

# ARQUITECTURA DEL PROGRAMARI

---

## Tema 3

**Enginyeria del Programari**  
ETS Enginyeria Informàtica  
DSIC – UPV

*Curs 2021-2022*

# Objectius

- Introduir el concepte d'arquitectura del sistema
- Descriure les principals característiques de **l'arquitectura multicapa.**

# Continguts

1. Introducció
2. Concepte d'Arquitectura del Programari
3. Tipus d'arquitectura: Multicapa
  - Arquitectura Multicapa
    - Arquitectura de 3 capes
4. Bibliografia

# INTRODUCCIÓ

---

*Programming in the small/medium/large*

# *“Programming in the small/medium/large”*

- Quan els sistemes programari creixen en grandària, es requereix una organització dels mateixos en subsistemes que els facen gestionables
- Al llarg de la història del desenvolupament de programari s'han utilitzat diferents estratègies per a gestionar la complexitat, generalment relacionades amb el disseny a diferents nivells d'abstracció
  - Mòduls (Mètodes estructurats)
  - Classes (Mètodes orientats a objectes)
  - ...

## **Problemes**

- Estes aproximacions són de baix nivell, són agrupacions de codi.
- Es requereix un mecanisme més abstracte ... que separe l'aplicació en blocs funcionals.

# ARQUITECTURA DEL PROGRAMARI

---

Arquitectura Multicapa

# Què s'entén per arquitectura de programari?

*L'arquitectura de programari*, té a veure amb el disseny i la implementació d'estructures de programari d'alt nivell. És el resultat d'assemblar un cert nombre d'elements arquitectònics de forma adequada per a satisfer la major funcionalitat i requeriments d'acompliment d'un sistema, així com requeriments no funcionals, com la confiabilitat, escalabilitat, portabilitat, i disponibilitat.

*Kruchten, Philippe*

# L'arquitectura de programari és important

- En l'etapa de descripció de **l'Arquitectura del Sistema** hem de proveir al sistema d'una organització global en **subsistemes**.
- **Tipus de Sistemes**
  - **Arquitectura de Sistemes**



# Tipus de Sistemes (entre molts altres...)

- **Sistemes Distribuïts:**

El sistema programari en el qual el processament d'informació es distribueix sobre diverses computadores en compte d'estar confinat en una única màquina.

- **Sistemes Personals:**

No són distribuïts i estan dissenyats per a executar-se en un ordinador personal o estació de treball.

- **Sistemes Encastats:**

Sistemes informàtics (maquinari + programari), usualment de temps real, integrats en un sistema d'enginyeria més general, en el qual realitzen funcions de control, processament i/o monitoratge.

## Arquitectures de Sistemes Distribuïts (entre altres)

- **Arquitectures Multi-processor:**

El sistema consta de múltiples processos que poden (o no) executar-se en diferent processadors.

- **Arquitectures Client/Servidor:**

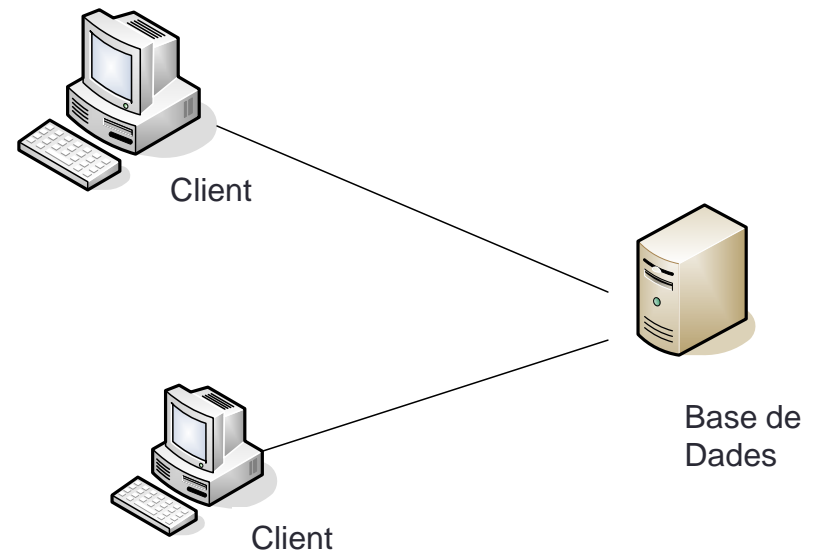
El sistema pot ser vist com un conjunt de serveis que es proporcionen als clients per part dels servidors. Els servidors i els clients es tracten de forma diferent.

- **Arquitectures d'objectes distribuïts:**

El sistema pot ser vist com un conjunt d'objectes que interaccionen i la localització dels quals és irrellevant. No hi ha distinció entre un proveïdor de serveis i l'usuari d'aquests serveis.

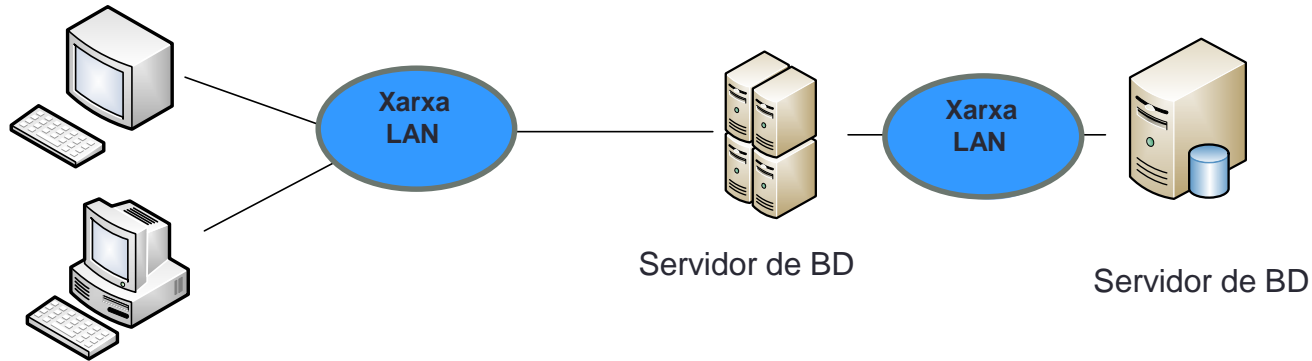
# Arquitectura Client/Servidor

- C/S distribueix una aplicació en 2 components especialitzats, l'execució dels quals es porta a terme en 1 o més equips:
  - El servidor (S) és un proveïdor de serveis.
  - El client (C) és un consumidor de serveis.
- C i S Interactuen per un mecanisme de pas de missatges:
  - Sol·licitud de servei.
  - Resposta.

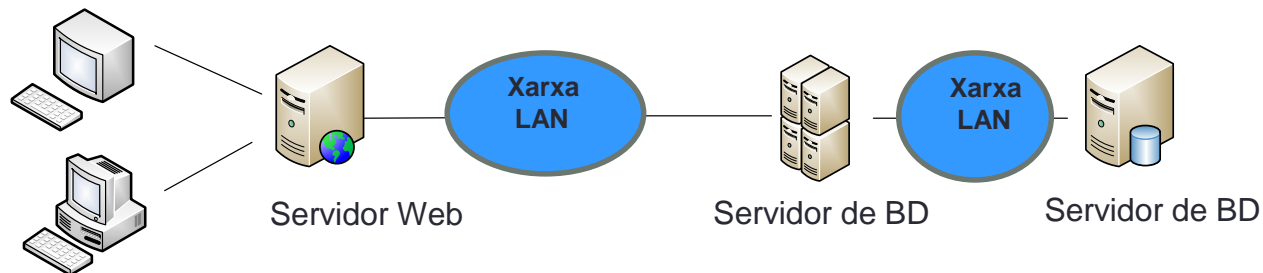


# Client -Servidor 3/N-nivells

- Arquitectura 3-nivells

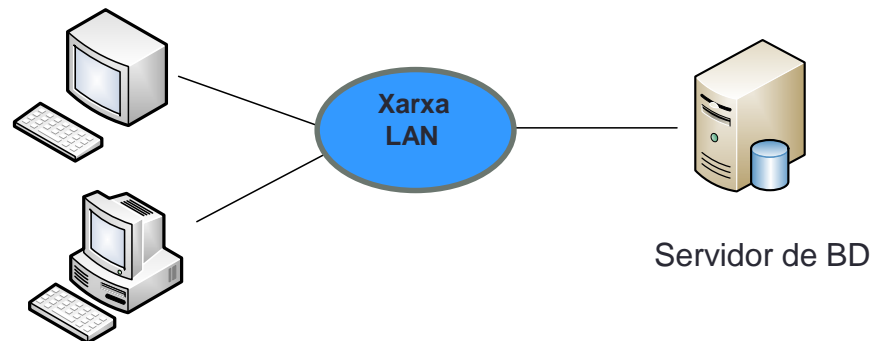
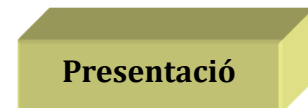


- Arquitectura de 4-nivells



# Nivells *versus* Capes

- **Capa** (*layer*) fa referència a una segmentació lògica de la solució, mentre que **nivell** (*tier*) es refereix a la segmentació o ubicació física.



# **\*\*Arquitectura Multicapa\*\***

Un **sistema per capes** és un conjunt ordenat de subsistemes, cadascun dels quals està construït en termes dels quals té per sota, i proporciona la base de la implementació d'aquells que estiguen per sobre d'ell.

- Els objectes de cada capa poden ser **independents** (recomanat) encara que solen haver-hi dependències entre objectes de diferents capes.
- Existeix una relació **client/servidor** entre les capes **inferiors** (que proporcionen serveis) i les capes **superiors** (que són usuaris d'aquests serveis).

# Arquitectura Multicapa – Visibilitat de les capes

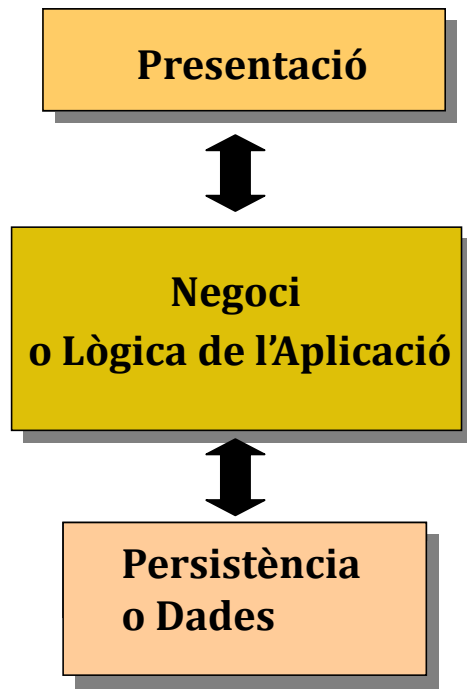
Les arquitectures basades en capes poden ser **obertes** o **tancades** segons la dependència existent entre capes.

- **obertes**: una capa pot utilitzar característiques de qualsevol capa a qualsevol nivell.
- **tancades**: una capa només utilitza característiques de la seua capa immediatament inferior.

Es recomana treballar amb arquitectures **tancades**, ja que redueixen les dependències entre capes i permeten que els canvis es facen amb facilitat perquè la interfície d'una d'elles només afecta a la capa següent.

# Arquitectura de Tres Capes

(Genèrica)



- **Presentació**

- Presentació dels resultats de computació a l'usuari i recollida d'entrades de l'usuari al sistema.

- **Lògica**

- Proporcionar la funcionalitat de l'aplicació

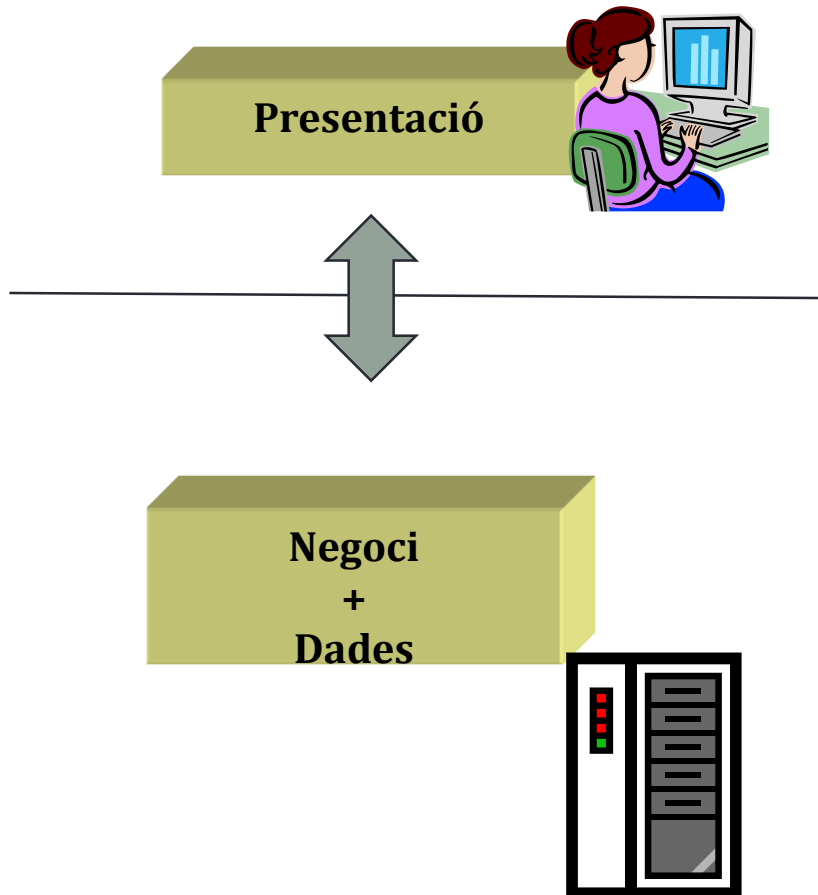
- **Dades**

- Proporcionar persistència a les dades, a través de bases de dades, fitxers...



## Altres possibilitats:

### Arquitectura de 2 capes: Clients Lleugers (*Thin clients*)



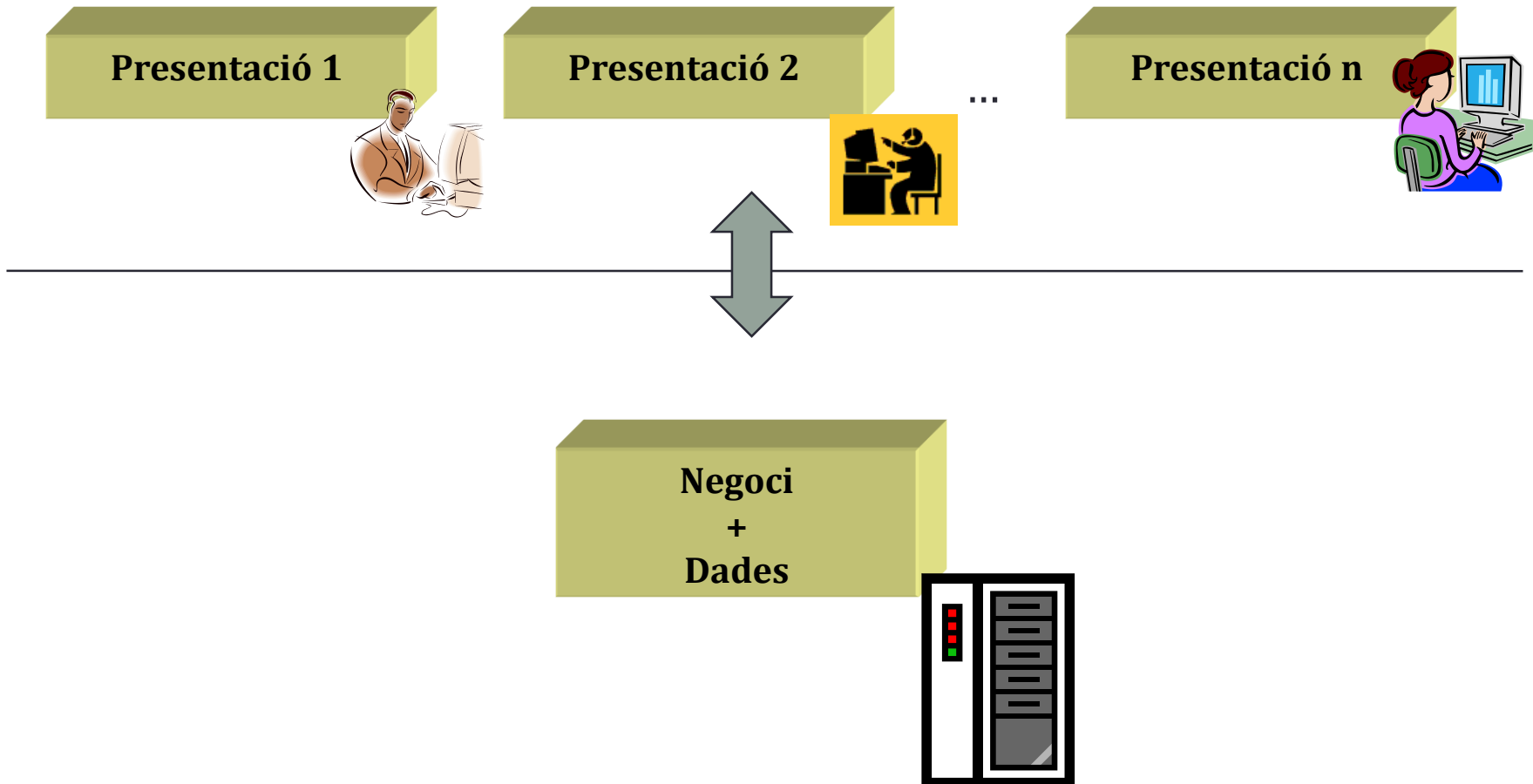
Útils para:

- Sistemes llegats (*legacy systems*) en els quals la separació de processament i gestió de dades és impracticable
- Aplicacions de maneig intensiu de dades (navegació i/o consultes per una BD) amb poc processament

## Altres possibilitats:

### Arquitectura de 2 capes: Clients Lleugers (*Thin clients*)

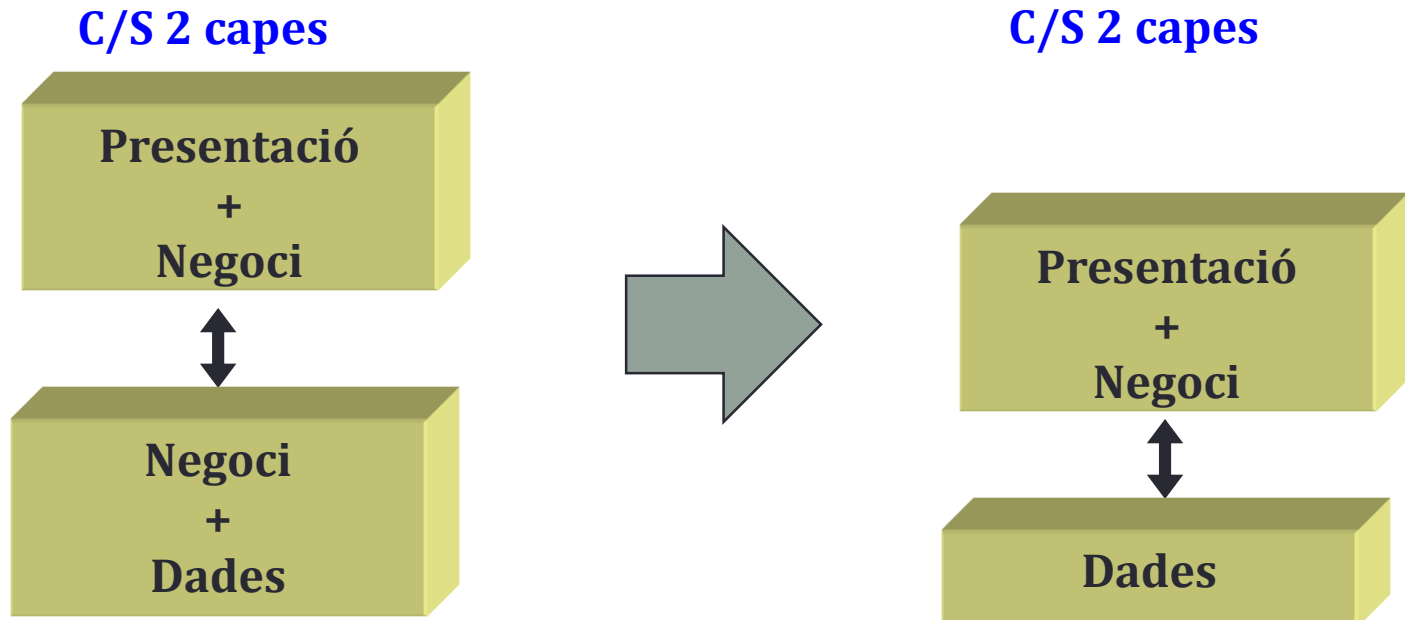
1 Aplicació – N plataformes:



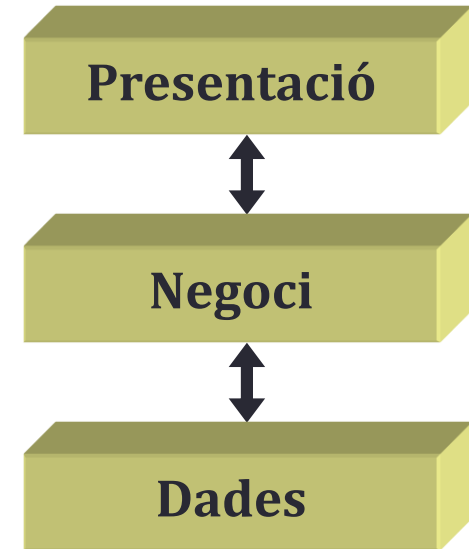
## Altres possibilitats:

### Arquitectura de 2 capes: Clients Gruixuts (*Fat clients*)

Part de la lògica (com p.e. validacions, regles de negoci) es passa al client



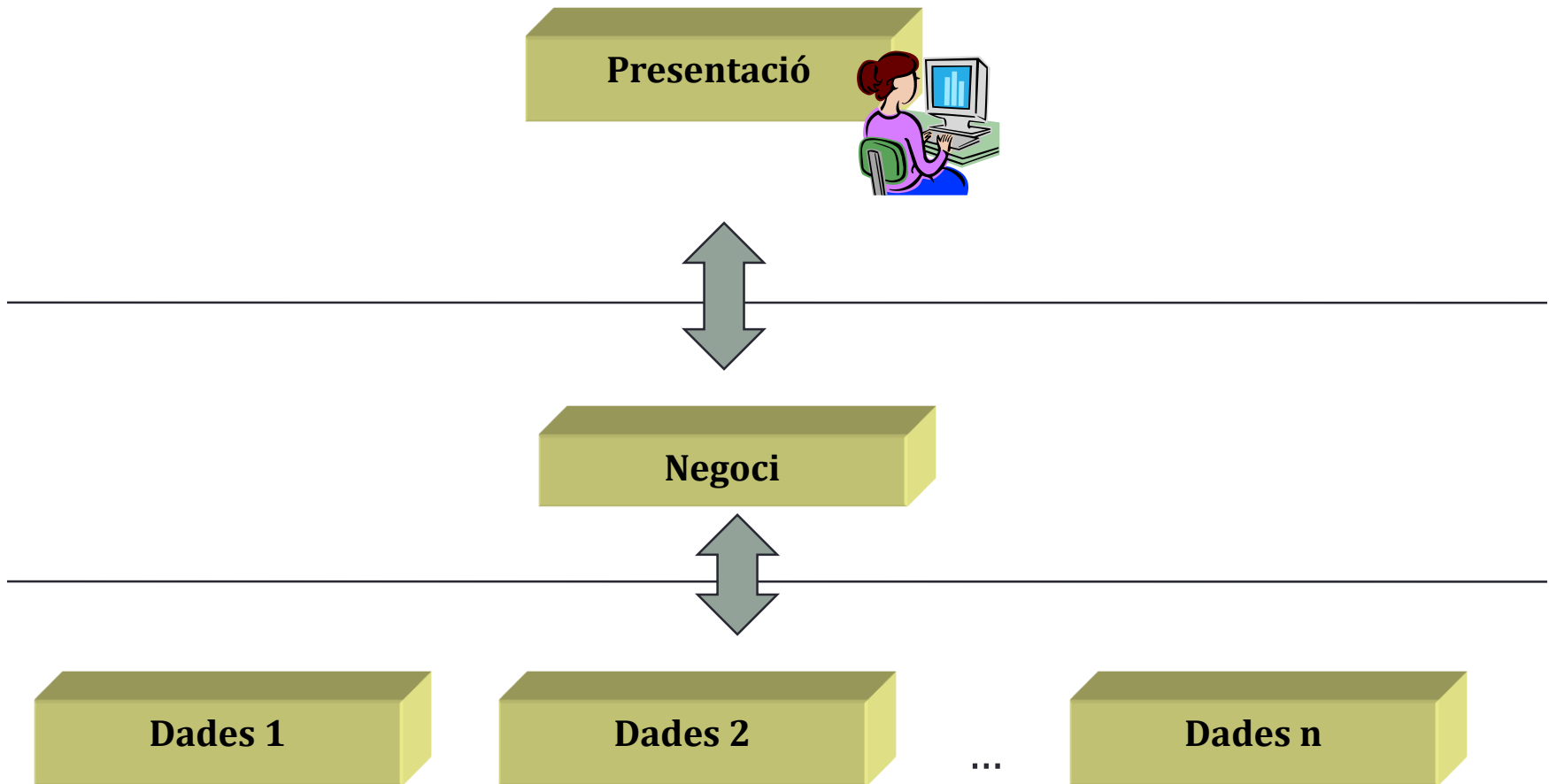
# Proposta: 3 capes



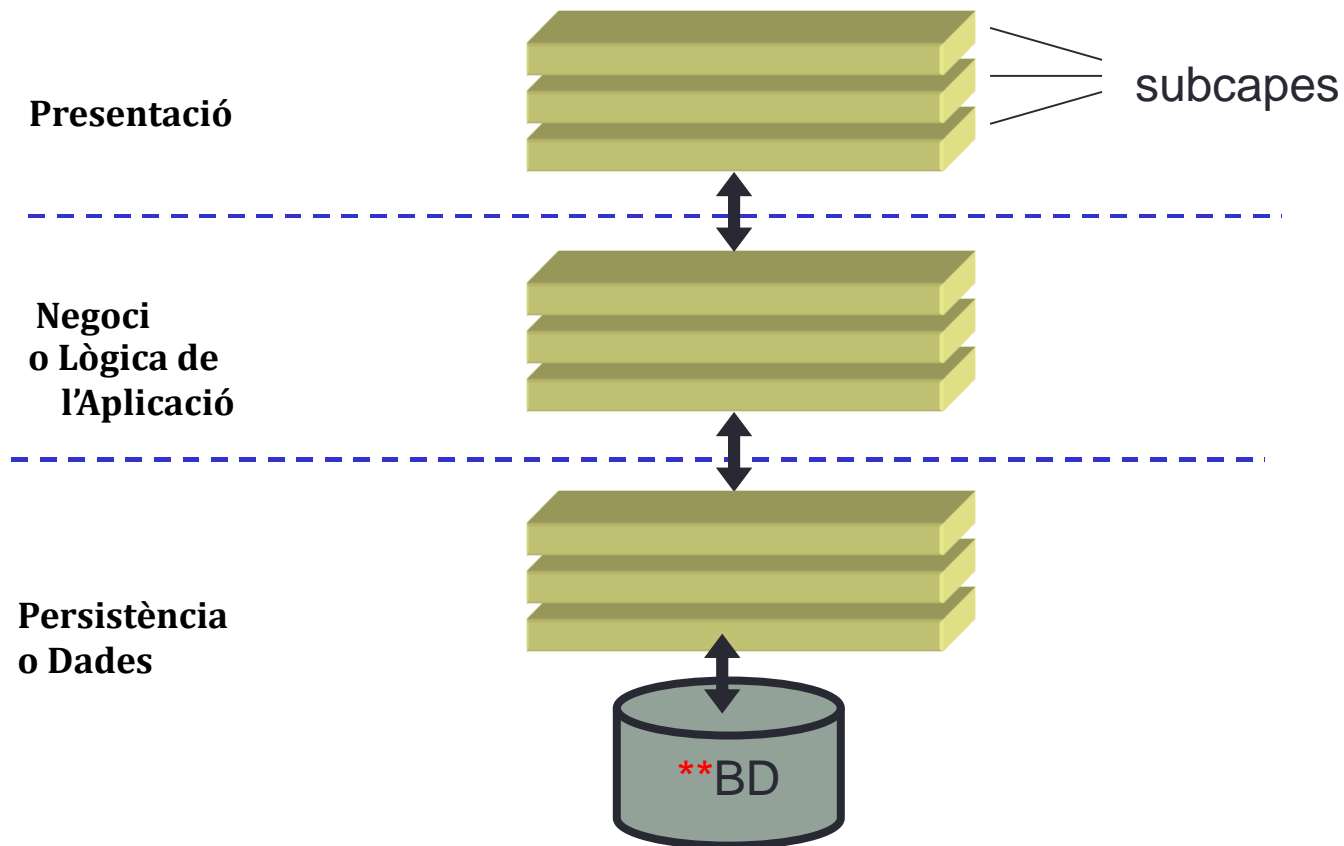
## Avantatges

- Aïllar la lògica de l'aplicació en components separats.
- Distribució de capes en diferents màquines o processos (Client/Servidor ---- Nivells vs. Capas)
- Possibilitat de desenvolupament en paral·lel.
- Dedicació de recursos a cadascuna de les capes.
- REUTILITZACIÓ ...

# Avantatges...



# Arquitectura de Tres Capes: variants



# Bibliografia

 **Alonso et al., Web Services: Concepts, Architectures and Applications, Springer, 2004.**

 **Capitol 1 i 2**

 **Sommerville, I. “Ingeniería del Software”. Ed. Pearson, 7 ed. , 2005.**

 **Capitol 11** *(disponible online des-de Polibuscador)*

 **Sommerville, I., Ingeniería del Software (9<sup>a</sup> ed.), Addison-Wesley, 2011**

 **Capitol 6**