Sistemas de Información

Introducción a los Sistemas de Información: El Modelo Cliente/Servidor

Agradecimientos: por su contribución a la realización de estas transparencias: Jesus Villamor Lugo y Simon Pickin (IT-UC3M), Juan José Gil Ríos (Terra.)

Índice

- Definición
- Concepto
- Clasificación
  - Servicio
  - Cómo se distribuye la aplicación entre C y S
- Arquitecturas multinivel

Definición

- *La arquitectura C/S es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realiza se efectúa con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema*

Cambio de paradigma

- **1ª Revolución: Cliente – servidor**
  - Del Mainframe a los sistemas C/S
  - Detonante: desarrollo HW, LAN
  - Resultado Mainframe - > C/S
- **2ª Revolución: Objetos distribuidos**:
  - Fragmentación del cliente y servidor en componentes
  - Detonante: desarrollo HW, WAN, Internet
  - Resultado C/S -> Sistemas 3 niveles o multinivel
- **Consecuencias para desarrolladores de los SI**
  - Incertidumbre
  - De sistemas propietarios a sistemas “a la carta”
  - Necesidad de conocer gran número de tecnologías

El Sistema de Información moderno y el modelo Cliente/Servidor

- El Sistema de Información moderno
  - Administra y despliega grandes redes
  - Ofrece estándares de interoperabilidad
  - Distribuye sus funcionalidades
  - Saca partido del modelo Cliente/Servidor
  - Muchas veces requiere habilidades híbridas
    - Procesamiento de transacciones, bases de datos, comunicaciones o conocimientos sobre GUI.

El Modelo Cliente/Servidor
El concepto

- Sistema distribuido donde el software está dividido entre
  - tareas servidor
  - tareas cliente
- Separación clara de responsabilidades
  - en base a la noción de servicio
- Papel del cliente:
  - inicia el diálogo
  - envía peticiones al servidor conforme a algún protocolo asimétrico
  - pide que el servidor actúe, o que le informe, o ambas cosas
- Papel del servidor:
  - espera pasivamente peticiones de los clientes
  - responde a las peticiones según su política

El Modelo Cliente/Servidor
Consecuencias

- Un servidor puede atender a muchos clientes
- Puede haber uno o varios servidores en un sistema
- Un servidor puede ser substituido por otro que ofrece (al menos) el mismo servicio sin afectar a los clientes
- Se puede ocultar a los clientes la ubicación del servidor
  - la ubicación no afecta la manera de utilizar los servicios
- El servidor puede regular el acceso a recursos compartidos
  - e.g. servidor X, servidor de impresión,...
- En el caso general, un objeto/componente/programa puede ser cliente, servidor o ambos

El Modelo Cliente/Servidor

Ventajas

- Base en la noción de servicio → buena estructura
  - acoplamiento cliente-servidor débil, comunicación por mensajes
  - interfaces claras, modularidad, flexibilidad
- Escalabilidad “vertical”
  - facilita: migrar a servidor más grande / veloz o servidores múltiples
- Escalabilidad “horizontal”
  - facilita: añadir clientes
- Hardware y plataformas software (SO) heterogéneos
  - despliegue independiente de cliente y servidor
  - clientes / servidores pueden usar el hardware y SO más adecuados para su función, ej. cliente barato, servidor rápido
- Robustez
  - servidor protegido contra fallos en el cliente

El Modelo Cliente Servidor
Dos puntos de vista

- Cliente y servidor como **entidades físicas**
  - Un servidor no es cliente; un cliente no es servidor
  - Granularidad al nivel de subsistema (gruesa)
  - Contexto: arquitecturas de aplicaciones comerciales
- Cliente y servidor como "**roles**"
  - La misma entidad puede actuar como cliente o servidor
  - Granularidad al nivel de objeto o componente (fina)
  - Contexto: tecnologías de objetos distribuidos

El Modelo Cliente Servidor (Entidades)
Clasificación 1: En función del servicio

- Servidores de **archivos**
  - Msg.: Peticiones de archivos
  - NFS, SAMBA,...
- Servidores de **bases de datos**
  - Msg.: Peticiones SQL
  - Oracle, Sybase, SQL Server,…
- Servidores de **transacciones**
  - Msg.: Transacción (Conjunto de peticiones SQL)
  - OLP,...
- Servidores de **objetos**
  - Msg.: Invocación a procedimientos remotos
  - servidores CORBA, OLE/DCOM,...
- Servidores **Web**
  - Msg.: Peticiones HTTP
  - servidores HTTP,...
- Servidores de ***groupware***
  - Msg.: Mensajes de groupware, *e-mails*
  - Lotus Notes, Exchange, etc.

El Modelo Cliente Servidor (Entidades)
Clasificación 2: ¿Cómo distribuir aplicación?

- **Cliente pesado** / servidor ligero
  - Mayor parte de la aplicación corre en el lado cliente
  - Servidor exporta datos en bruto
  - Clientes saben de organización de datos en el servidor
- **Cliente ligero** / servidor pesado
  - Mayor parte de la aplicación corre en el lado servidor
  - Servidor exporta métodos que operan sobre los datos
  - Cliente no es mucho más que el GUI

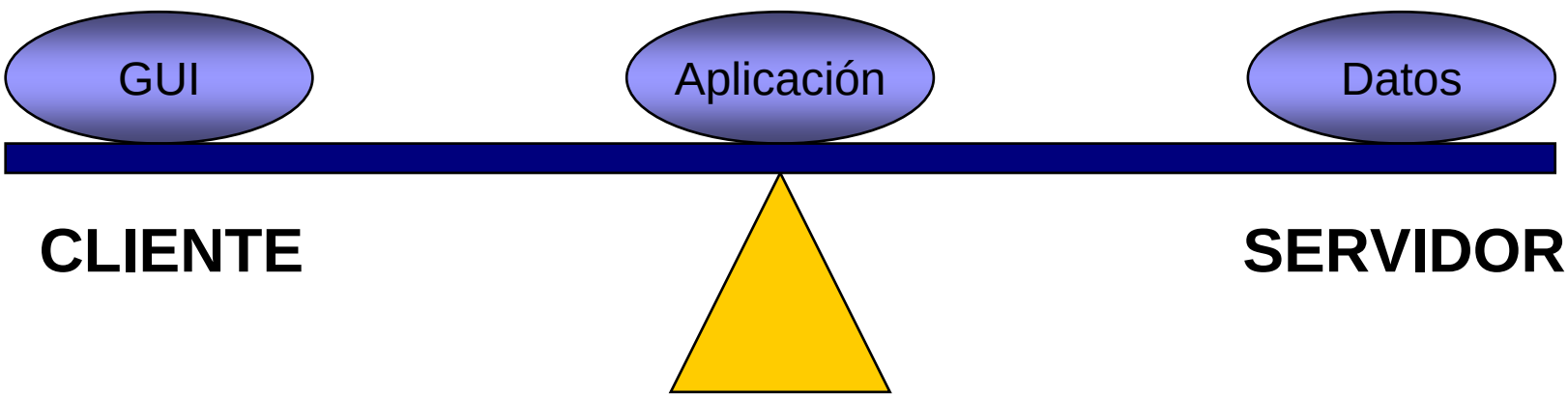
El Modelo Cliente Servidor (Entidades)
Clasificación 2: ¿Cómo distribuir aplicación?

Cliente pesado

- Servidor de archivos
- Servidor de BD
- Servidor de obj. distribuidos

Cliente ligero (servidor pesado)

- Servidor Web
- Servidor de transacciones
- Servidor de Groupware
- Servidor obj distribuidos



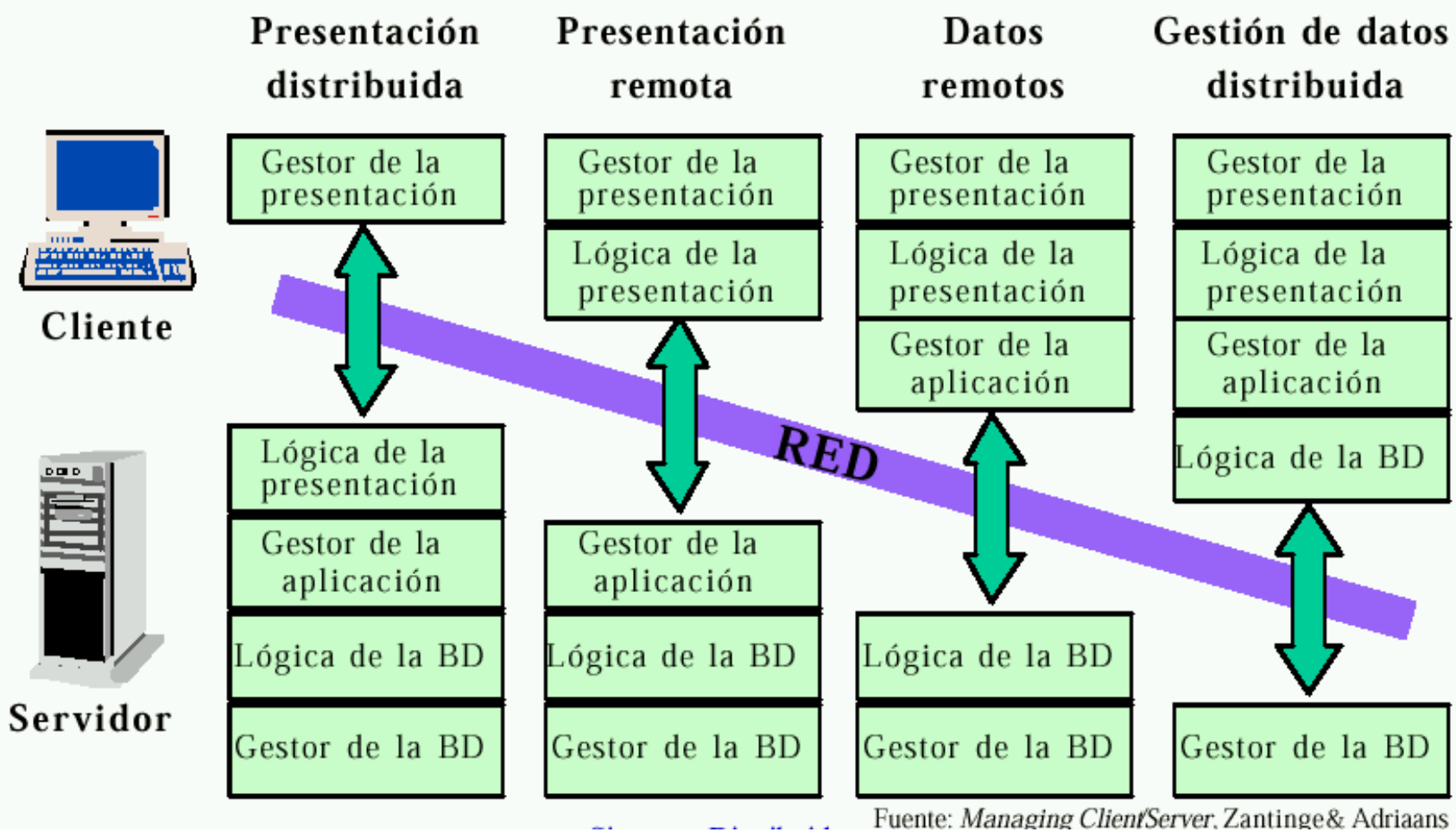
El Modelo Cliente Servidor (Entidades)
Arquitecturas de distintos niveles

- Aplicaciones comerciales se dividen en tres partes
  - acceso a datos
  - lógica de la aplicación (o lógica del negocio)
  - presentación (interfaz de usuario)
- **Acceso a datos**
  - gestión y acceso a datos persistentes
- **Presentación**
  - presentación de resultados al usuario de forma comprensible
- **Lógica del negocio**
  - el procesamiento

El Modelo Cliente Servidor (Entidades)
¿Dos niveles, tres niveles o multi-nivel?

- En todas las arquitecturas cliente-servidor
  - nivel del cliente (*client tier*): presentación
  - nivel de datos (*data tier*): acceso a datos
- **Arquitectura de dos niveles** (*2-tier*) : C-S clásico
  - lógica de la aplicación integrada
    - o bien con la presentación
    - o bien con el acceso a datos
    - o bien con ambos
- **Arquitectura de tres niveles** (*3-tier*)
  - lógica de la aplicación localizada en el nivel del medio, separada
    - tanto del acceso a datos
    - como de la presentación
- **Arquitectura multi-nivel** (*multi-tier*)
  - nivel del medio se divide en distintos niveles

El Modelo Cliente Servidor (Entidades)
Estrategias de distribución C/S



Fuente: *Managing Client/Server*, Zantinge& Adriaans

El Modelo Cliente Servidor (Entidades)
Ventajas del cliente ligero

- Menos **infraestructura** en el lado cliente
  - reduce costes puesto que hay muchos clientes, pocos servidores
- **Administración** más facil
  - es decir, configuración, mantenimiento, despliegue,…
  - puesto que hay menos servidores que clientes
- Menos **tráfico** en la red
  - debido a un nivel de servicio más abstracto ofrecido al cliente
- **Gestion** de recursos centralizado
  - ayuda a asegurar la integridad de los datos
  - mayor nivel de seguridad
  - mejor detección de fallos
- Más **evolutivo**, p.e. frente a un cambio del SGBD

El Modelo Cliente Servidor (Entidades)
Ventajas de las arquitecturas multi-nivel

- Todas las ventajas del cliente ligero
- Más **flexibilidad** y **escalabilidad**
- Niveles pueden **actualizarse / remplazarse** independientemente
  - Con cambios de requisitos
  - Con cambios de tecnología
- Un **control más fino de la carga** del servidor permite
  - evitar sobrecarga del servidor
  - equilibrar la carga entre servidores
  - conseguir tiempo de respuesta más bajo
    - pero al aumento del número de niveles puede aumentar el número de comunicaciones y por tanto el tiempo de respuesta, ¡ojo!
- Más facilidad para **depurar errores**
  - debido a una mayor modularidad

El Modelo Cliente Servidor (Entidades)
Diseño de un servidor

- Con estado o sin estado
  - con estado: más flexible, e.g. “carro de compra”
  - sin estado: más tolerancia a fallos (pero “cookies”)
- ¿Concurrencia?
  - multi-hilos: “hilos de servicio” viven en el mismo espacio de direcciones que el “hilo de escucha”
  - atención a la sincronización de hilos
- “Granjas” de servidores / réplica de servidores
  - transparente para el cliente, ¿cómo?
  - coherencia de servidores ≡ sincronización de hilos
  - distintas aplicaciones, distintos requisitos de sincronización

El Modelo Cliente Servidor
¿Qué es el *middleware*?

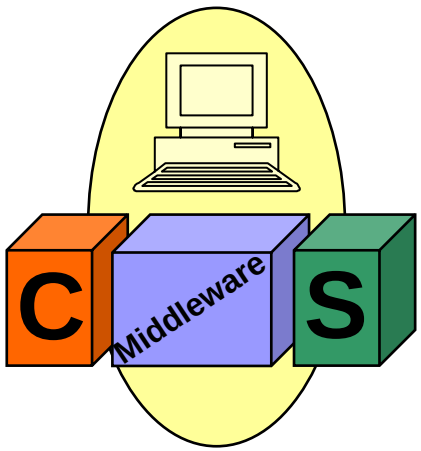
- Contexto del modelo cliente-servidor como entidades
  - la tecnología que conecta entre sí los niveles de una arquitectura multi-nivel
- Contexto del modelo cliente-servidor como *roles*
  - el software distribuido necesario para el soporte de interacciones entre clientes y servidores a través de una plataforma heterogénea.
- Empieza en el API de la parte del cliente y comprende
  - la transmisión de la solicitud a través de una red
  - la transmisión de la respuesta resultante del servidor

El Modelo Cliente Servidor

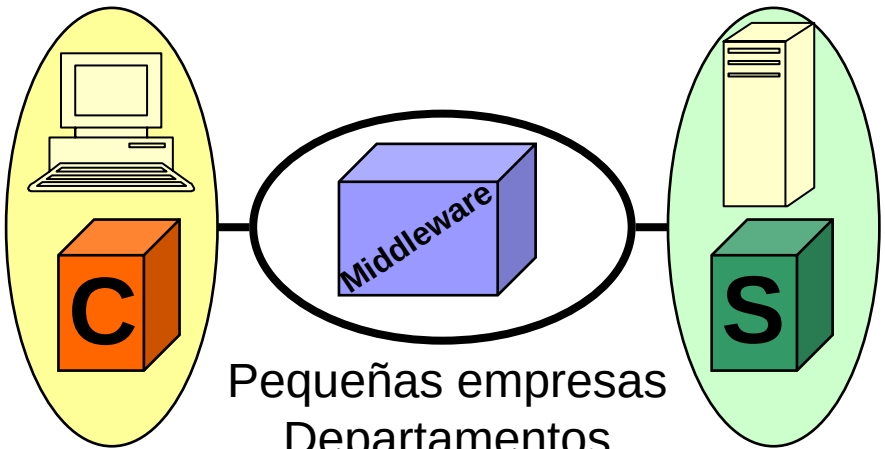
¿Qué es el *middleware*?

- *Middleware* general:
  - pilas de comunicación
  - directorios distribuidos
  - servicios de autenticación
  - llamadas a procedimiento remoto (RPC)
  - ...
- *Middleware* de servicios específicos
  - para bases de datos: ODBC, JDBC,...
  - para groupware: Lotus Notes,...
  - para objetos: CORBA 2, DCOM...
  - para componentes: CORBA 3, .NET...
  - para web: HTTP, SSL, SOAP
  - ...

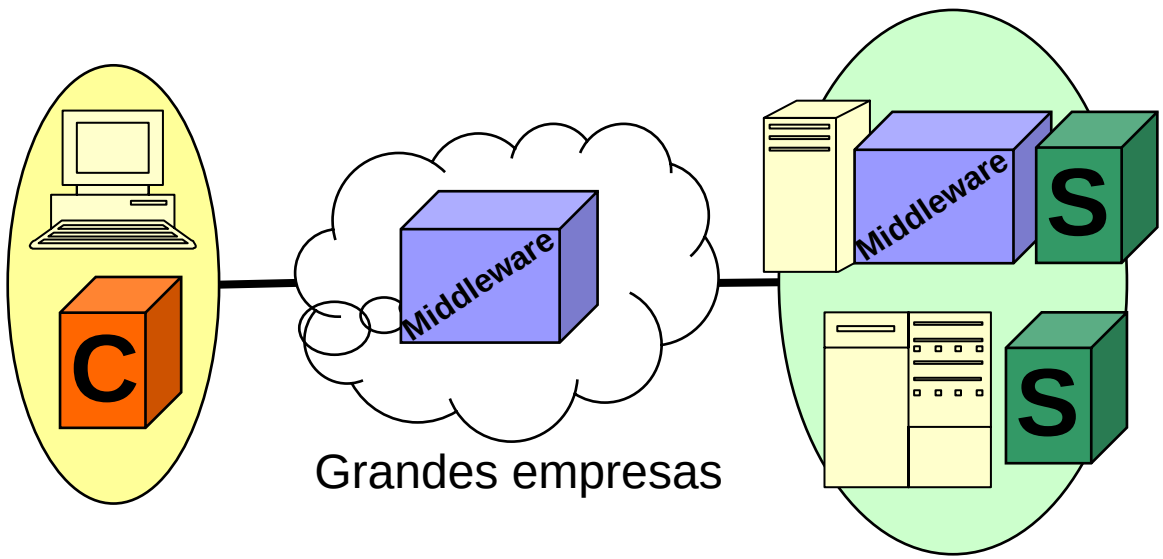
El Modelo Cliente Servidor (Roles)



Negocio familiar



Pequeñas empresas
Departamentos



Grandes empresas

El Modelo Cliente Servidor (Roles)