

# Sistemas Operativos

## Introducción a los Sistemas Distribuidos - Parte 1

Juan Carlos Fernández Caballero  
jfcaballero@uco.es

Asignatura "Sistemas Operativos"  
2º de Grado en Ingeniería en Informática  
Dpto. Informática y Análisis Numérico  
Universidad de Córdoba

28 de noviembre de 2018



- Transmitir una **visión general** de la naturaleza de los **sistemas distribuidos**.
- Comprender el **concepto de sistema distribuido y sus objetivos**.
- Asimilar las **ventajas e inconvenientes** de los sistemas **distribuidos** frente a los sistemas **centralizados**.
- **Concienciarse del amplio uso de sistemas distribuidos** en la sociedad actual en sistemas que van desde *Internet* hasta la *computación móvil*.
- Entender los **retos** que supone el **diseño de sistemas distribuidos y aplicaciones** que se basen en los mismos.

- 1.1. Definiciones.
  - 1.1.1. Sistemas distribuidos.
  - 1.1.2. Aplicaciones.
  - 1.1.3. Sistemas centralizados.
- 1.2. Ventajas e inconvenientes de los sistemas distribuidos.
  - 1.2.1. Ventajas SD frente a sistemas centralizados.
  - 1.2.2. Inconvenientes de los SD.
- 1.3. Tipos o paradigmas de sistemas distribuidos.
  - 1.3.1. *Internet* como sistema distribuido.
  - 1.3.2. *Intranets*.
  - 1.3.3. Computación móvil.
  - 1.3.4. Computación ubicua.

## 1.4. Aspectos y desafíos claves en los sistemas distribuidos.

- 1.4.1. Heterogeneidad.
- 1.4.2. Extensibilidad.
- 1.4.3. Seguridad.
- 1.4.4. Escalabilidad.
- 1.4.5. Tratamiento de Fallos.
- 1.4.6. Concurrencia.
- 1.4.7. Transparencia.

# Sistema Distribuido

- **Sistema Distribuido** → Aquel sistema en el que los **componentes hardware y software**, localizados en computadores unidos mediante **red** y a cualquier distancia, **comunican y coordinan** sus acciones mediante **paso de mensajes**.
- Proporciona **servicios** como un **todo**, como un **único sistema**.
- Se persigue la **transparencia** frente al usuario.
- Objetivo principal: **Compartir recursos**, tanto **hardware** como **software**, p.e. discos, impresoras, ficheros, bases de datos, imágenes, audio.



## Consecuencias la definición anterior

- **Concurrencia**

- Se **comparten recursos** por **diferentes orígenes paralelamente**.
- **Aumento de la capacidad de compartir recursos** mediante la **adición de más computadoras**.

- **Inexistencia de reloj global**

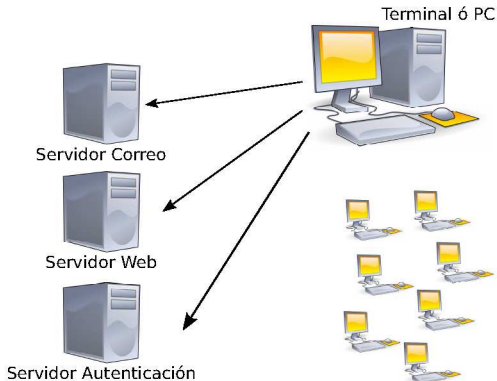
- La **coordinación no se basa en un reloj global**, sino en el **intercambio de mensajes** mediante **protocolos** y **mecanismos de concurrencia** sobre los recursos compartidos.

- **Fallos independientes**

- Surgen **nuevos tipos de fallos** que controlar, diferentes a los que ocurren en **sistemas aislados** (ordenador personal).
- **Un fallo en uno o varios nodos no deben inhabilitar el sistema**, son locales a un subconjunto de los componentes del sistema.
- Los errores se deben detectar, comunicar y **reestablecer en el menor tiempo posible**.



# Sistema Distribuido



## Aplicaciones de los Sistemas Distribuidos

- **Entornos empresariales:** redes corporativas e intranets.
- Entornos de **computación de altas prestaciones:** Procesamiento **paralelo** (cluster) como alternativa a costosos **supercomputadores**.
- **Servicios con alta disponibilidad:** Servidores de ficheros, correo electrónico, bases de datos distribuidas, ftp, news.
- **Aplicaciones multimedia:** Videoconferencia, televigilancia, juegos multiusuario, enseñanza asistida por ordenador.
- **Aplicaciones comerciales:** Reserva aerolíneas y trenes, operaciones bancarias
- **Sistemas ubicuos:** GPS automóviles, domótica, edificios inteligentes.



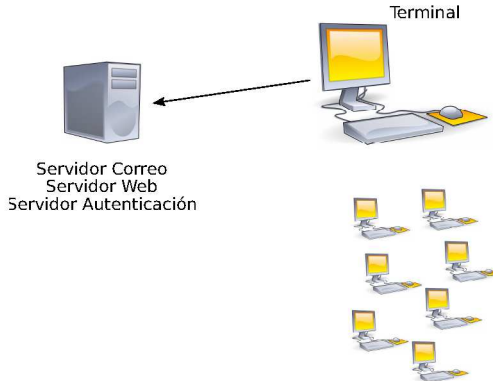


# Sistema Centralizado

- **Sistema Centralizado:** Aquel SD que **utiliza los recursos compartidos de una sola computadora**: Memoria, CPU, discos, periféricos, ficheros, bases de datos.
  - Si hay **terminales** conectados, todos ejecutan o **piden servicios sobre/a la misma máquina**.
- **Ventajas:** Entorno y **administración controlada y fácil de mantener**.
- **Desventajas:**
  - Velocidad de **repuestas lenta**, si la **conexión de la red** a la máquina central es **mala**, cuello de botella.
  - Necesidad de **mecanismo de respaldo** o copia del sistema ante **caídas** de la máquina central → **Muere todo el sistema**.
  - El **crecimiento** del sistema y de **petición de servicios** depende de los **equipos que soporte la máquina central**.



# Sistema Centralizado



## Ventajas SD descentralizados frente a SD centralizados

- **Economía:** Por normal general **cuesta menos crear un sistema de pequeños ordenadores** con similar **potencia** que un **único super-computador**.
- **Velocidad de procesamiento:** Se soporta el **procesamiento paralelo**.
- **Adaptación:** Se adapta a diferentes **distribuciones geográficas** y tipos de organizaciones.
- **Tolerancia a fallos:** Al estar distribuida la carga de trabajo en muchas máquinas **la falla de una de ellas no afecta a las demás**, el sistema sobrevive como un todo.
- **Crecimiento incremental:** **Admite crecimiento** tanto de **usuarios** como de **recursos (software y hardware)** a relativamente **bajo coste**.



# Inconvenientes de los SD, tanto descentralizados como centralizados

- **Heterogeneidad** en **hardware**, **sistemas operativos**, **lenguajes de programación** y desarrolladores diferentes, ¿cómo comunicarlos?.
- **Paso de mensajes basados en redes**:
  - Conlleva a **pérdida de mensajes** (fallas) → Necesidad de **protocolos** que sean tolerantes a los fallos.
  - **Cuellos de botella** en tráfico y en la capacidad de procesamiento de los servidores.
  - Existen **latencias** en la comunicación y procesamiento de envíos/-peticiones.



# Inconvenientes de los SD, tanto descentralizados como centralizados

- Problemas de seguridad:
  - Problemas de **confidencialidad** al viajar los datos por la red.
  - Problemas de **suplantación** al haber recursos compartidos entre múltiples usuarios.
- Estandarización: Necesidad de crear **estándares** de comunicaciones y **protocolos** que se cumplan por **todos** .



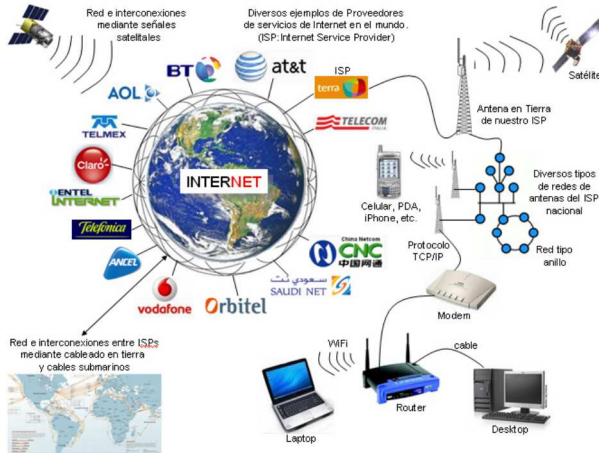
## Internet como Sistema Distribuido

**Internet:** Colección heterogénea de redes de computadores con las siguientes características:

- **Interacción** mediante **paso de mensajes** y **protocolos**.
- Ofrece **multitud de servicios** distintos: Contenido Web, transmisión de ficheros, vídeo *streaming*, *email*...
- Los **servicios son abiertos** → Se pueden mejorar y ampliar.
- **Heterogeneidad de redes:** Redes de área local (LAN) enlazadas por conexiones *backbone*, proveedores de servicio (ISP), enlaces a satélite, comunicación de telefonía móvil WAP, GSM, GPRS, UMTS/3G, 4G, tecnología Wifi, etc.



## Gran heterogeneidad



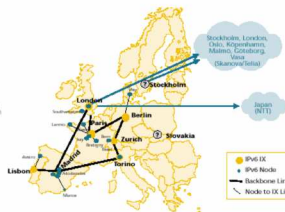
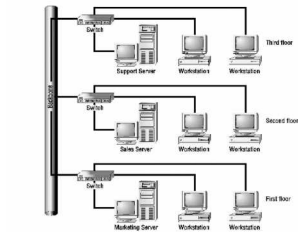
## Backbones o conexiones troncales entre redes

- **Backbone:** Es un tipo de **cableado** donde **confluyen las conexiones de varias redes**. También se conocen como conexiones troncales.
  - Principalmente proporciona interconexiones **entre redes y cuartos de comunicaciones**.
  - Incluye la **conexión vertical** entre redes y cuartos en edificios de varios pisos.
  - Componentes: **Cableado** y **hardware** para la **interconexión y adaptación de señales** entre redes → Gran ancho de banda.
- En términos más **globales** son los **enlaces entre regiones, países, continentes**.





## Backbones o conexiones troncales



## Hardware para Backbones



DCRS-9800 Series Dual Stack Backbone Routing Switch



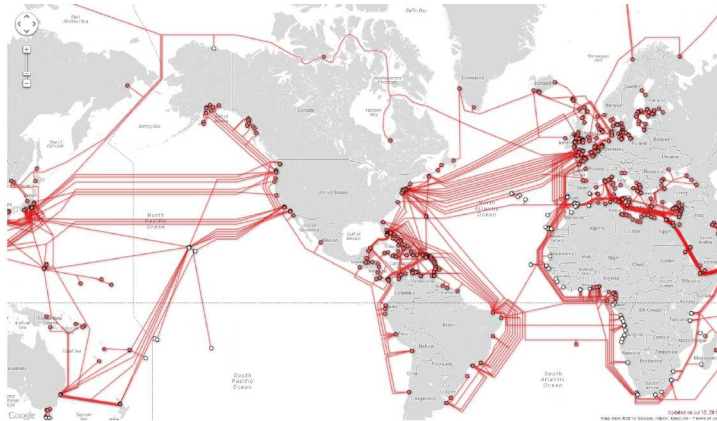
Digital China Networks DCRS-9800 Series are high performance wire speed chassis switch. DCRS-9816 is 18 slots chassis switch. It has 2 management slots and 16 business slots. With redundant power supply, fans and management modules, the Digital China DCRS-9800 Series ensures continuous operation and fully redundant system. The DCRS-9800 series is ideal for the backbone layer of campuses, enterprise networks and the aggregation layer of IP metropolitan networks.



## Definiciones

- Internet como Sistema Distribuido
- Intranets
- Computación móvil
- Computación ubicua

## Backbone mundial



## Intranets

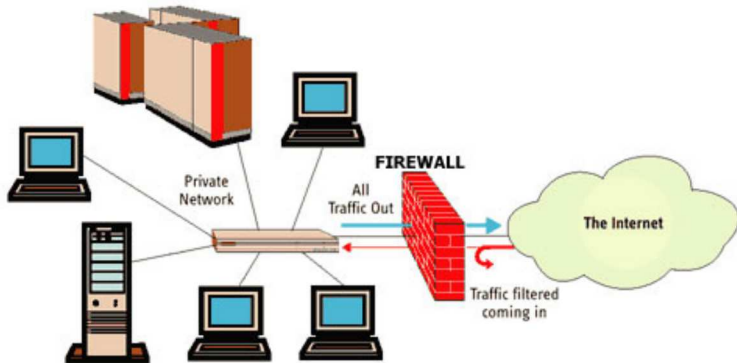
**Intranet:** Red de computadores **administrada separadamente** y configurada para cumplir unos fines y **políticas de seguridad local**.

- La configuración y seguridad es **responsabilidad de la organización** que la administra.
- Una intranet puede estar compuesta por **varias redes LAN enlazadas mediante backbone**.
- **Puede estar o no conectada a Internet**, a través de un *router* o **encaminador**, dependiendo de las políticas de acceso y seguridad.



## Intranets

Normalmente, para **proteger** una intranet se dispone de un **cortafuegos**, que **filtra los mensajes** que entran o salen y los **tipos de conexión entrantes/salientes** permitidos.



## Computación móvil

- **Computación Móvil:** Aquella que permite **tareas de cómputo** mientras el **usuario** está en **movimiento** o visitando otros **lugares distintos de su entorno habitual** o intranet.
  - Acceso a recursos mediante **dispositivos que se llevan consigo**.
  - Dichos dispositivos pueden acceder a **Internet** y a la **intranet** habitual del usuario (si la política de la intranet lo permite).
- Ejemplos: Portátiles, teléfonos móviles, cámaras digitales WIFI, relojes inteligentes, navegadores GPS de vehículos.

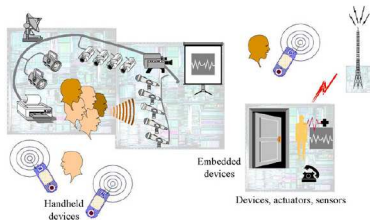


## Computación ubicua

- **Computación ubicua:** Aquella que permite la **utilización concertada** de muchos **dispositivos de computación pequeños** que están presentes en los **entornos físicos** de los usuarios **mientras se permanezcan en ellos:** hoteles, oficina, casa, restaurante.
- Es entendida como la **integración de la informática en el entorno de la persona.**  
Ejemplos: Domótica (iluminación y dispositivos mecánicos), acceso inalámbrico a impresora de hotel, proyectores y terminales, sensores de temperatura, monitoreo de sistemas de salud, dispositivos de diagnóstico, etc.
- La **computación móvil y ubicua** a veces son **difíciles de diferenciar** y se llegan a **complementar y solapar.**



## Computación móvil y ubicua





## Aspectos y desafíos en el diseño de sistemas distribuidos

- 1 Heterogeneidad.
- 2 Extensibilidad.
- 3 Seguridad.
- 4 Escalabilidad.
- 5 Tratamiento de Fallos.
- 6 Concurrencia.
- 7 Transparencia.



## Heterogeneidad

Un SD **debe permite acceder a servicios** y ejecutar aplicaciones sobre un **conjunto heterogéneo de redes y de computadoras**.

- Redes con diferentes **topologías**.
- **Hardware** diferente, desde computadoras hasta diferentes routers y cables de red.
- **Sistemas Operativos** diferentes (Windows, Linux, Mac OS, etc).
- Comunicación de programas implementados en diferentes **lenguajes de Programación** (C, C++, Java, PHP, Python).



## Heterogeneidad

Herramientas para lograr hacer **transparente la heterogeneidad**:

- 1 **Cumplimiento de protocolos** establecidos: TCP/IP, HTTP, SMTP, POP3, etc.
- 2 **Estándares** de **representación** y **estructuras** de datos. Ej: enteros y caracteres según el hardware o máquina utilice (**big-endian** y **little-endian** - <http://es.wikipedia.org/wiki/Endianness>) → **Mecanismos de conversión**.
- 3 **Código móvil**: Código que **se envía de un lugar a otro** y código ejecutable en cualquier plataforma.  
Ej: Applets de Java y Flash → Un Applet es un programa que se ejecuta en el navegador del usuario que visita una página web, es decir, se ejecuta en SU ordenador.



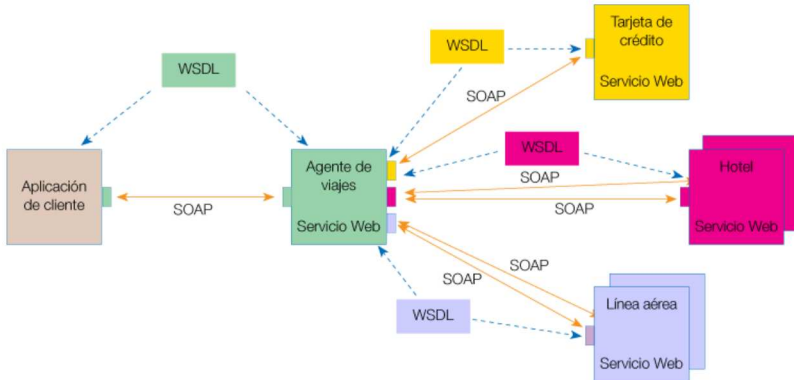
## Heterogeneidad

- ④ **Middleware:** *Capa software* que provee una **abstracción de programación** → CORBA, Java RMI, DCE, DCOM, Servicios Web (REST, SOAP, WSDL).
  - **Objetivo:** Lograr **transparencia** y librar la **heterogeneidad**.
  - Proporciona al programador una **API** conveniente para crear **aplicaciones distribuidas**.
  - Se **implementan sobre protocolos** de Internet.
  - **Soporte para abstracciones y servicios:** Procedimientos de invocación remota (RPC), comunicación en grupo, replicación de datos y streaming, servicios Web, etc.



# Heterogeneidad

Servicios Web en funcionamiento.



## Extensibilidad (a nivel software y de servicios)

- Tiene que ver con la definición de **Sistema abierto**:  
Aquel que puede **admitir de forma dinámica nuevas características y servicios**, e incluso **mejorar los ya existentes**, posibilitando su petición desde programas clientes diversos.
- **Sistema Extensible**:  
Aquel **Sistema Abierto** que puede **comunicarse** con otros sistemas abiertos.
- Para crear sistemas extensibles son necesarios:
  - Estándares.
  - Protocolos.
  - Publicar documentación, APIs, interfaces, *Request For Comments*.  
[https://es.wikipedia.org/wiki/Request\\_for\\_Comments](https://es.wikipedia.org/wiki/Request_for_Comments)



# Seguridad

**Seguridad:** Necesidad de **proteger los recursos** ofrecidos a **tres niveles**:

- **Confidencialidad:** Protección contra el **descubrimiento por individuos no autorizados** → **autenticación**.
- **Integridad:** Protección del canal contra la **alteración o corrupción** de los mensajes.
- Ataques de **denegación de servicios**. Bombardear un servicio con **múltiples peticiones** para inhabilitar su acceso.
- Seguridad de **código móvil**. Mecanismos para **detectar ejecutables mal intencionados** en applets, macros, etc.



## Escalabilidad (A nivel de hardware)

**Escalabilidad:** Un sistema es escalable si **un aumento de sus usuarios y/o recursos** no supone un **decremento de su efectividad**.

- **Pérdida de prestaciones:** Un **aumento** de los **datos y/o servicios (recursos)** no debe conllevar a peores tiempos de acceso y respuesta.
- **Cuellos de botella:** Un **aumento de la demanda de recursos** puede producir cuellos de botella que son necesarios evitar.
- **Solución:** **Adición de nuevas máquinas y algoritmos** para descentralizar los recursos e incrementar prestaciones.

Ejemplos: Sistema de nombres de dominio particionado entre servidores, Webs muy accedidas replicadas en varios servidores.





## Tratamiento de fallos

**Tratamiento de Fallos: Recuperación** frente a **fallos de funcionamiento** en el sistema.

- Suelen ser **parciales** de algunos componentes.
- Tratamiento de fallos habituales + **fallos de comunicación** (falla el **canal** o alguno de los **componentes intermedios**).



## Aspectos deseables en el tratamiento de fallos

- **Detección de fallos:** Uso de mecanismos como las **sumas de comprobación** o *checksums* en los receptores de mensajes.
- **Enmascaramiento de fallos:** **Ocultamiento** de fallos al **usuario**.  
Ejemplos: **Retransmitir** un mensaje que no llega al receptor, replica de discos por si uno falla.
- **Recuperación tras el fallo:** **Volver a estados estables** anteriores.  
Ejemplo: Inconsistencia de datos en servidor por causas hardware.



## Aspectos deseables en el tratamiento de fallos

- **Redundancia de recursos:**
  - Existencia de **rutas diferentes** entre cualesquiera dos **routers en Internet**.
  - Tablas del sistema de nombres de dominio (**DNS**) **replicadas** en diferentes servidores.
  - **Bases de datos replicadas en servidores**. Si uno falla se redirige a un cliente a otro servidor.
- **Disponibilidad:** Se desea un alto grado de disponibilidad, es decir, **aumentar la proporción de tiempo** en que el sistema y sus **servicios son utilizables**.



## Concurrencia

En un **sistema distribuido** los recursos se pueden compartir entre las máquinas que lo forman.

- Hay **acceso concurrente**, es decir, acceso a los mismos recursos al mismo tiempo → Necesidad de controlar la consistencia de los mismos en exclusión mutua: **semáforos**, **monitores**, **controladores hardware**, etc.
- **Aumento de desempeño**: Uso de hilos (ej: servidor de paginas Web).



# Transparencia

**Transparencia:** Ocultación al usuario y al programador de la separación de los componentes → Percepción de un **todo**.

- 1 **Transparencia de acceso:** Se debe permitir el **acceso a recursos locales y remotos empleando operaciones idénticas**. Ejemplo, contenidos de carpetas que se observan igual tanto si son locales como remotos, llamadas RPC.
- 2 **Transparencia de ubicación:** **No es necesario conocer ubicaciones físicas.**
- 3 **Transparencia de concurrencia:** Los procesos pueden **operar concurrentemente** sobre **recursos compartidos** sin interferencia visible al usuario.



## Transparencia

- ④ **Transparencia de replicación:** Múltiples ejemplares de recursos y servicios para **aumentar fiabilidad, prestaciones y disponibilidad** sin que el usuario lo note.
- ⑤ **Transparencia frente a fallos:** **Ocultación de fallos** software y hardware de cara al usuario.
- ⑥ **Transparencia de movilidad:** **Reubicación de recursos** sin afectar la operación de los usuarios (cambio de células viajando en tren usando el móvil).
- ⑦ **Transparencia de prestaciones:** Transparencia en la **reconfiguración y crecimiento** del sistema de cara al usuario final.



## Bibliografía



George Colouris, Jean Dollimore y Tim Kindberg.  
Sistemas Distribuidos. Conceptos y Diseño.  
Addison Wesley, 2001.



# Sistemas Operativos

## Introducción a los Sistemas Distribuidos - Parte 1

Juan Carlos Fernández Caballero  
jfcaballero@uco.es

Asignatura "Sistemas Operativos"  
2º de Grado en Ingeniería en Informática  
Dpto. Informática y Análisis Numérico  
Universidad de Córdoba

28 de noviembre de 2018

