

# Programación Distribuida y Tiempo Real

## Bases de Datos Distribuidas

### Clase Introductoria 2

#### 1. Diapositivas Tanenbaum-Van Steen

<https://www.distributed-systems.net/index.php/books/distributed-systems/>

(en ese mismo sitio pueden pedir una versión electrónica personal de la 3ra ed.)

Búsquedas, consultas, comentarios, etc.

#### 2. Características de los sistemas distribuidos (lo que *hay* en todos ellos)

2.1. Concurrencia y recursos compartidos (*Desafío* para Coulouris, cap. 1)

2.2. Memoria distribuida

2.3. No hay idea de *estado global a priori* (tiempo, por ejemplo)

2.4. Escalabilidad (*Objetivo* para Tanenbaum, cap. 1)

2.5. ¿? Seguridad (*Desafío* para Coulouris)

2.6. Apertura-Extensibilidad (*Objetivo* para Tanenbaum, *Desafío* para Coulouris)  
relación con otra característica:

2.7. Heterogeneidad (*Desafío* para Coulouris)

2.8. Transparencia (Cou-Chapter 1 slides.ppt, p. 8 y Tan-chap-01.ppt, p.4)

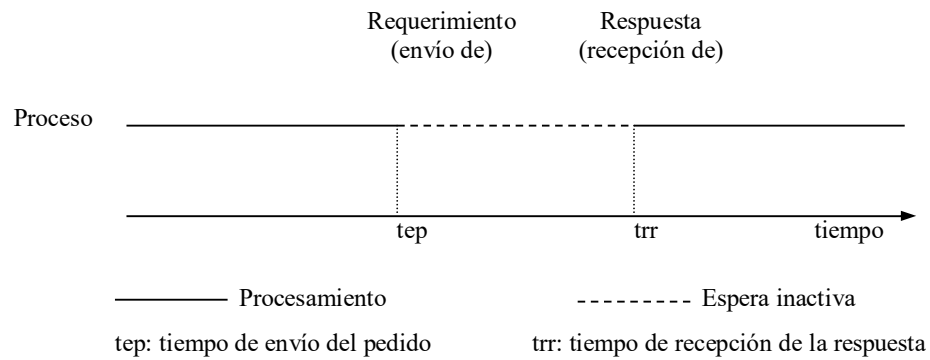
#### 3. Ideas generales del modelo de procesamiento cliente/servidor

3.1. ¿De dónde “sale” c/s?

3.2. ¿Por qué hay variaciones de c/s?

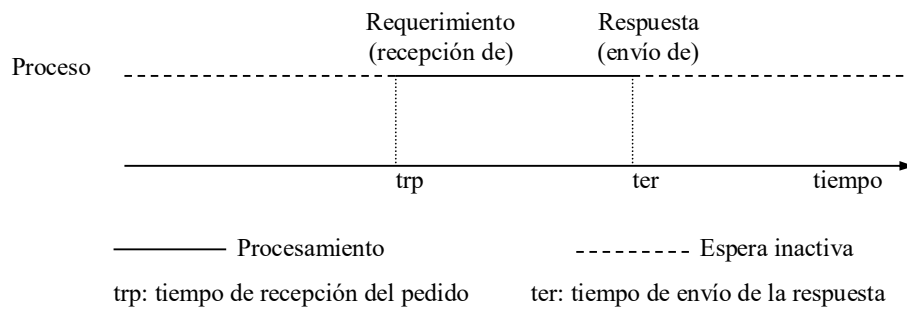
3.3. ¿Por qué no alcanzaría con c/s?

### 3.4. Esquema temporal de procesamiento para el cliente



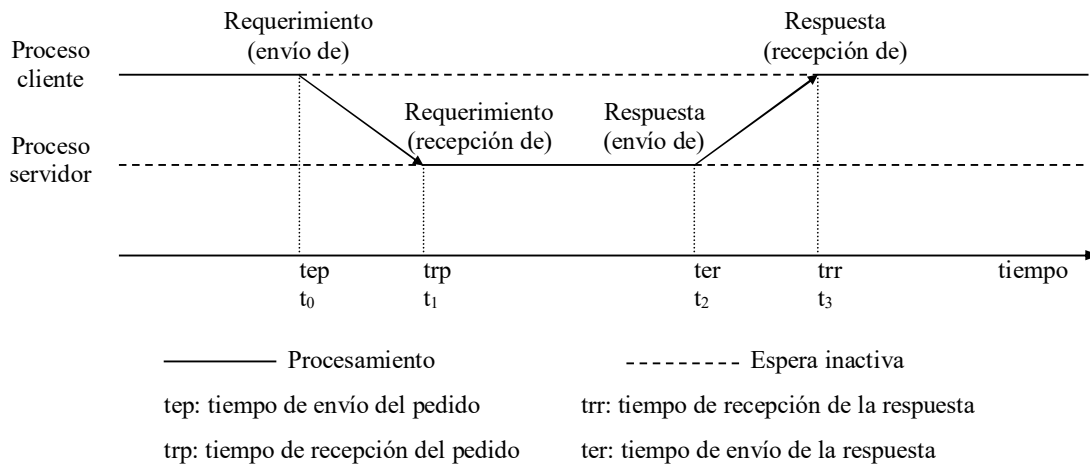
#### Proceso que Pide un Servicio

### 3.5. Esquema temporal de procesamiento para el servidor



#### Proceso que Proporciona un Servicio

### 3.6. Pero no es todo, están las comunicaciones...



#### Procesamiento Cliente/Servidor de un Requerimiento

#### 4. Caracterización del modelo de procesamiento cliente/servidor

- 4.1. Es un modelo más allá de la cantidad de máquinas, establece dos tipos de procesos que interactúan: cliente/s y servidor/es
- 4.2. Los clientes son activos en la interacción, *inician* con la petición
- 4.3. Los servidores son pasivos en la interacción, no conocen cuándo llegará una petición
- 4.4. Los clientes son los que usan o necesitan recursos que requieren al servidor
- 4.5. Los servidores tienen o administran los recursos, generalmente no los usan
- 4.6. Los clientes solamente tienen la visión de los recursos que necesitan
- 4.7. Los servidores tienen una visión más *amplia* del estado de todos los recursos
- 4.8. Ni los clientes ni los servidores deben necesariamente encargarse de la transferencia de la información entre ellos (y normalmente no lo hacen)
- 4.9. Tanto clientes como servidores tienen bien definida la *interfase*:
  - 4.9.1. Cómo se pide un servicio y qué datos son necesarios desde el cliente hacia el servidor
  - 4.9.2. Cómo se retorna la respuesta a un cliente, básicamente qué datos son devueltos al cliente en respuesta al servicio requerido

En este sentido, el modelo c/s es casi la forma *lógica* de procesamiento al menos en el inicio de los sistemas distribuidos, no se hace más (ni menos) que seguir el modelo de interacción de procesos de usuario con los sistemas operativos y las ideas más conceptuales de los protocolos de comunicación

#### 5. Ventajas y desventajas de los sistemas distribuidos con respecto a sistemas centralizados y a sistemas independientes

##### 5.1. Ventajas:

- 5.1.1. Costo/rendimiento
- 5.1.2. Distribución inherente del *problema*
- 5.1.3. Crecimiento incremental (o *granularidad* del crecimiento) del hardware
- 5.1.4. Recursos compartidos (frente a *sistemas independientes*)

##### 5.2. Desventajas:

- 5.2.1. Software (desde el existente hasta la ingeniería de software)
- 5.2.2. ¿? Seguridad (quizás por el software)
- 5.2.3. ¿? Características de las redes actuales (fallas, diferencias de rend.)
- 5.2.4. Administración

6. Tipos de sistemas distribuidos: slides.01.pdf, pp. 12-21

6.1. Distributed Computing Systems

6.1.1. Cluster computing

6.1.2. Grid computing ==> Cloud computing

**6.2. Distributed Information Systems**

6.2.1. Procesamiento de transacciones (p. 17)

6.2.2. Comunicación entre aplicaciones (quizás preexistentes)

6.3. Distributed Pervasive Systems