ARQUITECTURA DEL SOFTWARE

Tema 3

Ingeniería del Software

ETS Ingeniería Informática

DSIC – UPV

Objetivos

- Introducir el concepto de arquitectura del sistema
- Describir las principales características de la **arquitectura multicapa**.

Contenidos

1. Introducción
2. Concepto de “Arquitectura del Software”
    - Arquitectura Multicapa
        - Arquitectura de 3 capas
3. Bibliografía

INTRODUCCIÓN

Programming in the small/medium/large

"Programming in the small/medium/large"

- Cuando los sistemas software crecen en tamaño, se requiere una organización de los mismos en subsistemas que los hagan manejables

- A lo largo de la historia del desarrollo de software se han utilizado diferentes *estrategias para manejar la complejidad*, generalmente relacionadas con el diseño a diferentes niveles de abstracción
  - Módulos (Métodos estructurados)
  - Clases (Métodos orientados a objetos)
  - …

Problemas

- Estas aproximaciones son de bajo nivel, son meras agrupaciones de código.
- Se requiere un mecanismo más abstracto…. que separe la aplicación en bloques funcionales.

ARQUITECTURA DEL SOFTWARE

Arquitectura Multicapa

¿Qué se entiende por arquitectura de software?

*"La **arquitectura de software**, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad"*

Kruchten, Philippe

La arquitectura de software es importante

- En la etapa de descripción de la **Arquitectura del Sistema** debemos dotar al sistema de una organización global en **subsistemas.**

  - **Tipos de Sistemas**
    - **Arquitectura de Sistemas**

Tipos de Sistemas (entre otros muchos…)

- **Sistemas Distribuidos**:

  El sistema software en el que el procesamiento de información se distribuye sobre varias computadoras en vez de estar confinado en una única máquina.

- **Sistemas Personales**:

  No son distribuidos y están diseñados para ejecutarse en un ordenador personal o estación de trabajo.

- **Sistemas Empotrados**:

  Sistemas informáticos (hardware + software), usualmente de tiempo real, integrados en un sistema de ingeniería más general, en el que realizan funciones de control, procesamiento y/o monitorización.

Arquitecturas de Sistemas Distribuidos (entre otras)

- **Arquitecturas Multi-procesador:**

  El sistema consta de múltiples procesos que pueden (o no) ejecutarse en diferente procesadores.

- **Arquitecturas Cliente/Servidor**:

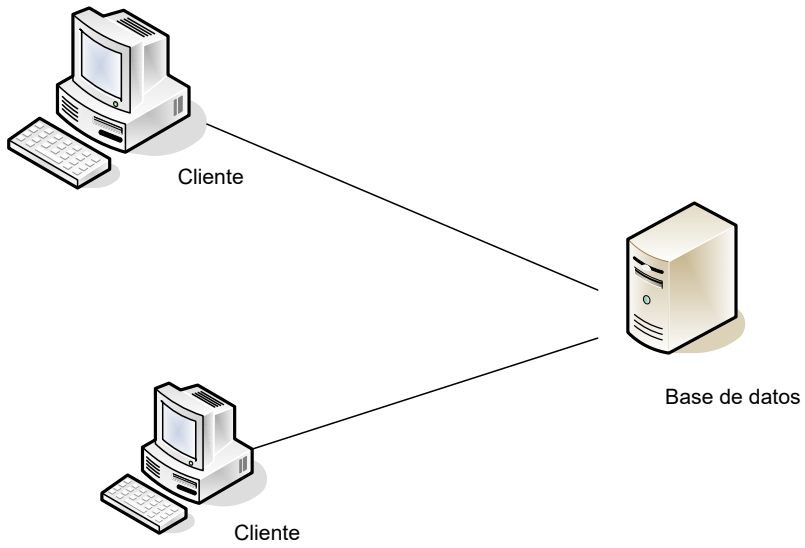
  El sistema puede ser visto como un conjunto de servicios que se proporcionan a los clientes por parte de los servidores. Los servidores y los clientes se tratan de forma diferente.

- **Arquitecturas de objetos distribuidos**:

  El sistema puede ser visto como un conjunto de objetos que interaccionan y cuya localización es irrelevante. No hay distinción entre un proveedor de servicios y el usuario de estos servicios.

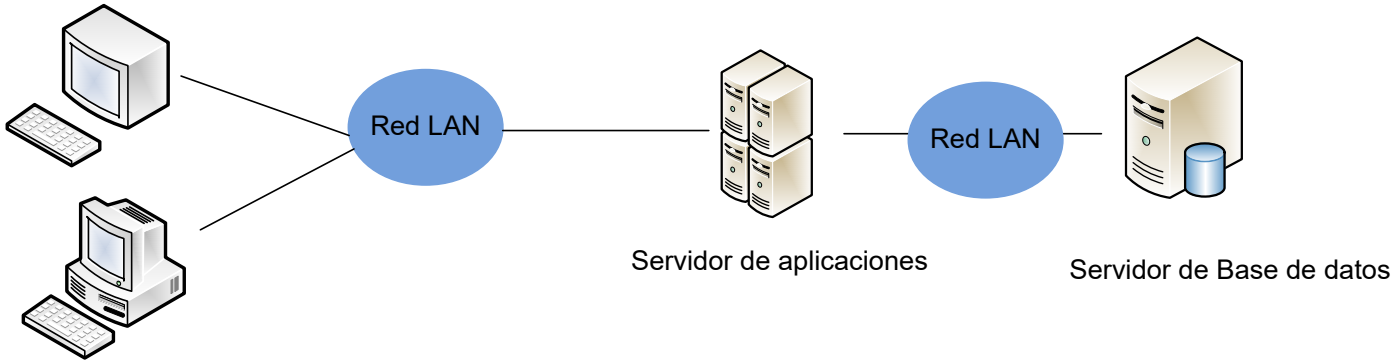
Arquitectura Cliente/Servidor

- C/S distribuye una aplicación en 2 componentes especializados cuya ejecución se lleva a cabo en 1 o más equipos:
  - El servidor (S) es un proveedor de servicios.
  - El cliente (C) es un consumidor de servicios.
- C y S Interactúan por un mecanismo de paso de mensajes:
  - Solicitud de servicio.
  - Respuesta.

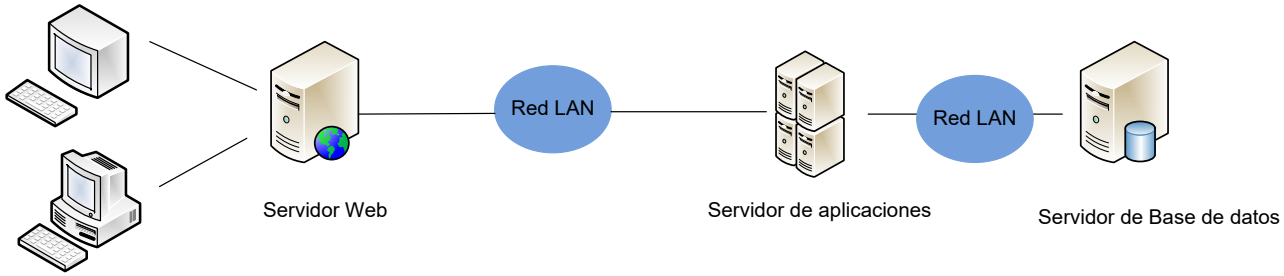


Ejemplo: *Cliente Servidor 3/N-niveles*

- Arquitectura 3-niveles

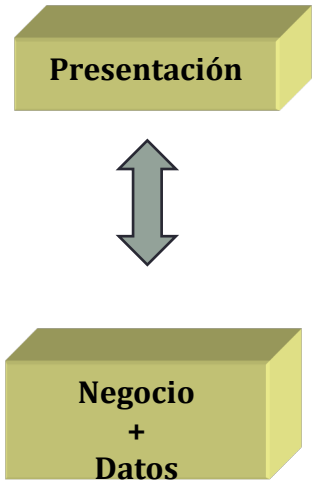


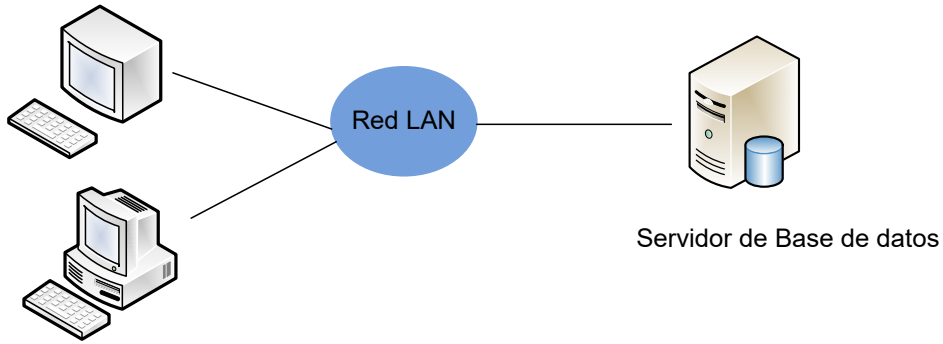
- Arquitectura de 4-niveles



Niveles *versus* Capas

- Capa (layer) hace referencia a una segmentación lógica de la solución, mientras que nivel (tier) se refiere a la segmentación o ubicación física.





** Arquitectura Multicapa **

Un **sistema por capas** es un conjunto ordenado de subsistemas, cada uno de los cuales está construido en términos de los que tiene por debajo, y proporciona la base de la implementación de aquellos que estén por encima de él.

- Los objetos de cada capa pueden ser **independientes** (recomendado) aunque suelen haber dependencias entre objetos de distintas capas.

- Existe una relación **cliente/servidor** entre las capas **inferiores** (que proporcionan servicios) y las capas **superiores** (que consumen dichos servicios).

Arquitectura Multicapa – Visibilidad de las capas

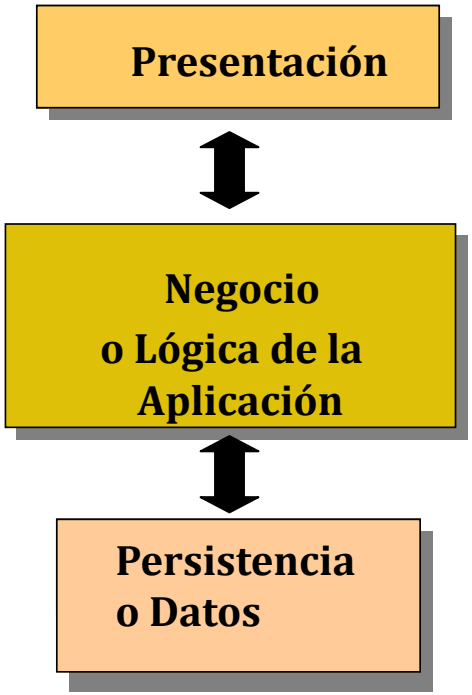
Las arquitecturas basadas en capas pueden ser **abiertas** o **cerradas** según la dependencia existente entre capas.

- **abiertas**: una capa puede utilizar características de cualquier capa a cualquier nivel.
- **cerradas**: una capa sólo utiliza características de su capa inmediatamente inferior.

Se recomienda trabajar con arquitecturas **cerradas,** ya que reducen las dependencias entre capas y permiten que los cambios se hagan con facilidad porque la interfaz de una de ellas sólo afecta a la capa siguiente.

Propuesta: Arquitectura de Tres Capas

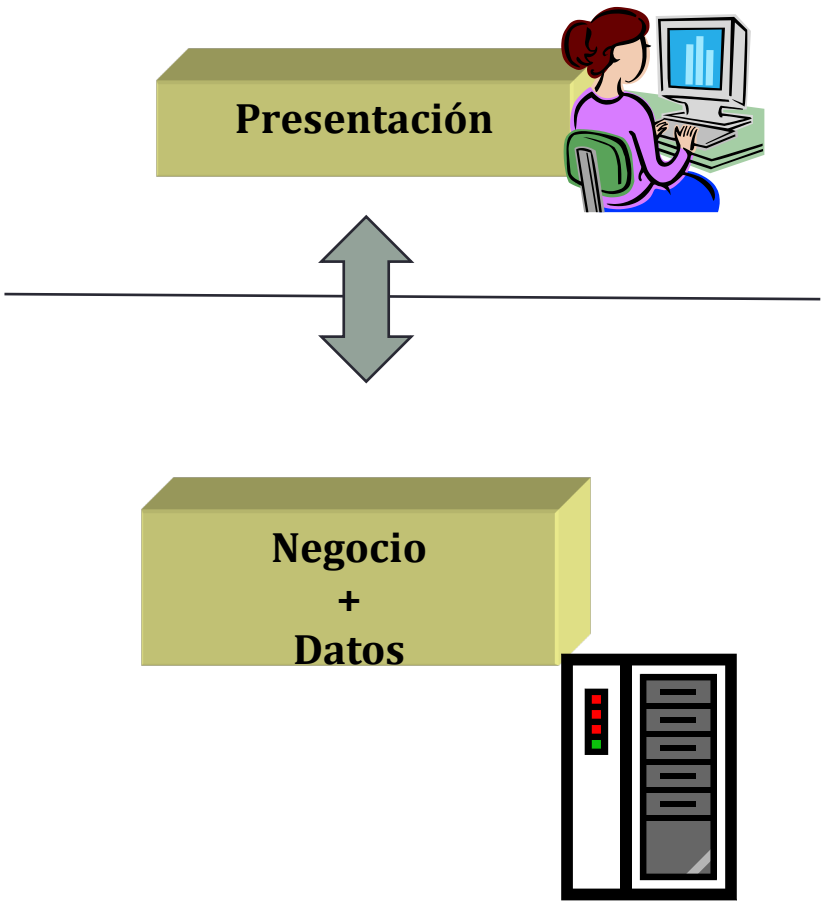
(Genérica)



- **Presentación**
  - Presentación de los resultados de computación al usuario y recogida de entradas del usuario al sistema.
- **Lógica**
  - Proporcionar la funcionalidad de la aplicación
- **Datos**
  - Proporcionar persistencia a los datos, a través de bases de datos, ficheros…

Otras posibilidades:

Arquitectura de 2 capas - Clientes Ligeros (Thin clients)



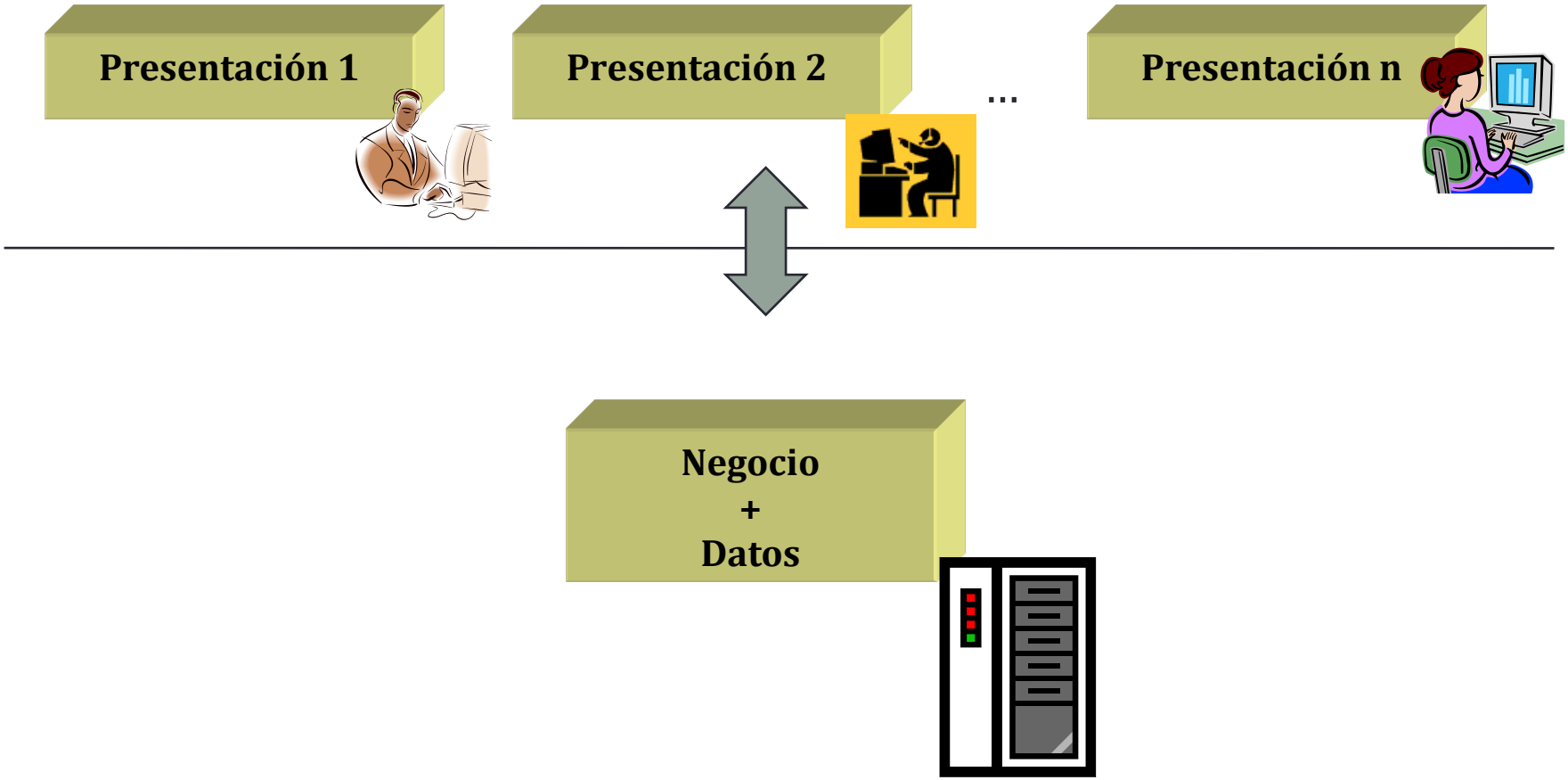
Útiles para:

- Sistemas legados en los que la separación de procesamiento y gestión de datos es impracticable
- Aplicaciones de manejo intensivo de datos (navegación y/o consultas por una BD) con poco procesamiento

Otras posibilidades:

Arquitectura de 2 capas: Clientes Ligeros (Thin clients)

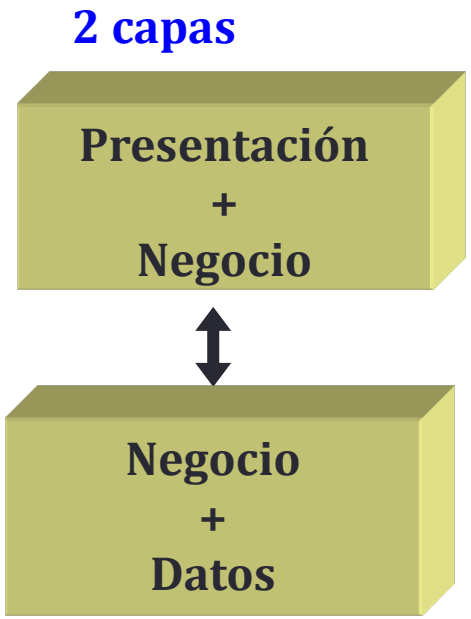
1 Aplicación – N plataformas

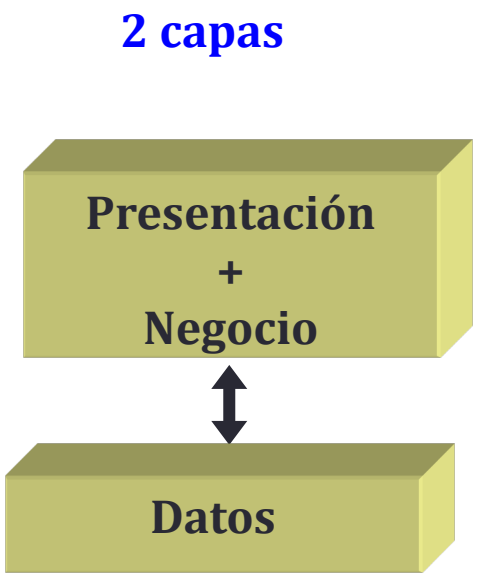


Otras posibilidades:

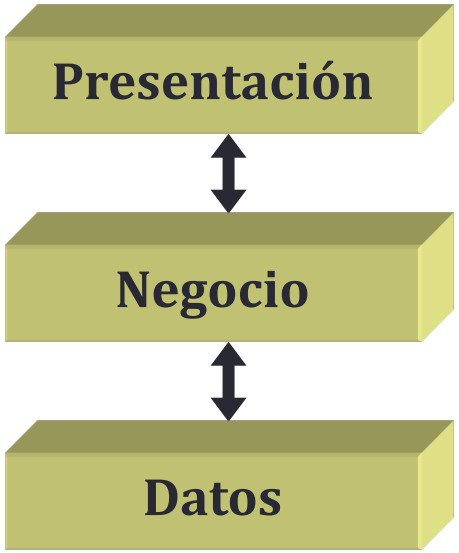
Arquitectura de 2 capas: Clientes Gruesos (Fat clients)

Parte de la lógica (e.g. validaciones, reglas de negocio) se pasa al cliente





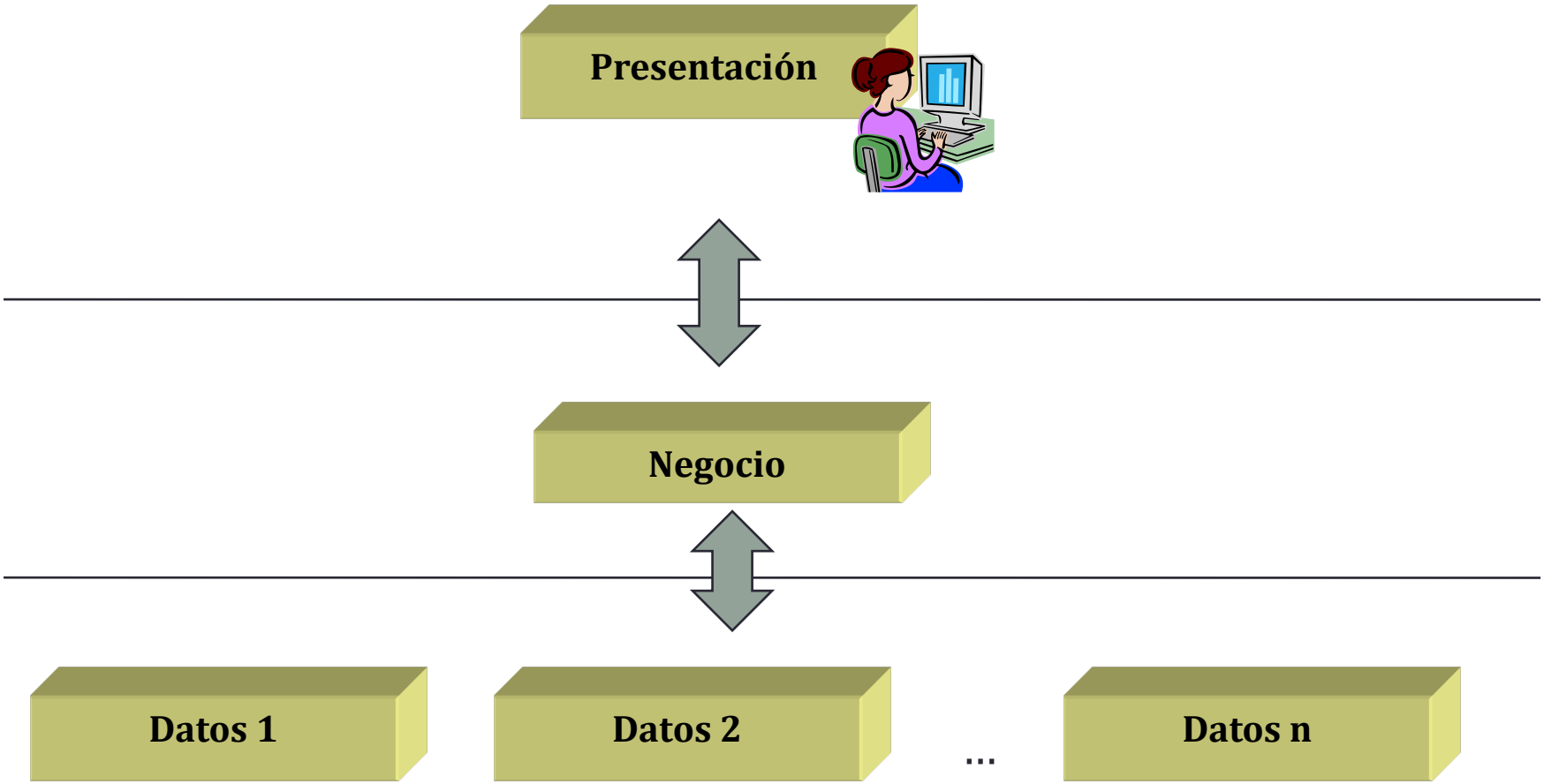
Propuesta: 3 capas



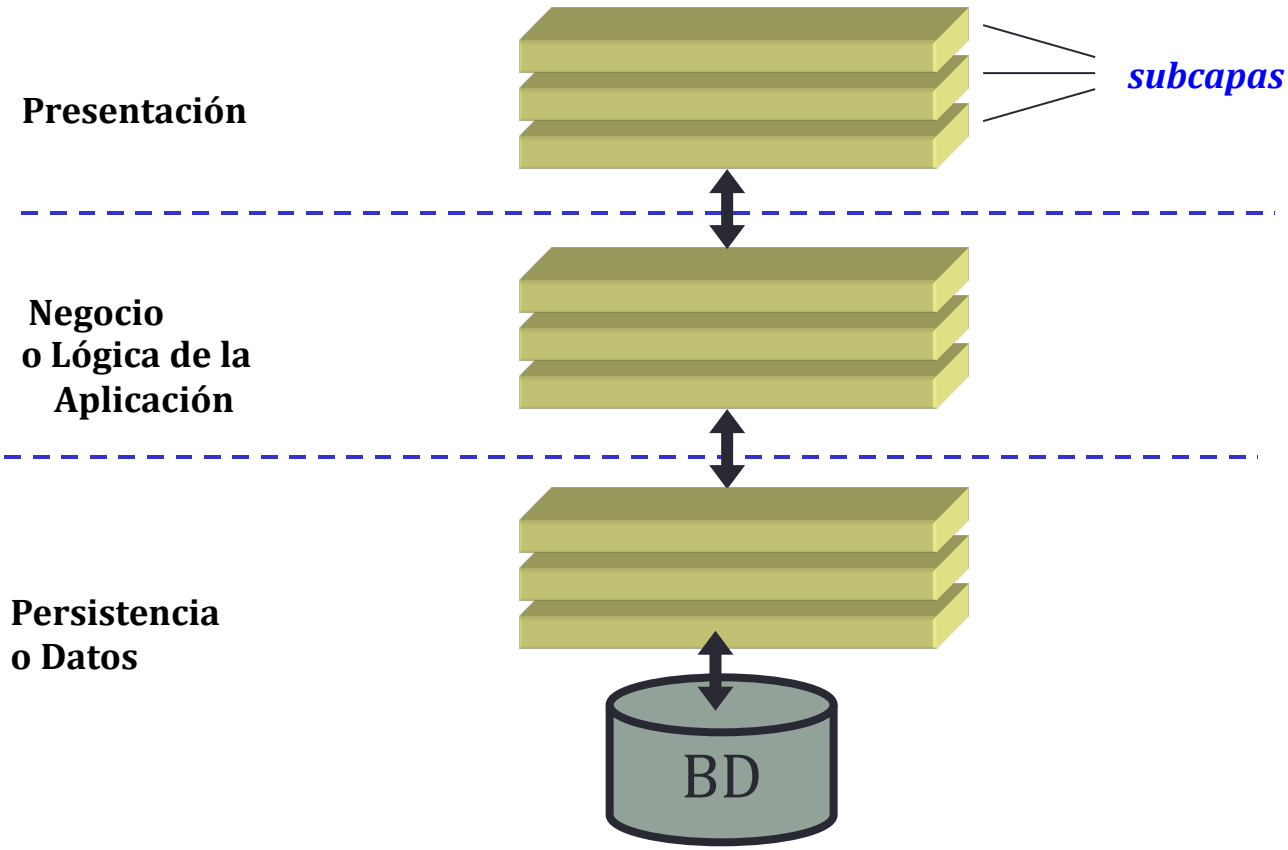
Ventajas

- Aislar la lógica de la aplicación en componentes separados.
- Distribución de capas en diferentes máquinas o procesos *(Cliente /Servidor --- niveles vs. capas)*
- Posibilidad de desarrollo en paralelo.
- Dedicación de recursos a cada una de las capas.
- REUTILIZACIÓN …

Ventajas…



Ventajas ...


Presentación
subcapas
Negocio
o Lógica de la
Aplicación
Persistencia
o Datos
BD

Bibliografía

- **Alonso et al., Web Services: Concepts, Architectures and Applications, Springer, 2004.**
  - **Capítulo 1 y 2**

- **Sommerville, I. "Ingeniería del Software". Ed. Pearson, 7 ed. , 2005.**
  - **Capítulo 11**

- **Sommerville, I., Ingeniería del Software (9ª ed.), Addison-Wesley, 2011. (Disponible online desde Polibuscador)**
  - **Capítulo 6**