

Fundamentos de Computación distribuida

sockets **RPC**
RMI

Contenido

introducción

Fundamentos de computación distribuida

Contenido

introducción

@ Programación SD

programación	T
1. Fundamentos de computación distribuida	1
2. Diseño de arquitecturas distribuidas	2
3. Tecnologías web y middleware	2
4. Seguridad	4
5. Coordinación y control de tiempo en SD	2.5
6. Sistemas de archivos distribuidos	2.5

introducción

resultados de aprendizaje

Contenido

introducción

- Comprender los conceptos de heterogeneidad, extensibilidad, escalabilidad, seguridad, concurrencia, tolerancia a fallos y transparencia en el contexto de los sistemas distribuidos.

Contenido

introducción

@ Fundamentos de computación distribuida

- Introducción a la computación distribuida
 - Evolución de los modelos de computación distribuida
 - Definiciones y propiedades
- Enfoques de sistemas distribuidos
 - SOR, SOD y Middleware

Contenido

introducción

- Ⓜ **Sistemas Distribuidos. Conceptos y Diseño**
G. Coulouris et al
Addison Wesley, 2001, 2012
Temas 4 y 5
- Ⓜ **Sistemas Distribuidos. Principios y paradigmas**
A.S. Tanenbaum
Prentice Hall , 2008
Temas 1 y 2
- Ⓜ **Computación Distribuida. Fundamentos y Aplicaciones**
M.L. Liu
Person Education , 2004
Temas 2,3,4,5,7 y 12
- Ⓜ **Service-Oriented Architecture: Concepts, technology and Design**
T. Erl
Prentice Hall, 2005
Temas 3,4,5 y 8

Contenido

introducción
computación

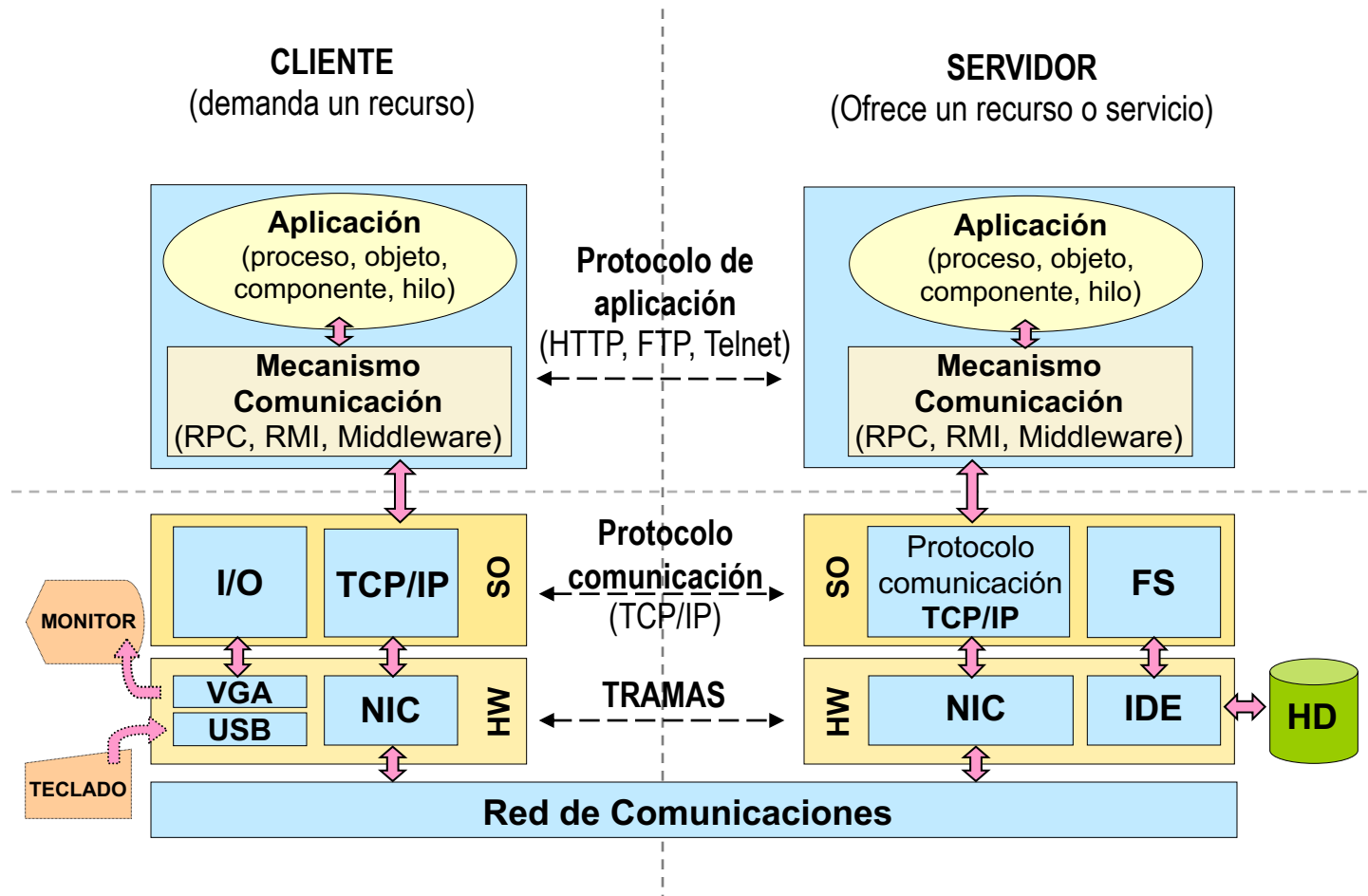
- ④ **Sistema Distribuido** → Elementos de computación independientes, interconectados, que comunican y coordinan sus acciones a través de una red de comunicaciones
- ④ Ejemplos de SD: Internet, intranets privadas, computación ubicua
- ④ **Computación Distribuida** → La que se desarrolla en un SD: servicios y aplicaciones de red

introducción

elementos de un sistema distribuido

Contenido

introducción
computación



Contenido

introducción
computación

@ Heterogeneidad

- Capacidad de los SD para estar compuestos por una **variedad** (de diferentes tipos) de componentes

- Estandarización
- Representación de datos
- Representación de código
- Representación de objetos
- Protocolos
- Integración
- Lenguajes intermedios
- UNIX, Windows

Hardware

- Representación datos

Red

- Ethernet, 802.11, ATM

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

- Capacidad de un SD de poder ser **extendido** pudiendo incorporar nuevos componentes:
 - Hardware
 - Redes
 - Computadores
 - Software
 - Aplicaciones
 - Servicios
 - Módulos

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

@ Escalabilidad

- Un SD es **escalable** si puede trabajar de forma correcta aunque se incrementen el número de:
 - Usuarios que lo utilizan
 - Recursos que se usan
 - Peticiones que se realizan a un servicio
 - Requerimientos de las aplicaciones
 - ...
- ¿Cómo se consigue?
 - Incorporación de forma dinámica de nuevos recursos HW/SW

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

@ Escalabilidad

@ Seguridad

- Entornos proclives a ataques externos
- Confidencialidad
- Integridad
- Disponibilidad
- Firewalls, SSL, HTTPS, Radius, Kerberos

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

@ Escalabilidad

@ Seguridad

@ Concurrencia y sincronización

- Posibilidad de que dos elementos del SD **accedan de forma simultánea** a un mismo recurso compartido
- Hay que garantizar el acceso concurrente para evitar inconsistencias
 - Acceso de forma controlada / exclusiva
 - Prioridad en los accesos a recursos
 - Secuenciación de las operaciones concurrentes

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

@ Escalabilidad

@ Seguridad

- Redundancia de componentes
- Sistemas de respaldo

@ Concurrencia y sincronización

@ Tolerancia a fallos

- Es necesario garantizar que el SD sea capaz de funcionar cuando uno de sus elemento falla – QoS (24x7)

Contenido

introducción
computación

@ Heterogeneidad

@ Extensibilidad

- De acceso: a recursos remotos como si fueran locales
- De ubicación/localización: a recursos remotos sin conocer su ubicación
- De movilidad: recurso cambia de ubicación sin que el usuario sea consciente
- De escalabilidad: el sistema crece en recursos sin que el usuario sea consciente
- Frente a fallos: el usuario no es consciente de fallos en HW/SW

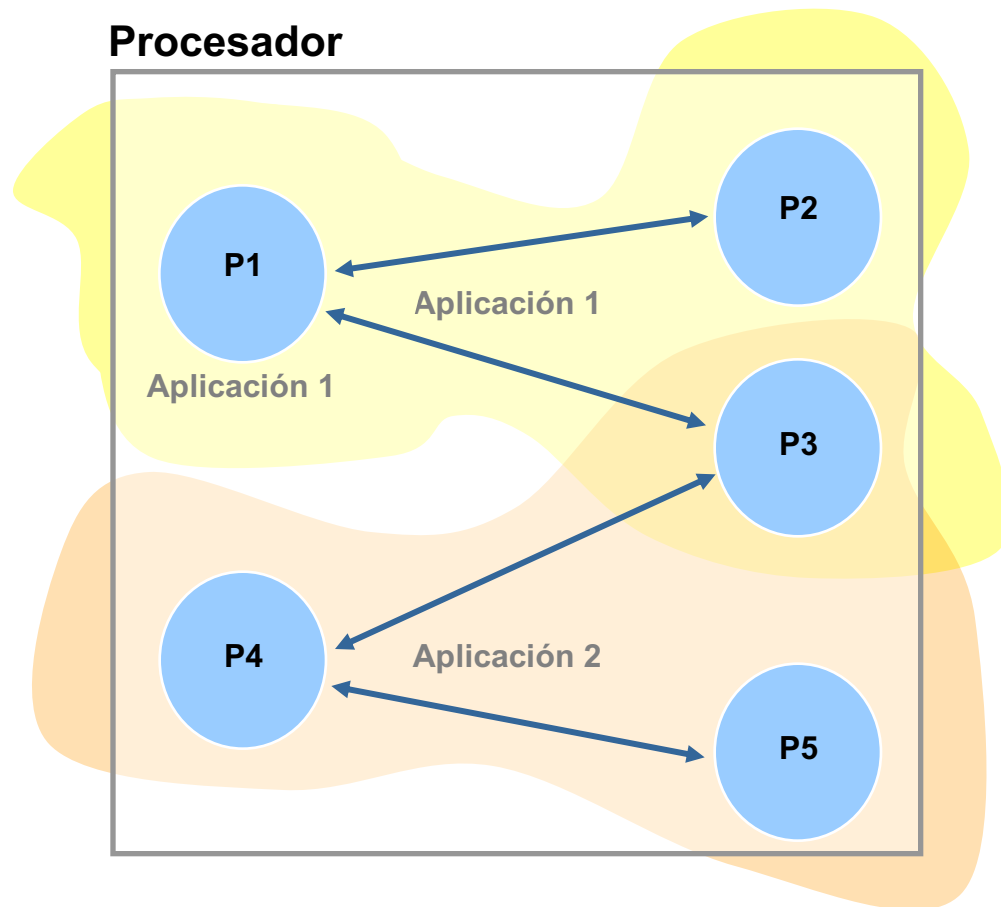
@ Concurrencia y sincronización

@ Tolerancia a fallos

@ Transparencia

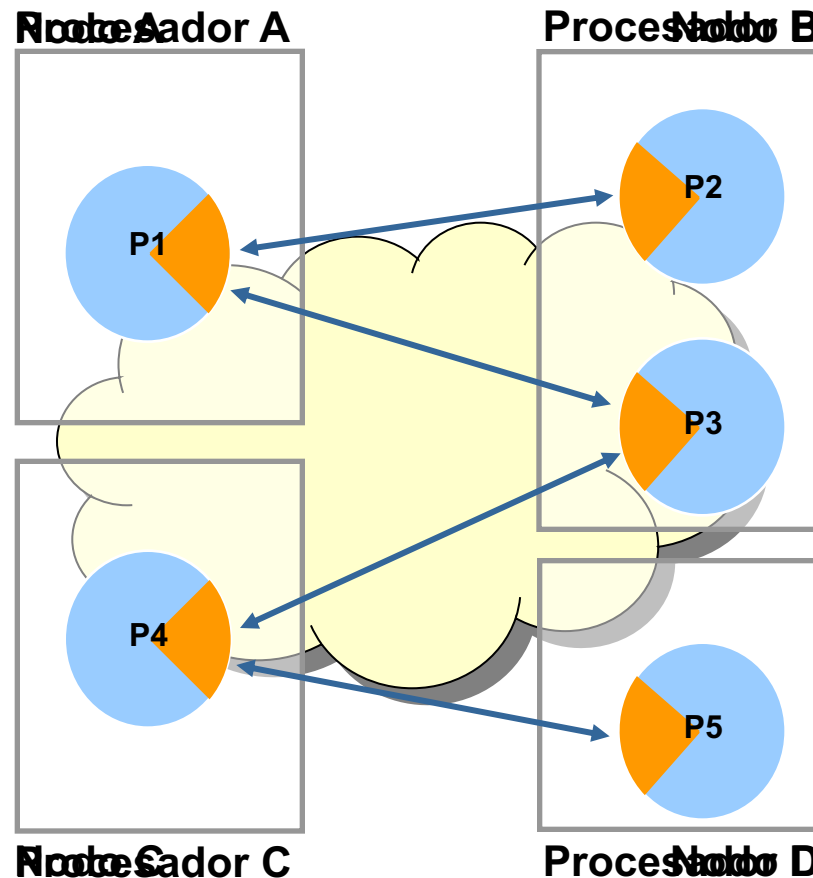
Contenido

introducción
computación



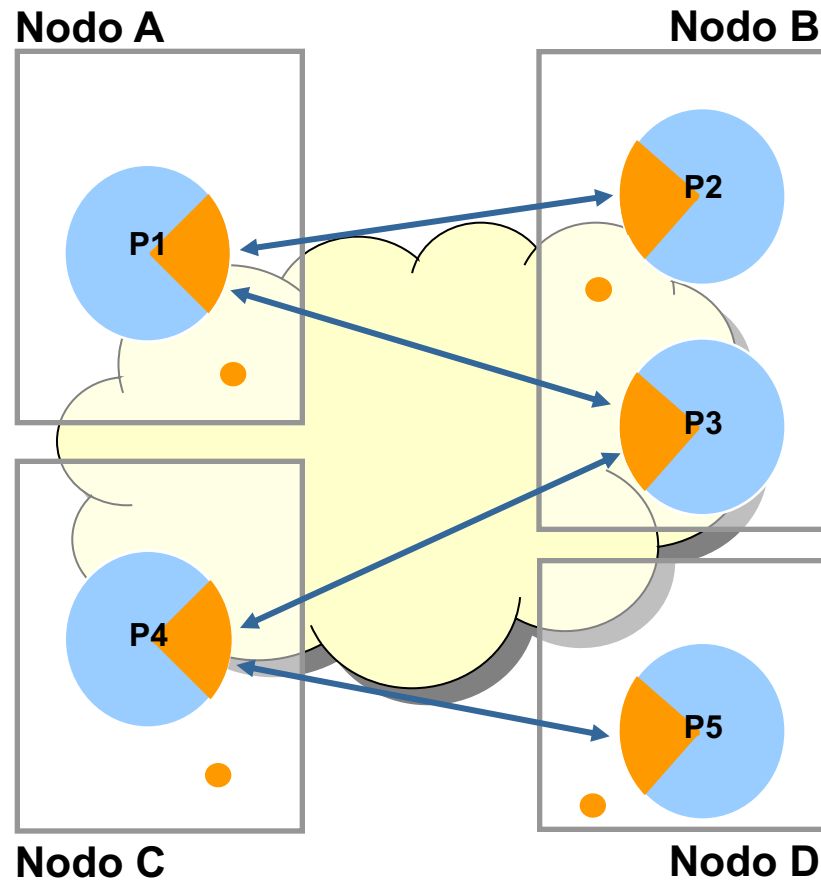
Contenido

introducción
computación



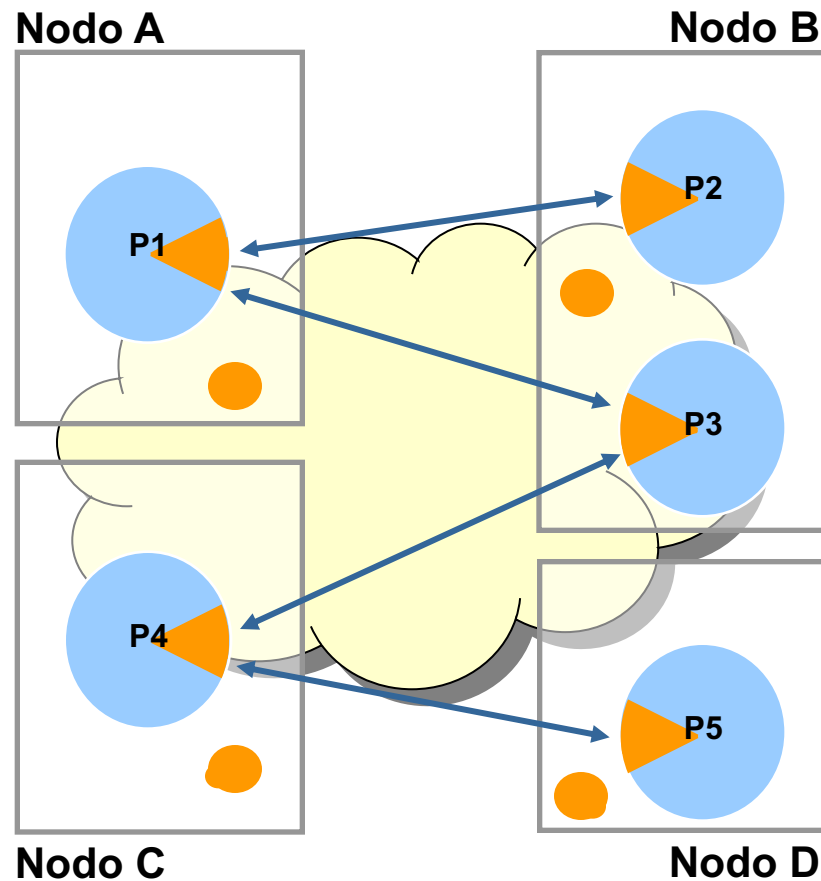
Contenido

introducción
computación



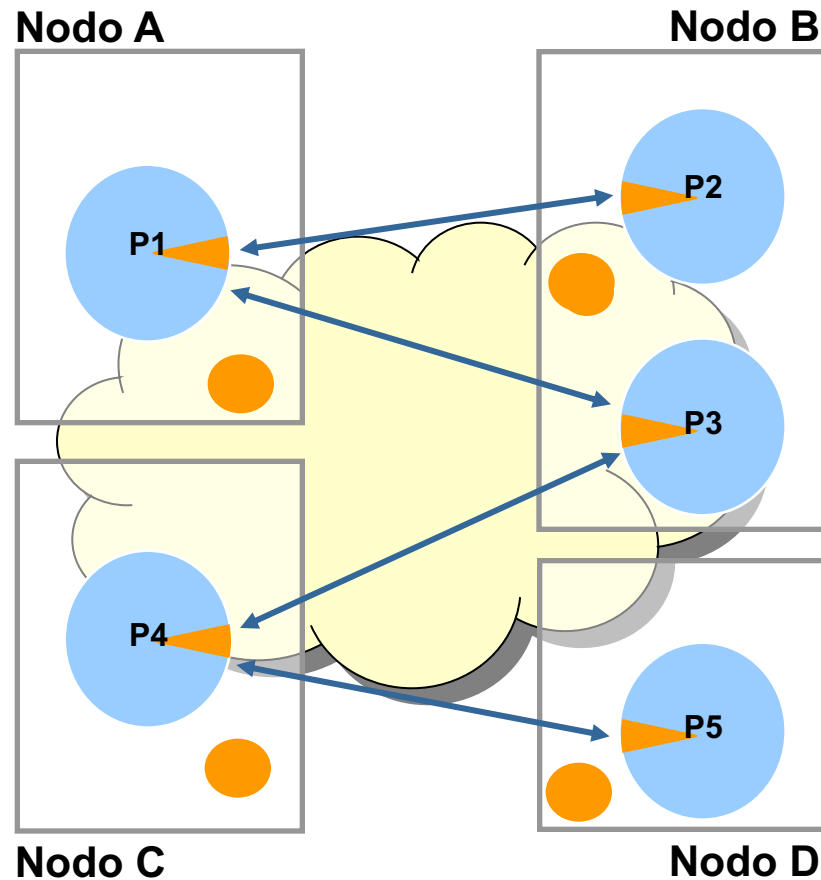
Contenido

introducción
computación



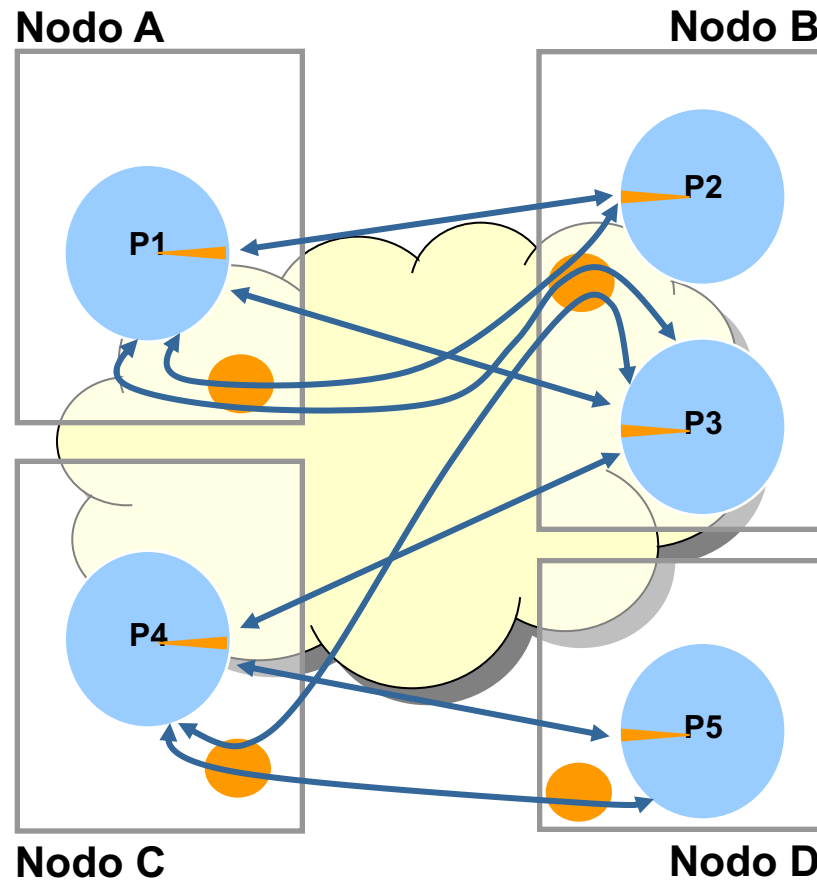
Contenido

introducción
computación



Contenido

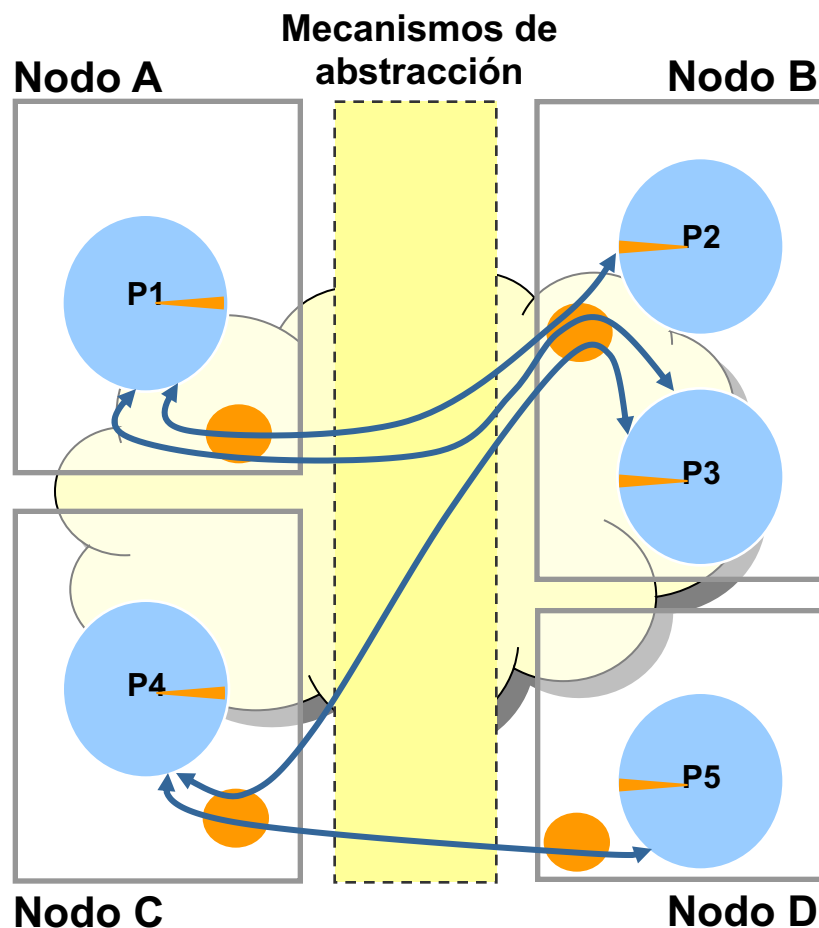
introducción
computación



Contenido

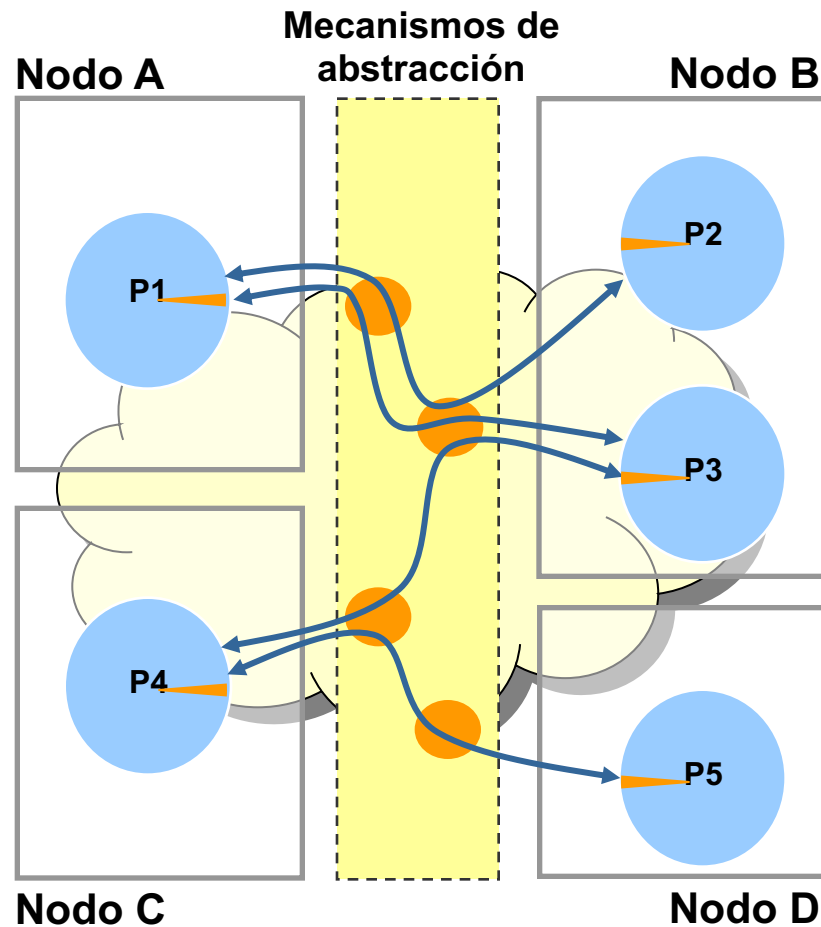
introducción

computación



Contenido

introducción
computación

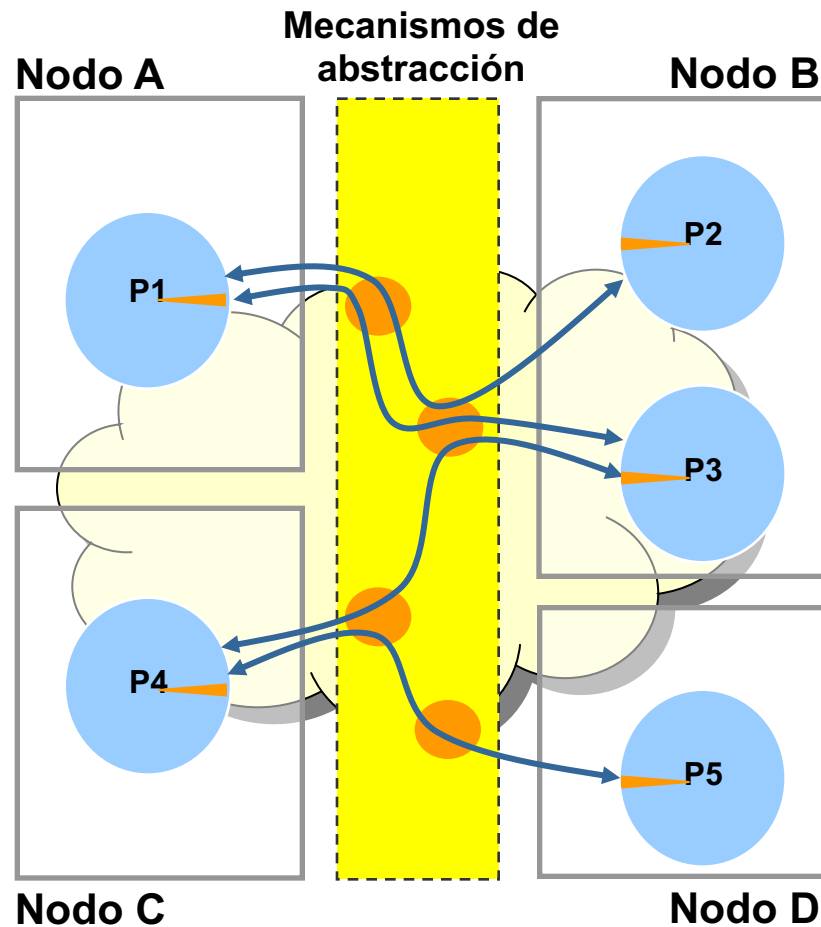


paradigmas de computación

evolución

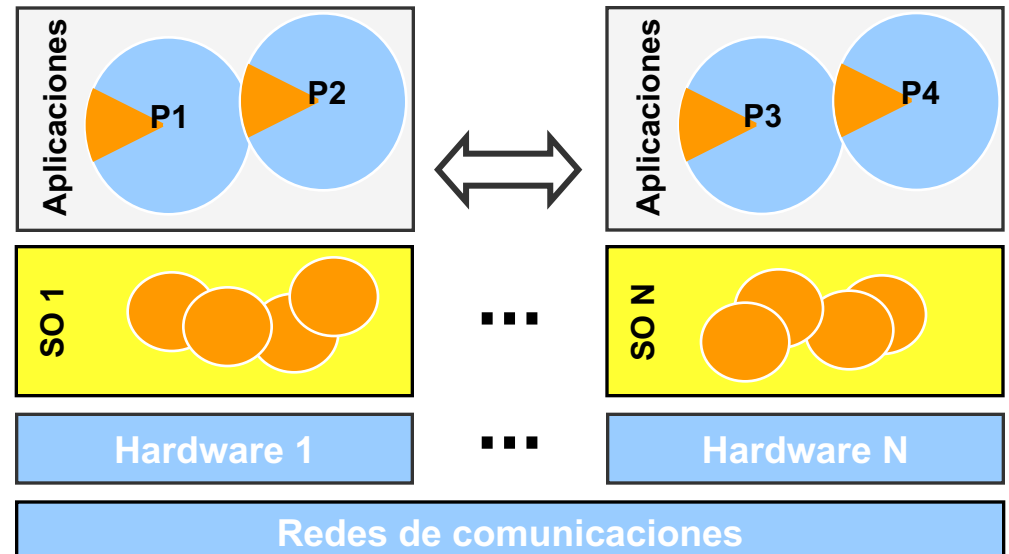
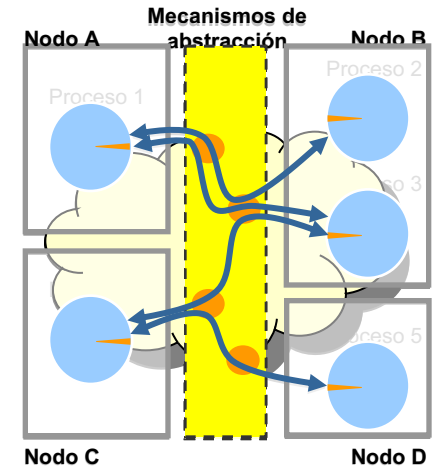
Contenido

- introducción
- computación
- arquitecturas
- comunicación
- conclusiones



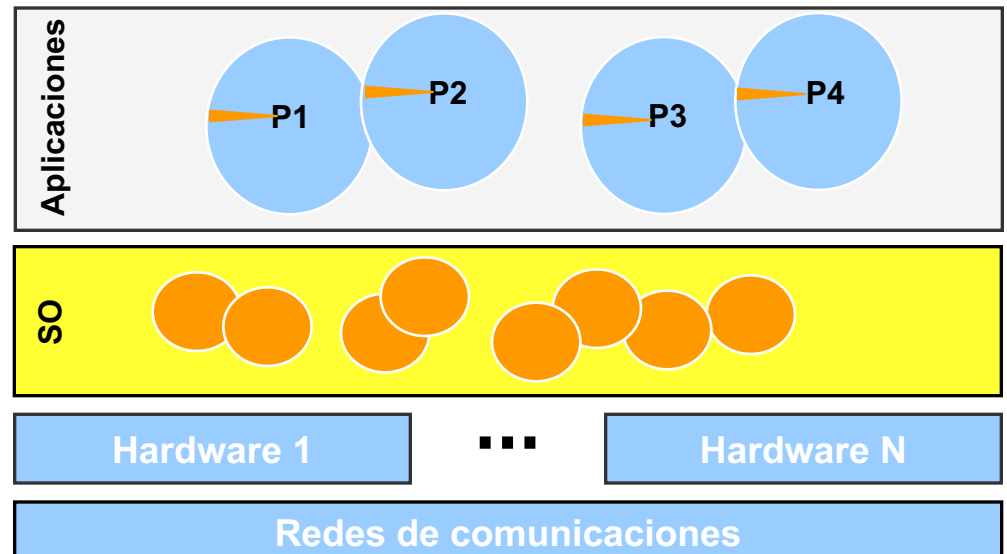
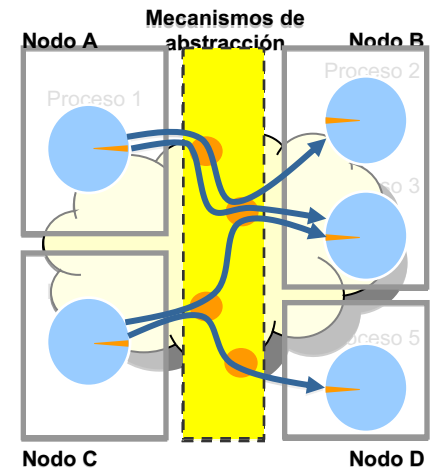
sistemas operativos en red

- Ubicación en el SO
- Heterogéneo → Específico del SO
- Ejemplos:
 - Linux, Windows, Novell NetWare
- Ventajas
 - Flexibilidad
 - SO → técnicas maduras
- Desventajas
 - Falta de transparencia
 - Mayor esfuerzo de integración



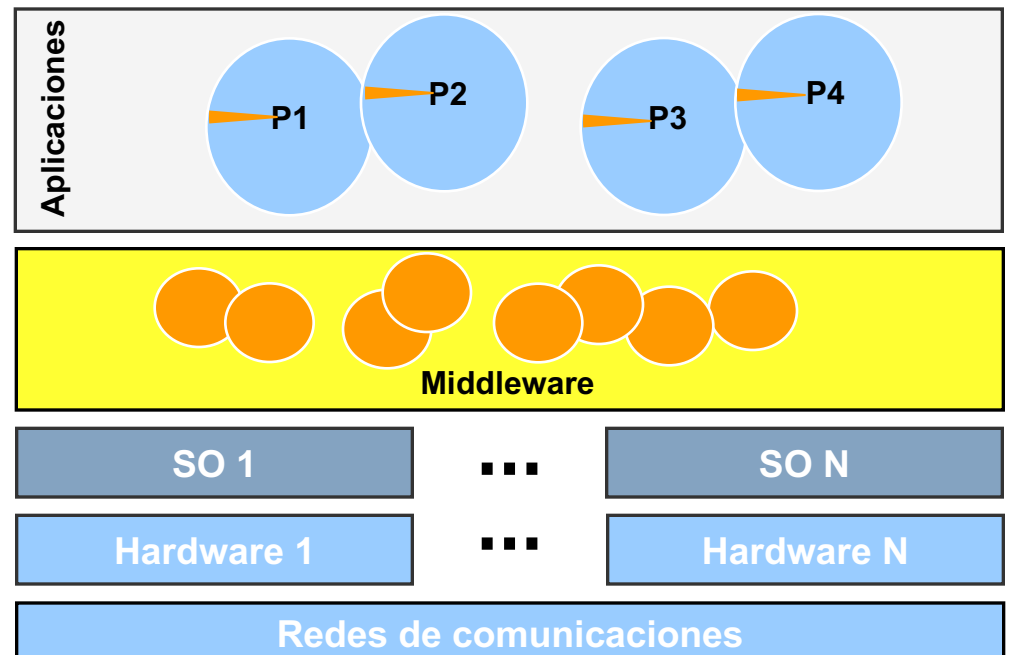
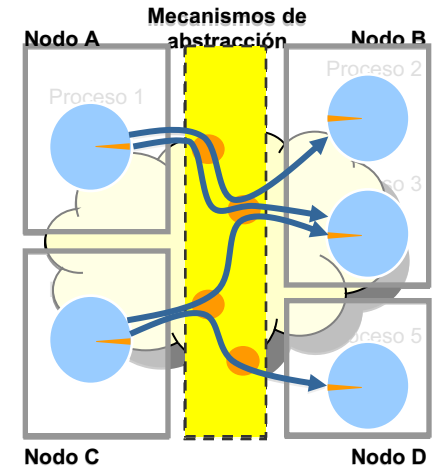
sistemas operativos distribuidos

- Ubicación en el SO
- Homogéneo → SO global
- Ejemplos:
 - Mach, Amoeba
- Ventajas
 - Transparencia
 - Escalabilidad
 - Facilidad de integración
- Desventajas
 - Técnicas complejas
 - Comunicaciones de alta velocidad
 - Competencia de mercado



middleware

- Enfoque mixto
 - Modelo conceptual → SOD
 - Infraestructuras → SOR
- Capa por encima del SO
- Homogéneo
- Ejemplos:
 - CORBA,
 - J2EE
 - .Net Framework
- Ventajas
 - Flexibilidad
 - Transparencia
 - Integración
 - Madurez
 - Escalabilidad
- Desventajas
 - Plataformas heterogéneas
 - Necesidad de estandarización



Fundamentos de Computación distribuida

sockets **RPC**
RMI