

Aplicaciones Distribuidas

◆ Informática III

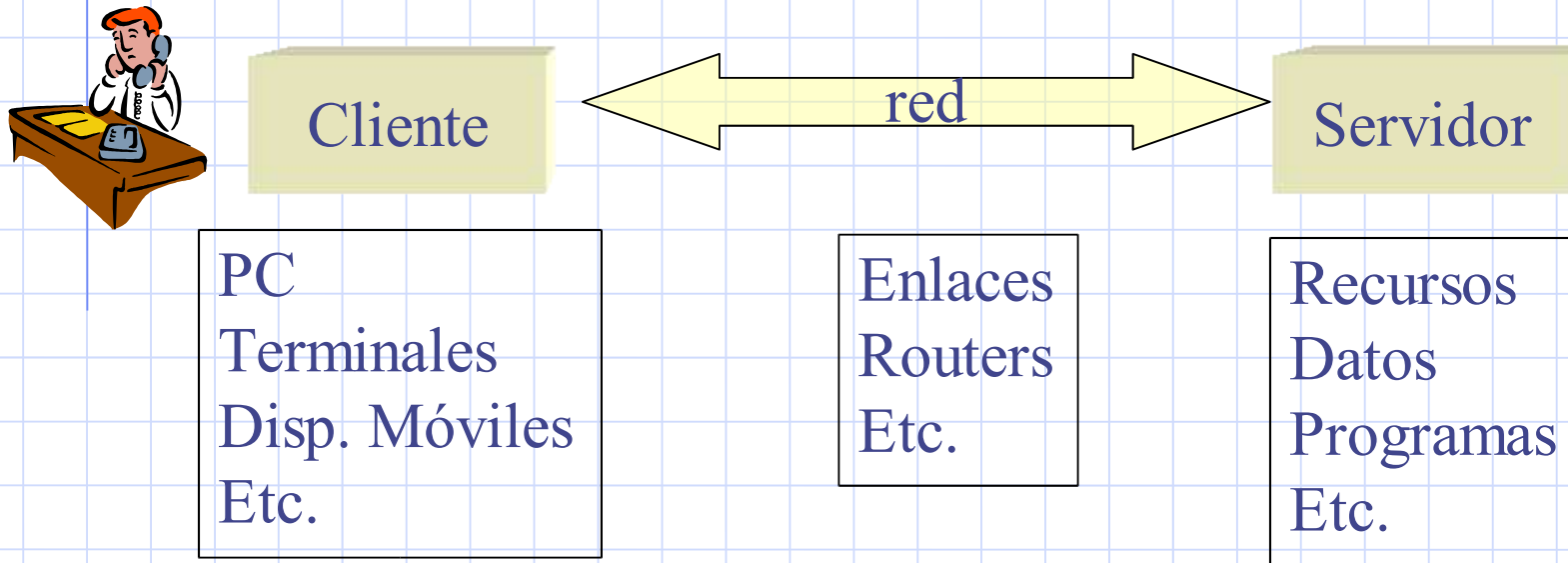
Temario

- ◆ Elementos arquitecturales
- ◆ Arquitecturas tradicionales
- ◆ Arquitecturas Cliente/Servidor
- ◆ Arquitecturas distribuidas

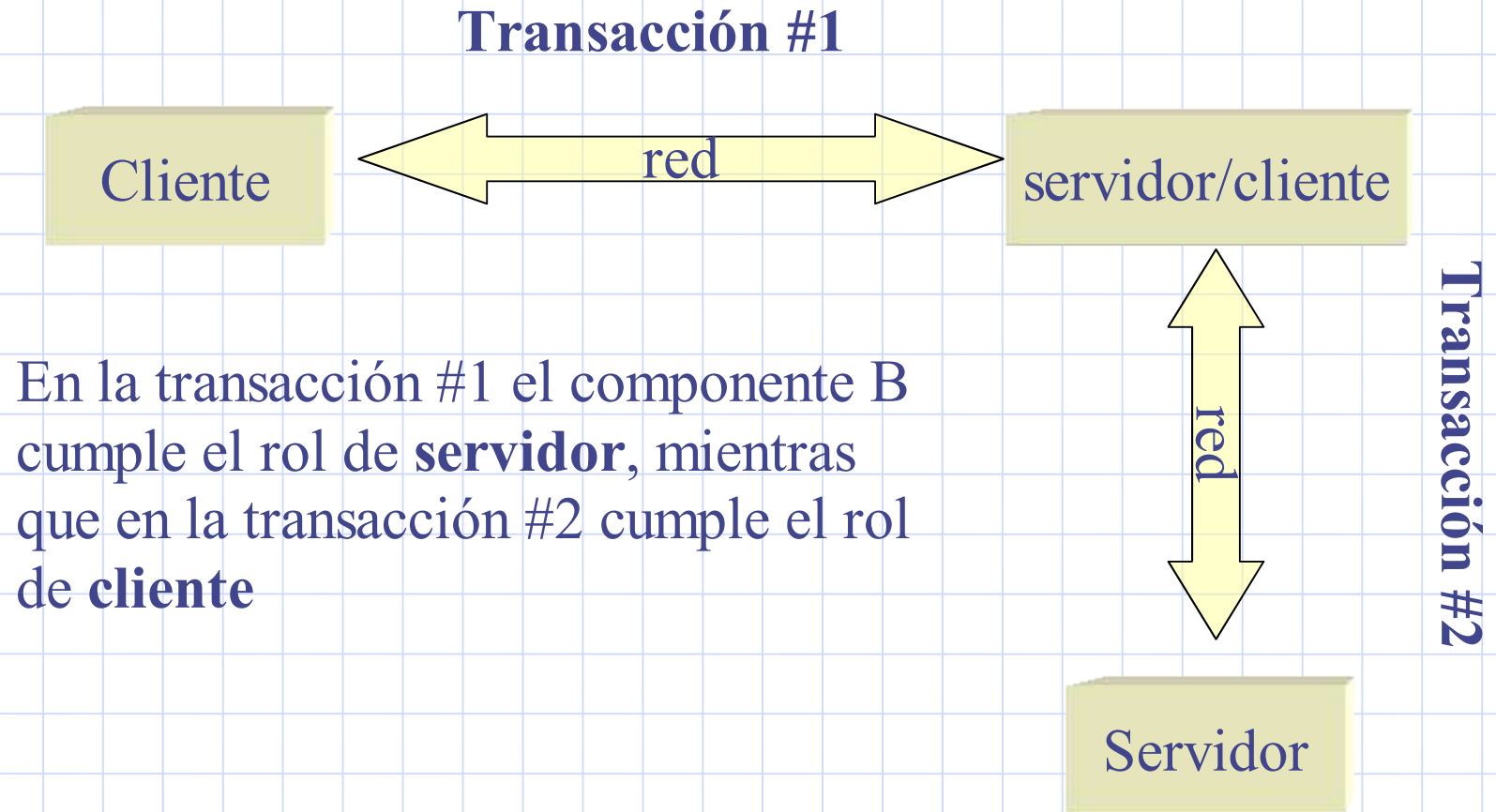
Elementos Arquitecturales

- ◆ Componentes de red
- ◆ Interfaces de usuario
- ◆ Administración de datos
- ◆ Administración del procesamiento

Componentes de Red



Roles



Interfaces de Usuario

El dispositivo de acceso de los usuarios humanos a un sistema distribuido se llama “*front-end*” o **interface de usuario**. Generalmente es un conjunto de pantalla, teclado y mouse, pero en forma creciente vemos nuevos dispositivos como teléfonos celulares, PDA’s, etc. Genéricamente se denominan “**cliente**”

Administración de Datos

- ◆ Es el “otro extremo” visto desde la interface de usuario
- ◆ Involucrado en el almacenamiento y recuperación de datos
- ◆ Las tecnologías aplicables incluyen archivos planos, bases de datos relacionales u orientadas a objetos, etc.

Administración del Procesamiento

- ◆ Todo aquello involucrado con la **transformación** de los datos: genéricamente conocido como **programas o aplicaciones**

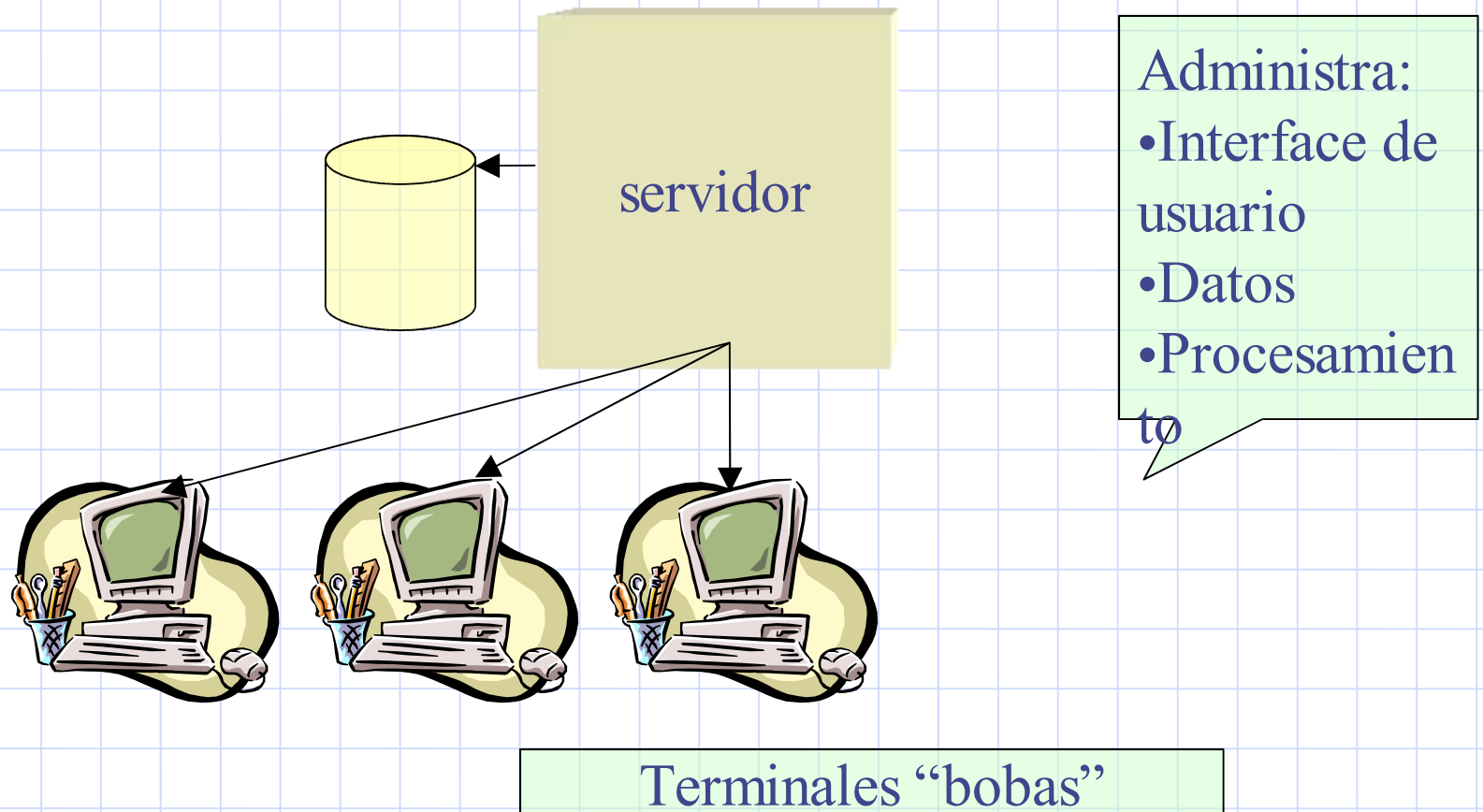
Arquitecturas Tradicionales

- ◆ Procesamiento centralizado
- ◆ Servidores de archivo

Procesamiento Centralizado

- ◆ Modelo surgido en los orígenes de la tecnología de la computación, durante la década de los '60.
- ◆ Toda la capacidad de cómputo está localizada en el servidor central
- ◆ Es el modelo de aplicación de mainframes y sigue en uso en la actualidad

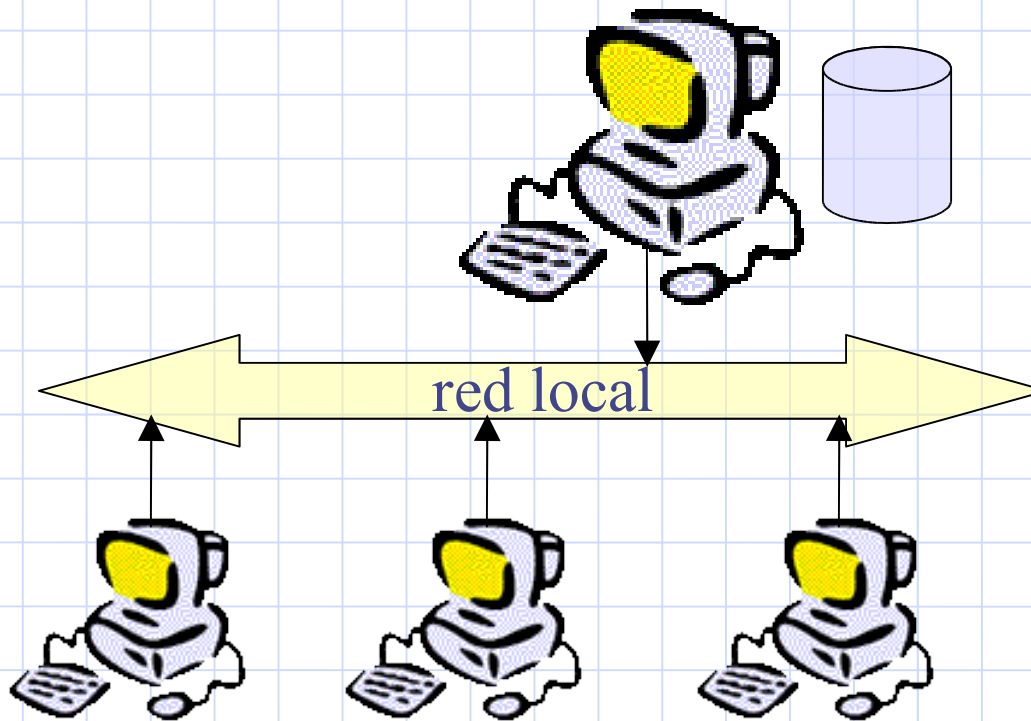
Modelo Arquitectural



Servidor de Archivos

- ◆ Durante los '80 se impuso por su bajo costo y sencillez
- ◆ Aprovecha la capacidad de procesamiento creciente de las PC's con la masificación de las redes locales
- ◆ El rol del servidor se limita al servicio de almacenamiento e impresión

Modelo Arquitectural



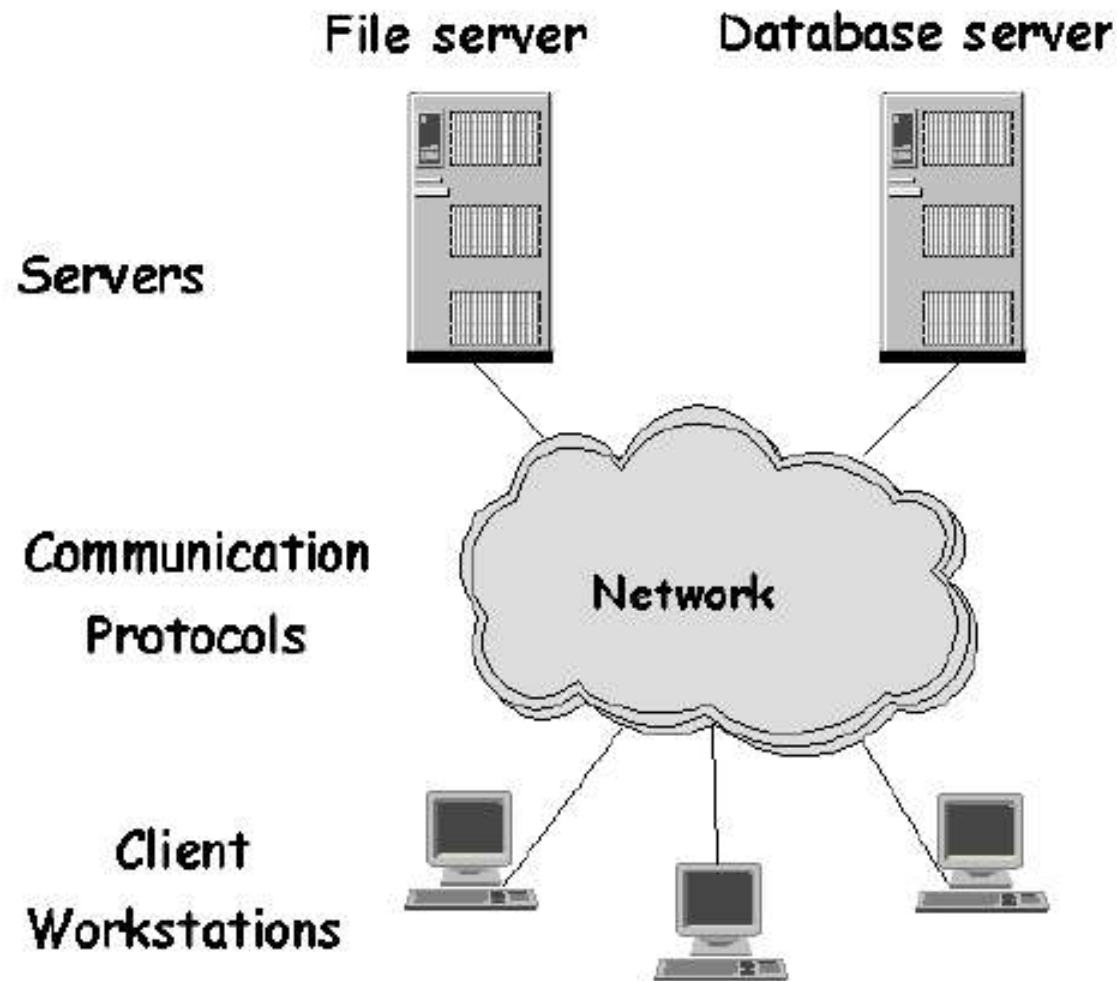
Arquitectura Cliente/Servidor

- ◆ Modelo de dos capas
- ◆ Aplicaciones web
- ◆ Modelo de tres capas

Modelo de Dos Capas

- ◆ Es la partición física y lógica de una aplicación en dos componentes que interactúan:
 - El lado "cliente"
 - El lado "servidor"
- ◆ La falta de cualquiera de ambas imposibilita el uso de la aplicación

Modelo Arquitectural



Modelo Cliente-Servidor

- ◆ La aplicación se particiona en una capa de presentación y lógica que corre del lado cliente, y otra de gestión de datos, que corre del lado servidor
- ◆ Requiere de dispositivos clientes con capacidad de procesamiento importante
- ◆ Tiene problemas de escalabilidad y administración

Aplicaciones Web

- ◆ El boom de Internet y el World Wide Web en los '90 dio gran impulso a un nuevo modelo de construcción de aplicaciones, conceptualmente similares al modelo centralizado pero con alcance global
- ◆ Las primeras aplicaciones de este tipo fueron formularios de búsqueda

Aplicaciones Web

- ◆ El servidor y el cliente dialogan utilizando un protocolo llamado HTTP (HiperText Transfer Protocol) diseñado por el CERN de Suiza en la década de los '80
- ◆ Este protocolo se basa en **pedidos** (operación GET) del cliente a los que el servidor responde enviando **documentos**

Aplicaciones Web

- ◆ El lado cliente es universal, centrado en programas conocidos como “navegadores” (browsers)
- ◆ Un navegador “interpreta” texto formateado en una codificación llamada HTML (HiperText Markup Language)
- ◆ El servidor es conocido como “Web Server”

HiperText Transfer Protocol

Servidor HTTP

Cliente: *Browser*

← Requiere hipervínculo

Envía respuesta →

NCSA, CERN
NetScape, Microsoft

Explorer, Netscape,
Mosaic, HotJava

Navegadores

- ◆ Los primeros fueron Lynx y Mosaic en entornos Unix
- ◆ Netscape primero e Internet Explorer después lograron gran difusión
- ◆ Existen navegadores (mas correctamente clientes html) para dispositivos móviles y no tradicionales

Servidores Web

- ◆ Son programas capaces de “hablar” el protocolo HTTP con los clientes
- ◆ Están diseñados para entregar eficientemente gran número de respuestas a clientes concurrentes
- ◆ El primer servidor fue NCSA httpd, hoy son muy utilizados Apache (open source), MS-IIS y Dominó, entre otros

HTML

```
<html>
<head>
<title>Escuela de Ingeniería Electrónica</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<frameset rows="73,*,18" framespacing="0" border="0" frameborder="0">
  <frame name="superior" scrolling="no" noresize target="contenido" src="titulo.htm">
  <frameset cols="133,*">
    <frame name="contenido" target="principal" src="index2.htm" scrolling="auto">
    <frame name="principal" src="Principal.htm" target="_self" scrolling="auto">
  </frameset>
  <frame name="inferior" scrolling="no" noresize target="contenido" src="pie%20de%20pagina.htm">
</frameset>
<body topmargin="0" leftmargin="0">
<p>Esta página usa marcos, pero su explorador no los admite.</p>
</body>
</noframes>
</frameset>
</html>
```


HTML Dinámico

- ◆ La gran difusión de la web impuso la necesidad de dar mayor interactividad a las páginas HTML
- ◆ Surge así un conjunto de técnicas basadas en formularios, en particular el CGI (Common Gateway Interface), que permitían la creación “dinámica” de páginas HTML

Código Activo

- ◆ Surge también un conjunto de herramientas de scripting utilizada para enriquecer la interactividad del modelo
- ◆ JavaScript, Java, ASP, JSP, etc. Son ejemplos de esta técnica

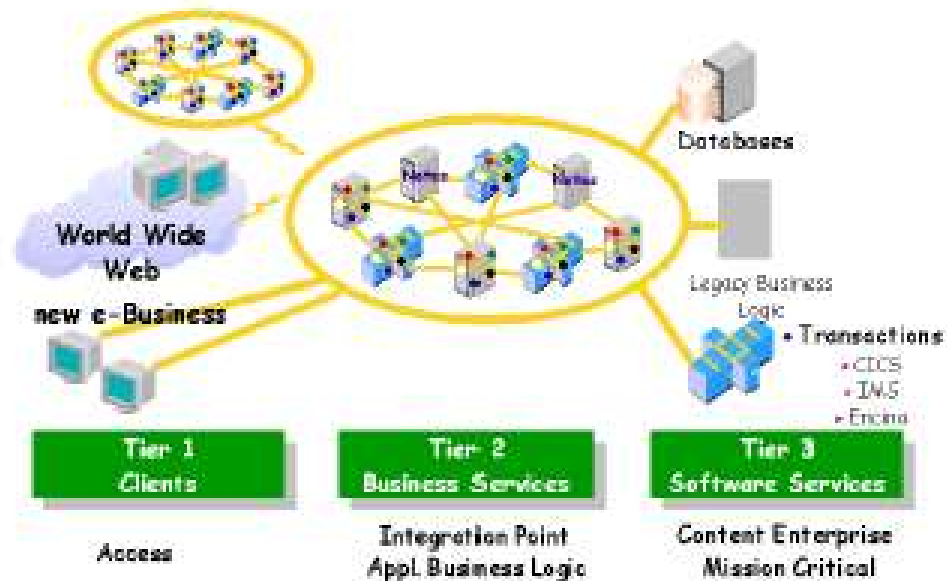
Aplicaciones Web

- ◆ Este modelo nacido para el intercambio de documentos académicos fue rápidamente adoptado para el desarrollo de los mas diversos programas
- ◆ Hoy se usa en aplicaciones comerciales, de información, de colaboración, etc.

Modelo de Tres Capas

- ◆ A partir de la adopción del modelo web, surge la necesidad de separar contextos
- ◆ Este modelo distingue tres constituyentes de la aplicación:
 - Capa de Presentación
 - Capa de Lógica
 - Capa de Datos

Modelo de Tres Capas



Distribución de Funcionalidad

	Cliente	Lógica	Datos
Protocolo	HTTP	Middleware	Propietario
Aplicación	Navegador	Web server, appl server	Base de datos
Dispositivo	PC, PDA, celular	Servidor	Servidor

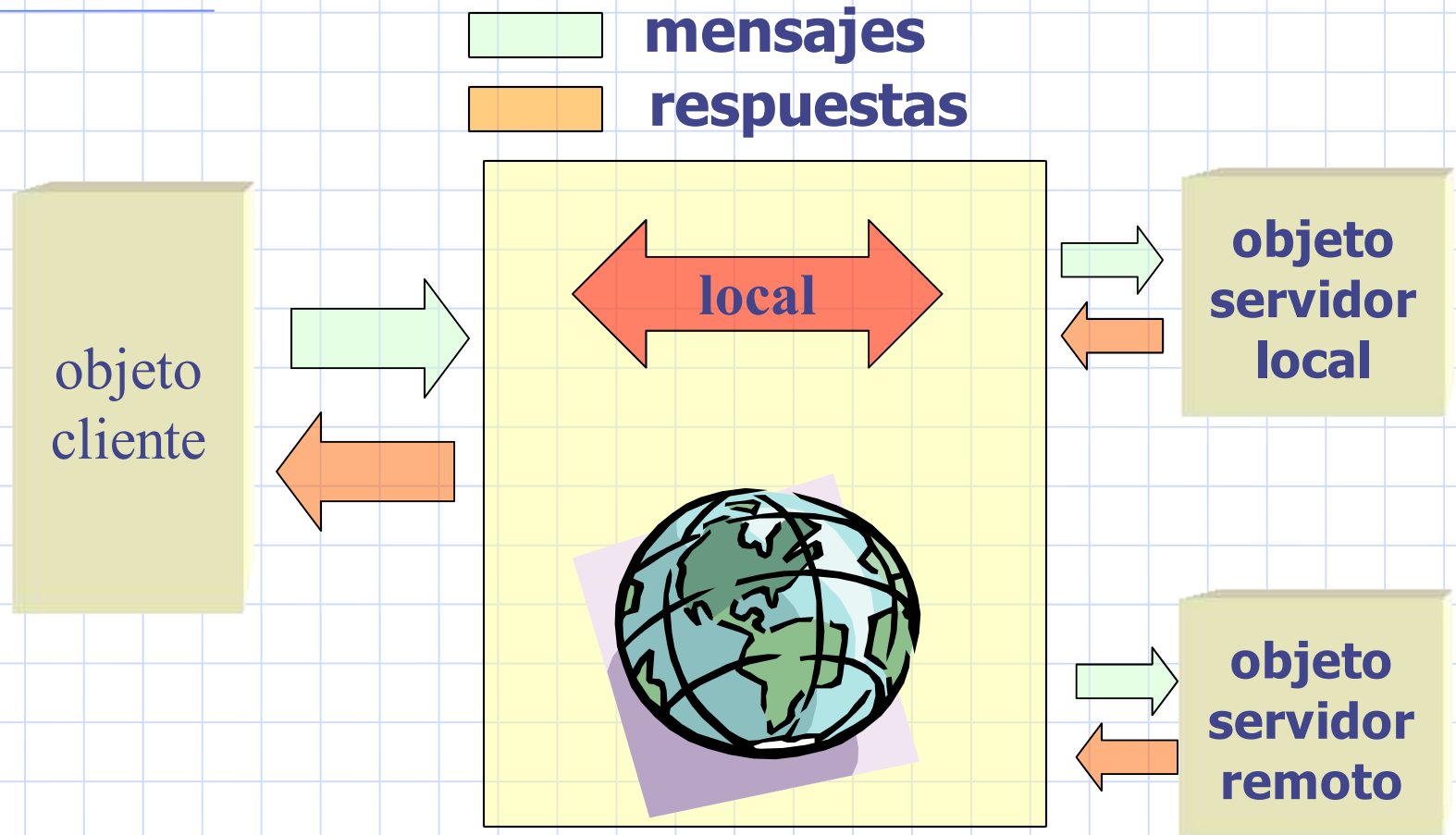
Arquitecturas Distribuidas

- ◆ Modelo de objetos remotos
- ◆ Middleware
- ◆ CORBA
- ◆ Java RMI
- ◆ DCOM
- ◆ Servicios Web

Modelo de Objetos Remotos

- ◆ Es la generalización del modelo tradicional de objetos para el diseño de software
- ◆ El paso de mensajes (interacción) entre objetos sigue siendo el componente central de la arquitectura
- ◆ Los objetos comunicantes pueden residir en distintas computadoras

Modelo de Objetos Remotos



Middleware

- ◆ Es el conjunto de servicios que permiten la construcción, despliegue y funcionamiento de aplicaciones distribuidas
- ◆ Existen distintos frameworks que proveen estos servicios
- ◆ Incluyen de servicios de localización, mensajería y activación

CORBA

- ◆ Es un framework de middleware creado por el Object Management Group para la interoperabilidad de programas escritos para distintas plataformas (Unix, Windows, VMS, Mac, etc.) en distintos lenguajes (C, C++, Java, COBOL, etc.)

Java RMI

- ◆ Remote Method Invocation es el mecanismo provisto por la plataforma Java para la interacción entre objetos localizados en distintas computadoras

DCOM

- ◆ Distributed Component Object Model es la infraestructura de middleware implementada en las plataformas Windows de Microsoft Corporation

Servicios Web

- ◆ Son componentes distribuidos que brindan servicios específicos y delimitados
- ◆ Permiten “ensamblar” una aplicación compleja combinando un conjunto de SW en una intranet o en Internet

CORBA

- ◆ Desarrollado en los '90 por la OMG, Common Object Request Broker Architecture permite la invocación de métodos en objetos ubicados en cualquier punto de una red
- ◆ A través de una especificación de interfaces es neutral frente al lenguaje y la plataforma de implementación de los objetos

CORBA

- ◆ Los mensajes entre objetos son conducidos por un Object Request Broker, que oculta los detalles de implementación de la interface de alto nivel
- ◆ Un objeto puede así invocar los servicios de otro sin conocimiento estricto de su locación, plataforma o lenguaje de implementación

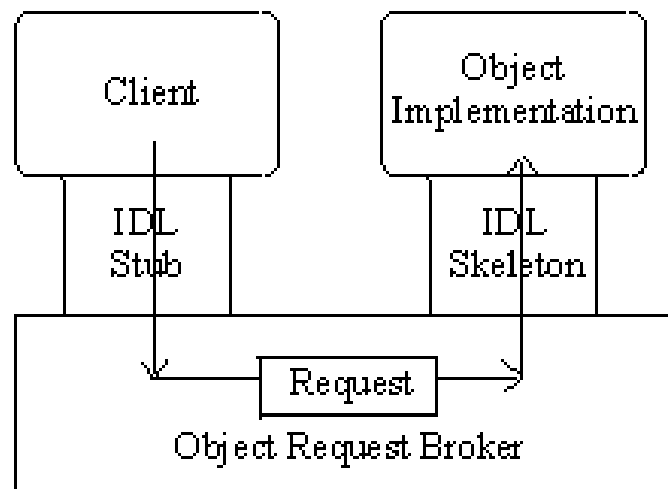


Figure 1: A request passing from client to object implementation

Copyright © 2000 Object Management Group

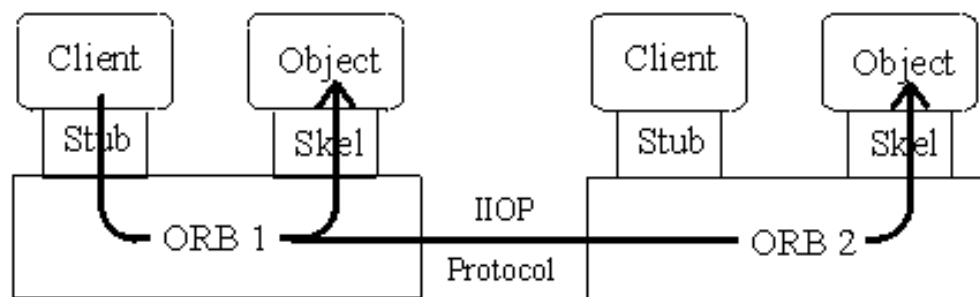
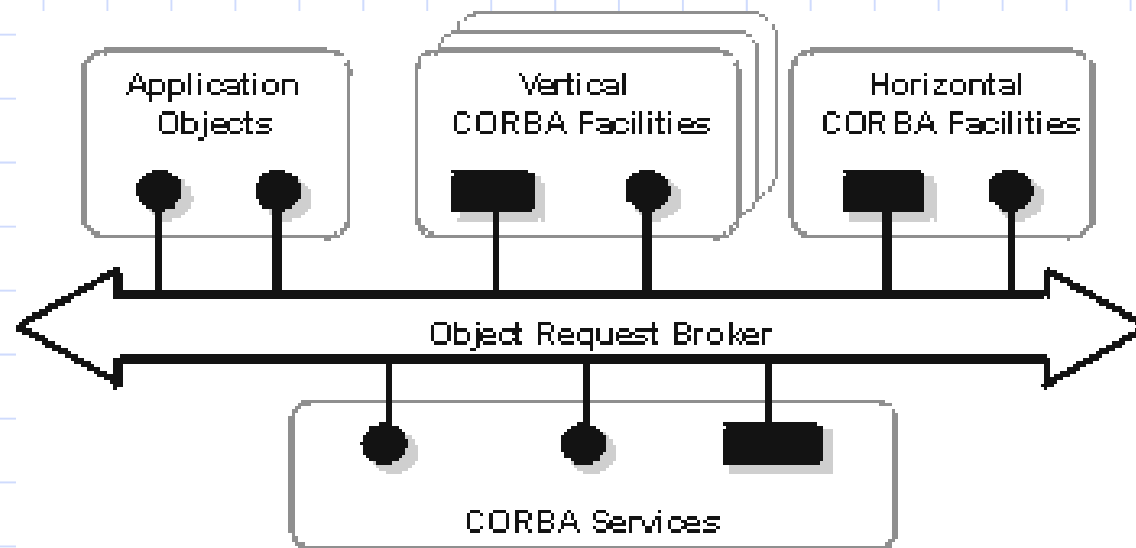
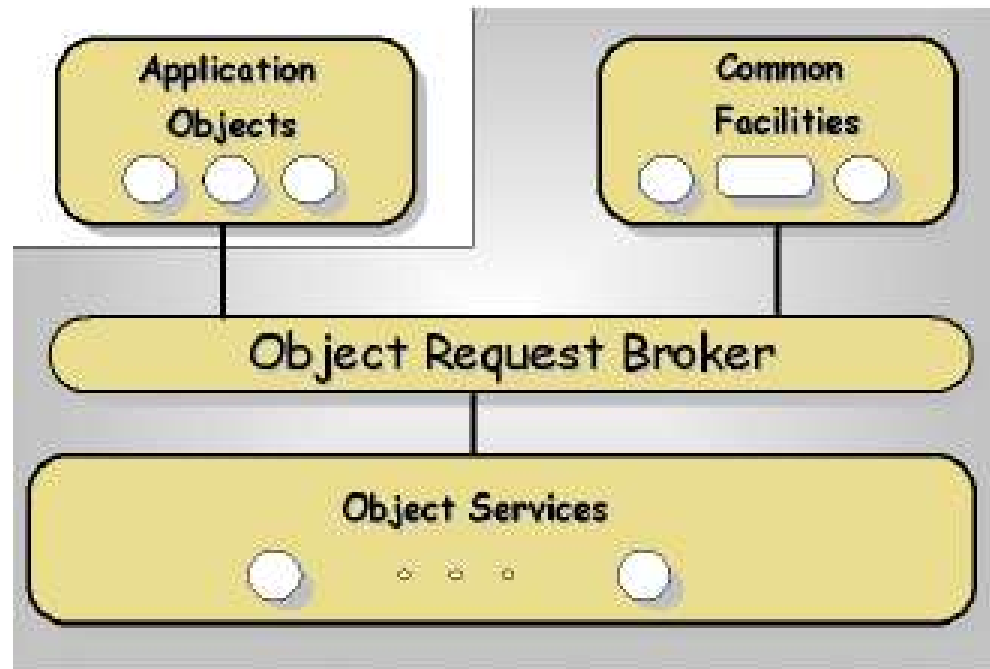


Figure 2: Interoperability uses ORB-to-ORB communication

Copyright © 2000 Object Management Group



CORBA: Arquitectura



Elementos Arquitecturales

- ◆ ORB: comunicación entre objetos
- ◆ Object Services: servicios agregados al ORB, tales como nombres, persistencia, seguridad y transacciones
- ◆ Common Facilities: servicios de aplicación, tales como componentes y documentos
- ◆ Application Objects: las aplicaciones

Object Request Broker

- ◆ Establece la relación cliente-servidor entre objetos: a partir de un requerimiento de un objeto cliente, el ORB localiza un objeto servidor adecuado, pasa los parámetros requeridos e invoca el método adecuado, retornando los resultados al cliente

Servicios de Objeto

- ◆ Ciclo de Vida: creación, copia, traslado y eliminación de objetos
- ◆ Persistencia: interface de almacenamiento persistente de objetos
- ◆ Nombres: permite la localización de objetos por su nombre, vinculado a un IOR (Interoperable Object Reference)
- ◆ Eventos: permite que un objeto registre su interés en conocer la ocurrencia de uno o mas eventos generados por otros objetos

Servicios de Objeto (cont.)

- ◆ Control de concurrencia: sincroniza el acceso de varios clientes a un recurso compartido
- ◆ Transacciones: brinda un servicio de operaciones atómicas ordenadas y reversibles
- ◆ Tiempo: provee una referencia temporal global única, permitiendo el ordenamiento de eventos
- ◆ Seguridad: provee autenticación, autorización, auditoría y herramientas de administración de la seguridad global

Facilidades Comunes

- ◆ También denominadas CORBAFacilities, proveen servicios a las aplicaciones centrados en el tipo y campo de aplicación de cada desarrollo.
- ◆ Incluyen (ente otros) agentes móviles, workflow, internacionalización, frameworks de objetos de negocio, etc.

Objetos Aplicación

- ◆ Encapsulan el comportamiento requerido para un determinado campo de aplicación: un sistema de control, un sitio de comercio electrónico o una aplicación de gestión corporativa