



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Arquitectura cliente/servidor

Programación en Internet
Curso 2007-2008

Programación en Internet – Curso 2007-2008

Índice

- Introducción
- Tipos de servidores
- Ventajas
- Desventajas
- Arquitectura de una aplicación de n-capas
- Separación de funciones
- Modelos de distribución
- Tecnologías de programación distribuida

Programación en Internet – Curso 2007-2008

Introducción

- Arquitectura de desarrollo de aplicaciones
- 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 (o pares)
 - Cada miembro de la red tiene responsabilidades similares

Programación en Internet – Curso 2007-2008

Introducción

- Se puede aplicar a programas que se ejecutan en un mismo ordenador
- Es más ventajoso para aplicaciones distribuidas a través de una red de ordenadores
- Diferentes requisitos para el cliente y el servidor: velocidad de procesamiento, memoria, capacidad de almacenamiento de datos, etc.

Programación en Internet – Curso 2007-2008

Introducción

- Clientes: inteligentes (*rich, fat o thick client*) o simples (*thin o lean client*)
 - El software de la parte cliente se conoce como *front-end*
 - Inicia la petición
 - Espera y recibe la respuesta del servidor
 - Se puede conectar a un grupo de servidores de forma simultánea
 - Ofrece un interfaz gráfico (entorno amigable) al usuario final

Programación en Internet – Curso 2007-2008

Introducción



Wyse Winterm 3200LE



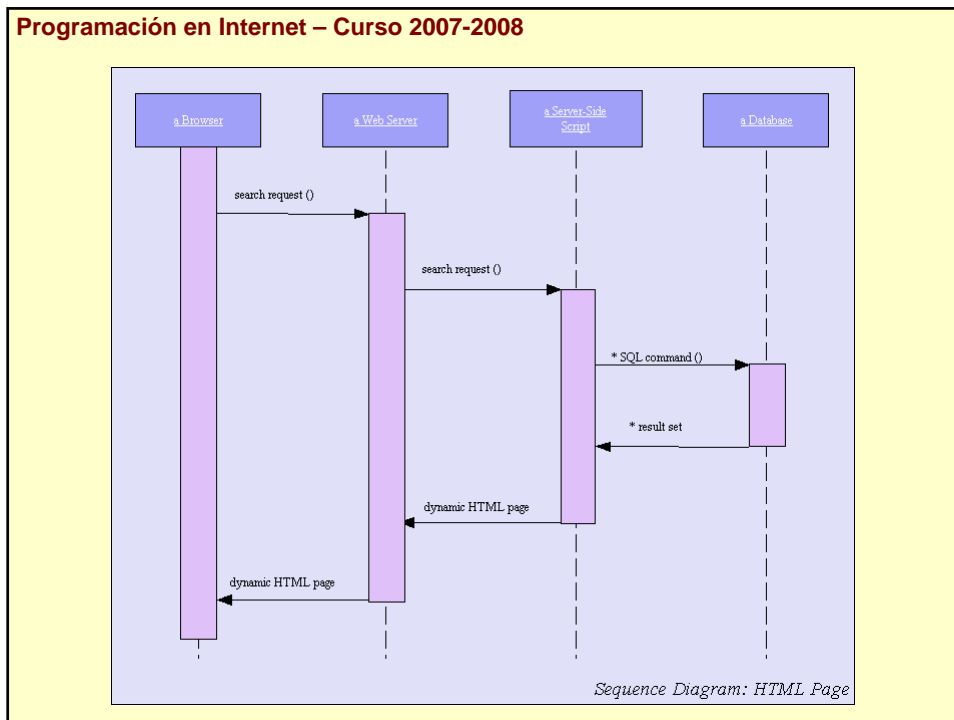
IBM 3151 Terminal

Introducción

- Servidor
 - El software de la parte servidor se conoce como *back-end*
 - Pasivo
 - Espera las peticiones de los clientes
 - Cuando recibe una petición, la procesa y devuelve una respuesta
 - Normalmente acepta un gran número de conexiones simultáneas de los clientes
 - No interacciona directamente con los usuarios finales

Introducción

- La interacción entre el cliente y el servidor se suele modelar mediante diagramas de secuencia de UML



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 2007-2008

Ventajas (I)

- Distribución de aplicaciones: concurrencia de procesos
- Permite emplear entornos heterogéneos y multivendedor: la plataforma no tiene por qué ser la misma (se comunican a través de un API conocido)
- Portabilidad
- Acceso independiente del lugar físico: empleo de un cliente “ligero” con requisitos mínimos de instalación

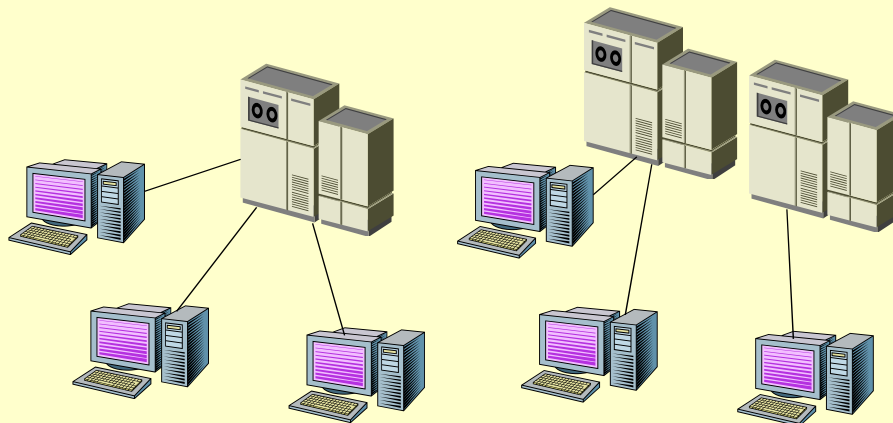
Programación en Internet – Curso 2007-2008

Ventajas (II)

- Escalabilidad:
 - La capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes
 - Un sistema es escalable si al incrementar los recursos empleados (por ejemplo, nº procesadores), el rendimiento crece proporcionalmente

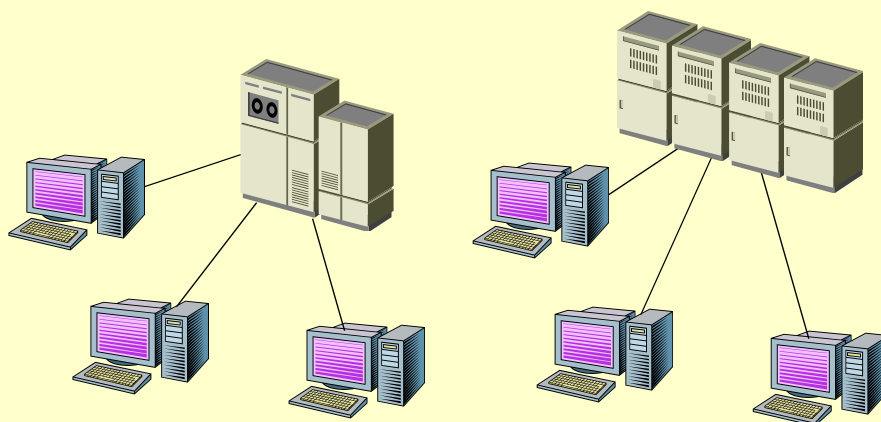
Ventajas (III)

- Escalabilidad horizontal (*scale out*)



Ventajas (y IV)

- Escalabilidad vertical (*scale up*)



Desventajas

- Aumenta la comunicación (congestión tráfico de red)
- Falta de robustez (caída del servidor)

Arquitectura de una aplicación de n capas

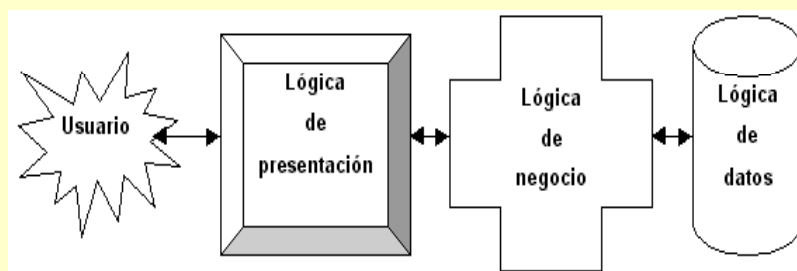
- N capas (*n-tiers*): separar los componentes de una aplicación en n niveles o capas lógicas
- Cualquier número de capas
- Lo más común: 1 (aplicación monolítica), 2 y 3
- No implica separación física en distintos ordenadores (aunque es lo normal): una aplicación de 3 capas puede existir en un único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

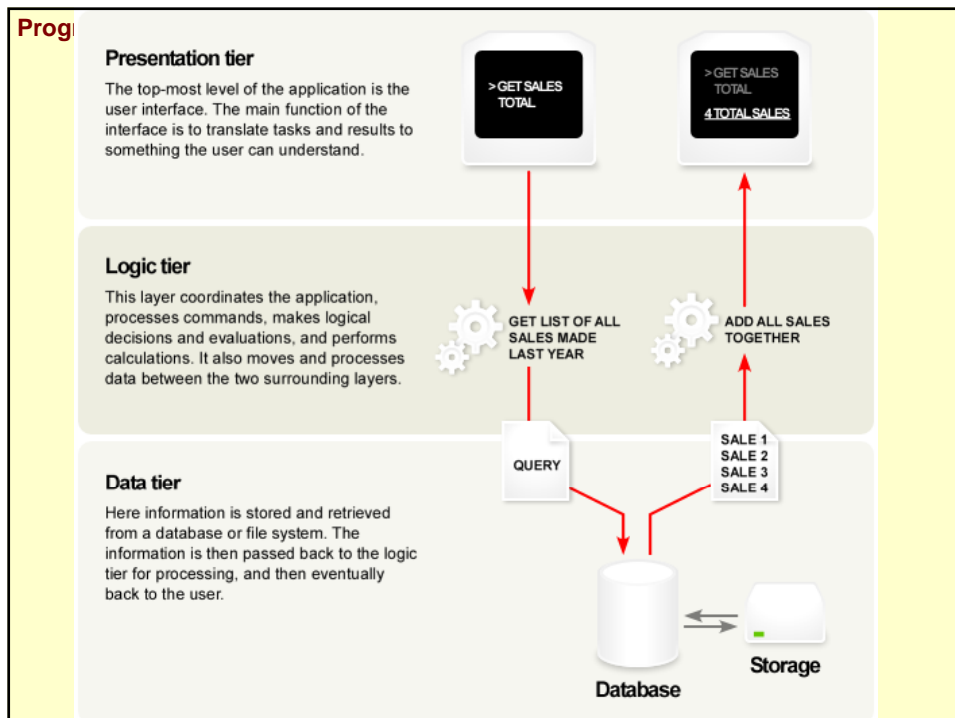
Separación de funciones (I)

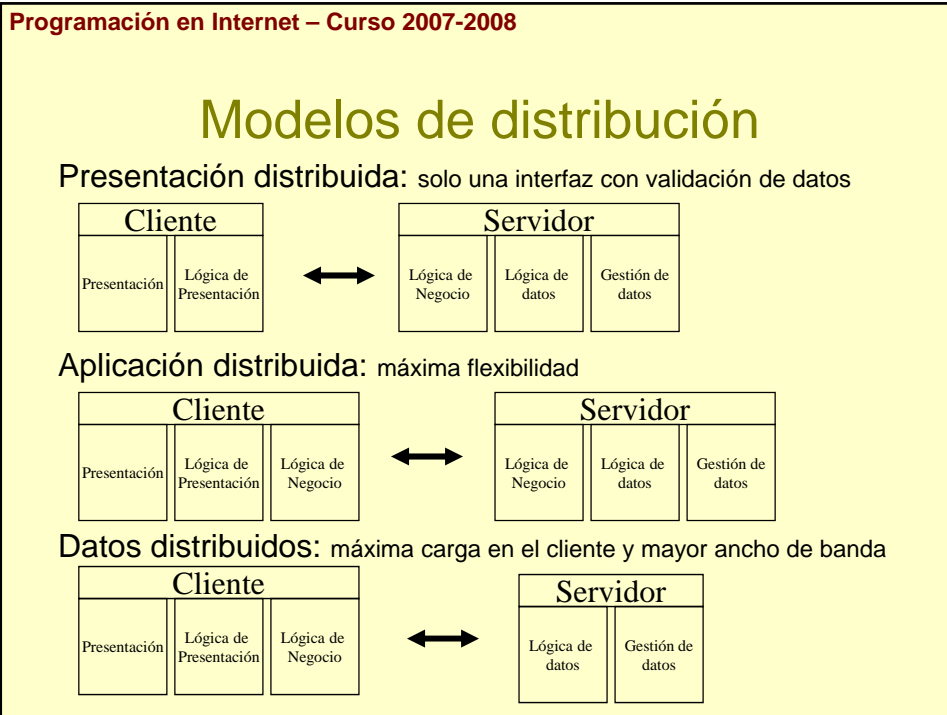
- Lógica de presentación
- Lógica de negocio (aplicación)
- Lógica de datos



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
 - Asegura su persistencia, integridad y disponibilidad





- Programación en Internet – Curso 2007-2008**
- ## 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 2007-2008

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 2007-2008

Tecnologías de programación distribuida (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

Programación en Internet – Curso 2007-2008

Tecnologías de programación distribuida (y III)

- **Servicios web**
 - Desarrollado por OASIS y W3C
 - La interoperabilidad se consigue mediante la adopción de estándares abiertos
 - Basado en XML: SOAP, WSDL y UDDI