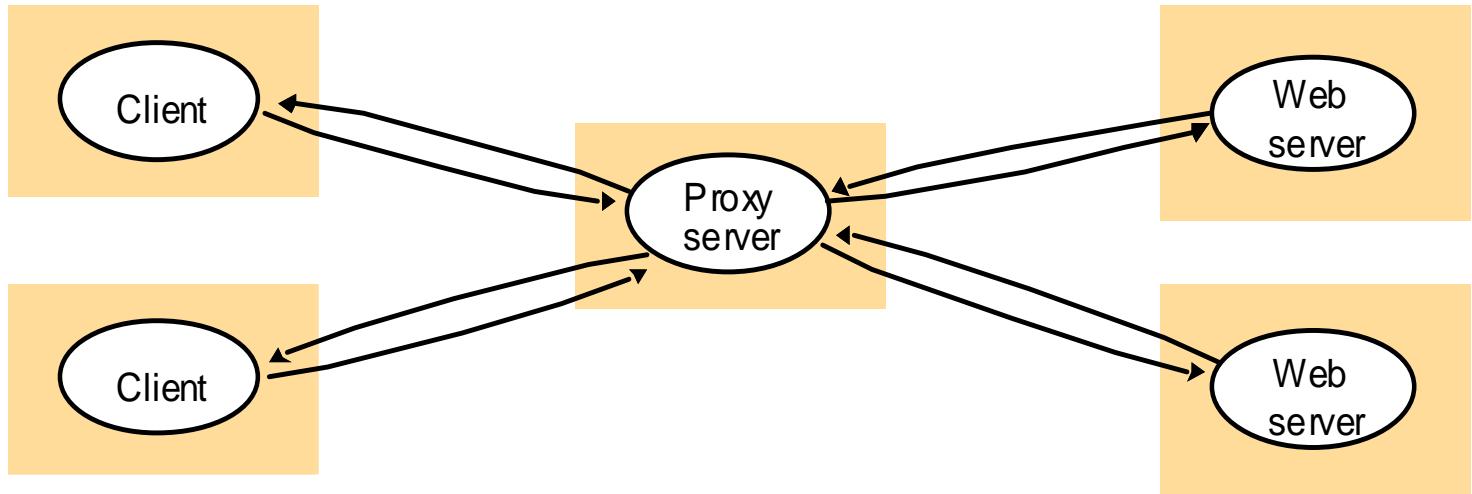
- Asignación de servicios a múltiples servidores
 - Almacenamiento en caché
 - Código móvil
 - Agentes móviles



Asignación de servicios a múltiples servidores



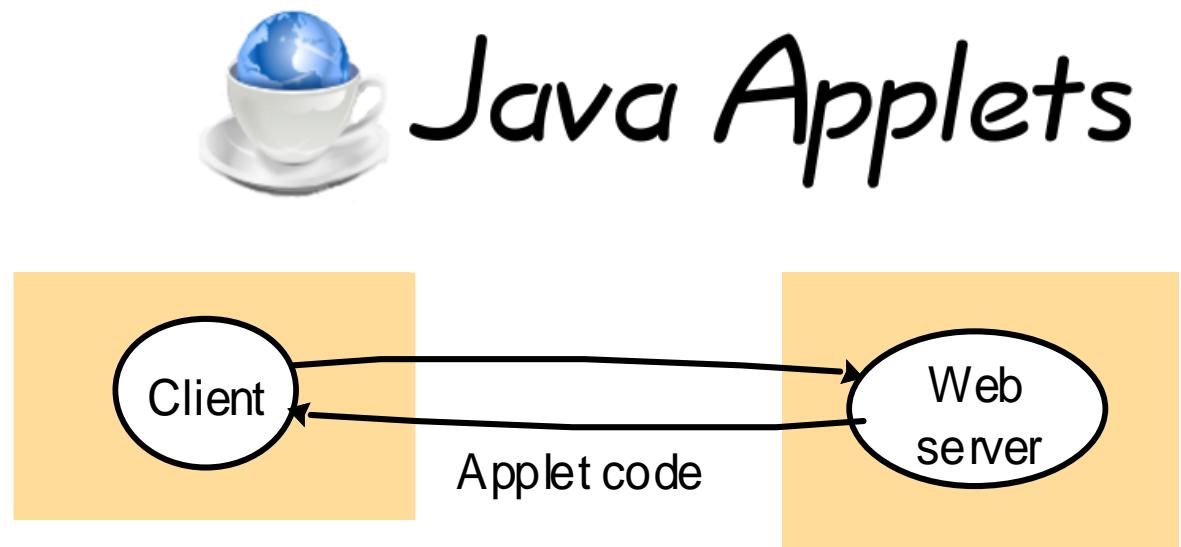
Almacenamiento en caché



Código móvil

- Applets

a)

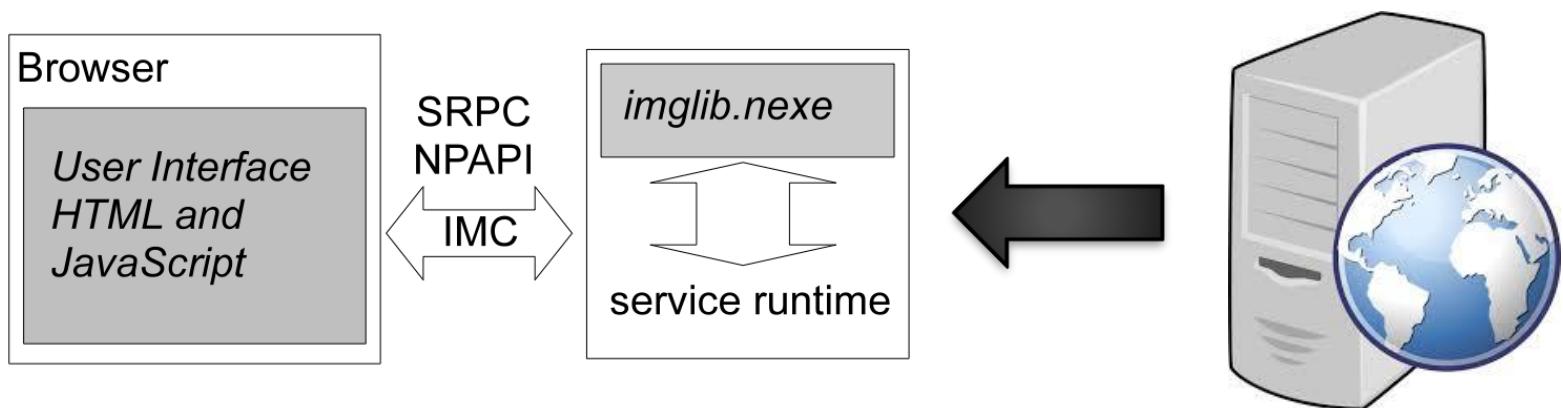


b)



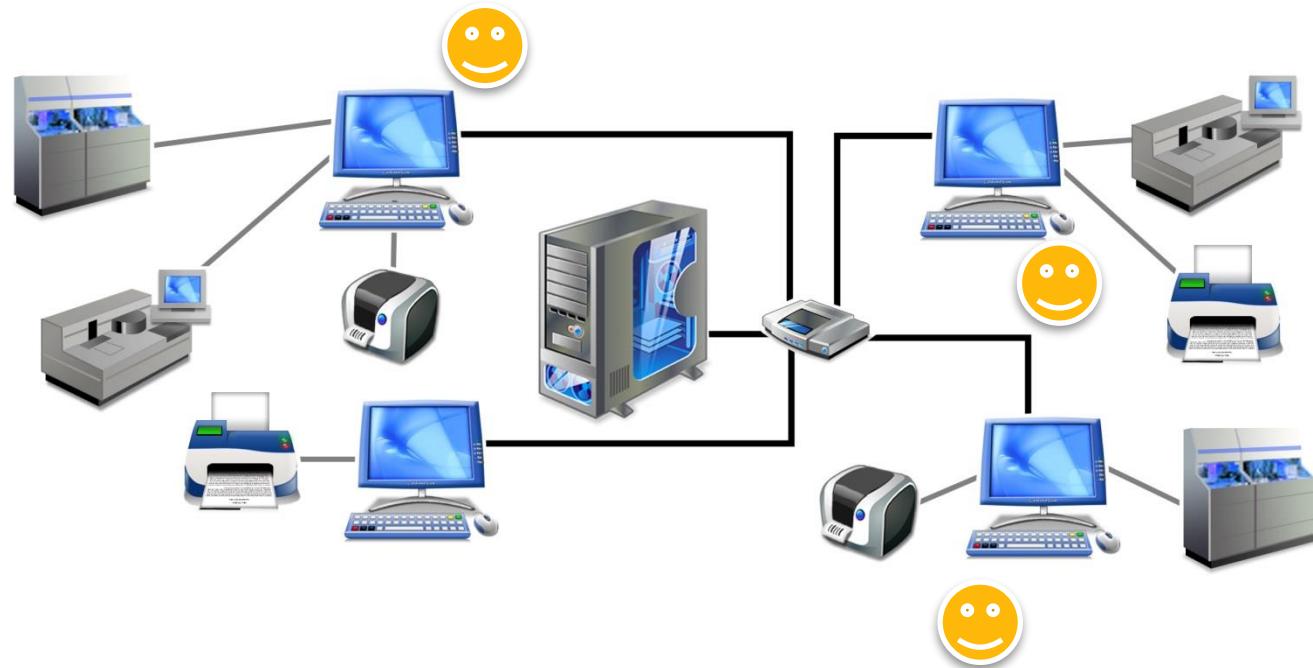
Código móvil

- Sandboxes



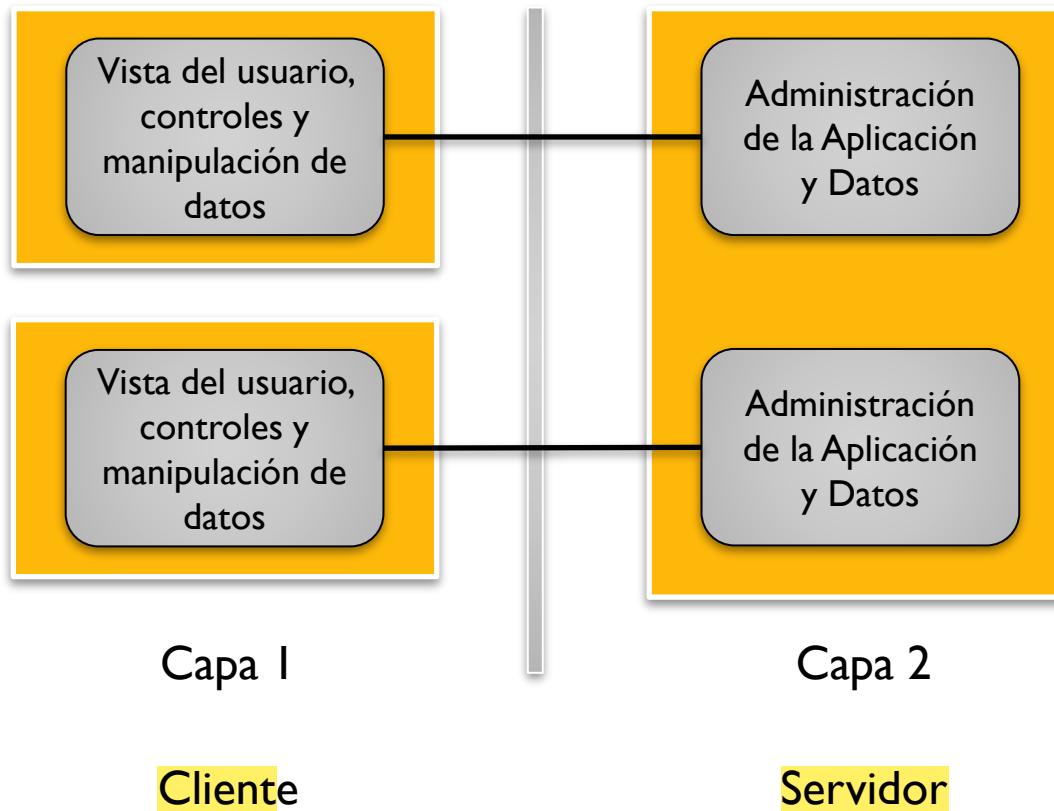
Agentes móviles

- Recolección de datos



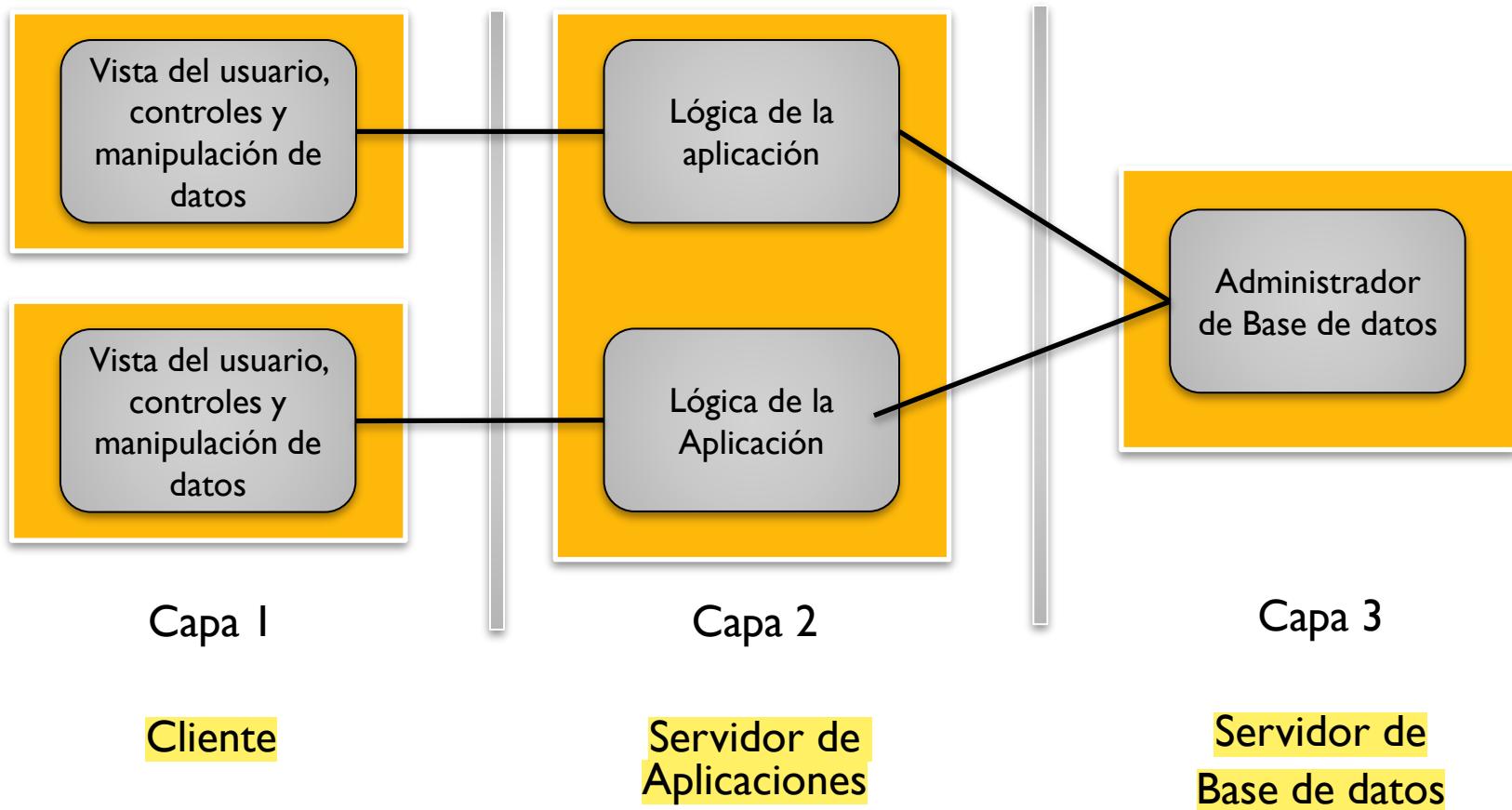
Patrones Arquitectónicos

- Arquitectura en 2 capas



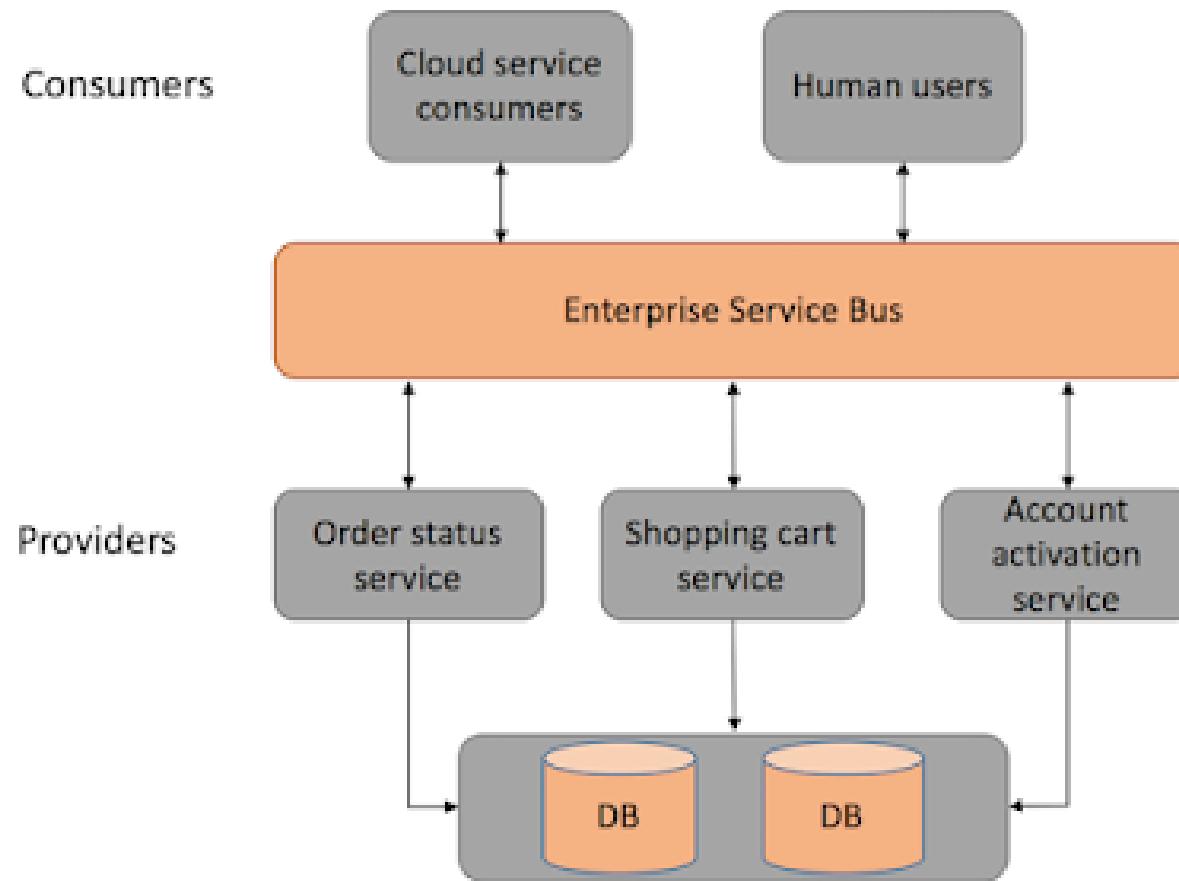
Patrones Arquitectónicos

- Arquitectura en 3 capas



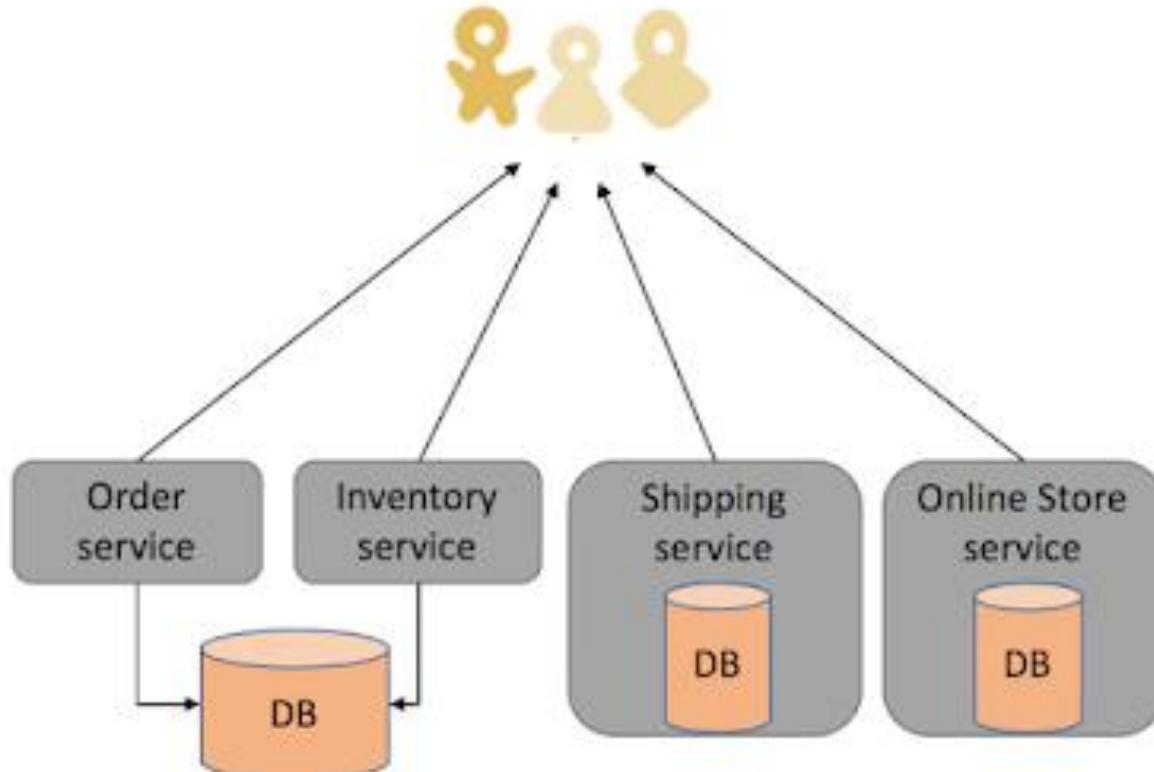
Patrones Arquitectónicos

- Arquitecturas orientadas a servicios



Patrones Arquitectónicos

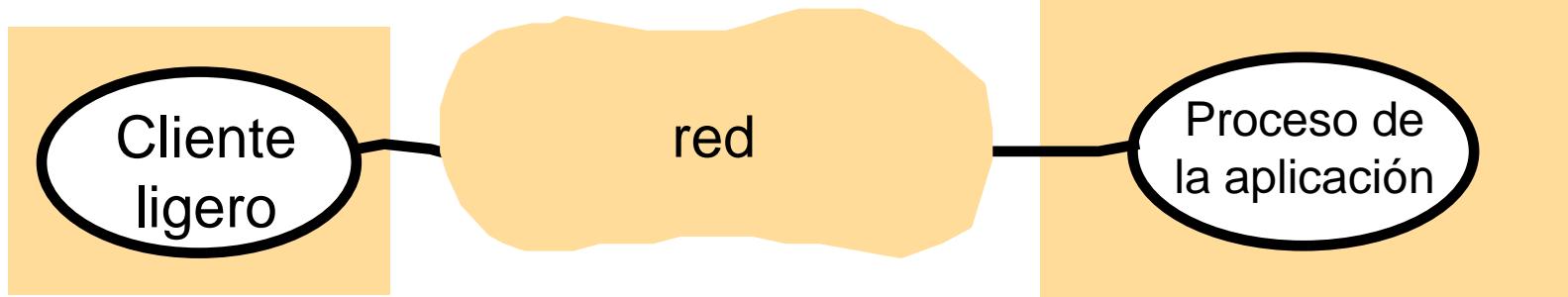
- Arquitecturas de microservicios



Patrones Arquitectónicos

- **Creadores ligeros**

Dispositivo en red



Servidor de cómputo



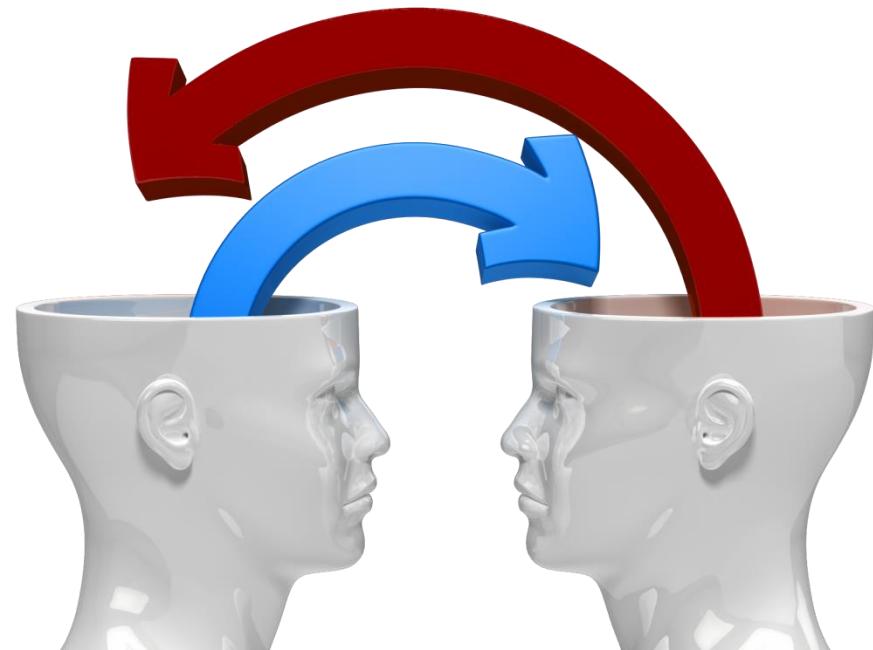
Modelos de sistemas distribuidos

- Modelos físicos
- Modelos arquitectónicos
- Modelos fundamentales



Modelos Fundamentales

- **Modelo de Interacción**
- **Modelo de Fallo**
- **Modelo de Seguridad**



Modelo de Interacción

- Factores que afectan la interacción en los SDs:

- El desempeño de los canales de comunicación es una limitante
- Es imposible mantener una noción global del tiempo



Modelo de Interacción

- Desempeño de los canales de comunicación:

- **Latencia**

- Tiempo que le toma al mensaje llegar a su destino
 - Retardo al acceder a la red.
 - Tiempo que toman los procesos y el SO en mandar y recibir el mensaje

- **Ancho de banda**

- **Fluctuación (jitter)**

- Variación en el tiempo invertido en completar el reparto de una serie de mensajes, ejemplo: aplicaciones multimedia



Modelo de Interacción

- Relojas de las computadoras y eventos de temporización

- SDs sincronos



- SDs asíncronos



Modelo de Interacción



- SDs síncronos

- El tiempo para ejecutar cada paso de un proceso tiene un **límite inferior** y uno **superior**
- Cada **mensaje** se recibe en un **tiempo limitado** conocido
- Cada **proceso** tiene un **reloj local** (basado en un reloj real)

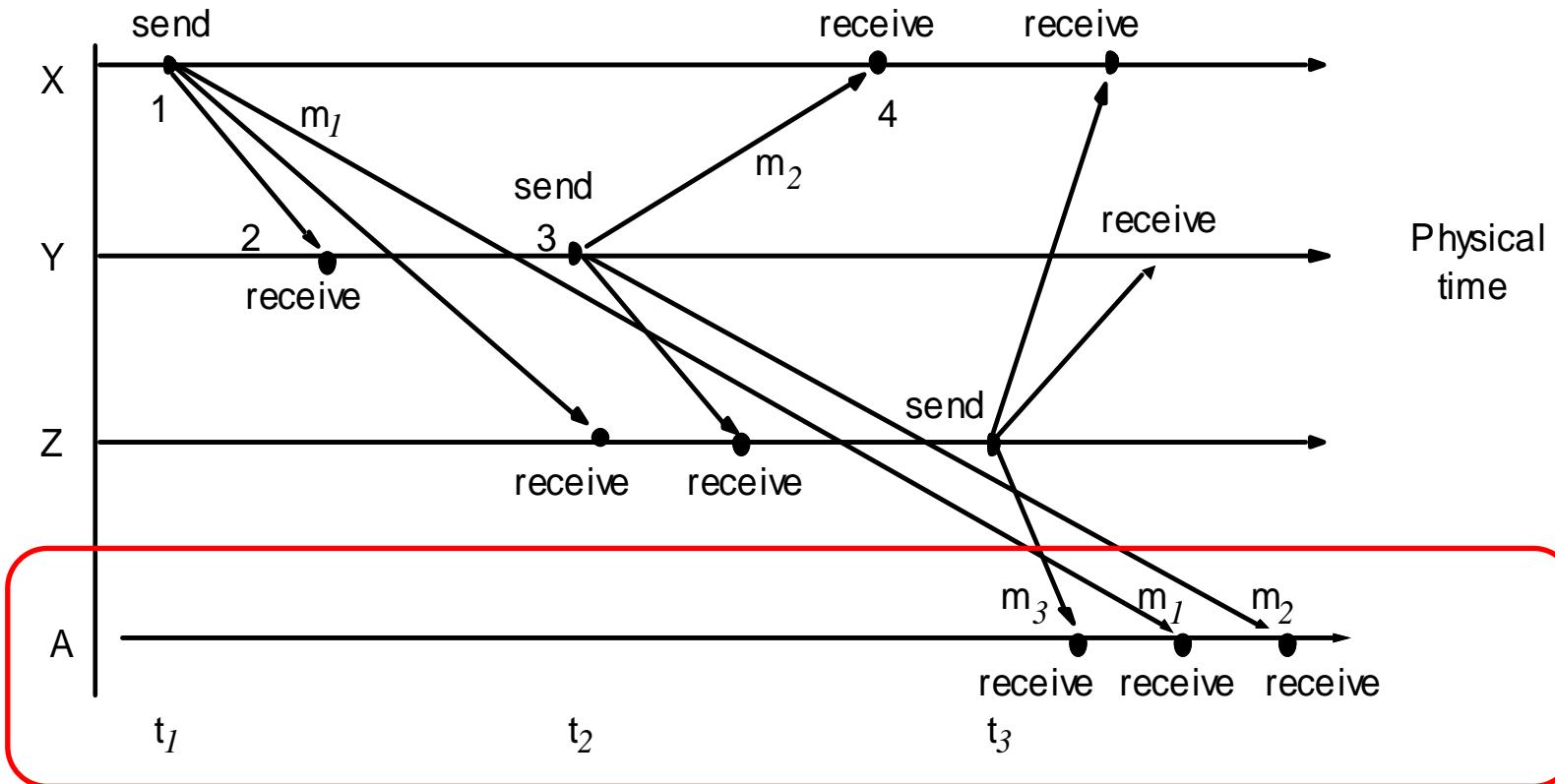
Modelo de Interacción



- SDs asíncronos
 - El tiempo de ejecución de cada paso de un proceso no tiene límites
 - La latencia en la transmisión de mensajes puede ser arbitraria
 - Los valores de los relojes locales son arbitrarios.

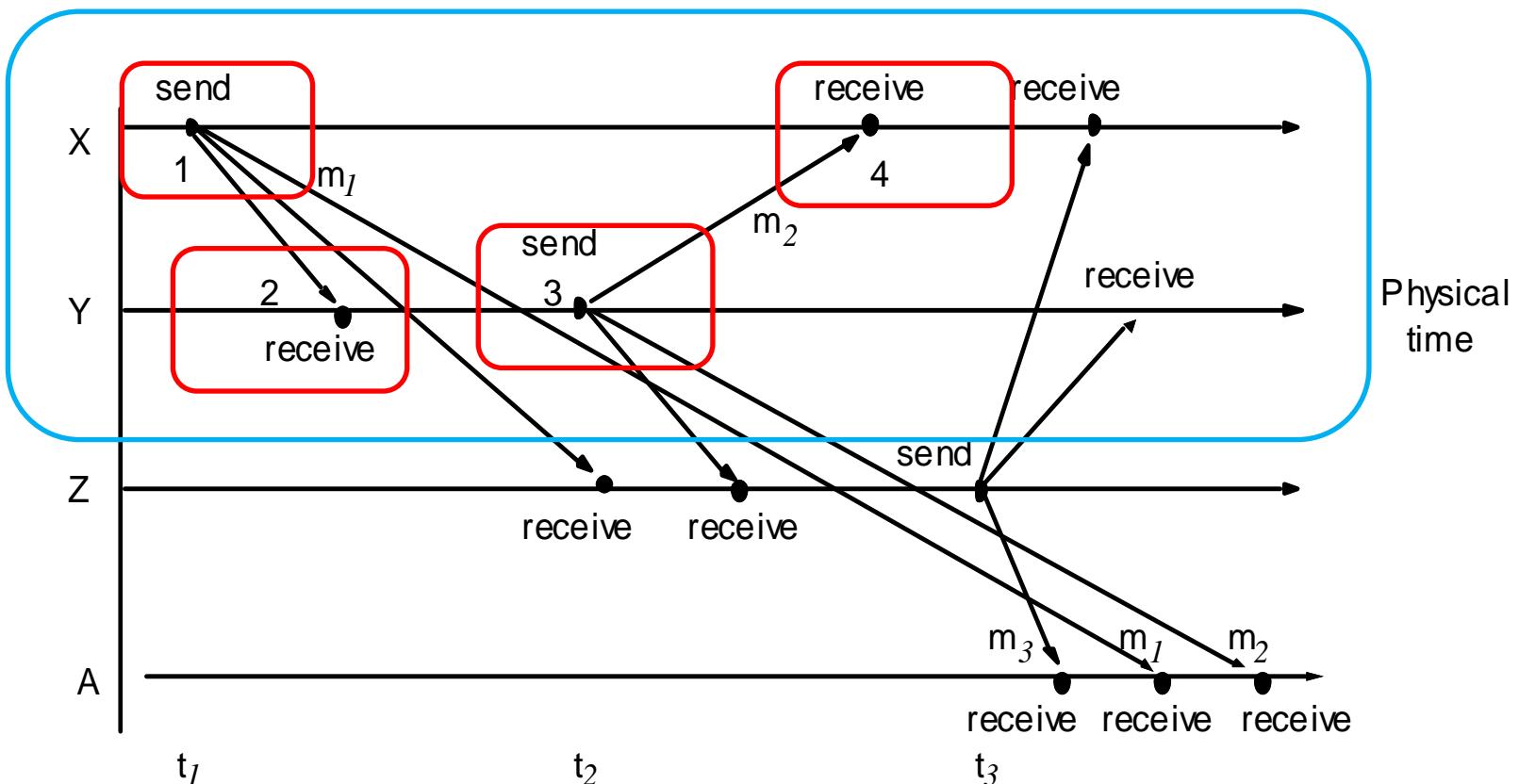
Ordenamiento de Eventos

- Envío de un email para acordar una Reunión



Ordenamiento de Eventos

- Tiempo lógico [Lamport, 1978]



Lectura adicional:

[Lamport, 1978] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System", Communications of the ACM, Vol 21(7), 1978, pp. 558-565

Modelos Fundamentales

- Modelo de Interacción
- **Modelo de Fallo**
- Modelo de Seguridad



Modelo de Fallo

- Fallos por omisión
 - Proceso
 - Comunicación
- Fallos de temporización
- Fallos arbitrarios
(bizantinas)



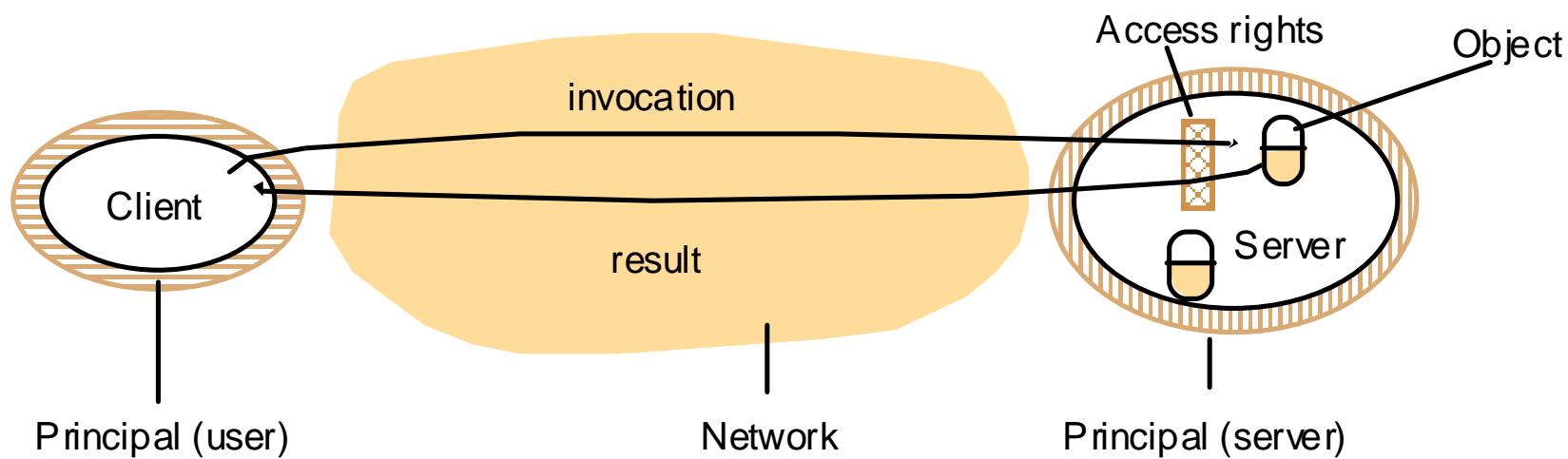
Modelos Fundamentales

- Modelo de Interacción
- Modelo de Fallo
- **Modelo de Seguridad**



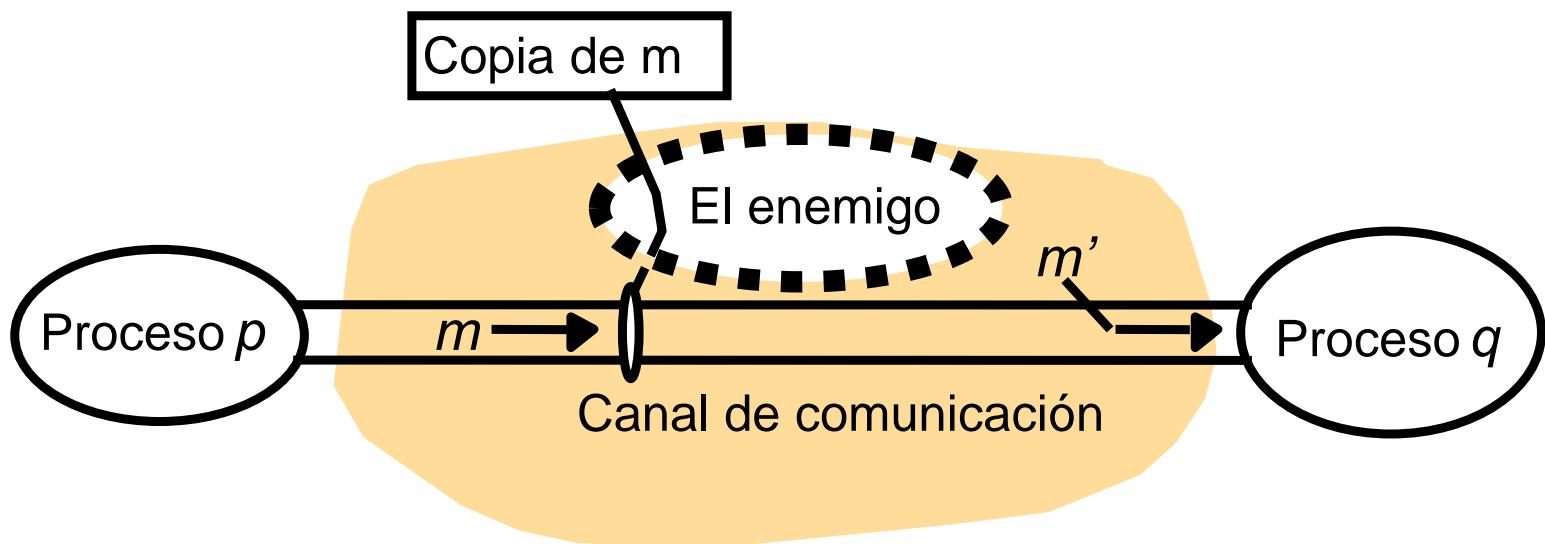
Modelo de Seguridad

- La seguridad de un SD depende de:
 - Proteger los objetos de los procesos



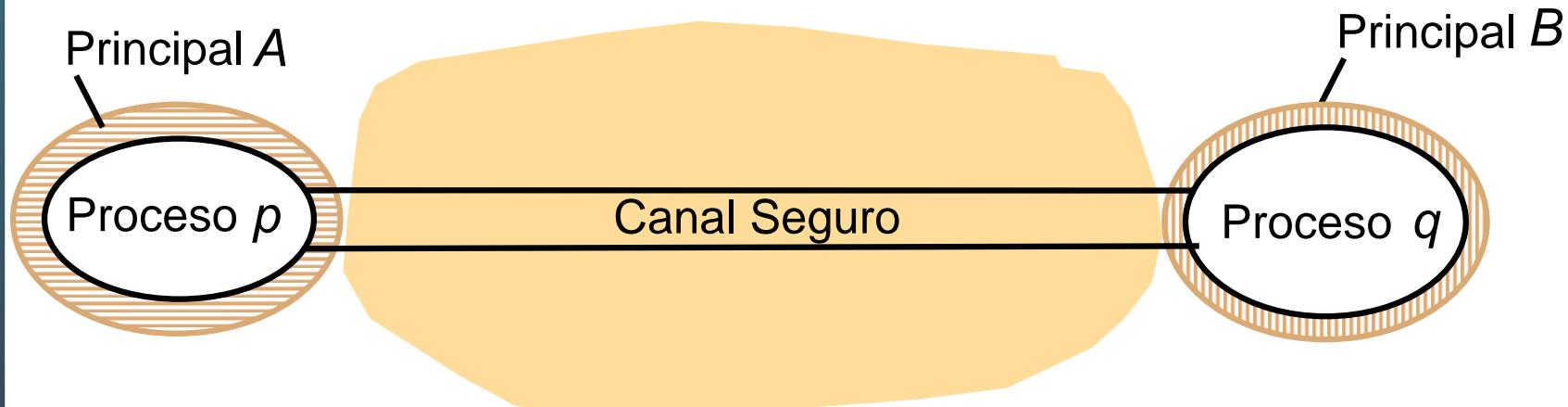
Modelo de Seguridad

- La seguridad de un SD depende de:
 - **Proteger los canales de comunicación**



Modelo de Seguridad

- Vencer amenazas de seguridad
 - Criptografía
 - Autenticación
 - Canales seguros (VPN, SSL)



Modelo de Seguridad

- Otras amenazas
 - Denegación de servicio (DoS)
 - Código móvil

