

CC5303 – Sistemas Distribuidos

# **1.- Introducción a Sistemas Distribuidos**

*Parte 1*

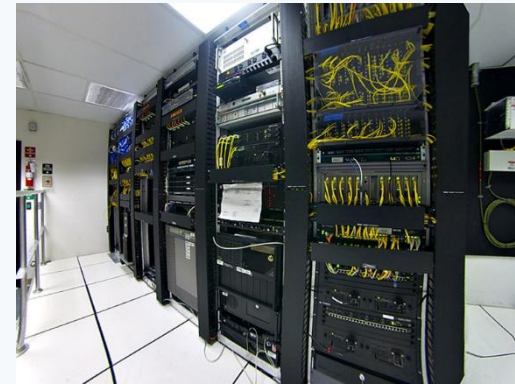
Sebastián Blasco V.

# Contenidos

1. Nociones y definiciones de SSDD
  1. Definiciones
  2. Usos y objetivos
  3. Organización de SSDD
2. Características de SSDD
  1. Ventajas, Desventajas, Problemas y Desafíos

# Qué es un SSDD?

- Suenan como sistemas distribuidos
  - La web
  - Redes inalámbricas
  - DNS
  - BitTorrent
  - Sistemas *Cloud*
  - Datacenters



**¿Qué tienen en común?**

# Definición SSDD

- *“Un Sistema Distribuido es una colección de computadores independientes que aparecen ante los usuarios como un único sistema coherente.”* (Andrew S. Tanenbaum)
- *“Un sistema en el cual tanto los componentes de hardware y software de un computador conectados en red se comunican y coordinan mediante paso de mensajes.”* (G.Coulouris, J.Dollimore, T.Kindberg)
- *“Aquel que le impide a uno continuar su trabajo cuando falla un computador del cual uno nunca ha oído hablar.”* (L.Lamport)

# Definición SSDD

- Cosas en común de las definiciones anteriores:
  - Son breves
  - Son mezquinas
  - Son ambiguas
- Nos dan la libertad para **tomar decisiones**
  - Diseño
  - Implementación
  - Mantenimiento
  - Algoritmos

# Definición SSDD

- Nuestra definición
  - *Un sistema distribuido es una colección de **entidades**, cada una **autónoma**, **programable** y **propenso a fallas** las que se comunican a través de un **medio no confiable** para coordinarse en la realización de una tarea específica.*
- **Entidades**: Un proceso en un computador o dispositivo.
- **Medios**: Conectividad física. Cableados o inalámbricos.
- **Libertad en su implementación práctica**

# Definición SSDD

- Trade off cosas buenas y malas.



## Beneficios

- Economía.
- Aumento en capacidad de procesamiento.
- Aplicaciones inherentemente distribuidas.
- Capacidad de crecimiento.
- Fiabilidad y disponibilidad.
- Compartir recursos y datos.



## Desventajas

- Fallas más frecuentes
  - Red, Latencia, pérdida de mensajes
- Interoperabilidad
  - Falta de estándares únicos
- Seguridad
- **Aumento de la complejidad**
- **Costos de administración**

# Definición SSDD

Implicancias básicas de **complejidad** al considerar usar un SSDD:

- *Concurrencia.*
  - Recursos compartidos. Acceso concurrente.
  - Sincronización.
- *Sin tiempo global.*
  - Cada computador tiene su propio reloj local.
  - Coordinación.
- *Fallas independientes.*
  - Posibilidad permanente de fallas en cada componente.
  - Detección de fallas y tolerancia.



# Definición SSDD

Es **complejo** construir un SSDD, pero es posible

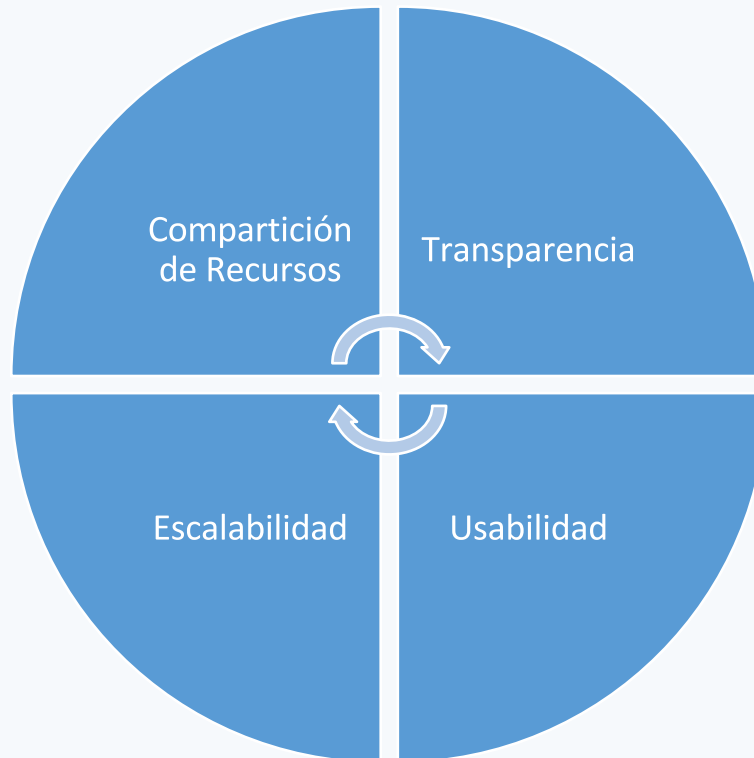
**Es posible construir un SD**

**$\Rightarrow ? \Rightarrow$**

**Es una buena idea construirlo**

# Definición SSDD

Aspectos de diseño básicos en un sistema distribuido



# Definición SSDD

## **Compartición de Recursos (Accesibilidad de recursos)**

- El objetivo básico de los sistemas distribuidos.
  - Usuarios deben tener acceso fácil a recursos compartidos y compartir sus propios recursos de forma controlada.
    - Impresoras
    - Computadores
    - Espacio de almacenamiento
    - Archivos, datos, páginas Web, redes, etc.
- ¿Por qué compartir recursos?
  - Costos, capacidad, escalabilidad

# Definición SSDD

## Transparencia

- Esconder el hecho que los recursos y procesos están físicamente distribuidos en diferentes máquinas.
- Un sistema distribuido que se presenta hacia el usuario como una sola máquina se dice ser **transparente**.

# Definición SSDD

## Transparencia

- La transparencia se aplica a varios aspectos de sistemas distribuidos:

Transparencia	Descripción
Acceso	Esconde las diferencias entre las <b>distintas representaciones</b> de datos y la formas en que éstos son accedidos por los usuarios.
Ubicación	Esconde la <b>ubicación física</b> de donde está situado un recurso.
Migración	Esconde el hecho que recursos puedan <b>moverse de una ubicación física a otra</b> sin afectar el modo en que se acceda.
Re-localización	Esconder el que los recursos son movidos de una ubicación a otra <b>mientras están en uso</b> y sin que el usuario note nada.
Replicación	Esconde el hecho que un recurso pueda estar <b>multiplicado</b> varias veces para mejorar su rendimiento y/o disponibilidad.
Concurrencia	Esconde que distintos usuarios están <b>compitiendo en paralelo por acceder</b> o usar recursos, o que un recurso está siendo compartido con otros usuarios
Fallas	El usuario no nota (o nunca supo) que un recurso ha dejado de funcionar (ni de si se ha recuperado).

# Definición SSDD

## Transparencia

- La transparencia tiene límites y grados y brindarla depende fuertemente de la capacidad de nuestros sistemas.
  - No siempre es posible esconder todo
    - Ejemplo: desfase temporal de comunicación entre puntos muy distantes (delay).
  - Siempre hay un trade-off entre transparencia y rendimiento
    - Ejemplo: Bases de datos replicadas en el mundo. Un caso particular son los servidores DNS que pueden tomar días en actualizarse, lo cual no se oculta al usuario.
  - Hay formas y formas (La correcta, la incorrecta... y al estilo MaxPower)...
    - Ejemplo: Insistencia de una aplicación por conectar a un servidor para “ocultar” la desconexión, termina degradando toda la red. ¿Qué estrategia sería mejor?

# Definición SSDD

## Escalabilidad

- Se refiere a la capacidad de crecer de un sistema.
- Típica limitación de cuello de botella en el servidor.
- Existen varias limitaciones típicas en la escalabilidad:

Concepto	Ejemplo
Servicios Centralizados	Un único servidor para múltiples usuarios
Información Centralizada	Un único directorio telefónico disponible a la vez
Algoritmos Centralizados	Un algoritmo de ruteo basado en la información completa

# Definición SSDD

## Escalabilidad

- La escalabilidad se puede dimensionar desde tres aspectos
  - **Tamaño:** la capacidad de agregar fácilmente más recursos y usuarios al sistema.
  - **Geográfica:** la capacidad de que usuarios y recursos se encuentren a distancias considerables
  - **Administración:** La capacidad de mantener una administración fácil a pesar del crecimiento
- Desafortunadamente los sistemas son escalables en alguna de las dimensiones sacrificando rendimiento en las otras



# Definición SSDD

## Escalabilidad

- Un SSDD es determinado por su software
- El desafío de escalar algoritmos se resuelve por medio de la descentralización de los mismos.
- Un algoritmo descentralizado normalmente se distingue de uno centralizado pues, en el primero:
  - Ninguna máquina tiene información del sistema completo
  - Las decisiones se basan en la información local
  - Un desperfecto localizado (de una máquina) no interrumpe la rutina general (El SSDD completo)
  - Se omite la existencia de un reloj global

# Definición SSDD

## Escalabilidad

- Básicamente, reconoceremos 3 técnicas para efectuar escalamiento:
  - Ocultar Latencias
    - Ocultar las esperas en peticiones remotas => Comm. Asíncrona
    - JS
  - Distribución
    - Dividir componentes en unidades más focalizadas
    - Zonas DNS
  - Replicación
    - Multiplicación de identidad de un componente
    - Caché

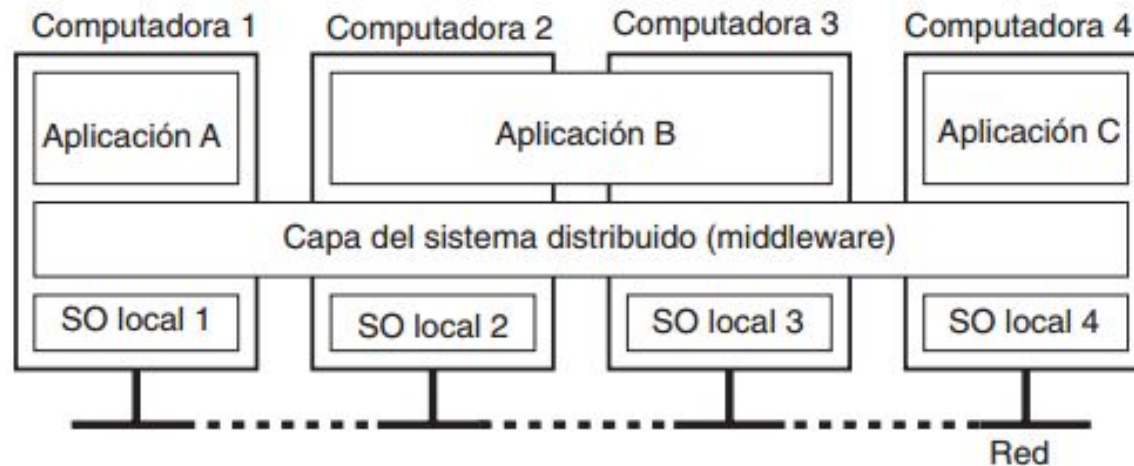
# Definición SSDD

## Usabilidad

- Los recursos y servicios son ofrecidos de acuerdo a reglas estándares describiendo la semántica y sintaxis.
- Usualmente estas reglas son descritas vía protocolos o interfaces.
  - HTTP
  - Web services
  - APIs
- Dos aspectos muy relevantes (en SSDD abiertos especialmente)
  - Completitud
  - Neutralidad
  - Facilidad de configuración

# SSDD en la práctica

Con el objetivo de dar soporte sobre máquinas heterogéneas, los SSDD se organizan en una capa de software denominada *middleware*



**La capa middleware ofrece a cada aplicación la misma interfaz, es decir, oculta las diferencias de hardware y software de cada máquina**

# Los 8 pecados en SSDD

Los SSDD difieren del software tradicional porque sus componentes están en efecto, distribuidos.

1. La red es confiable
2. La red es segura
3. La red es homogénea
4. La topología no cambia
5. La latencia es igual a cero
6. El ancho de banda es infinito
7. El costo de transporte es cero
8. Existe un administrador

# Tipos de SSDD

- Distinción HW y Software
- Viernes 9 de Septiembre