

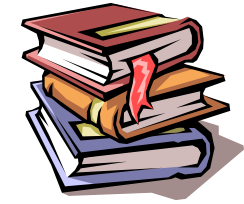


Desarrollos de Sistemas Distribuidos
Departamento de Lenguajes y Sistemas
Informáticos
E.T.S. de Ingeniería Informática
Universidad de Granada

Profesores:

José Luis Garrido Bullejos
Manuel Noguera García (teoría)
Carlos Rodríguez Domínguez (prácticas)

Temario de Teoría



- 1. Introducción a los Sistemas Distribuidos**
Conceptos. Características y objetivos. Paradigmas de aplicaciones distribuidas. Un modelo de referencia de computación distribuida.
- 2. Comunicación y Sincronización en Sistemas Distribuidos**
Sistemas de paso de mensajes. Comunicación Cliente/Servidor. Citas.
- 3. Coordinación**
Tiempo lógico. Relojes lógicos. Algoritmos distribuidos de coordinación.
- 4. Middlewares**
Comunicación en middlewares. Clasificación y tipos de middlewares. Middlewares básicos (RPC de Sun y RMI de Java).
- 5. Sistemas Cliente/Servidor**
Modelos Cliente/Servidor de n-etapas y configuraciones. Modelos Peer-to-Peer. Modelos funcional y de comportamiento. Diseño. Computación Móvil y Cloud
- 6. Aplicaciones distribuidas y cliente/servidor**
Arquitectura y metodología de desarrollo. Replicación. Otros *middlewares* avanzados (CORBA, SOA y Espacios de Tuplas). Sistemas abiertos.

Temario de Prácticas



- **Diseño e implementación de sistemas y aplicaciones distribuidas Cliente/Servidor y P2P con diferentes soportes:**
 - Llamadas a Sistemas Operativos.
 - Herramientas de programación: RPC de SUN.
 - Lenguaje de programación: RMI de Java.
 - Servicios web y dispositivos móviles.
 - Publish-Subscribe

Evaluación

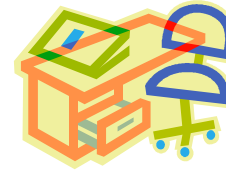
- **Evaluación continua:** control por tema y/o entrega de ejercicios de problemas planteados
- **Examen final:** 5/6 preguntas cortas y/o ejercicios de problemas.
- **Calificación:** 40% Teoría + 10% Portafolio (clase) + 50% Prácticas. Se aprueba superando al menos el 50% de cada parte.
- **Prerrequisitos Recomendados:** Sistemas Operativos I

y II

Bibliografía

1. Coulouris, G.F. et al: *Distributed Systems - Concepts and Design*. (4/e) Addison-Wesley, 2005.
2. Liu M.L.: *Distributed Computing – Concepts and Applications*. Addison-Wesley, 2003.
3. Bacon, J.: *Concurrent Systems: An Integrated Approach to Operating Systems, Distributed Systems and Databases (Open University Edition)*. Addison-Wesley, 2002.
4. Andrews, G.A.: *Concurrent Programming - Principles and Practice*. The Benjamin/Cummings Publishing Company, 1991.
5. Orfali, R.: *Client/Server Programming with Java and CORBA*. Wiley & Sons, 1996.

Tutorías



Profesor	Horario	Lugar
Manuel Noguera	M: 10 – 13h X: 10 – 13h	Despacho 22, 3ª planta (<i>ETSIIT</i>)
José Luis Garrido	X: 10 – 14h	Despacho 12, 3ª planta (<i>ETSIIT</i>)
Carlos Rodríguez	L: 11 – 14h J: 11 – 14h	Despacho 1-I15, 1ª planta (<i>CITIC</i>)

- (confirmar por **email**)

– Consultar Webs:

- <http://lsi.ugr.es/lsi/mnoguera>; <http://www.ugr.es/~mnoguera>
- <http://lsi.ugr.es/lsi/jgarrido>; <http://lsi.ugr.es/~jlgarrid>
- <http://lsi.ugr.es/lsi/crodriguez>; <http://www.ugr.es/~carlosrodriguez>



Desarrollo de Sistemas Distribuidos

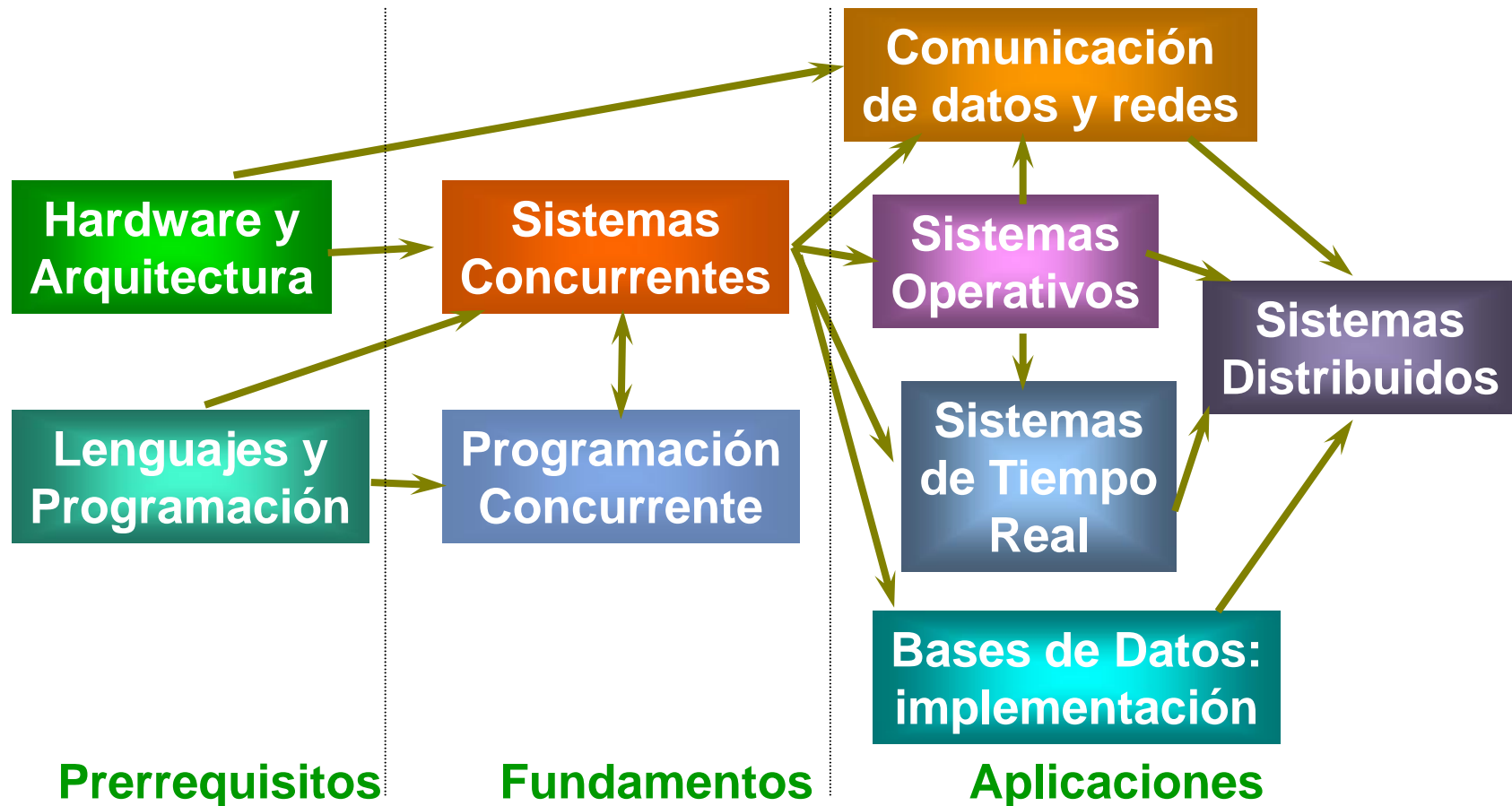
Tema 1

Introducción a los Sistema Distribuidos


Contenidos

1. Disciplinas relacionadas
2. Clasificación y definición
3. Características y objetivos
4. Paradigmas de aplicaciones distribuidas
5. Modelos de referencia

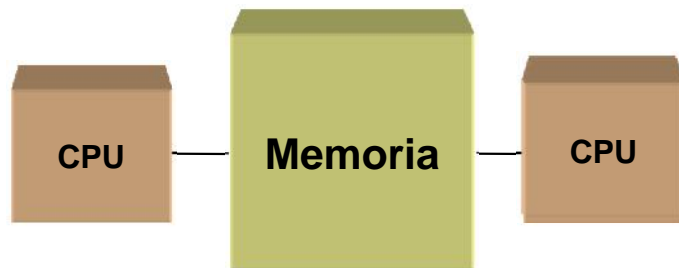
Disciplinas relacionadas



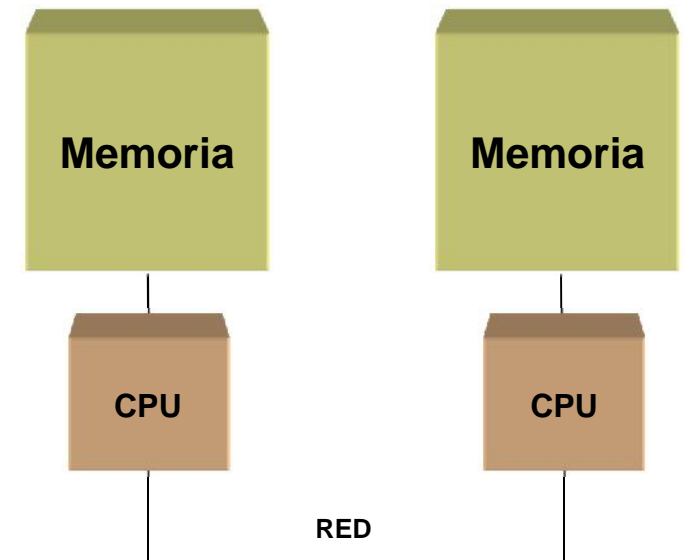
Clasificación Sistemas Distribuidos - *Hardware*

Arquitectura	Sistema Distribuido
Fuertemente acopladas (<i>"closely or tightly coupled"</i>)	Multicomputadoras Comunicaciones en microsegundos (Transputer, Cosmic Cube, Hipercubo, ...)
	LAN's Comunicaciones en milisegundos
Débilmente acopladas (<i>"loosely coupled"</i>)	WAN's, MANETs Comunicaciones en segundos

Modelos de Programación Concurrente - *Software*

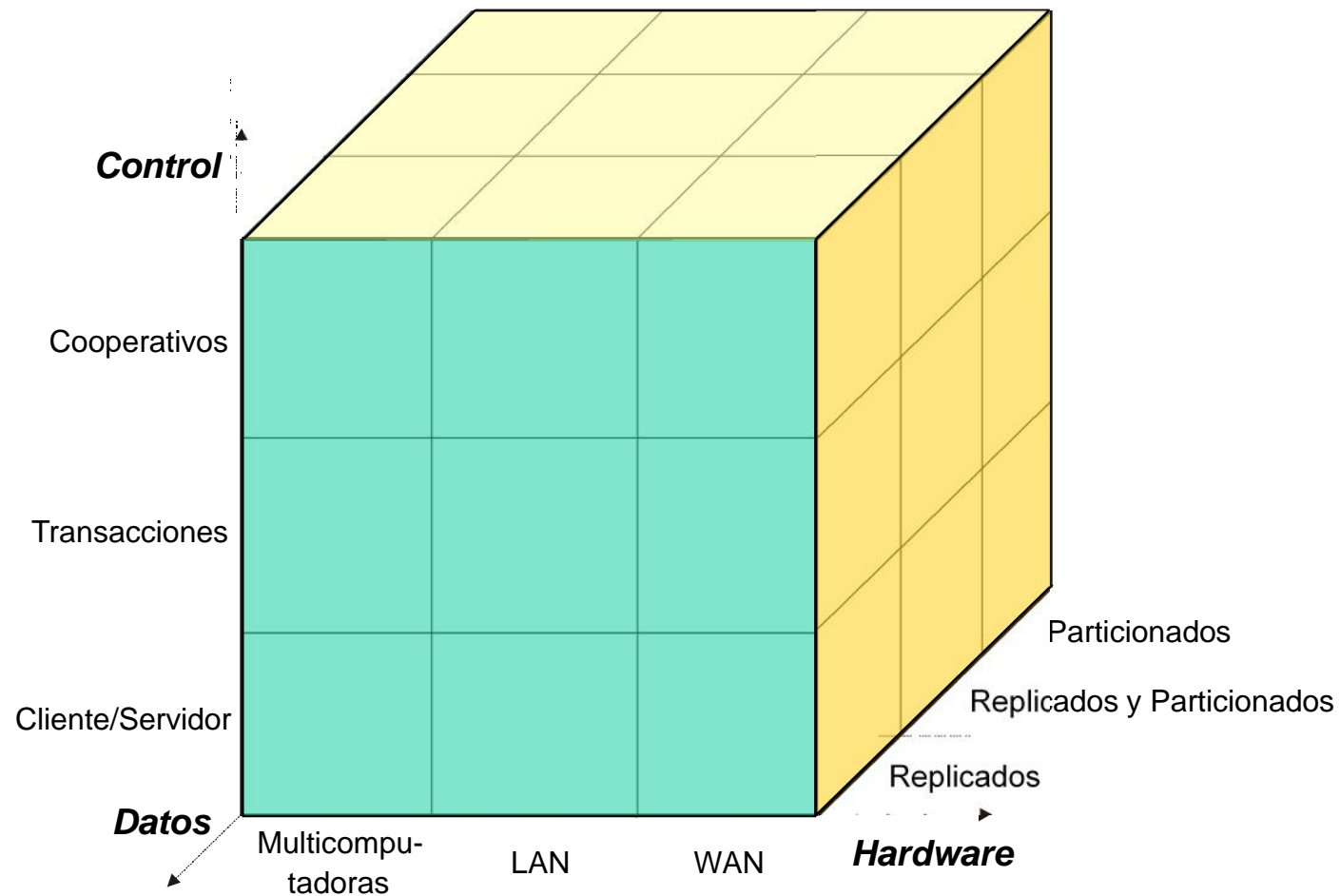


Memoria Compartida

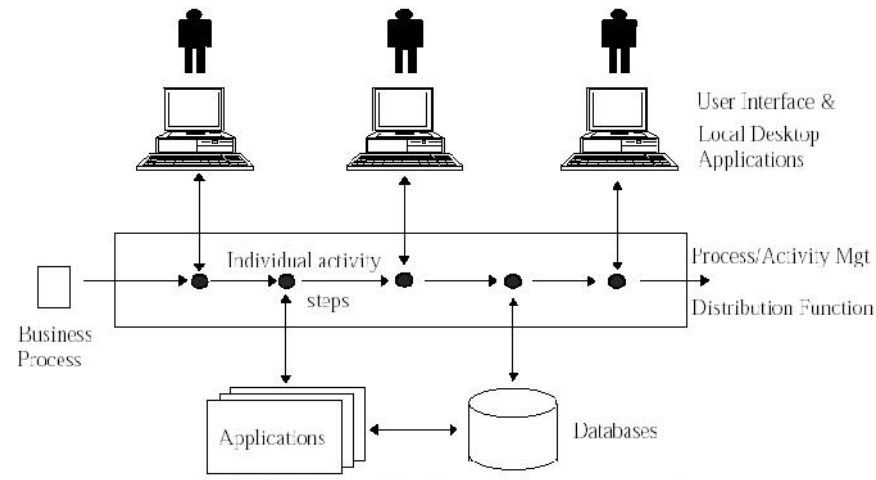


Memoria Distribuida

Modelo Genérico para tipos de Sistemas Distribuidos *Hw & Sw* (Adaptación del Modelo de *Enslow*)

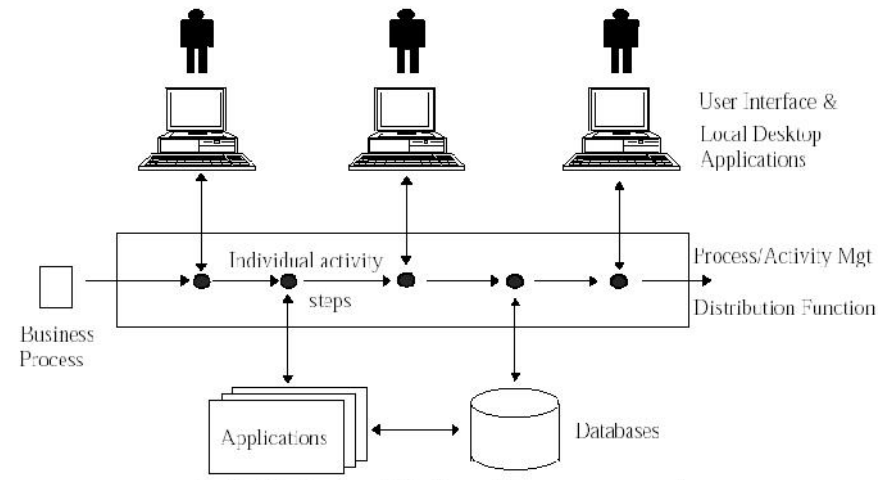


Definición de Sistema Distribuido

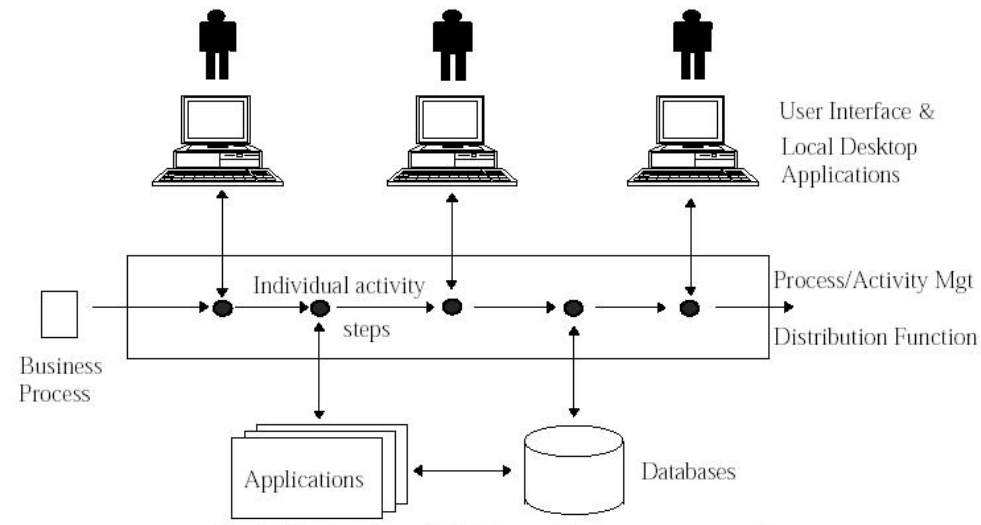


Conjunto de computadoras autónomas, pero enlazadas por una red y con software diseñado para producir y facilitar una computación integrada

Definición de Sistema Distribuido



Sistema en el que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes



Colección de computadoras independientes que se presentan ante los usuarios como un único sistema coherente

- *Aspecto hardware: Los computadores son independientes*
- *Aspecto software: los usuarios piensan que existe un único sistema*

Características y Objetivos

1. Compartición de recursos

- **Recurso** ð **hardware** ð **software**
 - En sistemas distribuidos, los recursos son gestionados por programas que ofrecen una **interfaz** de comunicación y se ejecutan en diferentes dispositivos de cómputo
- Tendencia actual al **control descentralizado**: *groupware*
- **Políticas y métodos** de manejo con requisitos comunes:
 - Mismo esquema de **denominación** (e.g. *Universal Resource Identifier* o URI)
 - **Asociación** de nombres de recursos a direcciones de comunicación
 - Coordinación acceso concurrente por **consistencia**

Características y Objetivos

2. Sistema abierto

- Determina si el sistema puede ser **ampliado** de varias formas
- Hace referencia tanto al hardware como software
- **Unix**, primer sistema abierto:
 - **Desarrolladores de aplicaciones** tienen acceso a todos los servicios del Sistema Operativo
 - **Vendedores de hardware y gestores de sistemas** ya que el sistema puede ser ampliado
 - **Vendedores software y usuarios** ya que es independiente del hardware
- La utilización de **protocolos estándares** (*“Inter Process Communication”* o IPC) de comunicación incrementa el ámbito de los sistemas abiertos

Características y Objetivos

3. Concurrencia

- **Concurrencia y paralelismo** surgen naturalmente en sistemas distribuidos:
 - Actividades **independientes** de los usuarios
 - **Independencia** de recursos hardware
 - Localización de procesos en diferentes computadoras

Características y Objetivos

4. Escalabilidad

- El **software de sistema y aplicación** no debería cambiar cuando el sistema aumenta
- El trabajo que implica el procesamiento de una única petición para acceder a un recurso compartido debe ser independiente del tamaño de la red
- En todo caso, las disminuciones de rendimiento del sistema deben ser moderada
- **Principal desafío:** diseñar el software del sistema distribuido de forma que permanezca **efectivo**. **Técnicas** a aplicar:
 - Replicación de datos
 - Utilización de cachés
 - Despliegue de servicios
 - Sistemas abiertos: introducción de servicios y reimplementación de los existentes, independencia de proveedores

Características y Objetivos

5. Tolerancia a fallos:

- El diseño se basa en dos propuestas:
 - Redundancia *hardware*
 - Recuperación *software*
- Cara a fallos *hardware* un SD proporciona un alto nivel de **disponibilidad** (medida tiempo que está disponible para su uso)
- Las redes hasta el momento no han sido redundantes y, por tanto, se busca un diseño seguro

Características y Objetivos

6. Transparencia:

- SD como un todo
- Es el principal objetivo en el diseño del software del sistema
- Formas de transparencia básicas:
 - **Acceso:** mismas operaciones sobre entidades locales y remotas
 - **Localización:** entidades accedidas sin importar localización
 - **Concurrencia:** operaciones concurrentes sobre la misma entidad sin interferir

Características y Objetivos

6. Transparencia:

- Formas de transparencia avanzadas:
 - **Replicación:** existencia de varias instancias de una misma entidad sin conocimiento por parte del programa o usuario. Copia de los archivos más usados en los diferentes servidores sin ser advertido por los usuarios
 - **Fallo:** ocultación fallos hardware y software
 - **Migración:** movimiento de entidades sin afectar a programas ni usuarios
 - **Rendimiento:** reconfiguración del sistema según carga
 - **Escalabilidad:** no cambia la estructura del sistema cuando éste o las aplicaciones se amplían

Paradigmas de Aplicaciones Distribuidas

- **Aplicaciones paralelas de alto rendimiento:**
 - Decrementan el **tiempo de respuesta**
 - Ventaja de **escalabilidad** de SD frente a sistemas multiprocesador o multicore
 - Se pueden clasificar en cuanto al **grano del paralelismo** (tiempo de computación entre comunicaciones):
 - Grano grueso (*“large- or coarse-grain”*)
 - Grano medio (*“medium-grain”*)
 - Grano fino (*“fine-grain”*)
 - Grano medio y grueso adecuado para SD débilmente acoplados

Paradigmas de Aplicaciones Distribuidas

- **Aplicaciones tolerantes a fallos:**
 - Incrementan **fiabilidad** (“*reliability*”) y **disponibilidad**
 - SD potencialmente seguros debido a la **propiedad de fallo parcial**:
 - Cuanto más débilmente acoplado más fiabilidad
 - Seguridad mediante la **replicación** de funciones o datos de las aplicaciones

Paradigmas de Aplicaciones Distribuidas

- **Aplicaciones con especialización funcional:**
 - Aplicación como **colección de servicios** (e.g. SO Amoeba)
 - Forma natural de diseñar estas aplicaciones con **implementaciones alternativas**: centralizada, distribuida, replicada
 - Proporcionan **alto rendimiento y seguridad**
 - Unos servicios de pueden comunicar con otros
 - **Fácil escalabilidad**

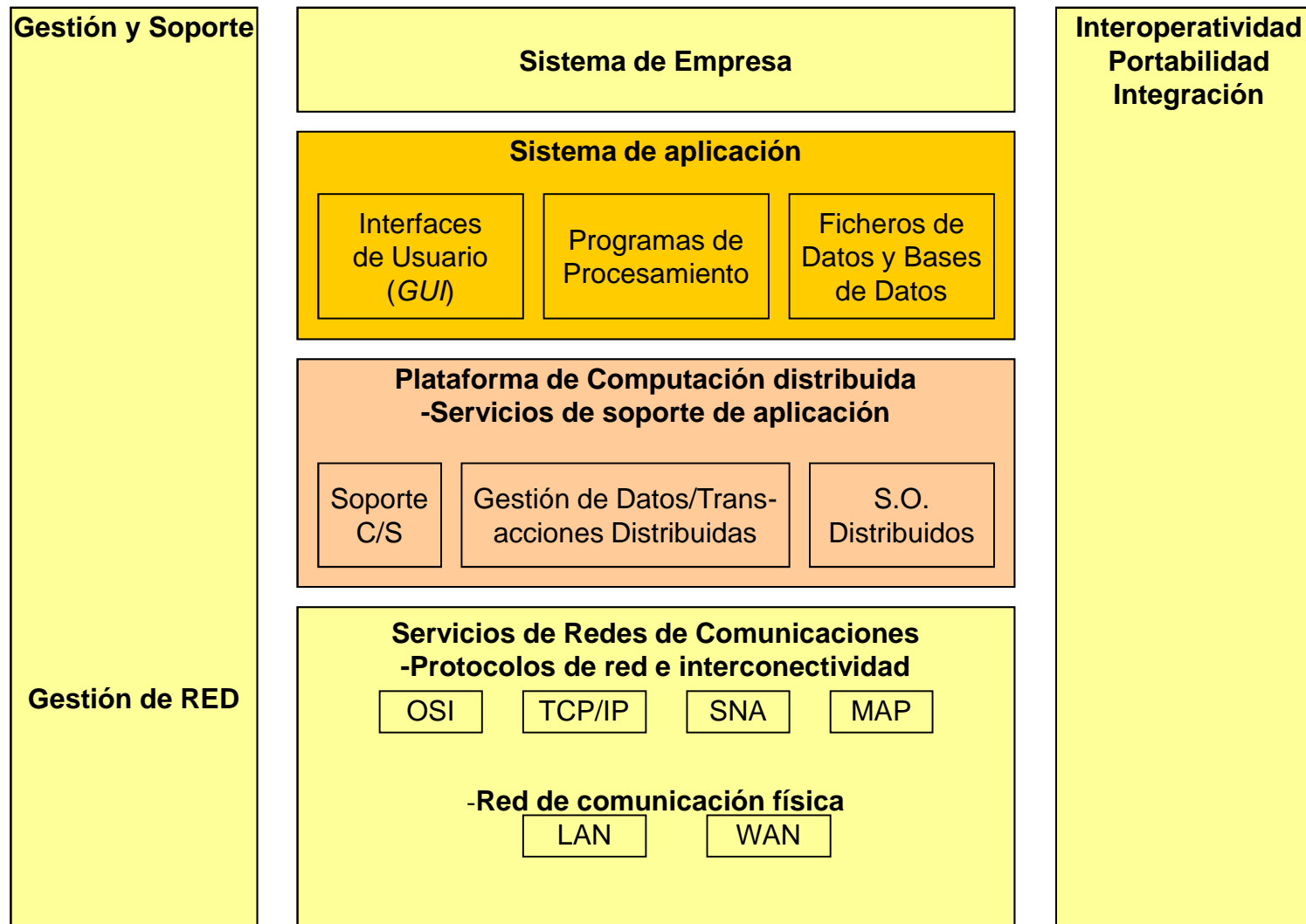
Paradigmas de Aplicaciones Distribuidas

- **Aplicaciones inherentemente distribuidas:**
 - Groupware: sistema basado en computadoras que soporta grupos de personas implicadas en tareas comunes y que proporciona una interfaz a un entorno compartido interactivo
 - Ejemplos más claros y exitosos:
 - Correo electrónico
 - *Workflow*: Software de gestión de flujos de trabajo

Modelo de referencia para sistemas distribuidos

- **Modelo de referencia:** “**Marco de trabajo** abstracto para la comprensión de las relaciones más relevantes entre las entidades de un determinado **entorno**” [SOA-OASIS 06]
- **Objetivo:** Desarrollar arquitecturas específicas o concretas, utilizando estándares consistentes o especificaciones que soporten el entorno en cuestión
- **Elementos:** Conjunto mínimo de conceptos, axiomas y relaciones en un dominio particular; independiente de estándares, tecnologías, implementaciones u otros detalles concretos específicos

Modelo de referencia para Sistemas Distribuidos



Modelo de referencia para Sistemas Distribuidos

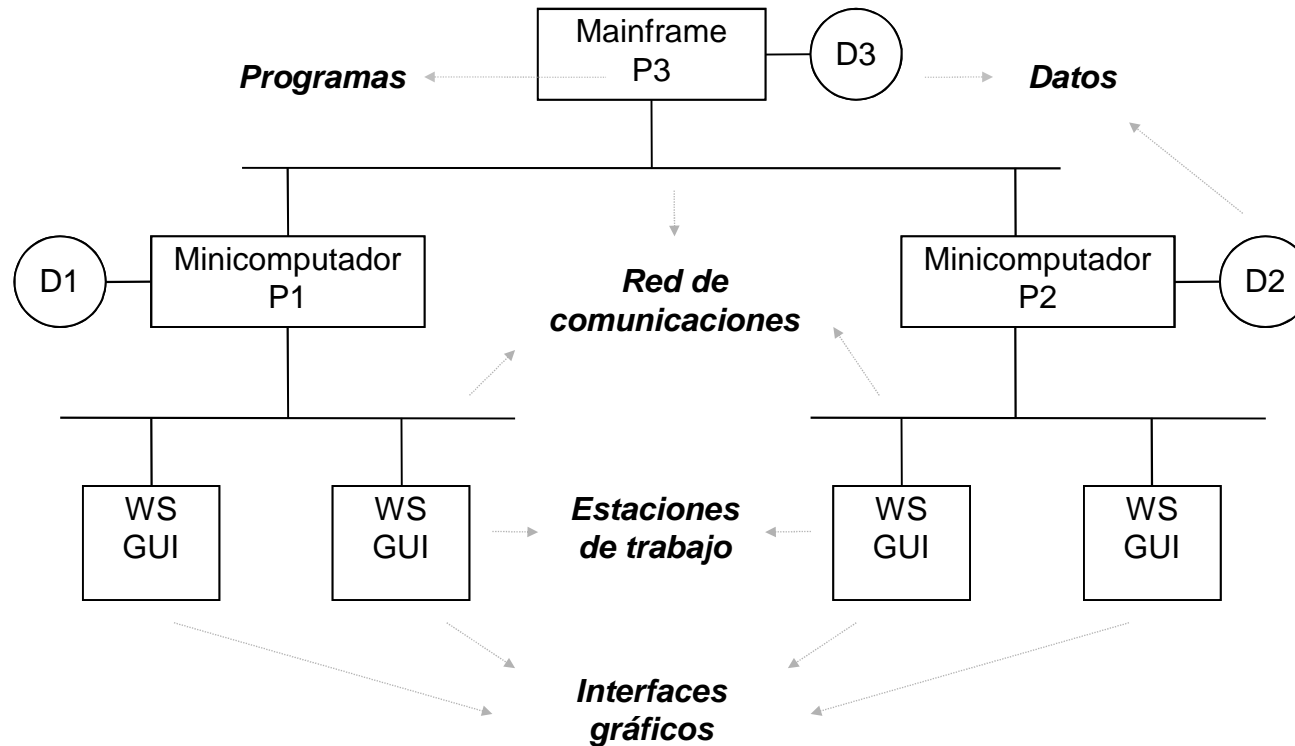
- **Sistemas de Empresa:**
 - Establecen **requisitos y líneas de trabajo** más allá de la utilización de la tecnología.
 - Pueden ser:
 - **Centralizados**
 - **Distribuidos:** las actividades se realizan en diferentes lugares

Modelo de referencia para Sistemas Distribuidos

- **Sistemas de Aplicación:**
 - Proporcionan **soporte automatizado** al sistema de empresa
 - Tecnologías de la información haciendo uso de los siguientes **componentes:**
 - Datos
 - Programas
 - Interfaces de usuario
 - **Configuraciones centralizadas** producen cuellos de botella en rendimiento y disponibilidad
 - **Inconveniente:** diseño de aplicaciones distribuidas es una tarea compleja debido a decisiones difíciles en cuanto a la elección de nuevas tecnologías de redes y servicios

Modelo de referencia para Sistemas Distribuidos

- **Sistemas de Aplicación:**
 - Configuración común:



Modelo de referencia para Sistemas Distribuidos

- **Plataforma de Computación Distribuida:**
 - Las aplicaciones distribuidas dependen de diversas tecnologías: computadoras, redes, servicios de ficheros,...
 - Se proporcionan dos **tipos de servicios**: servicios de red y de soporte a aplicaciones
 - **Servicios de red:**
 - Nivel más bajo de servicios: transferencia de información
 - Arquitectura de red que describe componentes físicos, funciones realizadas por componentes e interfaces entre estos
 - Modelo de referencia OSI desarrollado como estándar

Modelo de referencia para Sistemas Distribuidos

- **Plataforma de Computación Distribuida:**
 - **Servicios de soporte a aplicaciones:**
 - Responsables del nivel más alto de interconectividad entre aplicaciones
 - Se llevan cabo a través de:
 - Emuladores de terminal
 - **Sistemas Cliente/Servidor** basados en el modelo conceptual de interacción entre procesos que se ejecutan asíncronamente y posiblemente en diferentes computadoras, los cuales pueden ser consumidores de servicios (clientes) o suministradores de éstos (servidores)
 - » Su construcción requiere solucionar cuestiones técnicas y de gestión
 - » **RPC** (*“Remote Procedure Call”*) es el mecanismo básico
 - » Computación Cliente/Servidor = Computación distribuida siguiendo el modelo Cliente/Servidor

Modelo de referencia para Sistemas Distribuidos

- **Plataforma de Computación Distribuida:**
 - **Servicios de soporte a aplicaciones:**
 - Se llevan a cabo a través de:
 - **Sistemas de Gestión Distribuida de Datos y Transacciones:**
 - » Versión sofisticada de sistema cliente/servidor que permite almacenar, acceder y manipular de forma distribuida datos manteniendo la integridad
 - » **Niveles de transparencia:** lectura (localización), actualizaciones (replicación), ejecución de transacciones (conurrencia) y fallos
 - **Sistemas Operativos Distribuidos:**
 - » Sintetizan y amplían servicios de soporte a aplicaciones
 - » Funcionalidades: total transparencia, selección de servicio y encaminamiento automáticos

Modelo de referencia para Sistemas Distribuidos

- **Sistemas abiertos:**
 - **Interoperatividad:** sistemas pueden trabajar entre sí a través de interfaces bien definidas
 - **Portabilidad:** mover un sistema a otro entorno, i.e., desligar aplicaciones de plataformas
 - **Integración:** facilitar la utilización de un sistema proporcionando acceso uniforme a usuarios finales
 - Ejemplos: OSF's DCE, OMG's Corba

Modelo de referencia para Sistemas Distribuidos

- **Gestión y soporte:**
 - Relacionado con herramientas y técnicas para administrar un SD
 - **Gestión** aborda:
 - Cómo manejar y soportar tecnologías de computación distribuida
 - Cómo emplear estas tecnologías para mejorar los procesos de gestión

Modelo de referencia para Sistemas Distribuidos

- **Gestión y soporte:**

- **Soporte** aborda las siguientes cuestiones:

- Selección hardware/software
 - Configuración, instalación y mantenimiento
 - Copias de seguridad y recuperación de datos
 - Detección y corrección de fallos
 - Reconfiguración de los componentes de la aplicación en diferentes sitios
 - Administración de seguridad

Modelo de referencia RM-ODP

- **RM-ODP (“*Reference Model for Open Distributed Processing*”):**

ISO/IEC 10746

Modelo de referencia que proporciona un marco de trabajo para la estandarización del Procesamiento Distribuido Abierto, soporta distribución, independencia de plataforma y tecnología, y portabilidad, además de una arquitectura de empresa

Modelo de referencia RM-ODP

- Los principales elementos son:
 - Un enfoque de **modelado de objetos** para la especificación del sistema
 - Especificación a través de **puntos de vista** separados, pero relacionados
 - Definición de una **infraestructura de sistema** que proporciona transparencia en la distribución para aplicaciones
 - Un **marco de evaluación** para el estudio de la correspondencia de sistemas al estándar RM-ODP

Modelo de referencia RM-ODP

- Contiene cuatro recomendaciones básicas y estándares:
 - **Introducción:** ámbito, justificación, explicación de conceptos clave y arquitectura general
 - **Bases:** definición de conceptos y marco de análisis para normalización
 - **Arquitectura:** especificación de las características que califican a un SD como abierto, también define puntos de vista, subdivisión del sistema y la relación entre sus componentes
 - **Semántica arquitectónica:** contiene la formalización de los conceptos de modelado interpretando conceptos en términos de construcciones de diferentes técnicas estándar de descripción formal

Modelo de referencia RM-ODP

- Un **punto de vista** (“*viewpoint*”) es una subdivisión de la especificación de un sistema. Cada uno de los siguientes puntos de vista utiliza los mismos conceptos básicos:
 - **Empresa:** propósito, ámbito y políticas del sistema que describen los requisitos de negocio y cómo alcanzarlos
 - **Información:** semántica de la información y su procesamiento
 - **Computacional:** descomposición funcional del sistema en objetos que interaccionan a través de interfaces
 - **Ingeniería:** mecanismos y funciones para soportar interacciones distribuidas entre objetos
 - **Tecnología:** elección de tecnologías para procesamiento, funcionalidad y presentación de la información

Modelo de referencia RM-ODP

- Puntos de vista para la especificación del sistema y entorno:

