

Sistemas distribuidos

Grado en Ingeniería Informática

Tema 1.1: Introducción a los Sistemas Distribuidos

Departamento de Ingeniería Informática
Universidad de Cádiz

Escuela Superior de Ingeniería
Dpto. de Ingeniería Informática



Versión 2.1

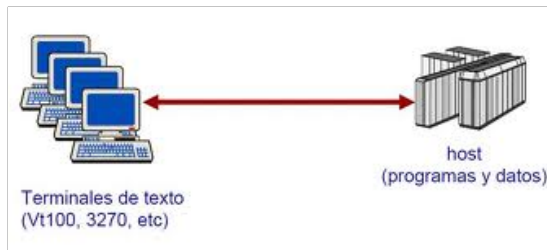
Indice

- 1 Evolución
- 2 Definiciones
- 3 Características
- 4 Ejemplos

Sección 1 | Evolución

Evolución de los sistemas

- **Sistemas centralizados:** recursos centralizados y acceso mediante terminales.



Evolución

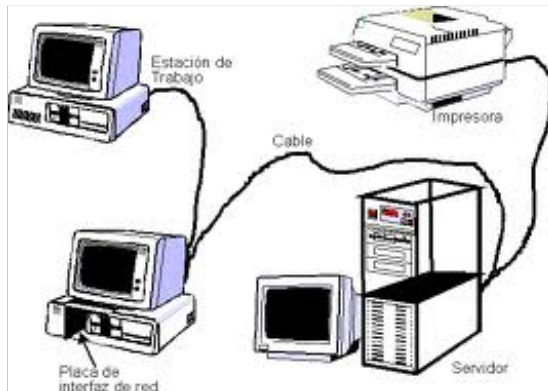
de los sistemas

- **Sistemas unipersonales PC**, computación en un solo nodo.



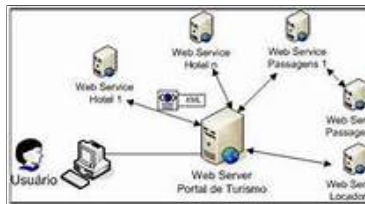
Evolución de los sistemas

- **Sistemas en red** múltiples elementos de cómputo independientes unidos por una red



Evolución de los sistemas

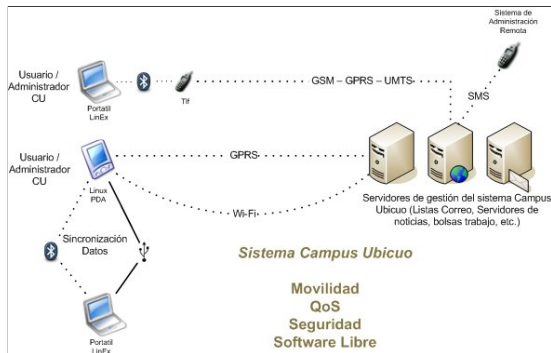
- **Sistemas distribuidos:** visión de sistema único a partir de un conjunto de elementos interconectados.



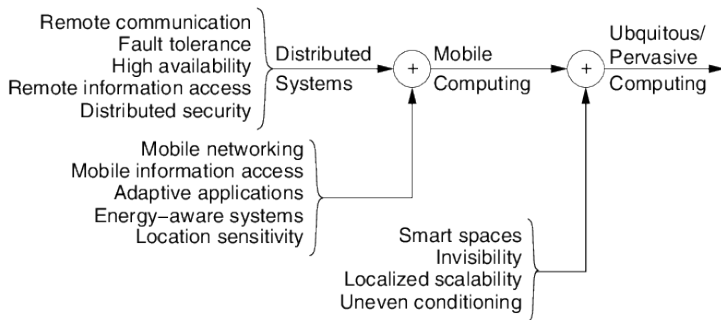
Evolución

de los sistemas

- **Sistemas ubicuos:** concepto introducido por Mark Weiser en 1991. Se define como la capacidad de computo **distribuida** en el entorno y con la que el usuario interactúa de forma **inconsciente**.



Evolución de los sistemas



Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. IEEE Personal communications, 8(4), 10-17.

Evolución

Diferencias

Diferencia entre sistemas en red y sistemas distribuidos

Como podemos ver, la diferencia fundamental entre los **sistemas en red** y los **sistemas distribuidos** es la **transparencia**. En el sistema distribuido, la composición y estructura de la red le pasa desapercibida al usuario, es decir, ni la ve ni le importa.

Solamente le importan los recursos disponibles o, a veces, simplemente el tipo de los recursos disponibles, sin tener en cuenta en qué máquina están realmente ubicados.

Sección 2 | Definiciones

Definiciones

Diferentes definiciones

Coulouris et al.: Aquel sistema en el que los componentes localizados en una red de computadores se **comunican** y **coordinan sus acciones** únicamente mediante el paso de mensajes

- Comunicación a través de una red
- Concurrencia
- No hay reloj central

Definiciones

Diferentes definiciones

Tanenbaum et al.: Una colección de computadoras independientes que aparecen ante los usuarios del sistema como una **única computadora**

- Sistema operativo distribuido
- Transparencia
- Virtualización

Definiciones

Diferentes definiciones

van Steen et al.: Componente software que asegura que una colección de computadoras independientes aparece ante los usuarios como un **único sistema coherente**

- Existencia de un software de unión = Middleware
- Formado por elementos de computación independientes

Definiciones

Diferentes definiciones

Lesli Lamport: You know you have one (Sistema Distribuido) when the crash of a computer you've never heard of stops you from getting any work done

Sección 3 | Características

Características

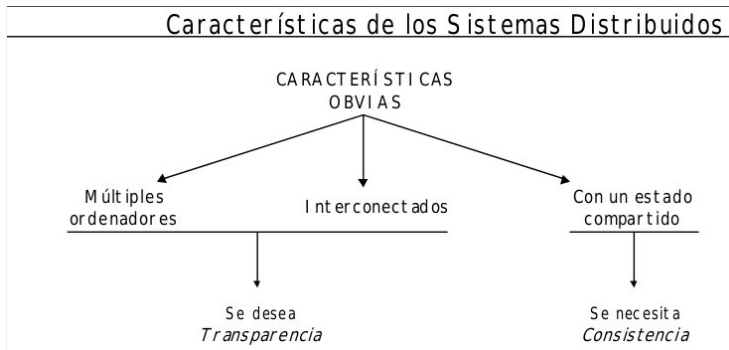
de manera informal

Algunos conceptos

- Se **ocultan a los usuarios** las diferencias entre las máquinas y la complejidad de los mecanismos de comunicación entre máquinas.
- Los usuarios **acceden a los SSDD de forma homogénea** sea cual sea el lugar desde el que lo hagan.
- Los SSDD deben ser relativamente **sencillos de ampliar**, por lo que deben escalar bien.
- Los servicios que ofrecen los SSDD deben estar **siempre disponibles**, aunque determinadas partes de los sistemas fallen o no estén disponibles por reparación, ampliación, sustitución, etc.
- Los SSDD habitualmente se **estructuran en niveles**, siendo la estructuración como “middleware” una de las más frecuentes.

Características

de los sistemas distribuidos



Características

Detallando los conceptos

Un sistema distribuido puede verse como un sistema formado por varios ordenadores haciendo algo conjuntamente, de lo que se desprenden tres características inmediatas

- **Compuesto por múltiples ordenadores.**
- **Hay interconexión entre ellos.**
- **Tienen un estado compartido.**

Características

Compuesto por múltiples ordenadores

Un sistema distribuido está compuesto de más de un sistema independiente, cada uno con una o más CPU's, memoria local, memoria secundaria (discos) y, en general, conexiones con periféricos de acceso inmediato (on line)

Características

Hay interconexión entre ellos

Parece claro que si varios ordenadores distintos van a colaborar en la realización de tareas, **deben comunicarse y sincronizarse** entre ellos, por lo que debe haber alguna línea o red de interconexión.

Características

Tienen un estado compartido

Si los ordenadores realizan un trabajo conjuntamente, deben mantener un estado compartido, es decir, todos los ordenadores tienen la misma visión del estado del sistema distribuido (tablas, bases de datos del sistema, de servidores, etc).

Características

Objetivo de construir un SD

Entre los objetivos de construir un SD podemos encontrar:

- Compartir recursos tanto hardware (discos, impresoras) como software (archivos, bases de datos)
- Compartir datos es esencial en muchas aplicaciones
 - Equipos de desarrollo comparten herramientas y datos
 - Aplicaciones comerciales ofrecen a usuarios accesos a datos compartidos
 - Trabajo cooperativo en algunas empresas
- Existen aplicaciones inherentemente distribuidas
 - Por ejemplo una cadena de supermercados con varias tiendas y almacenes
 - Sistemas de billetes de líneas aéreas

Consistencia y Transparencia

algunas características

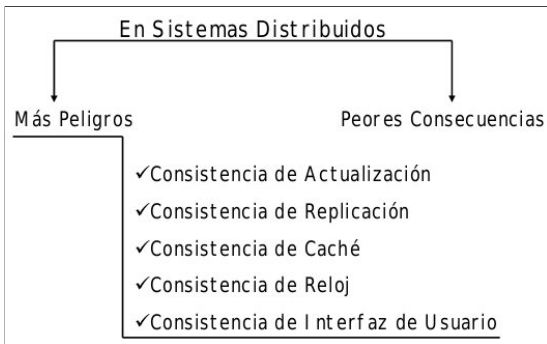
La construcción de un sistema distribuido que se comporte según esperan los usuarios, va a requerir considerar otro conjunto de características que se pueden resumir en estas dos:

- **Consistencia** es una necesidad imperativa, pues sin ella, simplemente es que el sistema no funciona.
- **Transparencia** Ocultar el hecho de que los procesos y los recursos están físicamente distribuidos sobre diferentes ordenadores

Consistencia

Información sobre tipos de consistencia

- En los SSDD el problema de la **inconsistencia** cobra una mayor dimensión, tanto por la importancia como por la cantidad de situaciones en que pueden producirse problemas
- Un sistema distribuido es **un único sistema** formado por múltiples máquinas independientes. Ya que es un único sistema, debe tener un **único estado global** compartido por todos los equipos que lo componen



Consistencia

de Actualización (1/5)

- **Problema:** Cuando varios procesos acceden concurrentemente a **un dato para actualizarlo** se puede producir una **inconsistencia**, porque la actualización de todo el dato en su conjunto no se realiza como una única operación atómica en exclusión mutua (Ejemplo. Bases de datos)
- ¿Solución?

Consistencia

de Actualización (1/5)

Solución: Este tipo de inconsistencia se evita utilizando **Transacciones**. Se utilizan las siglas **ACID** para referirse a las propiedades, en inglés, de las transacciones: Atomicity, Consistency, Isolation, Durability.

- **Atomicidad** es la propiedad que asegura la operación se ha realizado o no
- **Consistencia** Se asegura que sólo se empieza aquello que se puede acabar
- **Aislamiento** Una operación no puede afectar a otras
- **Durabilidad** Una vez realizada la operación esta persistirá

Consistencia

de Réplica (2/5)

Cuando un conjunto de datos debe de **mantenerse replicado** en varias estaciones.

Cuando un conjunto de datos debe mantenerse replicado en varias estaciones.

Cuando hay
modificación
en uno de ellos



MULTICAST

Si no llega a alguno



INCONSISTENCIA

Ejemplo: Juego mult iusuario en red.

Consistencia

de Cache (3/5)

- **Situación:** Cuando un cliente accede a un recurso (un fichero de datos), se pueden guardar copias de estos datos en una memoria local del cliente (memoria caché) para facilitar su acceso en posteriores referencias, **evitando tener que transferir** de nuevo los datos por la red.
- **Problema:** surge cuando un cliente actualiza datos que también residen en las memorias caché de otros clientes. En ese momento se dice que las copias que están en **tras cachés quedan anticuadas**.
- **Solución:** Hay **distintas técnicas** para asegurar la consistencia de las cachés, y se suelen tratar en la gestión de memoria de los sistemas operativos distribuidos y en las arquitecturas de sistemas multiprocesadores.

Consistencia

de Reloj (4/5)

- **Situación:** Muchos de los algoritmos utilizados en aplicaciones y programación de sistemas dependen de unas **marcas de tiempo o timestamps** que indican el momento en el que ha sucedido un evento.
- **Problema:** no resulta fácil mantener **la misma hora física** en todos los ordenadores o componentes de la red simultáneamente
- **¿Solución?**

Consistencia

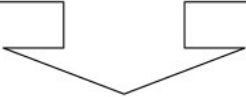
de Reloj (4/5)

- **Solución:** Una posibilidad consiste en enviar la hora por la red a todos los ordenadores, pues a pesar de que la transmisión en sí ya requiere un tiempo, esto se podría solucionar fácilmente si se pudiera añadir el tiempo de transmisión a la hora recibida en cada estación; **pero no es así, pues el tiempo de transmisión en una red es algo bastante impredecible.**

Consistencia

de Interfaz de Usuario (5/5)

En una aplicación interactiva distribuida, a veces, se pulsa un botón del ratón
y no cambia nada en la pantalla!



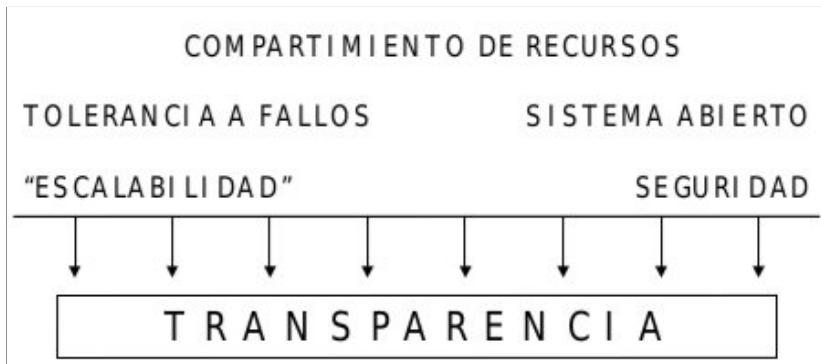
INCONSISTENCIA DE INTERFAZ

El retardo no debe ser mayor de 0,1 s.

Para dar la impresión de disponer de una máquina dedicada

Transparencia

- En el sistema se pueden producir fallos, pero el usuario no los va a notar; el sistema podrá crecer, pero para ello no habrá que parar el sistema; se podrán compartir recursos, pero no será necesario saber dónde están o a dónde se han movido



Transparencia

Tipos de transparencia

Transparencia	Descripción
Acceso	Oculto diferencias en la representación de los datos y en cómo se accede a los recursos
Ubicación	Oculto dónde se ubican los recursos
Migración	Oculto el hecho de que un recurso puede migrar de un lugar a otro
Reubicación	Oculto el hecho de que un recurso puede moverse de un lugar a otro mientras se utiliza
Replicación	Oculto el hecho de que un recurso puede tener más de una réplica, haciendo indistinguible a los usuarios la réplica que realmente utilizan
Concurrencia	Oculto el hecho de que un recurso pueda ser utilizado simultáneamente por más de un usuario.
Fallos	Oculto el fallo y la recuperación de los recursos
Persistencia	Oculto el hecho de que un recurso esté ubicado en memoria volátil o en memoria persistente.

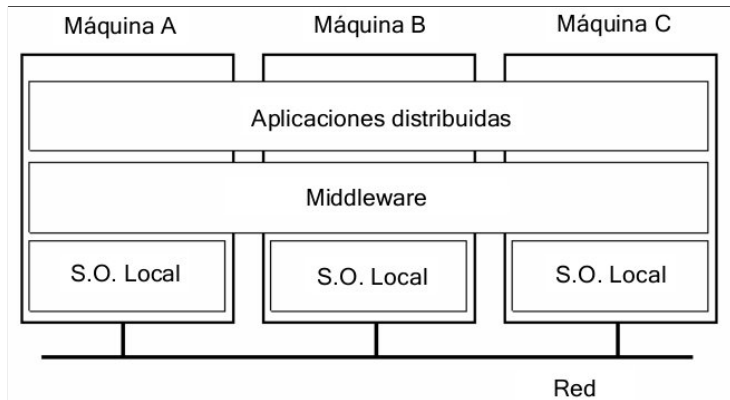
Middleware

El software distribuido requerido para **facilitar** las interacciones cliente-servidor se denomina middleware. El **acceso transparente a servicios y recursos** no locales distribuidos a través de una red se provee a través del middleware, que sirve como marco para la comunicaciones entre las porciones cliente y servidor de un sistema

Middleware

estructura

Un sistema distribuido está organizado como un “middleware”



Observar que la capa de middleware se extiende sobre múltiples máquinas

Middleware

Tipos de Middleware

- Protocolos de nivel de transporte de Internet
 - Ambos permiten un paso de mensajes básico
 - UDP: con fallos por omisión
 - TCP: garantiza la entrega en condiciones normales, pero al coste de una bajada de rendimiento
 - Generalmente, usamos TCP sobre IP, o TCP/IP
- Aunque TCP/UDP abstraen del nivel de red, no abstraen perfectamente de los niveles de hardware y SO
 - Distinto almacenamiento de números (little endian/big endian)
 - Distinta codificación de caracteres (ASCII/Unicode)

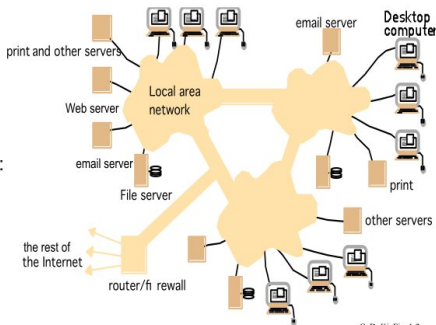
Sección 4 | Ejemplos

Ejemplos

Intranet

Intranets (CoDoKi, Fig. 1.2)

- a single authority
- protected access
 - a firewall
 - total isolation
- may be worldwide
- typical services:
 - infrastructure services: file service, name service
 - application services



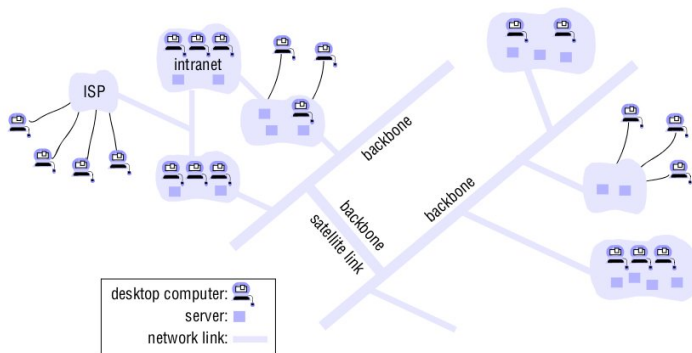
CoDoKi, Fig. 1.2

Figure 1.2 A typical intranet

Ejemplos

Internet

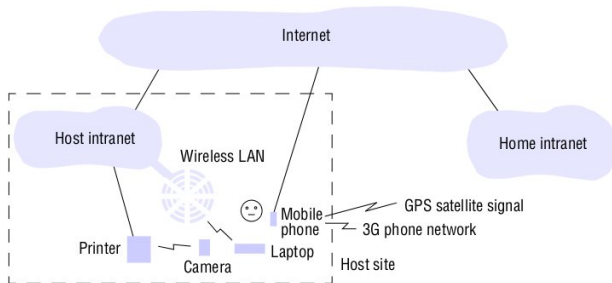
Figure 1.3 A typical portion of the Internet



Ejemplos

Sistemas móviles y ubicuos

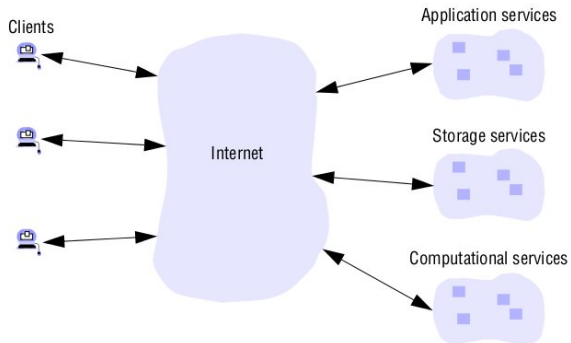
Figure 1.4 Portable and handheld devices in a distributed system



Ejemplos

Cloud Computing

Figure 1.5 Cloud computing



Ejemplos

Amazon EC2

