Sistemas Distribuidos

Módulo 1

Introducción a los Sistemas Distribuidos

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

Conceptos Generales

Concurrencia - está fuertemente relacionado con la utilización de dispositivos únicos

Computación Paralela - La computación paralela se orienta a resolver rápidamente una tarea empleando múltiples procesadores simultáneamente.

Computación Distribuida - Un sistema distribuido es una colección de computadoras autónomas que están conectadas unas con otras y cooperan compartiendo recursos.

KMC © 2020

Computación en Paralelo

Características

- Una aplicación es dividida en subtareas que son resueltas simultáneamente.
- Se considera una aplicación por vez y el objetivo es el ***speed-up*** de procesamiento de la misma.
- Los programas usualmente corren en arquitecturas homogéneas y pueden tener memoria compartida.

KMC © 2020

Computación Distribuida

Problemas a resolver

- Administración del acceso a recursos compartidos
- Heterogeneidad operativa (HW, SO y los lenguajes).
- Seguridad.

Características

- Múltiples recursos en locaciones físicamente distantes.
- Corren múltiples aplicaciones a la vez.
- Generalmente heterogéneos.
- El sistema distribuido luzca como una ***única máquina para los usuarios***.
- Sistemas distribuidos ***no tienen memoria compartida*** (a nivel de hardware)

KMC © 2020

Motivaciones para la Computación Paralela y Distribuida

- Rendimiento absoluto.
- Relación precio/rendimiento.
- Razones tecnológicas.
- Aplicaciones con paralelismo o distribución inherentes.
- Recursos compartidos.
- Crecimiento incremental.
- Otras razones: balance de carga, utilización de capacidad ociosa.

KMC © 2020

Rendimiento de Aplicaciones Simples

Métrica obvia: **"tiempo de corrida"** (o costo de ejecución).

Speed-up:

$$\text{speed-up}(P) = T_1 / T(P)$$

donde:

T(P): tiempo de corrida del programa paralelo en **P** procesadores.

T_1: tiempo de corrida de un programa secuencial de referencia.

En general, este último es el programa secuencial más rápido que soluciona el problema.

Eficiencia:

$$\text{eficiencia}(P) = \text{speed-up}(P) / P$$

donde:

P: número de procesadores

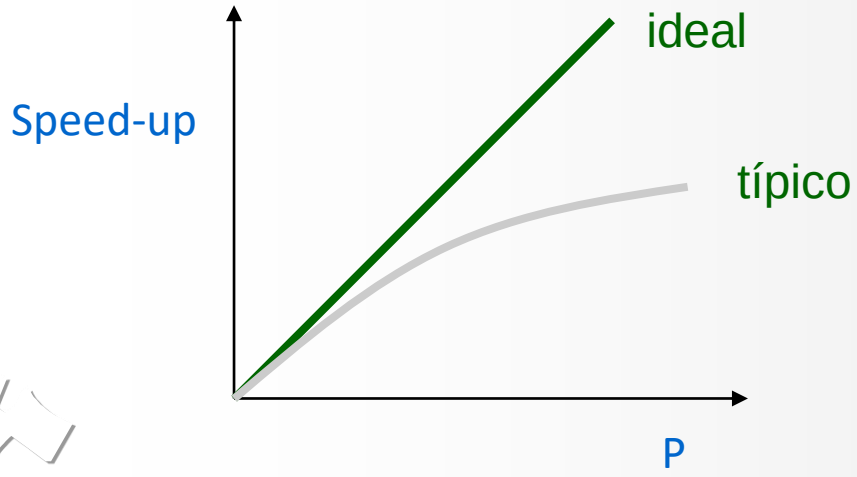
KMC © 2020

Rendimiento de Aplicaciones Simples

Idealmente se espera que el speed-up crezca linealmente y la eficiencia sea 1(uno) para todo P.

Hay casos donde el speed-up es ***superlineal*** o sea que k procesadores resuelven una tarea en menos que *un k-ésimo* del tiempo de corrida secuencial.

Comportamiento explicable por el aumento del tamaño del caché.



KMC © 2020

Rendimiento de Aplicaciones Simples

Razones de la diferencia entre el speed-up ideal y típico:

Ley de Amdahl:

- 1er. Observación
  - ✓cada computación contiene una porción serial de ejecución, es decir, alguna parte ***s*** del código no es posible paralelizarlo;
- 2da. Observación (ley Gustafson-Barsis)
  - ✓establece que se usan programas paralelo muy frecuentemente para resolver instancias más grandes de un problema que su contraparte secuencial; así en la medida que el número de procesadores crece, T_1 crece mientras que ***s*** permanece casi constante, en la práctica T_1/s no es una constante.

KMC © 2020

Rendimiento de Aplicaciones Simples

- **Administración de tareas y balance de carga**: El manejo de un conjunto de tareas induce cierta sobrecarga.
- **Comunicación y sincronización**: La paralelización introduce la necesidad de comunicación y sincronización. Los costos de comunicación son medidos en términos de **latencia** y **ancho de banda**.

Los costos de comunicación pueden ser reducidos pero no evitados.

KMC © 2020

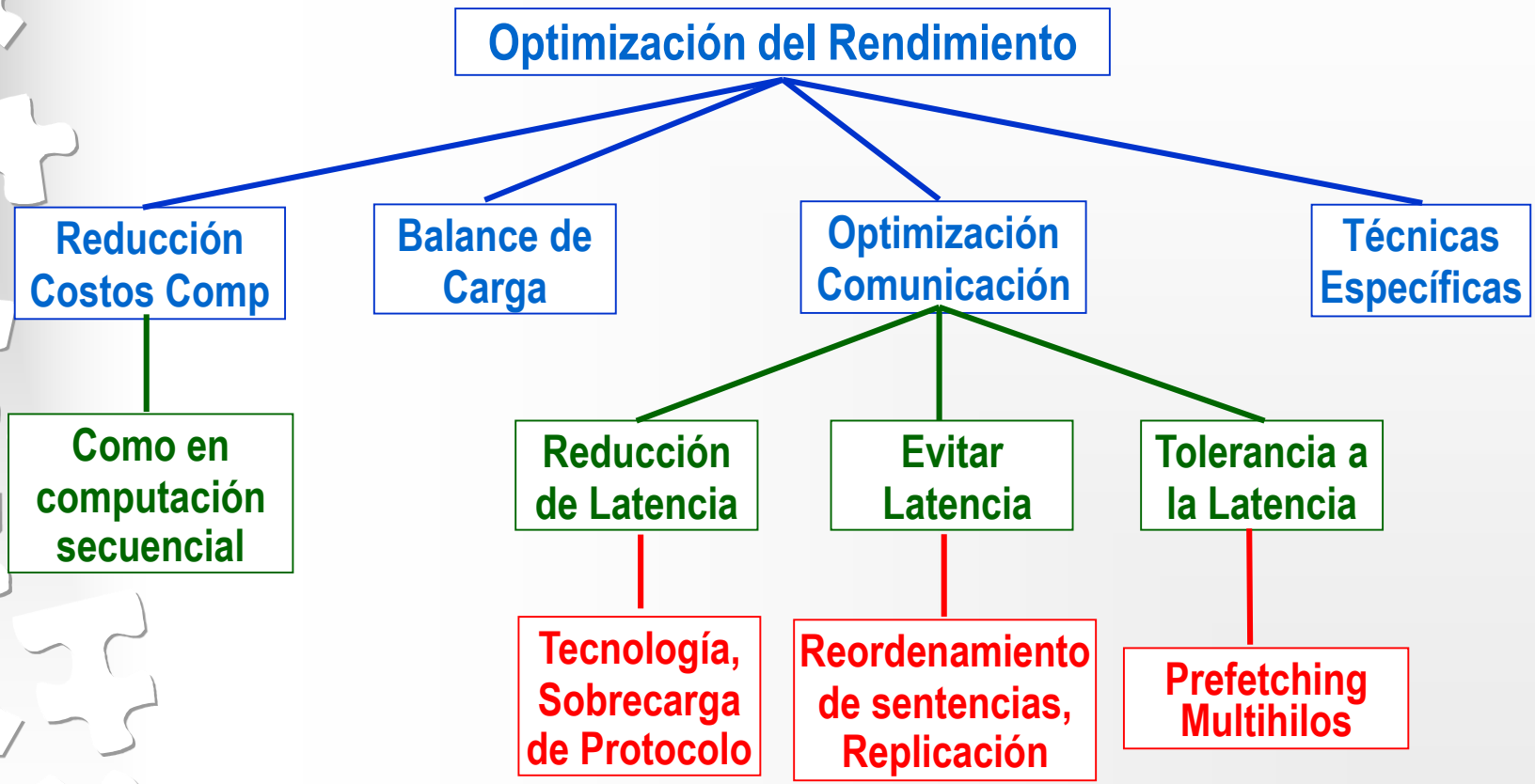
Rendimiento en Aplicaciones Múltiples

Consideraciones

- Tiempo de respuesta
- Procesamiento total (throughput)
- Utilización de recursos
- Calidad de servicios

KMC © 2020

Optimización del Rendimiento



KMC © 2020

Complejidad de Diseño de Programa

- Granularidad o tamaño de la tarea - Puede ser *fina*, *media* o *gruesa*.
- *Planificación de tareas* en sentido espacial y temporal y la *distribución de datos*.
- El *manejo de las comunicaciones* y la *sincronización* (llamado en conjunto *coordinación*). La *correctitud* es fundamental dado que la coordinación puede generar efectos no deseados.
- *Interbloqueo*.

Problemas Intratables

Por ejemplo: dentro del grupo de problemas intratables resulta el referido a la planificación de tareas, mas específicamente el de mapear un grafo de programa a la arquitectura.

KMC © 2020

Portabilidad del Código y del Rendimiento

Un programa es ***portable*** si corre en una variedad de arquitecturas, inclusive las futuras. Ventajas:

- El esfuerzo de escribir un programa.
- Pasar fácilmente a arquitecturas más potentes si es necesario más poder de computación.
- Pasar fácilmente a arquitecturas alternativas si el sistema original *capotó*.
- Los programas pueden ser desarrollados en plataformas relativamente baratas.

Para tener en cuenta es *portabilidad del rendimiento*

El problema es que no se pueden aprovechar las especificidades de las arquitecturas.

KMC © 2020

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

Sistemas Distribuidos

Definiciones

«Un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora» Tanenbaum.

«Sistemas Distribuidos son aquellos en los cuales los componentes de hardware y software están ubicados en computadoras de una red y se comunican y coordinan sus acciones solamente por medio de mensajes» Coulouris.

KMC © 2020

Sistemas Distribuidos

DESVENTAJAS de los sistemas distribuidos

- ***Software***: Hay poco software disponible para sistemas distribuidos. La algorítmica es menos controlable.
- ***Redes***: Se pueden saturar o causar otros problemas
- ***Seguridad***

Limitaciones que crean problemas tecnológicos en los SD.

- ✓No existe una memoria global (cada nodo tiene su memoria local).
- ✓Establecer un estado global es complejo.
- ✓No se puede asegurar un tiempo global.

KMC © 2020

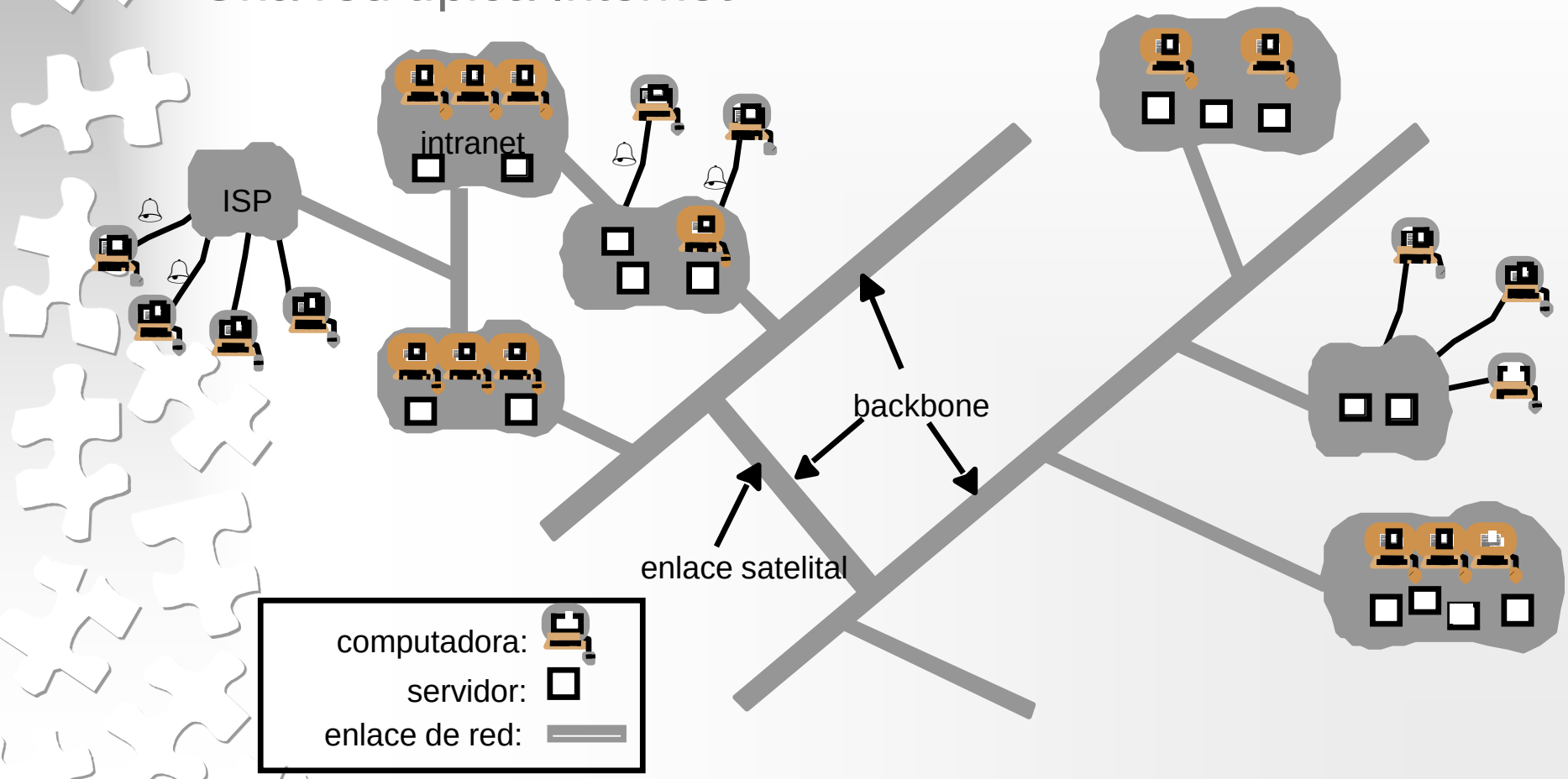
Sistemas Distribuidos: Tendencias

- ✓ Tecnología de red ubicua.
- ✓ Computación ubicuo y la movilidad del usuario.
- ✓ El incremento en la demanda de servicios multimedia.
- ✓ La vista de sistemas distribuidos como utilidad.

KMC © 2020

Sistemas Distribuidos: Ejemplos

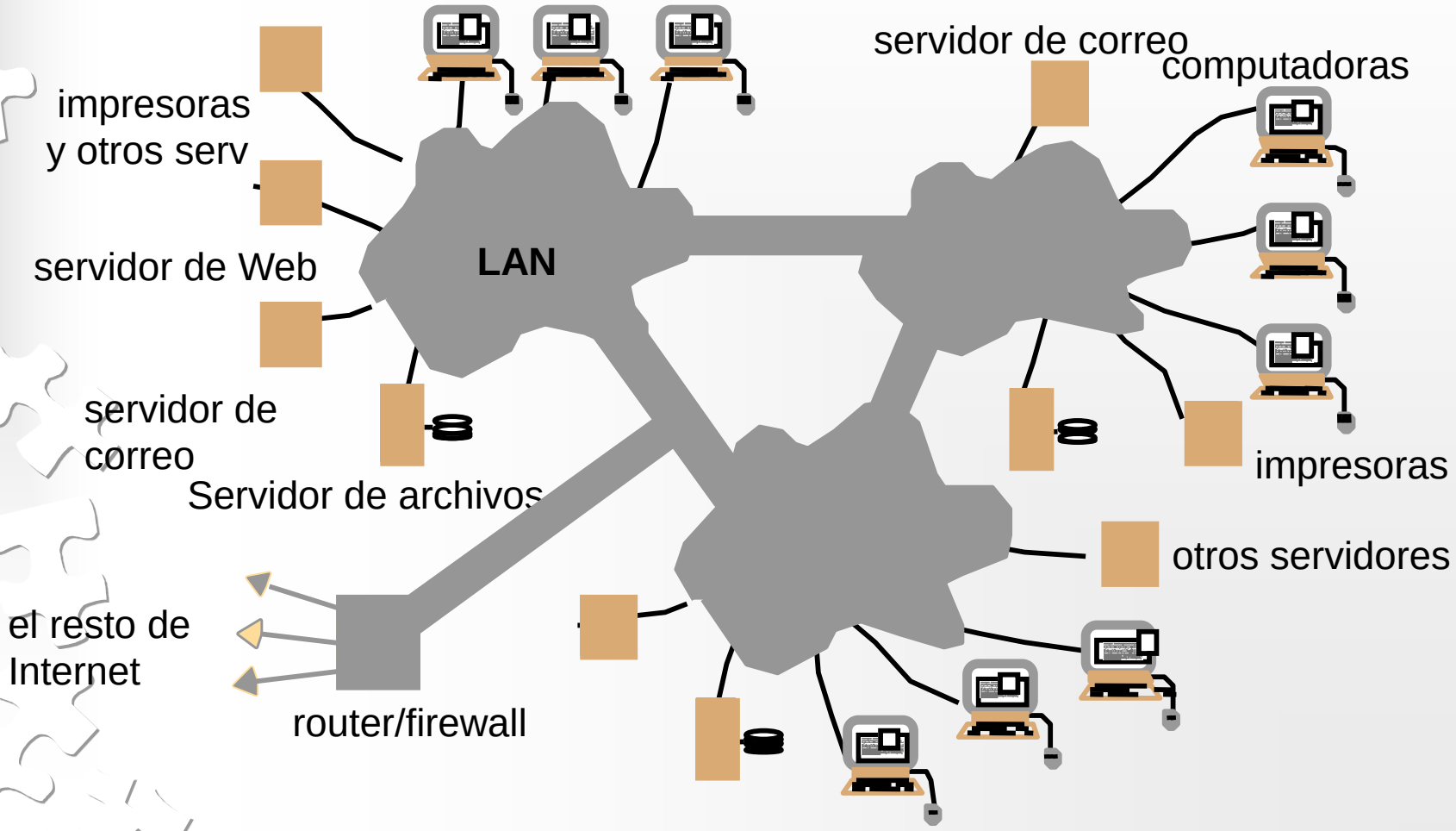
Una red típica Internet



KMC © 2020

Sistemas Distribuidos: Ejemplos

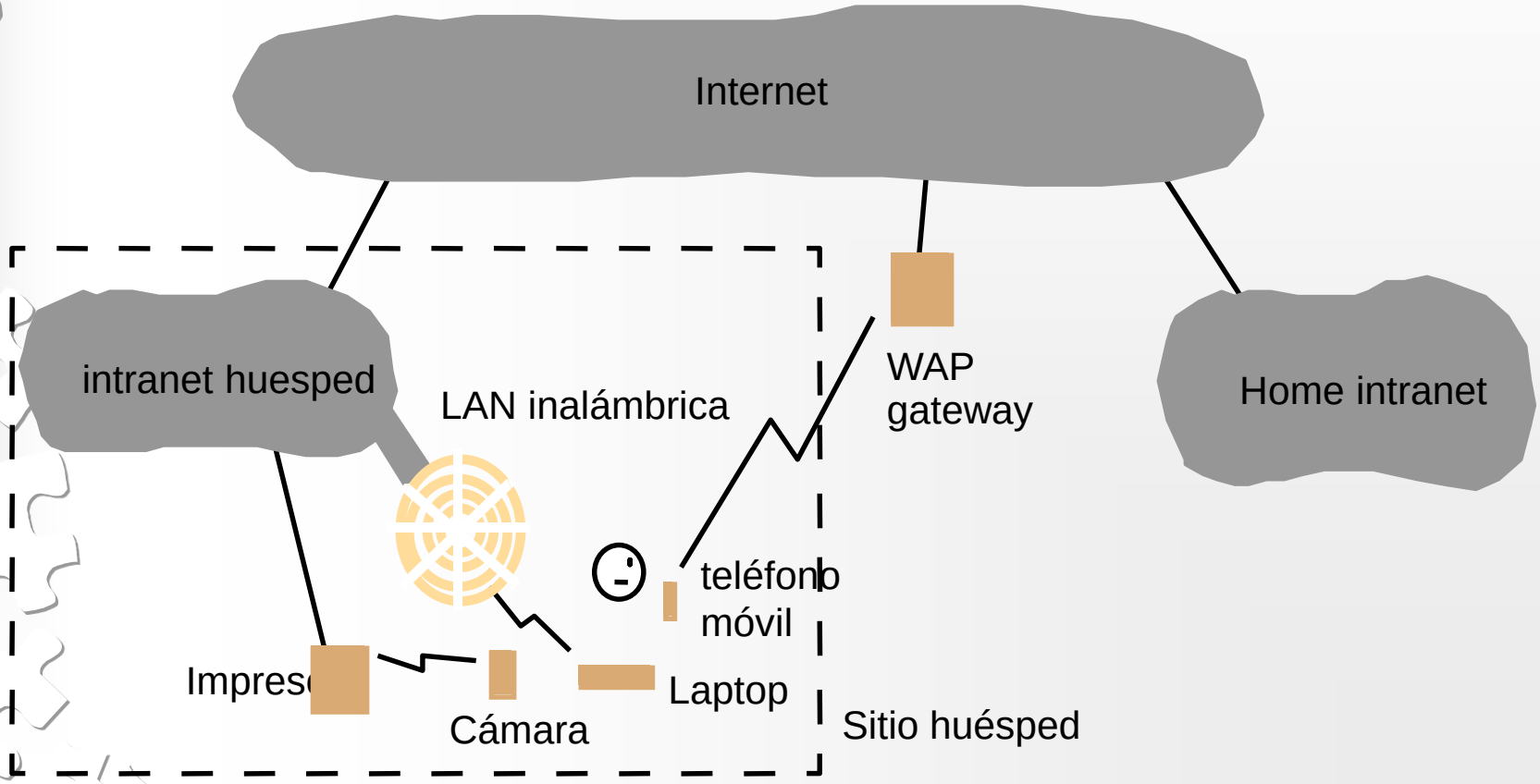
Una intranet típica



KMC © 2020

Sistemas Distribuidos: Ejemplos

Dispositivos portables y manuales en un sistema distribuido



KMC © 2020

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

Sistemas Distribuidos - Desafíos

- Heterogeneidad
- Extensibilidad
- Seguridad
- Escalabilidad
- Manejo de Fallas
- Concurrencia
- Transparencia
- Calidad de Servicio

KMC © 2020

SD- Desafío: HETEROGENEIDAD

Un sistema es ***heterogéneo*** si está compuesto por hardware y software distinto.

Muchos sistemas distribuidos son heterogéneos, mientras que programas paralelos son escritos frecuentemente para máquinas homogéneas.

Aquí aparece la noción de **interoperabilidad**: denota la habilidad de diferentes componentes, posiblemente de distintos proveedores, para interactuar. Estas partes pueden ser hardware o software.

Los componentes, para interoperar, deben respetar determinadas interfaces estándares. (IDL)

KMC © 2020

SD- Desafío: ESCALABILIDAD

- Problemas

Concepto	Ejemplo
Servicios Centralizados	Un único servidor para todos los usuarios.
Datos Centralizados	Una sola guía telefónica en línea.
Algoritmos Centralizados	Ruteo basado en información completa.

Algoritmos Distribuidos

1.- Información **PARCIAL ESTADO** sistema
2.- Decisión con información local
3.- Falla de una máquina
4.- Ausencia de **RELOJ GLOBAL**

KMC © 2020

SD- Desafío: Manejo de Fallos

Las fallas en los sistemas distribuidos son parciales, esto es, algunos componentes pueden fallar y otros funcionar correctamente.

- Detección de fallas
- Enmascaramiento de fallas
- Tolerancia de fallas
- Recuperación de fallas
- Redundancia

KMC © 2020

SD- Desafío: TRANSPARENCIA

Ocultación al usuario y al programador de aplicaciones de la separación de los componentes en un sistema distribuido, de forma que se perciba el sistema como un todo más que como una colección de componentes independientes.

TRANSPARENCIA	DESCRIPCIÓN
Acceso	Esconde diferencias en la representación de datos y como un recurso es accedido.
Locación	Esconde la locación del recurso.
Concurrencia	Esconde que un recurso pueda ser compartido por varios usuarios competidores.
Replicación	Esconde desde donde es utilizado un recurso compartido por varios usuarios competidores.
Fallas	Esconde la falla y recuperación de un recurso.
Migración (Movilidad)	Esconde el movimiento de un recurso a otra locación.
Relocación	Esconde que un recurso pueda ser movido a otra locación mientras está en uso.
Persistencia	Esconde si un recurso (software) esta en memoria o disco.

KMC © 2020

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

SD: Conceptos de Software

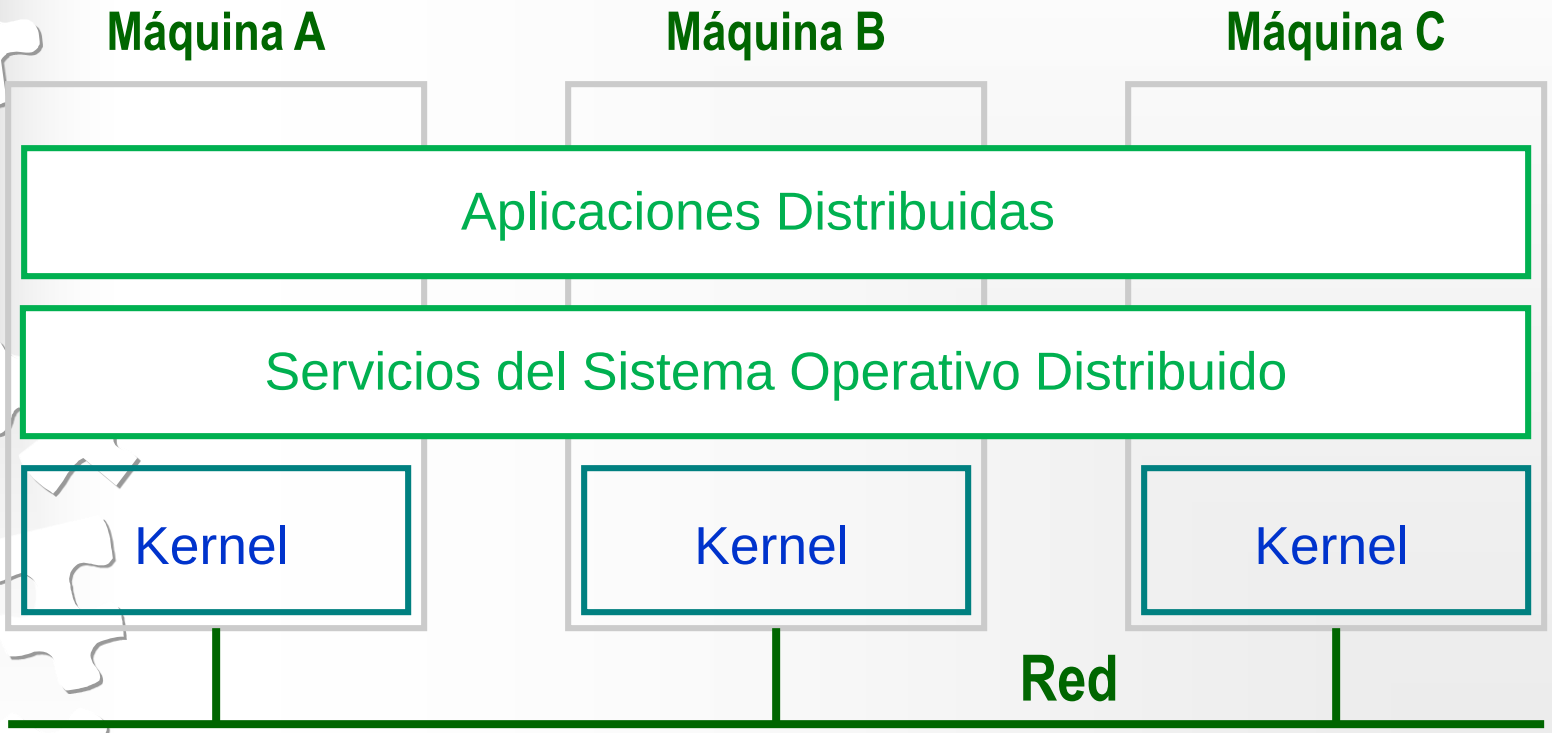
- SOD (Sistemas Operativos Distribuidos)
- SOR (Sistemas Operativos de Red)
- Middleware

Sistema	Descripción	Objetivo Principal
SOD	Sistemas operativos fuertemente acoplados para multiprocesadores y multicomputadoras homogéneas	Esconde y maneja los recursos de hardware
SOR	Sistemas operativos flojamente acoplados para multicomputadoras heterogéneas (LAN y WAN).	Ofrece servicios locales a clientes remotos
Middleware	Capa adicional sobre un SOR implementando servicios de propósito general.	Provee distribución transparente

KMC © 2020

SD: Conceptos de Software

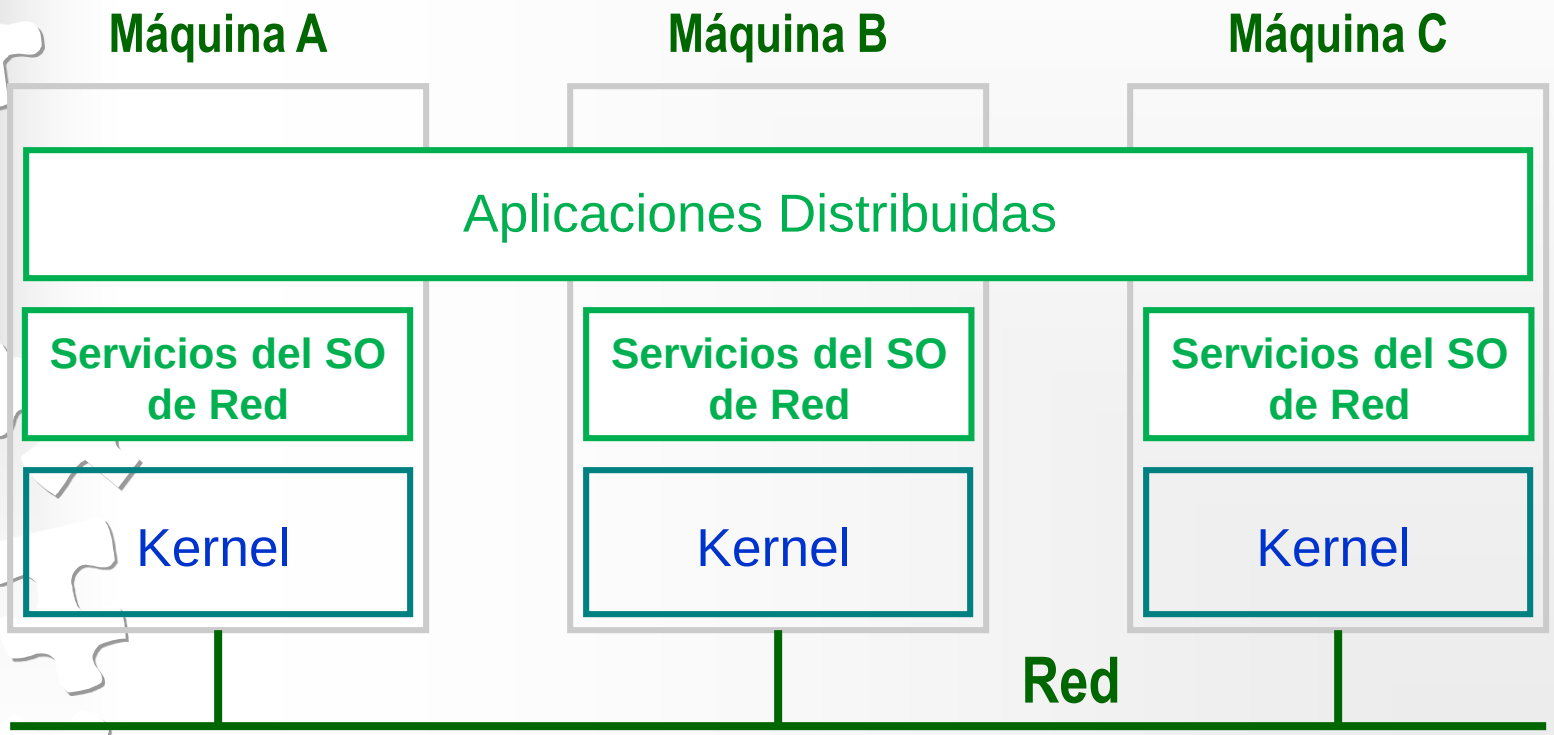
Sistemas Operativos Multicomputadora - 1



KMC © 2020

SD: Conceptos de Software

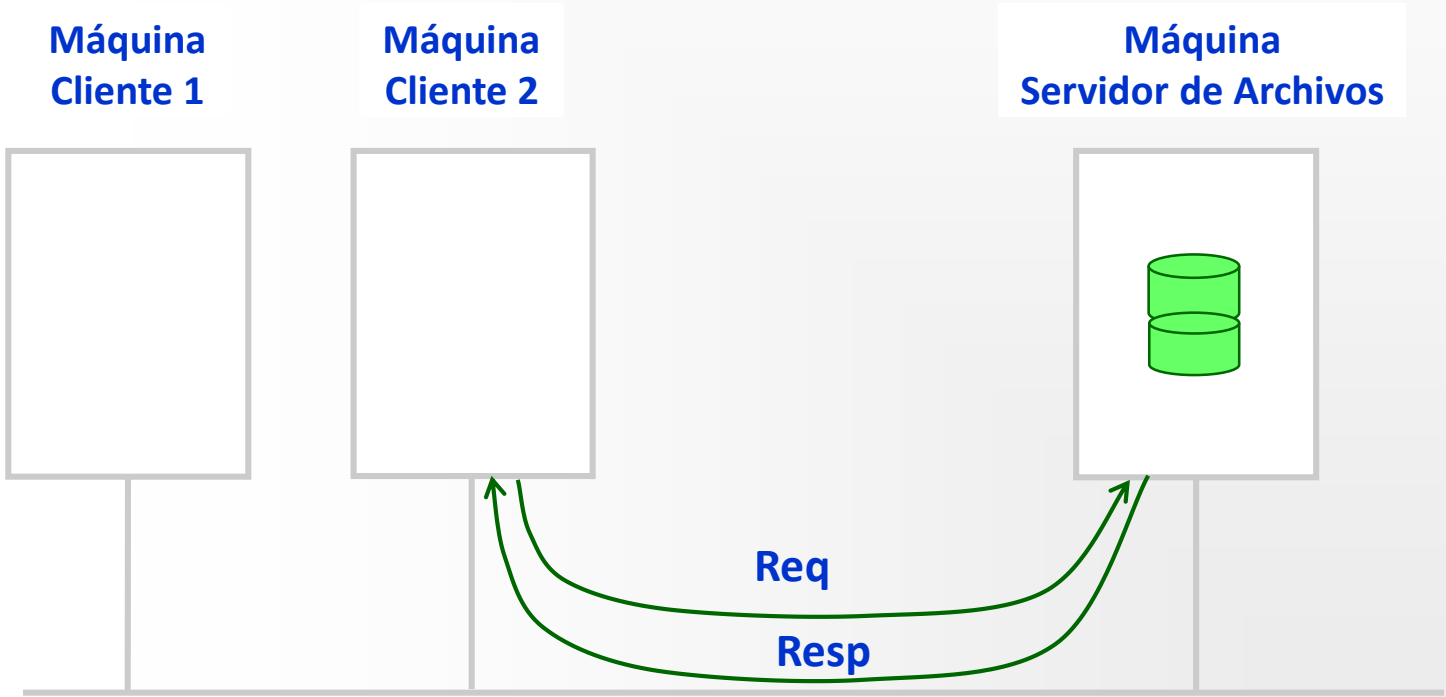
Sistema Operativo de Red



KMC © 2020

SD: Conceptos de Software

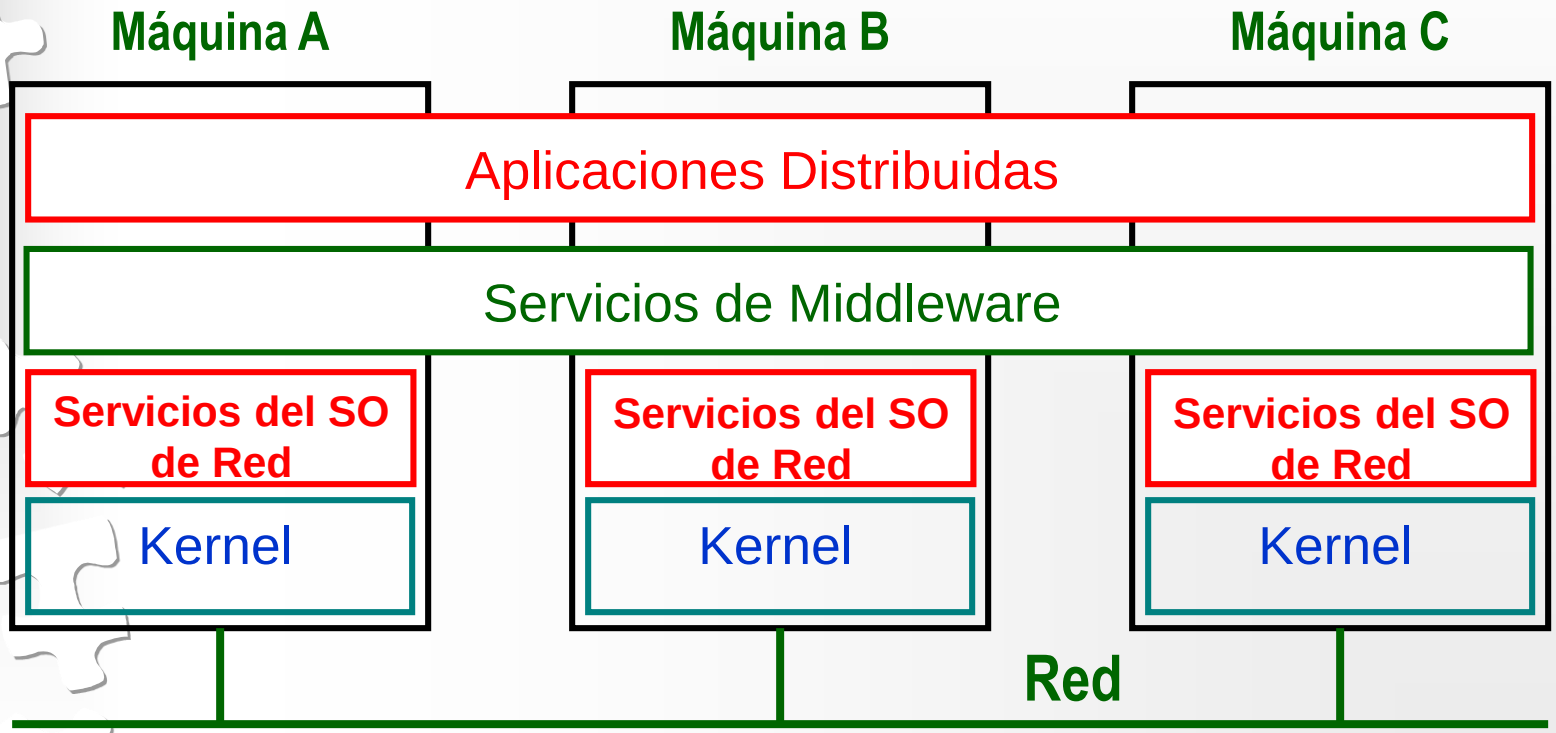
Sistema Operativo de Red



KMC © 2020

SD: Conceptos de Software

POSICIÓN DEL MIDDLEWARE



KMC © 2020

SD: Conceptos de Software

COMPARACIÓN ENTRE SISTEMAS

Item	SO Distribuido		SO de Red	SO basado en Middleware
	Multiproces.	Multicompu.		
Grado de transparencia	Muy alto	Alto	Bajo	Alto
Igual SO en todos los nodos	Si	Si	No	No
Número de copias de SO	1	N	N	N
Base para comunicaciones	Memoria compartida	Mensajes	Archivos	Modelo específico
Manejo de Recursos	Global, central	Global, distribuido	Por nodo	Por nodo
Escalabilidad	No	Moderada	Si	Varía
Apertura	Cerrado	Cerrado	Abierto	Abierto

KMC © 2020

Agenda

1. Conceptos Generales
2. Sistemas Distribuidos.
    1. Definición
    2. Desafíos
    3. Conceptos de Software
    4. Modelos de Sistemas

KMC © 2020

Sistemas Distribuidos: Modelos de Sistemas

- Modelos FÍSICOS – capturan la composición del hardware de un sistema en términos de las computadoras y las redes de interconexión.
- Modelos ARQUITECTÓNICOS – describen el sistema en términos de las tareas computacionales y de comunicación realizadas por los elementos.
- Modelos FUNDAMENTALES – describen una perspectiva abstracta para examinar un aspecto individual de un sistema distribuido.
  - ⇒Modelo de Interacción
  - ⇒Modelo de Fallo
  - ⇒Modelo de Seguridad

KMC © 2020

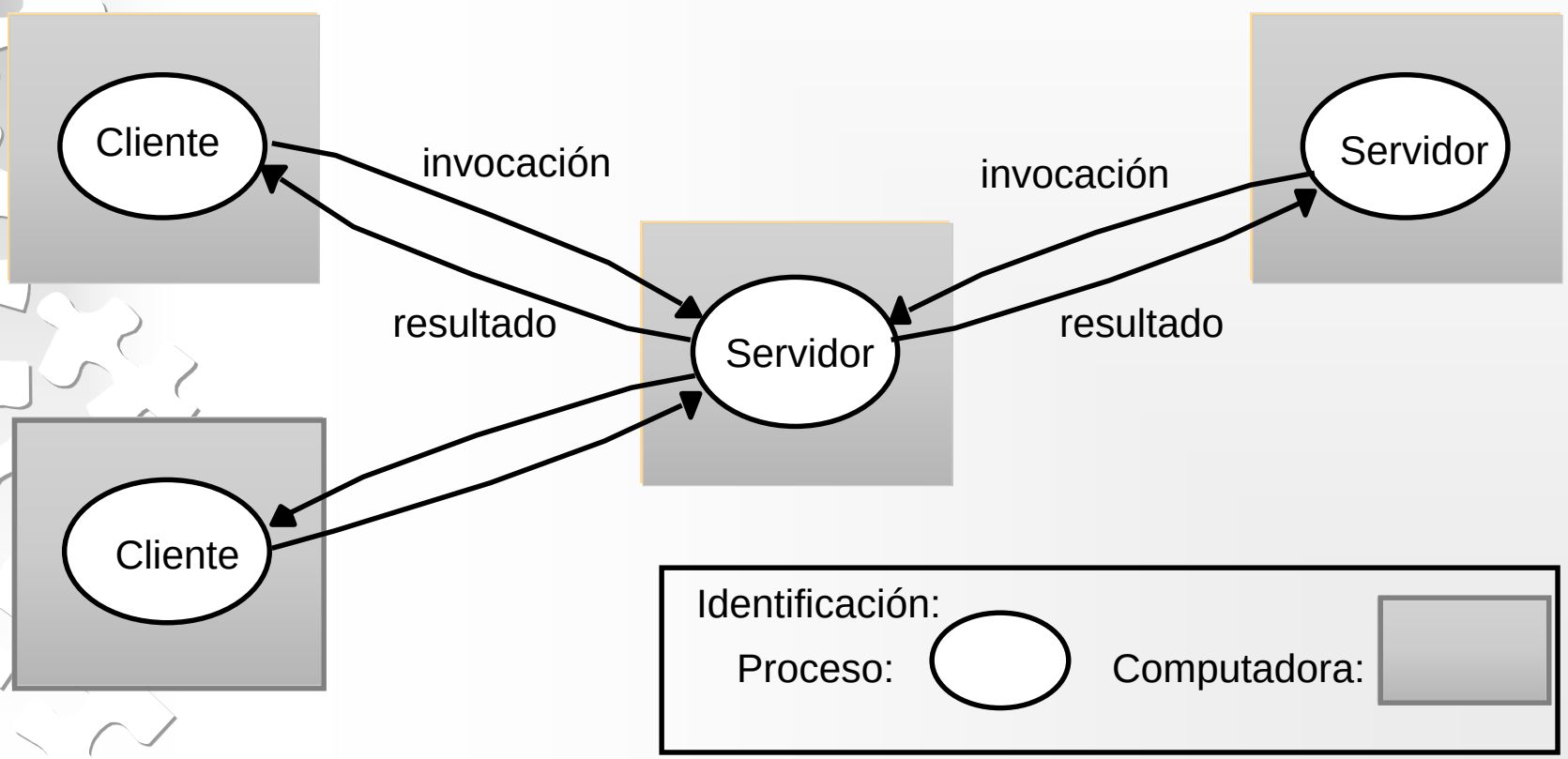
SD Modelo de Sistema: ARQUITECTÓNICO

- Elementos Arquitectónicos
  - ⇒Entidades
  - ⇒Paradigmas de comunicación
  - ⇒Roles y responsabilidades
  - ⇒Mapeo sobre la infraestructura física

KMC © 2020

SD Arquitectónico: Roles y Responsabilidades

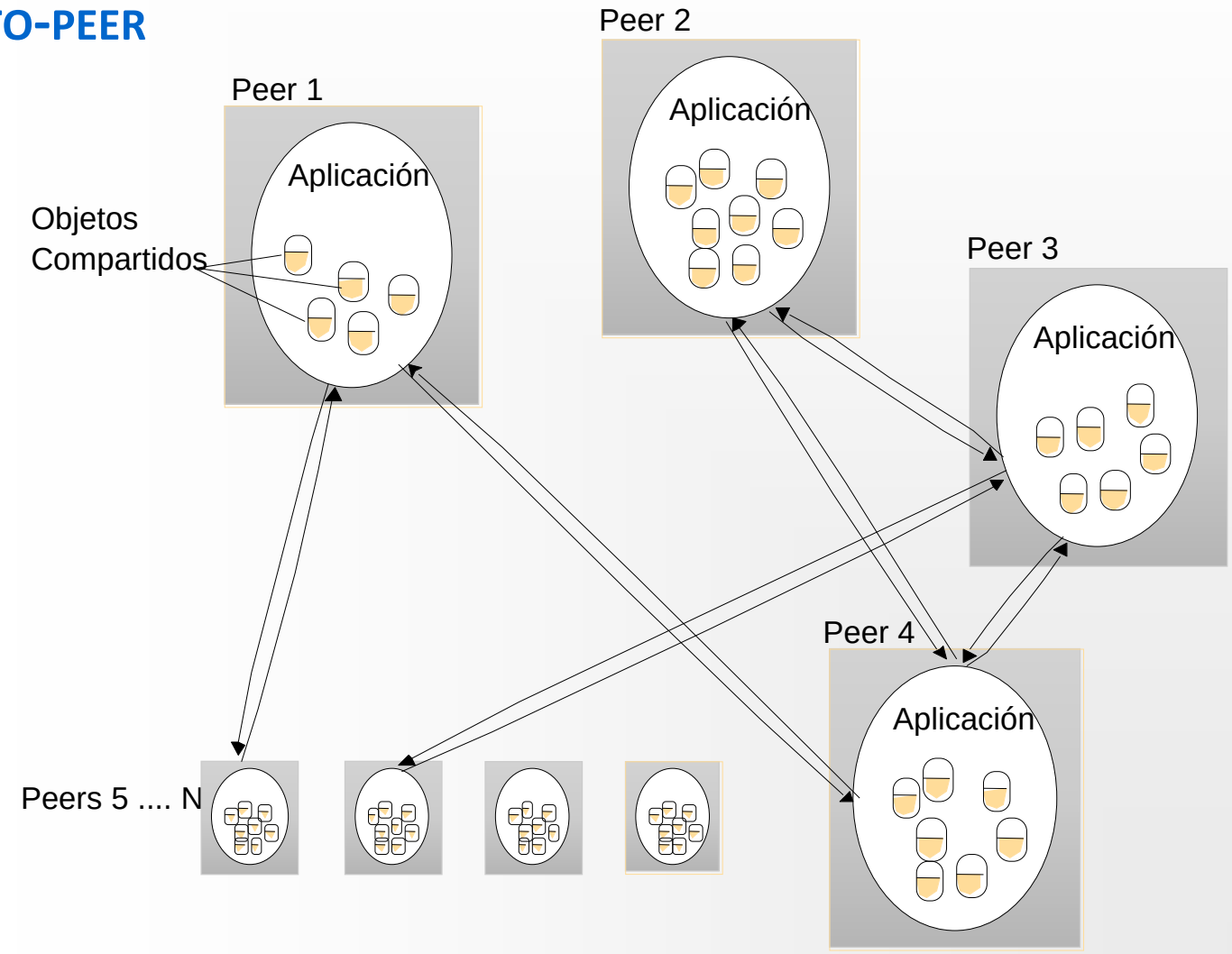
Cliente-Servidor



KMC © 2020

SD Arquitectónico: Roles y Responsabilidades

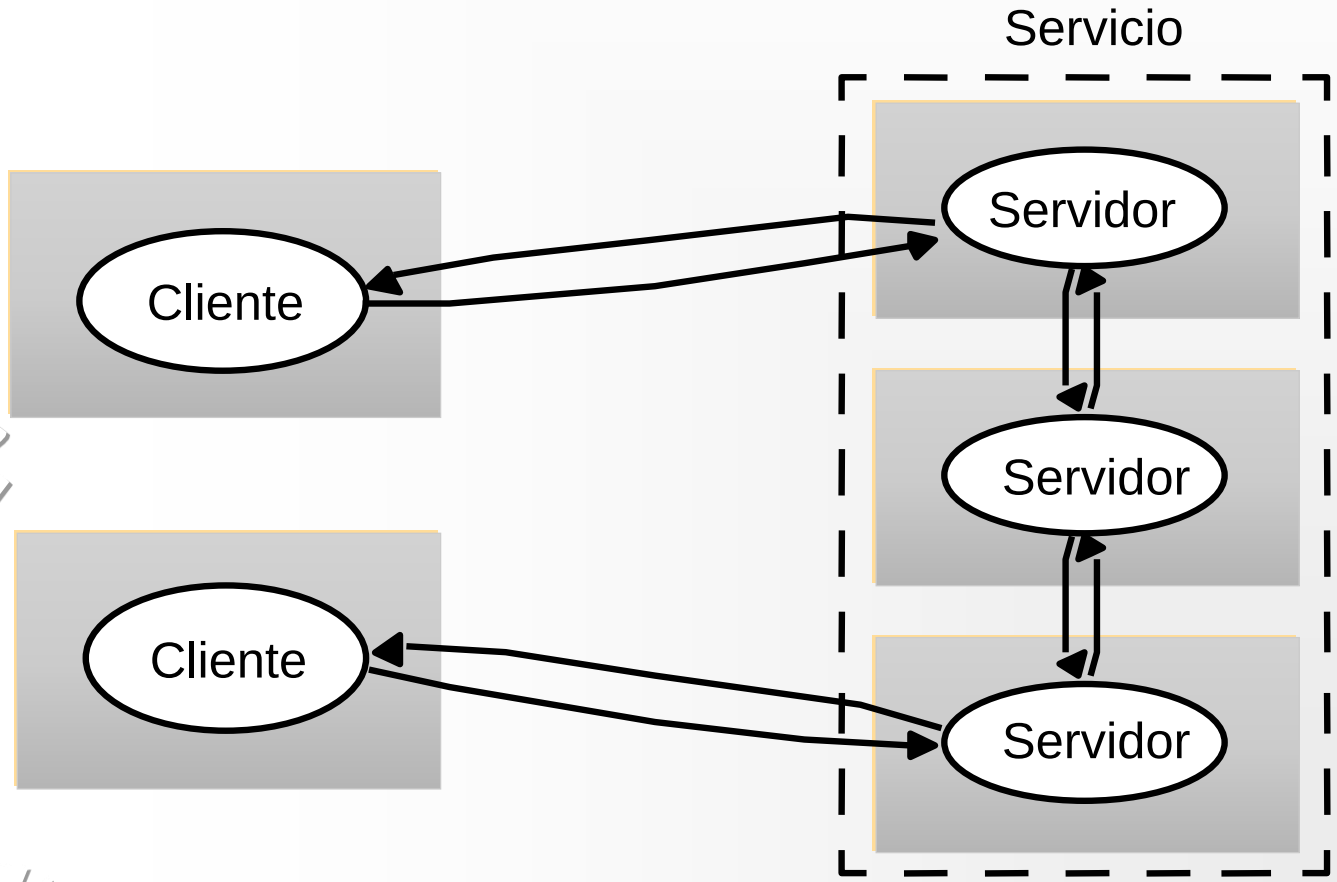
Peer-to-peer



KMC © 2020

SD Arquitectónico: Ubicación

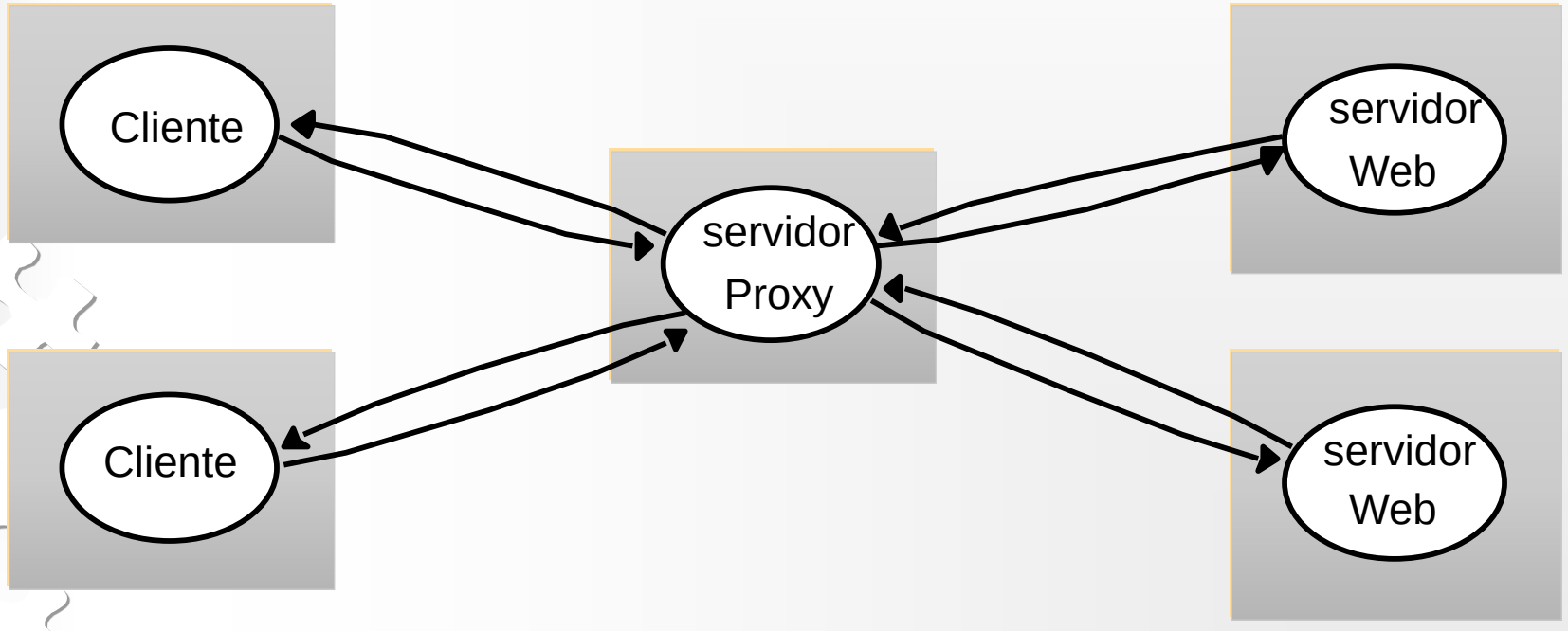
Servicio provisto por múltiples servidores



KMC © 2020

SD Arquitectónico: Ubicación

Cache – Ejemplo Servidor Proxy

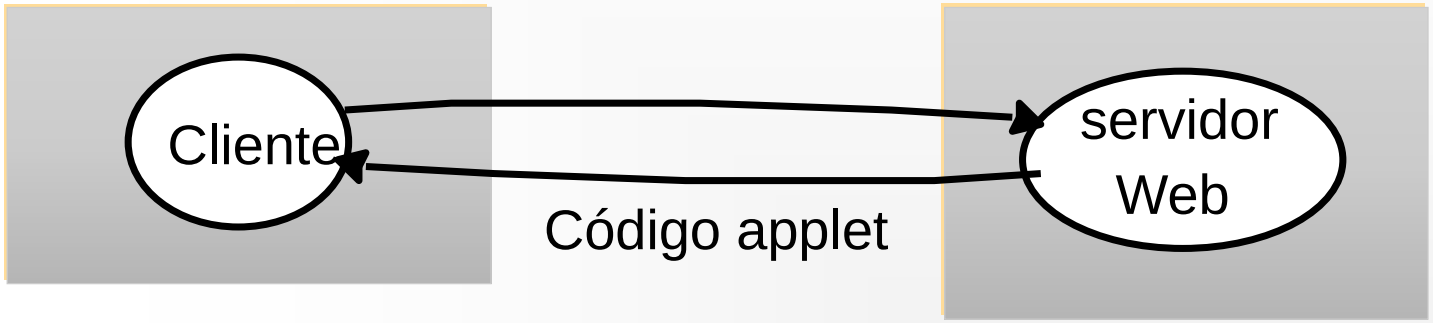


KMC © 2020

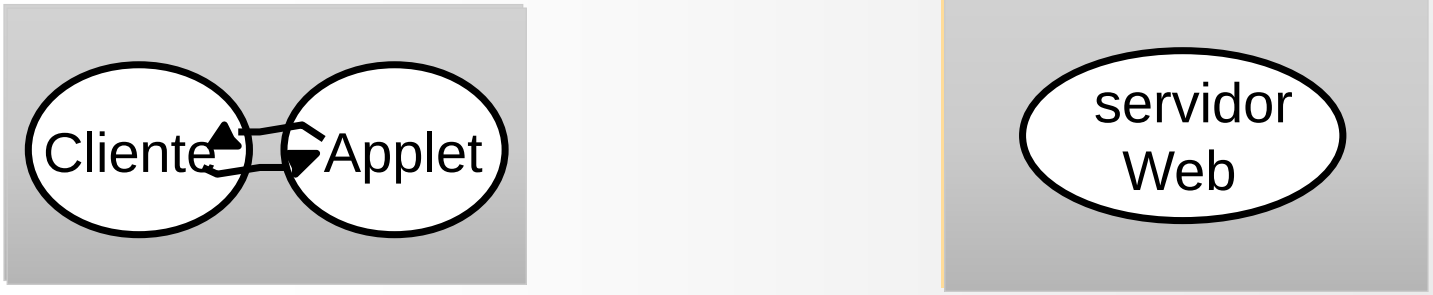
SD Arquitectónico: Ubicación

Código Móvil

a) El requerimiento del cliente resulta en la bajada de un código applet



b) El cliente interactúa con el applet



KMC © 2020

SD Modelo de Sistema: ARQUITECTÓNICO

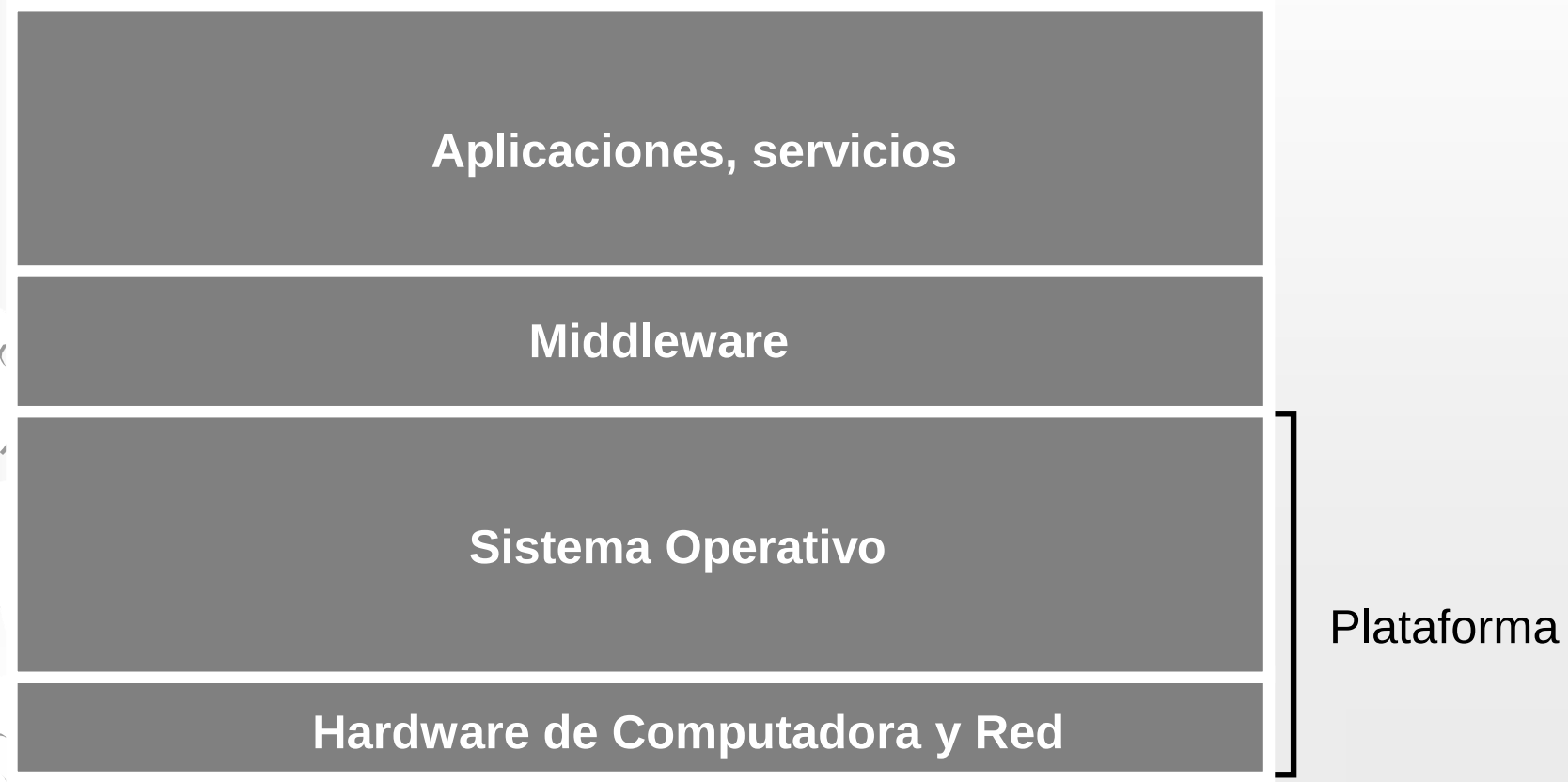
Patrones Arquitectónicos

- Capas
- Tiers
- Clientes Delgados

KMC © 2020

SD Modelo de Sistema: Arquitectónico - Patrones

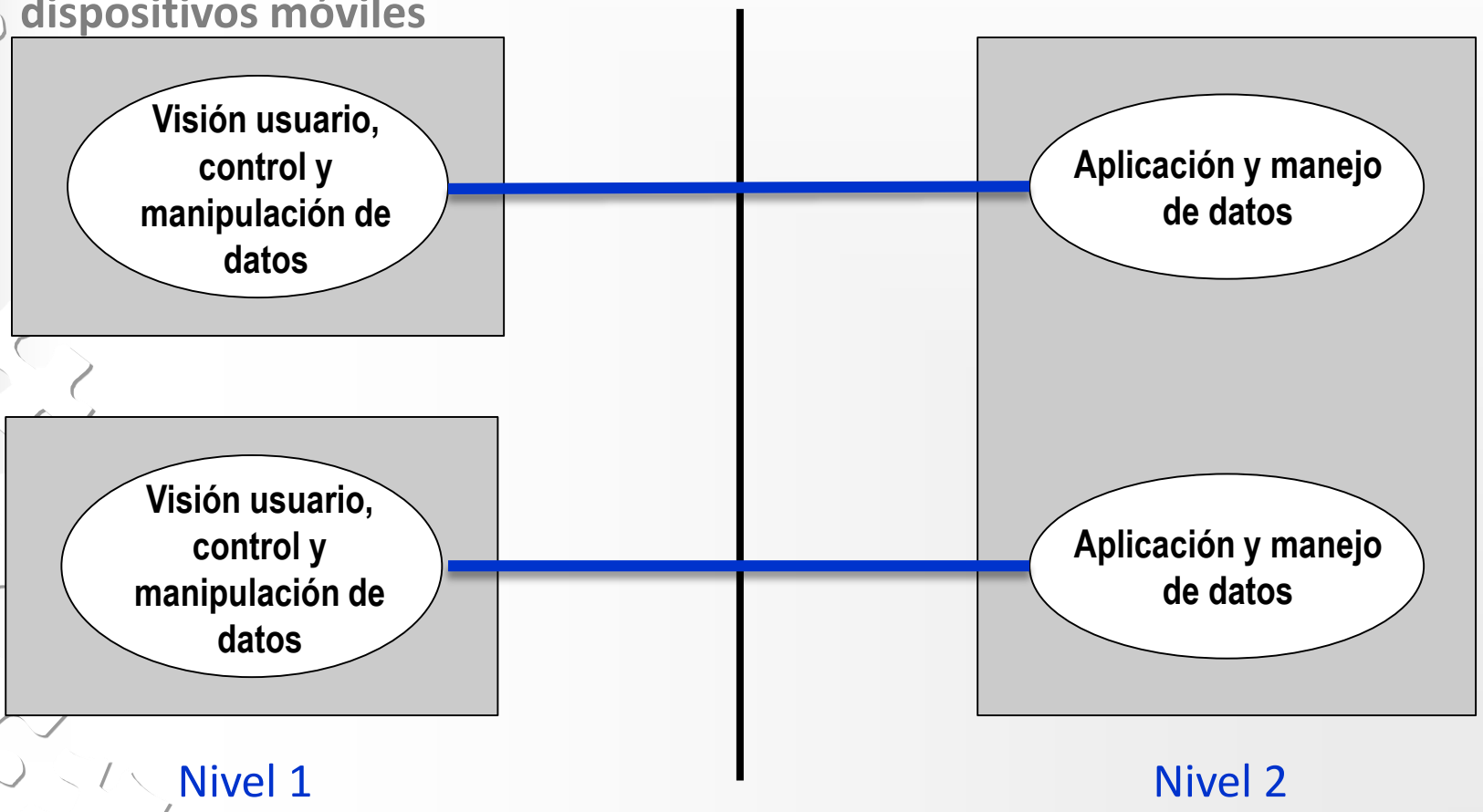
Capas de Software y Hardware



KMC © 2020

SD Modelo de Sistema: Arquitectónico - Patrones

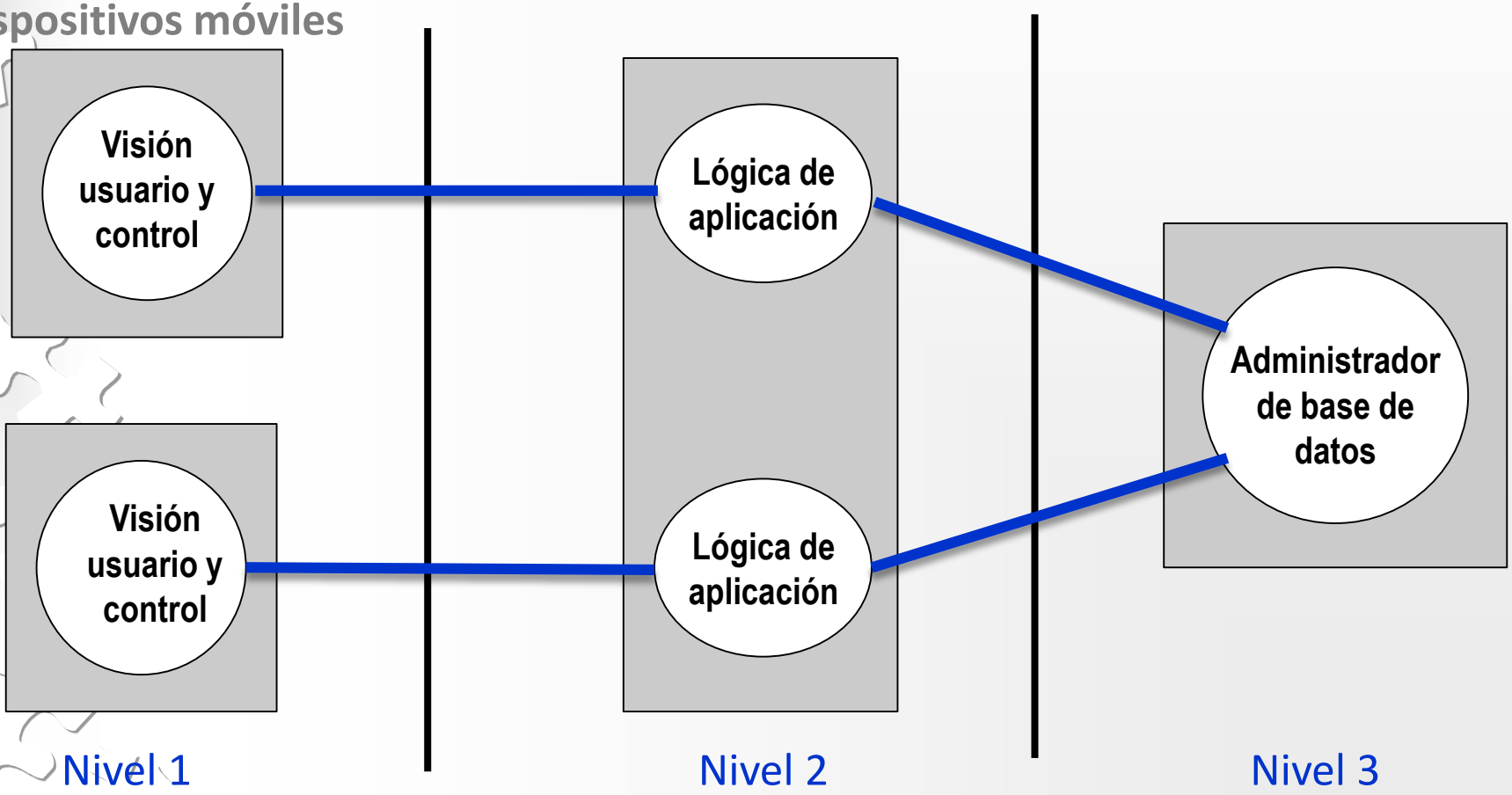
Dos-Tiers



KMC © 2020

SD Modelo de Sistema: Arquitectónico - Patrones

Tres-Tiers



KMC © 2020

SD Modelo de Sistema: Arquitectónico - Patrones

Clientes Delgados

Red de computadoras o PCs

servidor de cómputo

KMC © 2020

SD Modelo de Sistema: Arquitectónico – Patrones

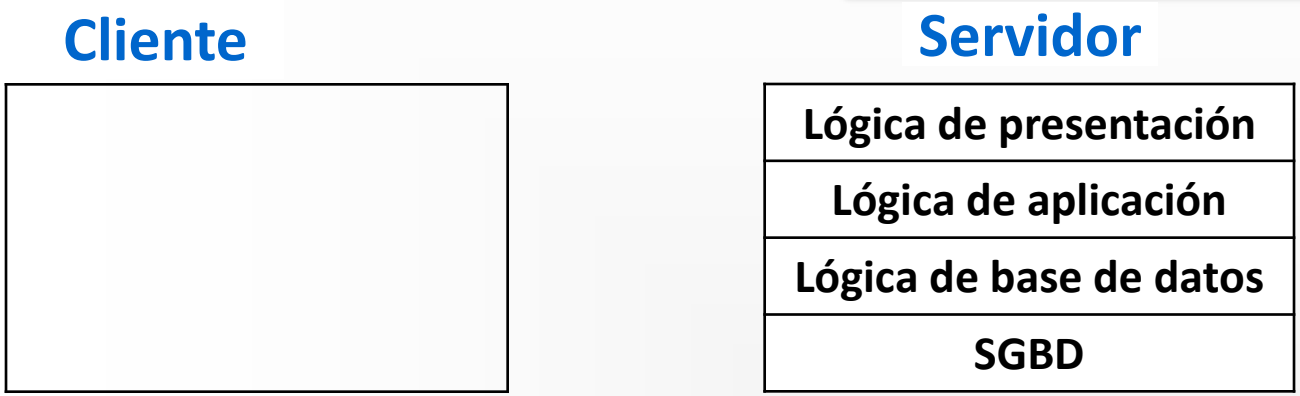
Cliente-Servidor

- Las funciones reales de la aplicación pueden repartirse entre cliente y servidor de forma que:
  - Se optimicen los recursos de la red y de la plataforma.
  - Se optimice la capacidad de los usuarios para realizar varias tareas.
  - Se optimice la capacidad para cooperar el uno con el otro en el uso de recursos compartidos

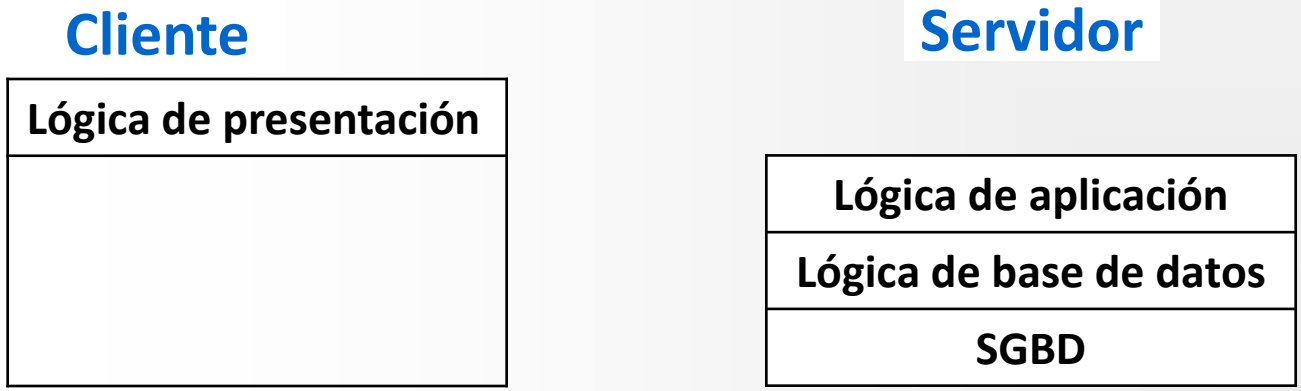
KMC © 2020

SD Modelo de Sistema: Arquitectónico – Patrones

Cliente-Servidor



(a) Proceso basado en una máquina central

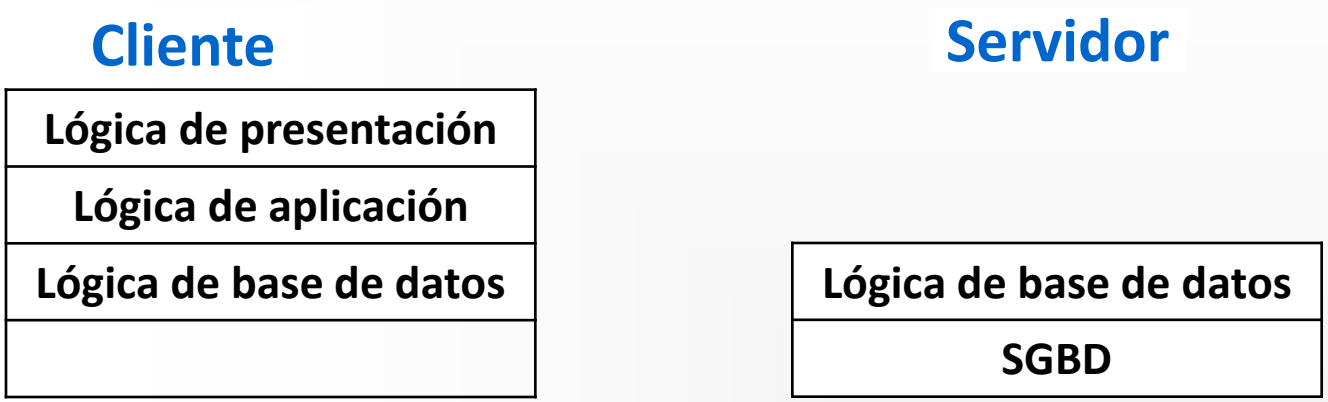


(b) Proceso basado en el servidor

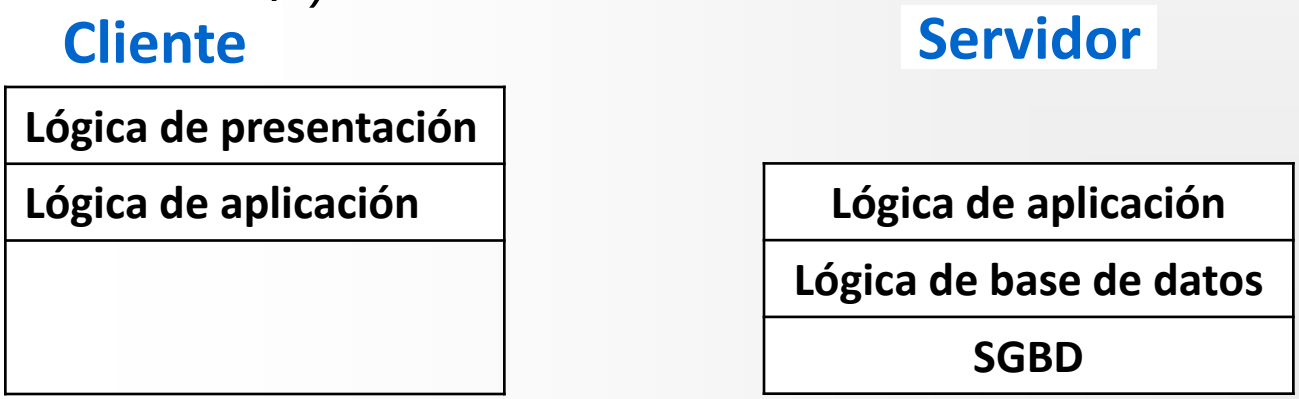
KMC © 2020

SD Modelo de Sistema: Arquitectónico – Patrones

Cliente-Servidor



(c) Proceso basado en el cliente



(d) Proceso cooperativo

KMC © 2020

Sistemas Distribuidos: Modelos Fundamentales

El objetivo de un modelo es:

- Hacer explícitas todas las premisas relevantes sobre los sistemas que estamos modelando.
- Hacer generalizaciones respecto a lo que es posible o no, dadas las premisas anteriores.

KMC © 2020

Sistemas Distribuidos: Modelos Fundamentales

Modelo de Interacción

Los sistemas distribuidos están compuestos por varios procesos, interactuando de manera compleja.

- Las prestaciones de las comunicaciones son con frecuencia una característica limitante.
  - Latencia (demora entre el inicio de la transmisión y el comienzo de la recepción)
  - *Ancho de banda*
  - *Jitter* es la variación en el tiempo invertido en completa el reparto de una serie de mensajes.
- No es posible mantener una única noción global del tiempo

KMC © 2020

Sistemas Distribuidos: Modelos Fundamentales

Modelo de Interacción

Sistemas distribuidos síncronos

1.- el tiempo de ejecución de cada etapa de un proceso tiene ciertos límites inferior y superior conocidos.

2.- cada mensaje transmitido sobre un canal se recibe en un tiempo limitado conocido.

3.- cada proceso tiene un reloj local cuya tasa de deriva sobre el tiempo real tiene un límite conocido.

Sistemas distribuidos asíncronos

No tiene límite para la velocidad de ejecución de un proceso, demora en la transmisión de un mensaje y deriva del reloj.

KMC © 2020

Sistemas Distribuidos: Modelos Fundamentales

Modelo de Fallo

- Fallos por omisión
  - ✓De procesos (fallo-parada, timeouts)
  - ✓De comunicaciones (fallo omisión de envío, de recepción, de canal)
- Fallos arbitrarios (fallo bizantino)
- Fallos de temporización se aplican a los sistemas distribuidos síncronos
- Enmascaramiento de fallos
- Fiabilidad y comunicación uno a uno
  - ✓Validez
  - ✓Integridad

KMC © 2020

Sistemas Distribuidos: Modelos Fundamentales

Modelo de Seguridad

La seguridad de un sistema distribuido puede lograrse asegurando los procesos y los canales empleados para sus interacciones y protegiendo los objetos que encapsulan contra el acceso no autorizado

KMC © 2020

Resumen

Conceptos Generales

- La computación paralela está orientada a resolver un problema de manera eficiente, dividiéndolo en subtareas para alcanzar un objetivo.
- La computación distribuida está orientada a que varias aplicaciones puedan compartir recursos y colaborar entre ellas.
- Las motivaciones para el crecimientos de estas áreas se encuentra en el uso eficiente de los recursos, el rendimiento, el crecimiento incremental, etc.

KMC © 2020

Resumen

Sistemas Distribuidos

- ¿Qué es? Dos definiciones fueron presentadas, con similitudes y diferencias de acuerdo al contexto dónde se lo ubique.
- Estos sistemas tienen ventajas, pero también desventajas y limitaciones como:
  - No existe una memoria global.
  - Establecer un estado global es complejo.
  - No se puede asegurar un tiempo global.
- El diseño de un sistema distribuido requiere considerar desafíos para el diseño como: heterogenidad, transparencia, extensibilidad, …

KMC © 2020

Resumen

Sistemas Distribuidos

- Otro aspecto al considerar el diseño es el modelo arquitectónico. Esta vista considera los siguientes aspectos:
  - los componentes y cómo estos se comunican entre sí,
  - los roles que representa y en dónde ubicarán cada uno de los elementos.
  - Uno de los modelos más ampliamente utilizados es el modelo cliente-servidor
- Los modelos fundamentales ayudan a razonar sobre las propiedades del sistema distribuido en términos de, por ejemplo, rendimiento, confiabilidad y seguridad.

KMC © 2020

Bibliografía:

- Leopold, C; “Parallel and Distributed Computing: A Survey of Models, Paradigms and Approaches”, John Wiley & Son, Inc, 2001.
- Tanenbaum, A.S.; van Steen, Maarten; “Distributed Systems: Principles and Paradigms”. 3rd. Edition, 2017. 2nd Edition, Prentice Hall, 2007
- Coulouris,G.F.; Dollimore, J. y T. Kindberg; “Distributed Systems: Concepts and Design”. 5th Edition Addison Wesley, 2011.

KMC © 2020