

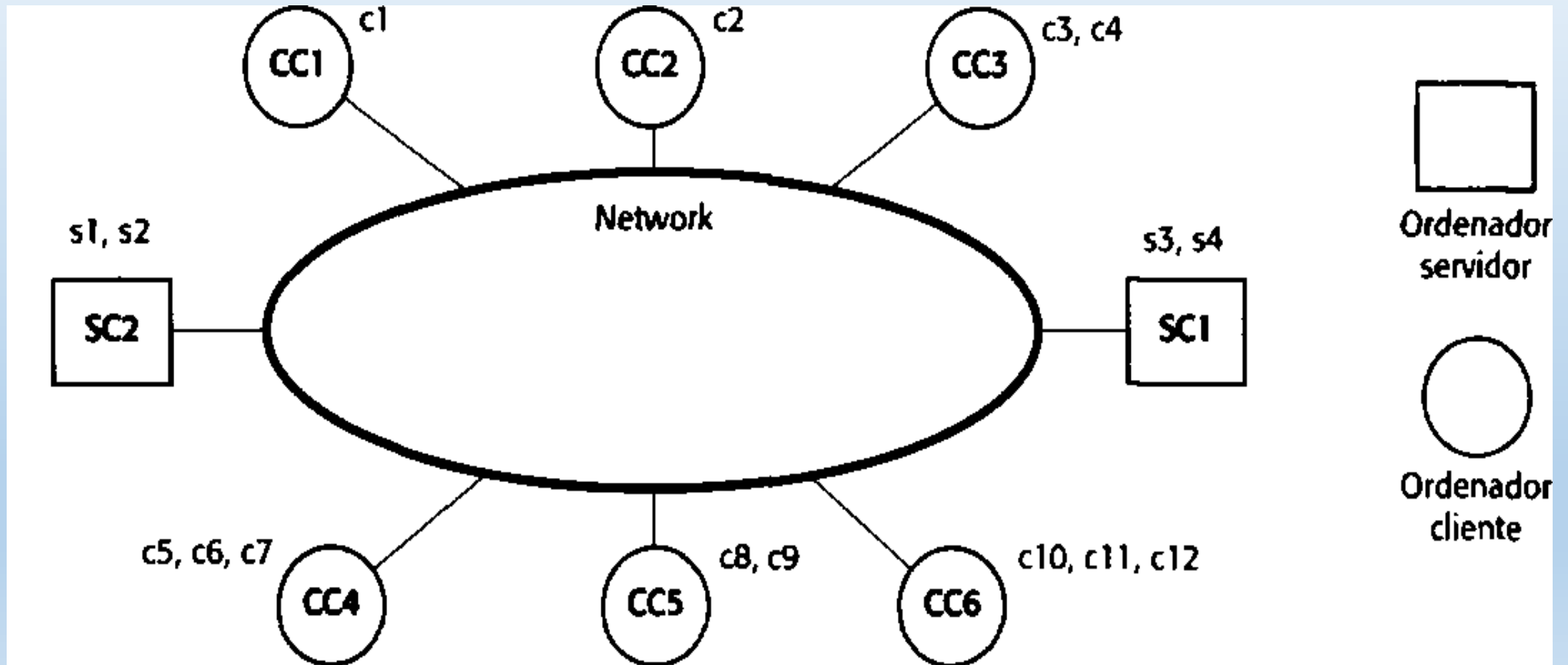
# ARQUITECTURAS CLIENTE – SERVIDOR

En una arquitectura cliente-servidor, una aplicación se modela como un conjunto de servicios proporcionados por los servidores y un conjunto de clientes que usan estos servicios.

- \* Los clientes necesitan conocer qué servidores están disponibles, pero normalmente no conocen la existencia de otros clientes.
- \* Clientes y servidores son procesos diferentes.
- \* Varios procesos servidores pueden ejecutarse sobre un único procesador servidor; por lo tanto, no hay necesariamente una correspondencia 1:1 entre procesos y procesadores en el sistema.

# ARQUITECTURAS CLIENTE-SERVIDOR

La figura muestra la arquitectura física de un sistema con seis computadores cliente y dos computadores servidor



# ARQUITECTURAS CLIENTE-SERVIDOR

Esa arquitectura FISICA puede ejecutar los procesos cliente y servidor que se muestran en la figura siguiente.

- Acá, Cuando hablamos de *clientes* y *servidores*, nos referimos a los **procesos lógicos**, en vez de a las **computadoras físicas** sobre las que se ejecutan, como era el caso anterior.

# ARQUITECTURAS CLIENTE-SERVIDOR

El diseño de sistemas cliente-servidor debería reflejar la estructura lógica de la aplicación que se está desarrollando.

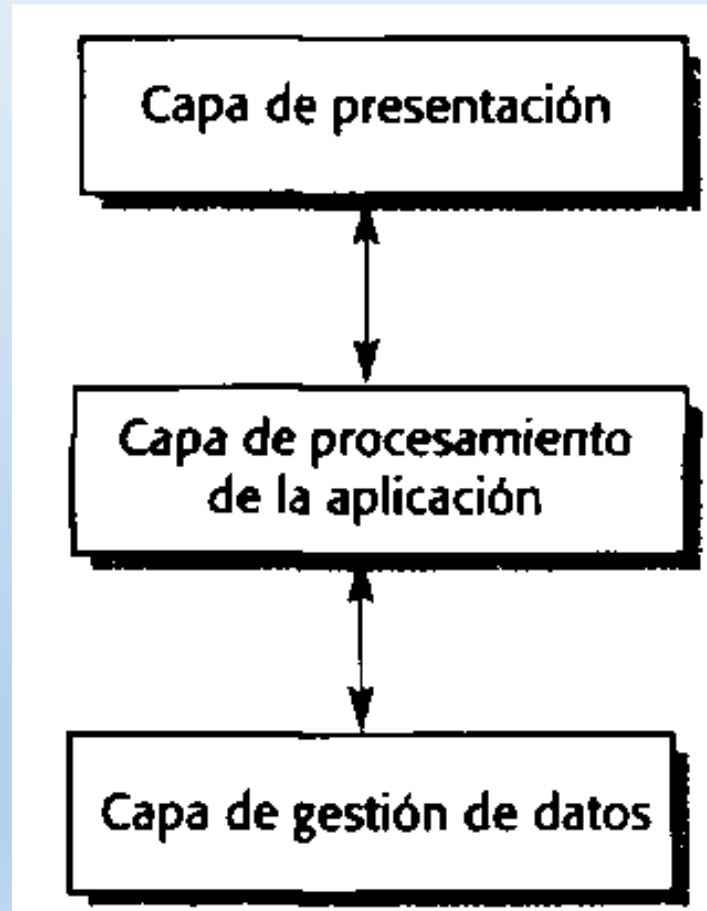
- \* Una forma de ver una aplicación se ilustra en la Figura siguiente, que muestra una aplicación estructurada en tres capas.

- \* La capa de presentación está relacionada con la presentación de la información al usuario y con toda la interacción con él.

La capa de procesamiento de la aplicación está relacionada con la implementación de la lógica de la aplicación.

- \* La capa de gestión de datos está relacionada con todas las operaciones sobre la base de datos.

# CAPAS DE LAS APLICACIONES CS



# ARQUITECTURAS CLIENTE - SERVIDOR

La arquitectura cliente-servidor más simple se denomina arquitectura cliente-servidor de dos capas.

- \* Se organiza como un servidor (o múltiples servidores idénticos) y un conjunto de clientes.

- \* Como se ilustra en la Figura siguiente, las arquitecturas cliente/servidor de dos capas pueden ser de dos tipos:

1. ***Modelo de cliente ligero (thin-client).*** Acá todo el procesamiento de las aplicaciones y la gestión de los datos se lleva a cabo en el servidor.

- \* El cliente simplemente es responsable de la capa de presentación del software.

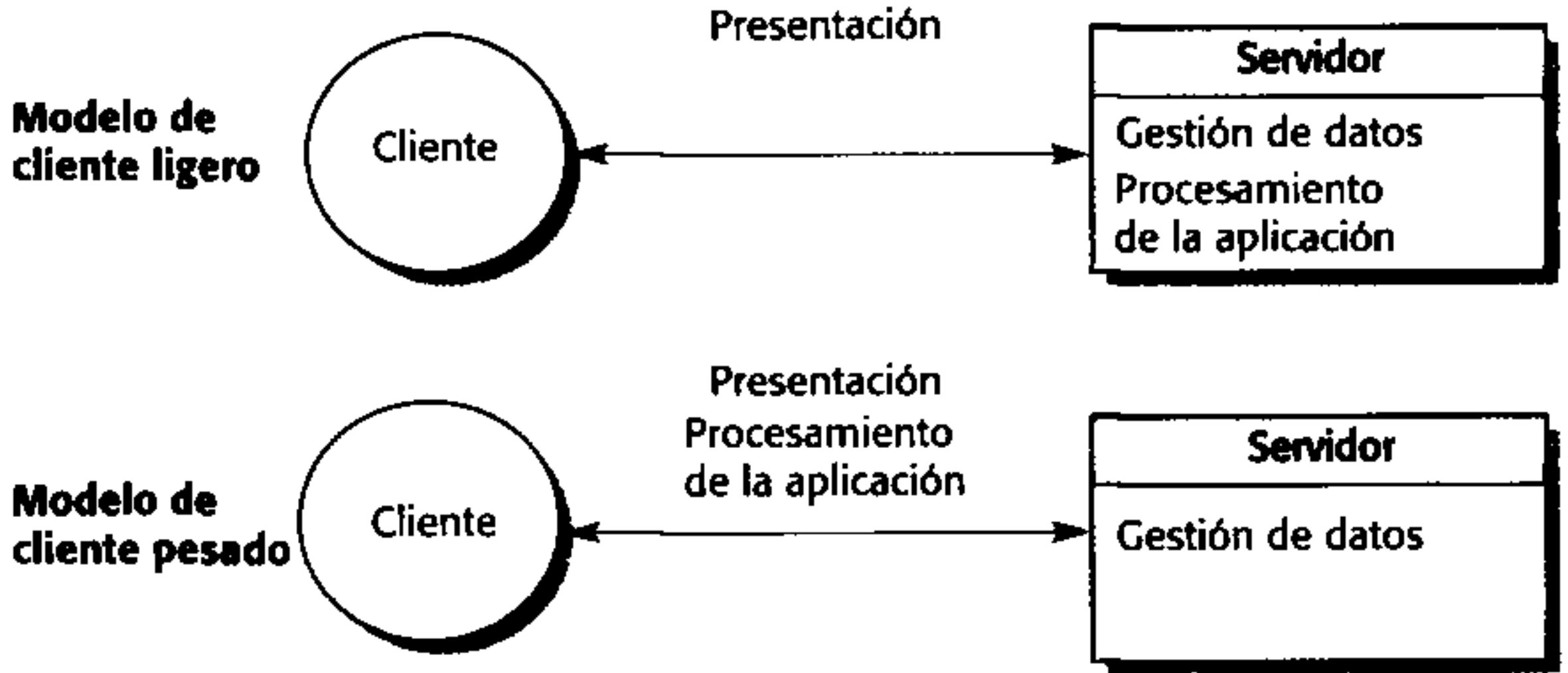
# ARQUITECTURAS CLIENTE – SERVIDOR

2. *Modelo de cliente rico (fat-client)*. En este modelo, el servidor solamente es responsable de la gestión de los datos.

\*El software del cliente implementa la lógica de la aplicación y las interacciones con el usuario del sistema.

# ARQUITECTURAS CLIENTE – SERVIDOR

## Clientes Livianos y Pesados





# MODELO CLIENTE LIGERO/POBRE/DELGADO

Una arquitectura de dos capas con clientes ligeros es la más simple que se utiliza cuando los sistemas heredados centralizados, evolucionan a una arquitectura cliente-servidor.

- \* La interfaz de usuario para estos sistemas se migra a PC's, y la aplicación en sí misma actúa como un servidor y maneja todo el procesamiento de la aplicación y gestión de datos.

- \* Un modelo de cliente ligero también puede implementarse cuando los clientes son dispositivos de red sencillos en lugar de PC's o estaciones de trabajo.

- \* El dispositivo de red ejecuta un navegador de Internet y la interfaz de usuario es implementada a través de ese sistema.

# MODELO CLIENTE LIGERO/POBRE/DELGADO

Una gran **desventaja** del modelo de **cliente ligero** es que ubica una **elevada carga de procesamiento**, tanto en el **servidor** como en la **red**.

\* El servidor es **responsable de todos los cálculos** y esto puede implicar la **generación de un tráfico significativo en la red** entre el cliente y el servidor.

\* Los dispositivos de computación modernos disponen de una gran cantidad de potencia de procesamiento, que es desperdiciada en la aproximación de cliente ligero, liviano o pobre.

# MODELO CLIENTE RICO/GORDO/PESADO

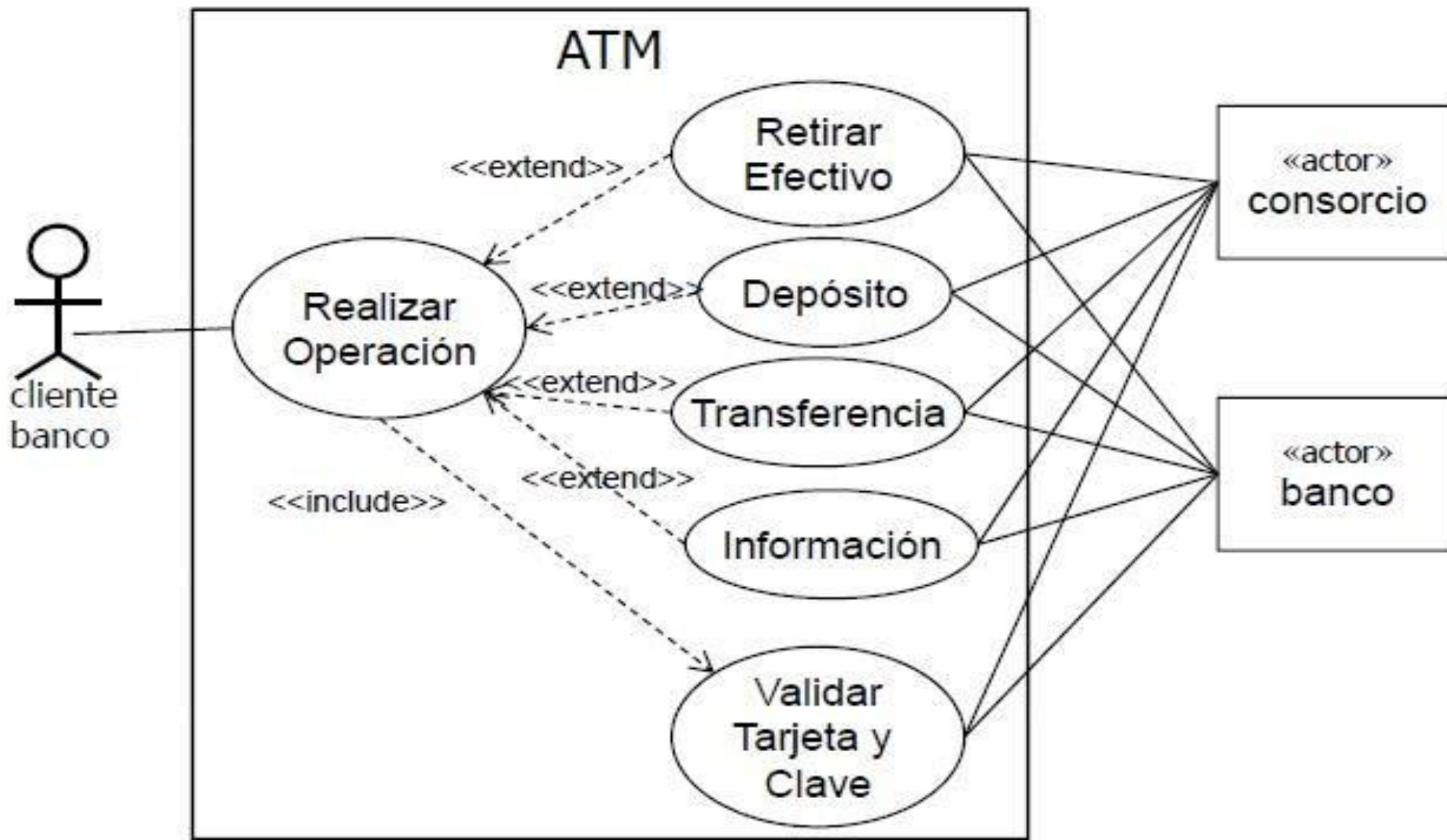
El modelo de cliente rico, pesado o gordo hace uso de esta potencia de procesamiento disponible y distribuye, tanto el procesamiento de la lógica de la aplicación, como la presentación al cliente.

- \* El servidor es esencialmente de transacciones.
- \* Gestiona todas las transacciones de la base de datos.
- \* Ejemplo: arquitectura de los sistemas bancarios ATM (cajeros automáticos), en donde cada Cajero es un cliente y el servidor es un ***mainframe*** que procesa la cuenta del cliente en la base de datos.

# MODELO CLIENTE RICO/GORDO/PESADO

**ATM** son las siglas en inglés que corresponden a cajero automático. Es una computadora especializada que le permite administrar su dinero de manera conveniente. ... Habitualmente puede acceder a la mayoría de los servicios en un **ATM** operado por su propio banco.

# MODELO CLIENTE RICO/GORDO/PESADO



# MODELO CLIENTE RICO/GORDO/PESADO

El **hardware** de los **cajeros automáticos** realiza una gran cantidad de **procesamiento relacionado con el cliente** y asociado a la transacción.

- \* Los **cajeros automáticos** no se conectan directamente con la base de datos de clientes sino con un **monitor de teleproceso**.

- \* Un **monitor de teleproceso** o **gestor de transacciones** es un **sistema middleware** que organiza las **comunicaciones** con los **clientes remotos** y **serializa las transacciones** de los clientes para su procesamiento en la base de datos.

- \* El uso de **transacciones serializadas** significa que el **sistema puede recuperarse de los defectos sin corromper los datos** del sistema.

# MODELO CLIENTE RICO/GORDO/PESADO

Aunque el modelo de cliente rico distribuye el procesamiento de forma más efectiva que un modelo de cliente ligero, la gestión del sistema es más compleja.

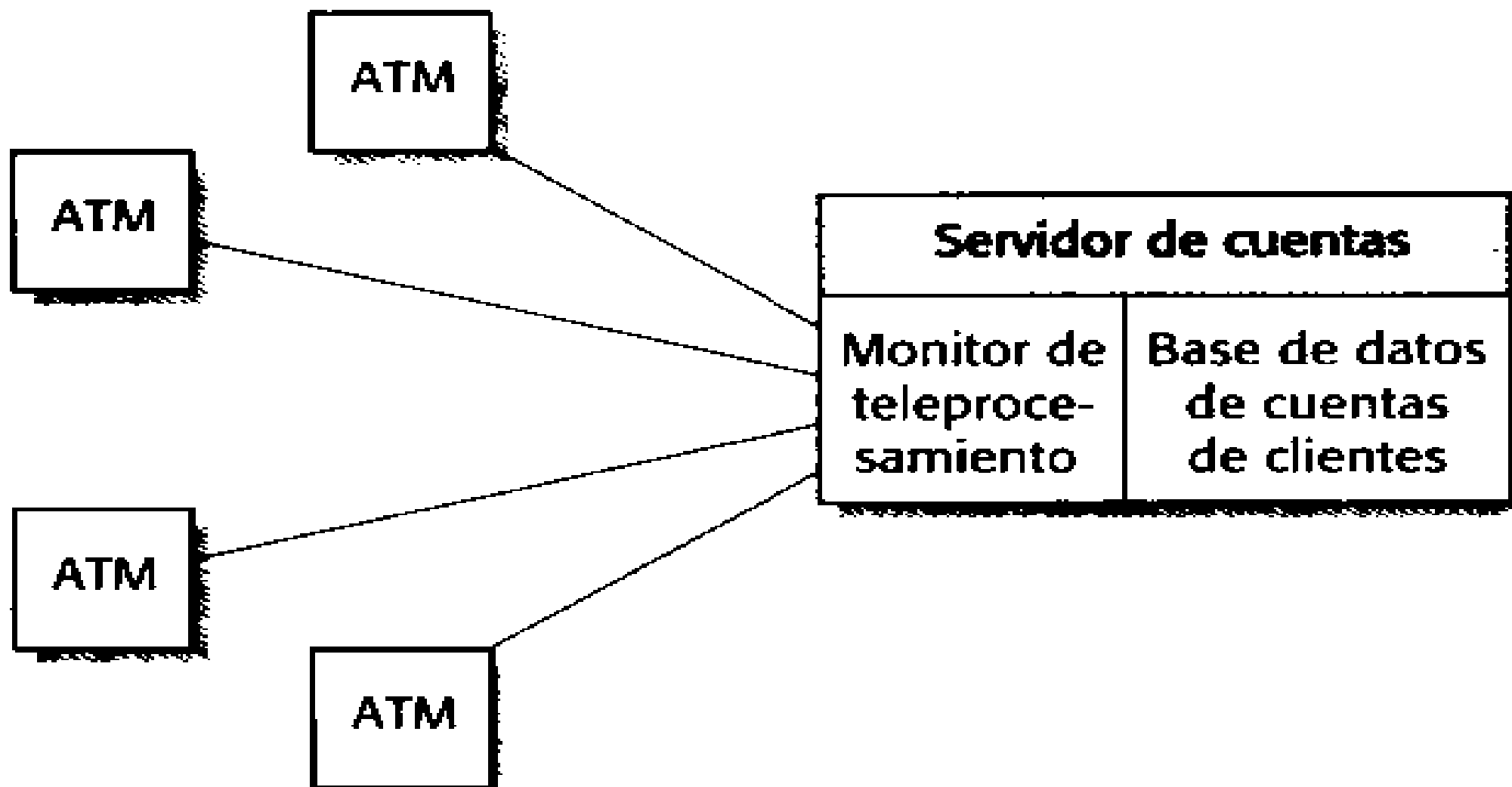
- \* La funcionalidad de la aplicación se expande entre varias computadoras.

- \* Cuando la aplicación software tiene que ser modificada, esto implica la reinstalación en cada computadora cliente.

- \* Esto puede significar un costo importante si hay cientos de clientes en el sistema.

# MODELO CLIENTE RICO/GORDO/PESADO

## Sistema Cliente – Servidor de Cajeros Automáticos (ATM)





# MODELO CLIENTE

## SERVIDOR DE 2 CAPAS

El problema con una aproximación cliente-servidor de dos capas es que las tres capas lógicas presentación procesamiento de la aplicación y gestión de los datos deben asociarse con dos computadoras: el **cliente** y el **servidor**.

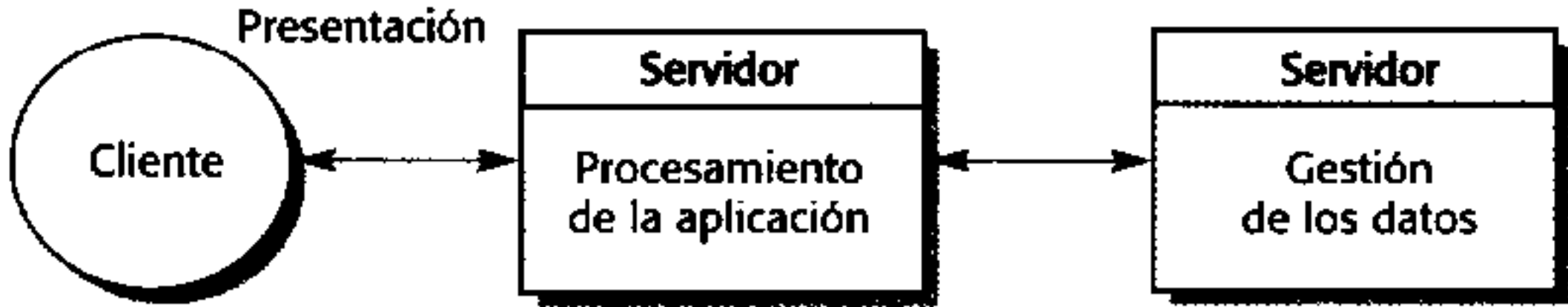
\* Aquí puede haber problemas con la escalabilidad y rendimiento si se elige el modelo de cliente ligero, o problemas con la gestión del sistema si se usa el modelo de cliente rico.

# MODELO CLIENTE

## SERVIDOR DE 3 CAPAS

Para evitar estos problemas, una aproximación alternativa es usar una arquitectura ***cliente-servidor de tres capas***.

\* Aquí, la presentación, el procesamiento de la aplicación y la gestión de los datos son procesos lógicamente separados que se ejecutan sobre procesadores diferentes.



# MODELO CLIENTE

## SERVIDOR DE 3 CAPAS

Un sistema bancario por Internet es un ejemplo de una arquitectura cliente-servidor de tres capas.

- \* La base de datos de clientes del banco (usualmente ubicada sobre una computadora mainframe) proporciona servicios de gestión de datos; un servidor web proporciona los servicios de aplicación tales como facilidades para transferir efectivo, generar estados de cuenta, pagar facturas, y así sucesivamente.

- \* La propia computadora del usuario con un navegador de Internet es el cliente.

# MODELO CLIENTE

## SERVIDOR DE 3 CAPAS

El sistema es escalable, porque es relativamente fácil añadir nuevos servidores web, a medida que el número de clientes crece.

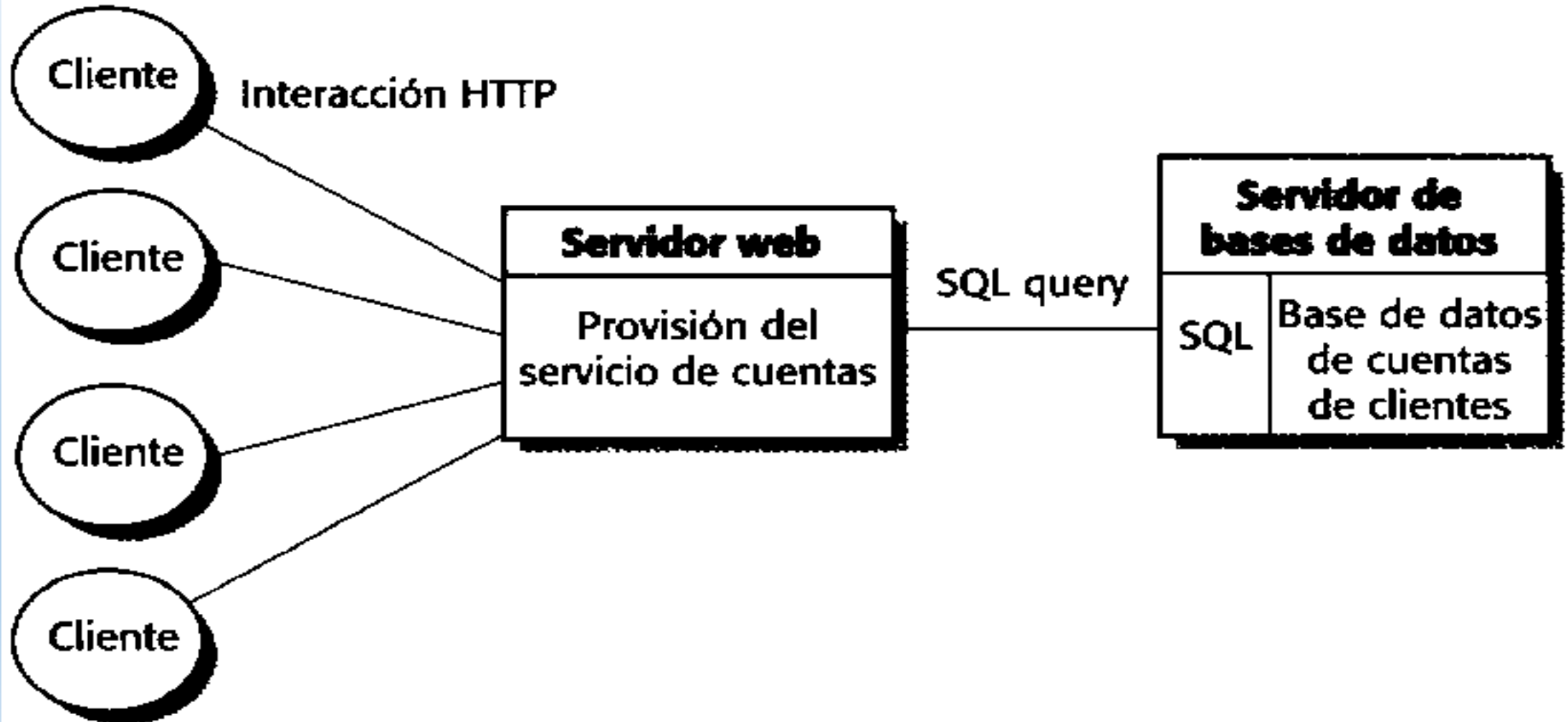
- \* El uso de una arquitectura de tres capas permite optimizar la transferencia de información entre el servidor web y el servidor de la base de datos.

- \* Las comunicaciones entre estos sistemas pueden usar protocolos de comunicación de bajo nivel muy rápidos.

- \* Para recuperar información de la base de datos se utiliza un middleware eficiente que soporte consultas a la base de datos en SQL (Structured Query Language).

# MODELO CLIENTE SERVIDOR DE 3 CAPAS

Figura de Modelo de Arquitectura Cliente – Servidor  
de 3 Capas (Sistema Bancario en Internet)



# Modelo Cliente - Servidor

## Resumen del uso de diferentes arquitecturas Cliente-Servidor

**Arquitectura C/S de dos capas con clientes ligeros**

Aplicaciones de sistemas heredados en donde no es práctico separar el procesamiento de la aplicación y la gestión de los datos.

Aplicaciones que requieren cálculos intensivos tales como compiladores con poca o ninguna gestión de los datos.

Aplicaciones que requieran manejar una gran cantidad de datos (navegar y consultar) con poco o ningún procesamiento de la aplicación.

**Arquitectura C/S de dos capas con clientes ricos**

Aplicaciones en donde el procesamiento de la aplicación se proporciona por software comercial (por ejemplo, Microsoft Excel) sobre el cliente.

Aplicaciones que requieren un procesamiento de datos computacionalmente intensivo (por ejemplo, visualización de datos).

Aplicaciones con una funcionalidad para el usuario final relativamente estable usada en un entorno de gestión del sistema bien establecido.

**Arquitectura C/S de tres capas o multicapa**

Aplicaciones a gran escala con cientos o miles de clientes.

Aplicaciones en donde tanto los datos como la aplicación son volátiles.

Aplicaciones en donde se integran datos de múltiples fuentes.



# SERVICIOS PROXY Y CACHE

Las cachés pueden estar en cada cliente o en un servidor proxy que puede compartirse desde varios clientes.

❑ Los navegadores mantienen una caché de las páginas visitadas recientemente, y de otros recursos web, en el sistema local de ficheros

del cliente; utilizan una petición HTTP para comprobar si dichas páginas han sido actualizadas.

❑ Los servidores proxy para el web proporcionan una caché compartida de recursos web a las máquinas cliente de uno o más sitios.