



Departamento de Lenguajes y  
Sistemas Informáticos



Universitat d'Alacant  
Universidad de Alicante

# Arquitectura cliente/servidor

Programación en Internet  
Curso 2004-2005

Programación en Internet – Curso 2004-2005

## Índice

- Introducción
- Tipos de servidores
- Ventajas
- Separación de funciones
- Modelos de distribución
- Tecnologías de programación distribuida
- Descripción de un sistema cliente/servidor

Programación en Internet – Curso 2004-2005

## Introducción

- Arquitectura de red
- Ordenador: cliente (solicita) o servidor (responde)
  - Se refiere a una transacción en particular (un ordenador puede adoptar los dos papeles, pero en transacciones distintas)
- Comunicación: protocolo
- Punto a punto (*peer-to-peer*) o red entre iguales
- Clientes: inteligentes (*rich*) o simples (*thin*)
- Aplicaciones distribuidas

Programación en Internet – Curso 2004-2005

## Introducción



Wyse Winterm 3200LE



IBM 3151 Terminal

Programación en Internet – Curso 2004-2005

## Tipos de servidores

- Servidor de ficheros
- Servidor de impresoras
- Servidor de red
- Servidor de bases de datos
- Servidor de aplicaciones
- Servidor de servicios de Internet (web, FTP, correo, etc.)

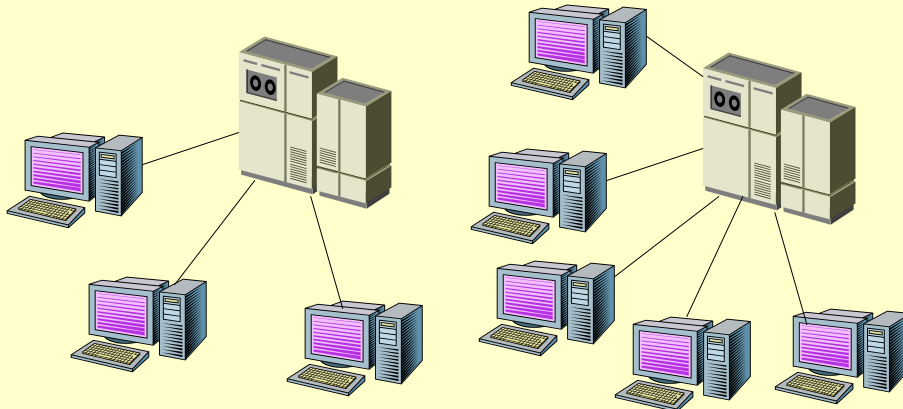
Programación en Internet – Curso 2004-2005

## Ventajas (I)

- Distribución de aplicaciones:  
conurrencia de procesos
- Portabilidad
- Acceso independiente del lugar físico:  
empleo de un cliente “ligero” con  
requisitos mínimos de instalación

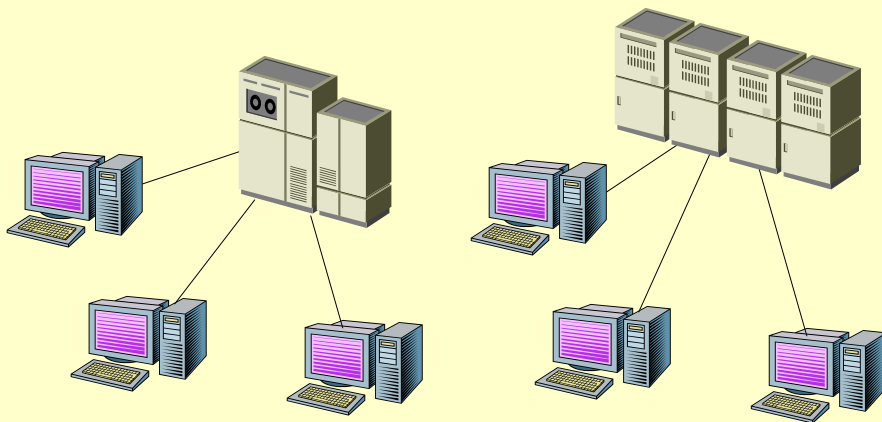
## Ventajas (II)

- Escalabilidad: horizontal y vertical



## Ventajas (y III)

- Escalabilidad: horizontal y vertical



## Arquitectura de una aplicación de n capas

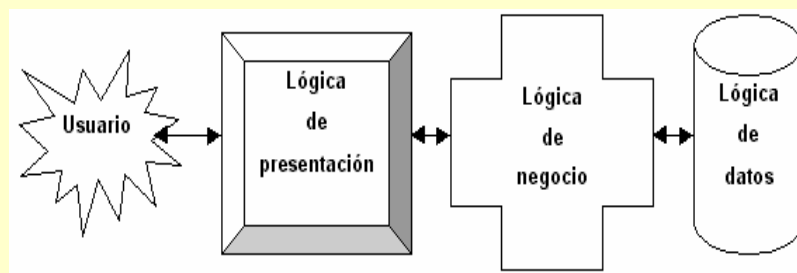
- N capas (*n-tiers*): separar los componentes de una aplicación en n niveles o capas lógicas
- Cualquier número de capas
- Lo más común: 1 (aplicación monolítica), 2 y 3
- No implica separación física en distintos ordenadores (aunque es lo normal): una aplicación de 3 capas puede existir en un único ordenador

## Ventajas

- Permite modificar una capa sin tener que modificar toda la aplicación
  - Aplicación 1 capa: existe un gran acoplamiento en la aplicación
- Simplifica la administración de los sistemas
- Disponibilidad inmediata de cambios en la lógica de aplicación o datos
- Posibilita el balanceo de la carga de trabajo entre distintos ordenadores

## Separación de funciones (I)

- Lógica de presentación
- Lógica de negocio (aplicación)
- Lógica de datos

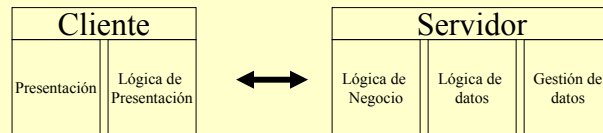


## Separación de funciones (y II)

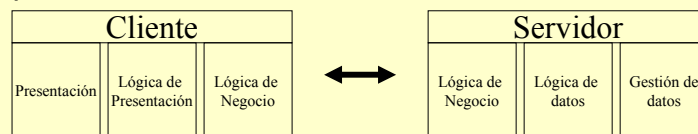
- Lógica de presentación:
  - Interfaz de usuario a través del cual interacciona con la aplicación
  - Interfaz gráfico (GUI) o basado en texto
- Lógica de negocio:
  - El núcleo de la aplicación
  - Implementa la funcionalidad de la aplicación
- Lógica de datos:
  - Gestiona la persistencia de los datos

## Modelos de distribución

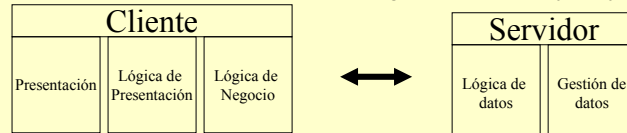
Presentación distribuida: solo una interfaz con validación de datos



Aplicación distribuida: máxima flexibilidad



Datos distribuidos: máxima carga en el cliente y mayor ancho de banda



## Dos y tres capas

- Sistema de dos niveles o capas:
  - Todos los casos anteriores
  - Un cliente y un servidor con la lógica de negocio y de datos
- Sistema de tres niveles o capas:
  - Un cliente con la lógica de presentación
  - Un servidor o servicio con la lógica de negocio
  - Un segundo servidor o servicio con la lógica de datos
  - Ventajas: mayor independencia, portabilidad y escalabilidad

Programación en Internet – Curso 2004-2005

## Tecnologías de programación distribuida (I)

- Diversas tecnologías que permiten a un programa que se ejecuta en un ordenador ejecutar código que reside en otro ordenador
- **RPC (*Remote Procedure Call*)**
  - Estándar de Internet (RFC 1831) basado en TCP/IP
  - Ejemplo de aplicación: NFS (*Network File System*), que permite acceder a sistemas de ficheros situados en otros ordenadores de forma transparente
- **RMI (*Remote Method Invocation*)**
  - Conjunto de protocolos desarrollados por Sun Microsystems
  - Exclusivo para Java
  - Permite que un objeto desarrollado con Java se comunique remotamente con otro objeto basado en Java

Programación en Internet – Curso 2004-2005

## Tecnologías de programación distribuida (y II)

- **CORBA (*Common Object Request Broker Architecture*)**
  - Arquitectura desarrollada por OMG (*Object Management Group*)
  - Comunicación remota entre objetos independientemente del lenguaje de programación con el que han sido desarrollados o del sistema operativo en el que se ejecutan
- **DCOM (*Distributed Component Object Model*)**
  - Extensión de COM desarrollada por Microsoft
  - Permite a los componentes COM comunicarse de forma remota
  - Basado en RPC
  - Permite la comunicación entre objetos desarrollados con distintos lenguajes de programación, pero su empleo se encuentra limitado a Microsoft Windows



## Descripción de un sistema

- Parte cliente (*front-end*) y servidor (*back-end*)
- Diferentes requisitos de máquina: velocidad de procesamiento, capacidad de almacenamiento de datos, etc.
- Entorno heterogéneo y multivendedor: la plataforma no tiene por qué ser la misma (se comunican a través de un API conocido)
- Cliente: interfaz gráfica de usuario, entorno amigable