

## Índice

- Introducción
- Arquitectura de una capa
  - Características
  - Ventajas e inconvenientes
- Arquitectura de dos capas
  - Características
  - Ventajas e inconvenientes
  - Patrones relacionados
- Arquitectura multicapa
  - Características
  - Ventajas e inconvenientes
  - Patrones relacionados
- Patrón controlador frontal
- Patrón controlador aplicación
- Patrón MVC
- Patrón transferencia
- Patrón *Data Access Object*
- Patrón delegado del negocio
- Patrón servicio de aplicación
- Patrón objeto del negocio
- Patrón almacén del dominio

## Introducción

- No nos preocuparemos de obtener buenos diseños a nivel componentes de cada capa
- De ello se ocupan disciplinas como:
  - Interacción persona-computadora
  - Programación
  - Patrones de diseño
  - Bases de datos

## Introducción

- Nosotros nos preocuparemos de dar un buen sistema de información desde el punto de vista *arquitectónico*
- Arquitectura\* es la organización fundamental de un sistema, expresado en sus componentes, sus relaciones entre ellos, y en el entorno y principios que guían su diseño y evolución

\*IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems

## Introducción

- Los sistemas de información tienen una gran relevancia en informática
- Según la ACM\* incluyen:
  - Modelos y principios
  - Gestión de bases de datos
  - Almacenamiento y recuperación de información
  - Aplicaciones de sistemas de información
  - Interfaces y presentación de la información

\*<http://www.acm.org/class/1998/>

## Introducción

- Por eso, veremos los principales patrones de la arquitectura multicapa en sistemas de información
- Quizás una denominación más actual puede ser *patrones de arquitectura de aplicaciones empresariales*

## Introducción

- Las características de las aplicaciones empresariales son:
  - Manejan una gran cantidad de datos persistentes
  - Estos datos son accedidos concurrentemente
  - Hay una gran cantidad de lógica del negocio, que representa la funcionalidad de la aplicación

## Introducción

- El acceso se produce a través de elaboradas interfaces de usuario
- Suelen tener necesidades de integración con otras aplicaciones empresariales de arquitectura heterogénea

## Introducción

- Según Christopher Alexander, “un *patrón* describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces”

## Introducción

- Aunque Alexander se refería a patrones en ciudades y edificios, lo que dice también es válido para patrones de diseño OO
- Podemos decir que los patrones de diseño:
  - Son soluciones simples y elegantes a problemas específicos del diseño de software OO.
  - Representan soluciones que han sido desarrolladas y han ido evolucionando a través del tiempo.

## Introducción

- Los patrones de diseño no tienen en cuenta cuestiones tales como:
  - Estructuras de datos.
  - Diseños específicos de un dominio.
- Son descripciones de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto

## Introducción

- Cada patrón de diseño identifica:
  - Las clases e instancias participantes.
  - Los roles y colaboraciones de dichas clases e instancias.
  - La distribución de responsabilidades

## Introducción

- Algunas fuentes de patrones:
  - GRASP\*, de Craig Larman.
  - Los patrones *Gang of Four* (GoF), de Eric Gamma et al.
  - *Core J2EE patterns*, de Alur et al.
  - *Patterns of Enterprise Application Architecture*, de Fowler et al.

\**General Responsibility Assignment Software Patterns*

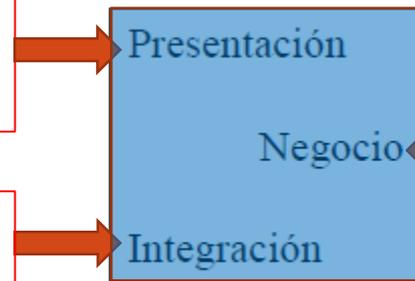
## Arquitectura de una capa

### Características

- La arquitectura de una capa no divide al sistema en presentación, negocio e integración

La capa de presentación encapsula toda la lógica de presentación (Interfaz) necesaria para dar servicio a los clientes que acceden al sistema

La capa de Integración es responsable de la comunicación con recursos (Datos) y sistemas externos

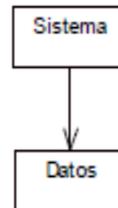


La capa de negocio proporciona los servicios (Funcionalidades) del sistema

Arquitectura de una capa

# Arquitectura de una capa

## Características

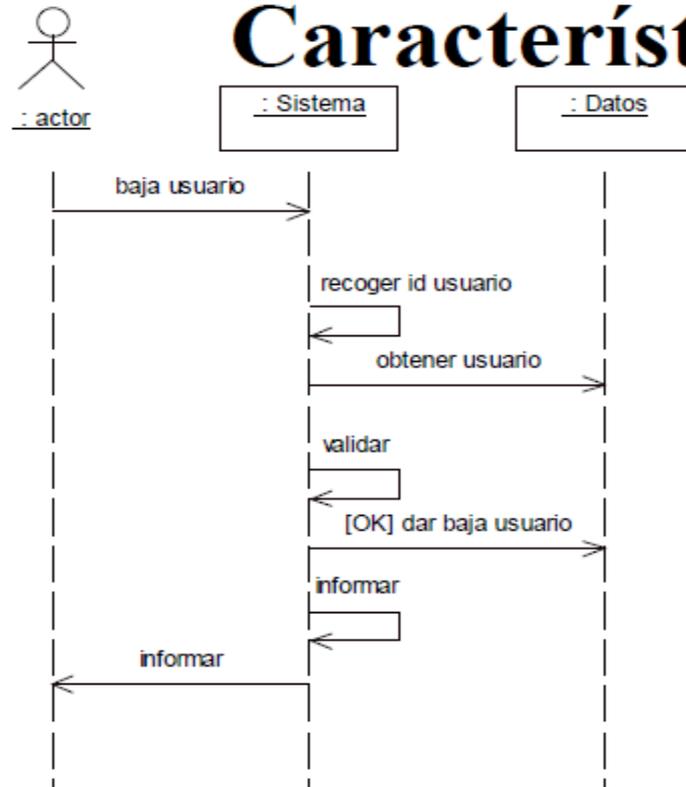


Clases del sistema

## Arquitecturas Multicapa

# Arquitectura de una capa

## Características



Comportamiento del sistema

# Arquitectura de una capa

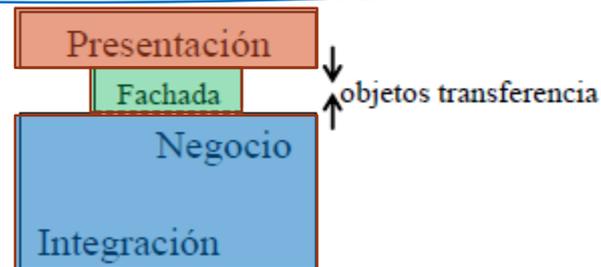
## Ventajas e inconvenientes

- Ventajas
  - Sencillez conceptual
- Inconvenientes
  - No se puede modificar ni la interfaz de usuario, ni la lógica del negocio ni la representación de los datos sin afectar a las demás capas
  - Complicación fáctica

# Arquitectura de dos capas

## Características

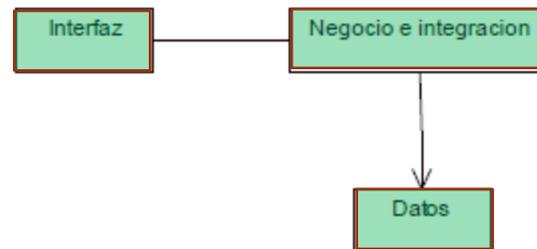
- La arquitectura de dos capas diferencia entre la capa de presentación y el resto del sistema
- No diferencia negocio de integración



Arquitectura de dos capas

# Arquitectura de dos capas

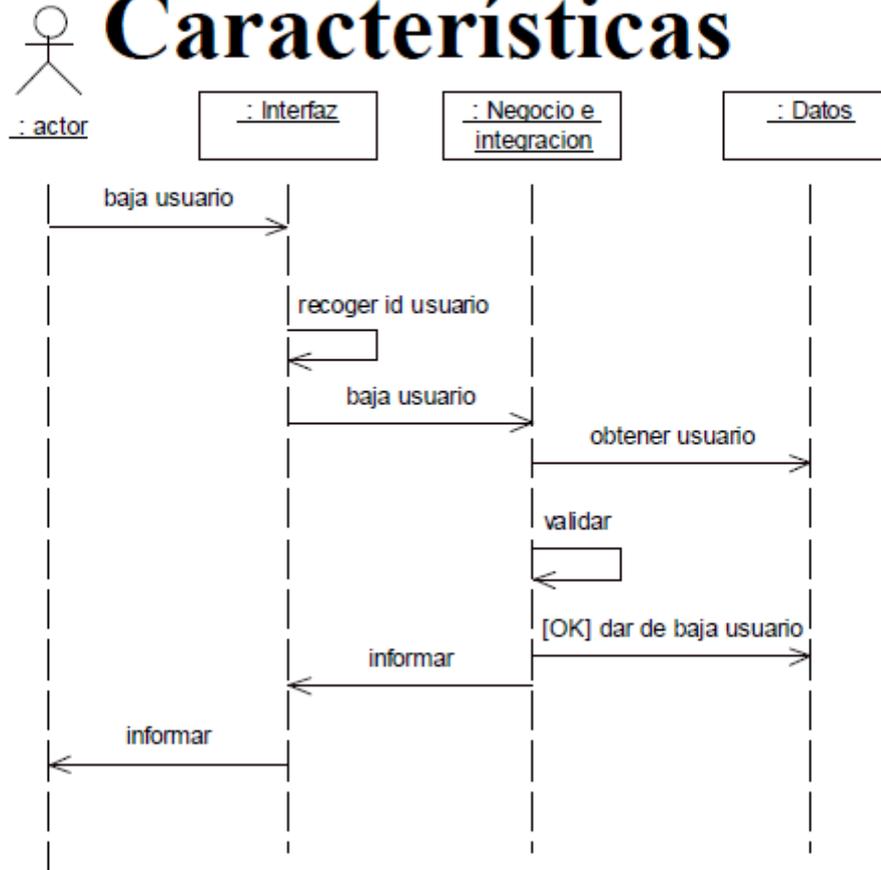
## Características



**Clases del sistema**

# Arquitectura de dos capas

## Características



Comportamiento del sistema

# Arquitectura de dos capas

## Ventajas e inconvenientes

- Ventajas
  - Permite cambios en el interfaz de usuario o en el resto del sistema sin interferencias mutuas
  - Simplicidad fáctica
- Inconvenientes
  - Mayor complicación arquitectónica que la arquitectura de una capa
  - No se puede modificar la lógica del negocio o la representación de los datos sin interferencias mutuas

# Arquitectura de dos capas

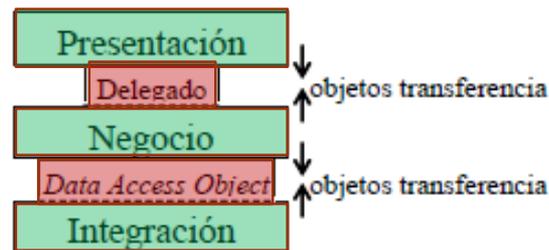
## Patrones relacionados

- Aunque no es estrictamente necesario, suele utilizarse:
  - MVC (Modelo Vista Controlador)

# Arquitectura multicapa

## Características

- La arquitectura multicapa considera una capa de presentación, otra de negocio, y otra de integración



Arquitectura multicapa

# Arquitectura multicapa

## Características

- *La capa de presentación* encapsula toda la lógica de presentación necesaria para dar servicio a los clientes que acceden al sistema
- *La capa de negocio* proporciona los servicios del sistema
- *La capa de integración* es responsable de la comunicación con recursos y sistemas externos



Datos

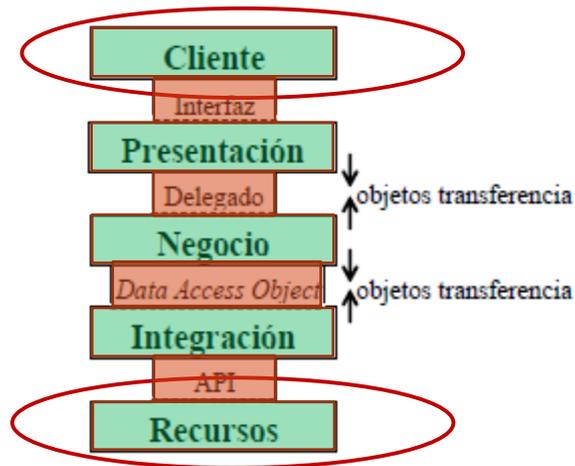
## Arquitectura multicapa

### Características

- En realidad, la arquitectura es de cinco capas, ya que incluye las capas de clientes y recursos
- La capa de clientes representa a todos los dispositivos o clientes del sistema que acceden al mismo. Está sobre la capa de presentación
- La capa de recursos contiene los datos del negocio y recursos externos. Está bajo la capa de integración

# Arquitectura multicapa

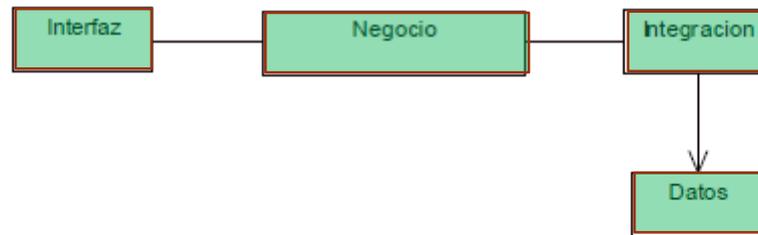
## Características



Arquitectura multicapa

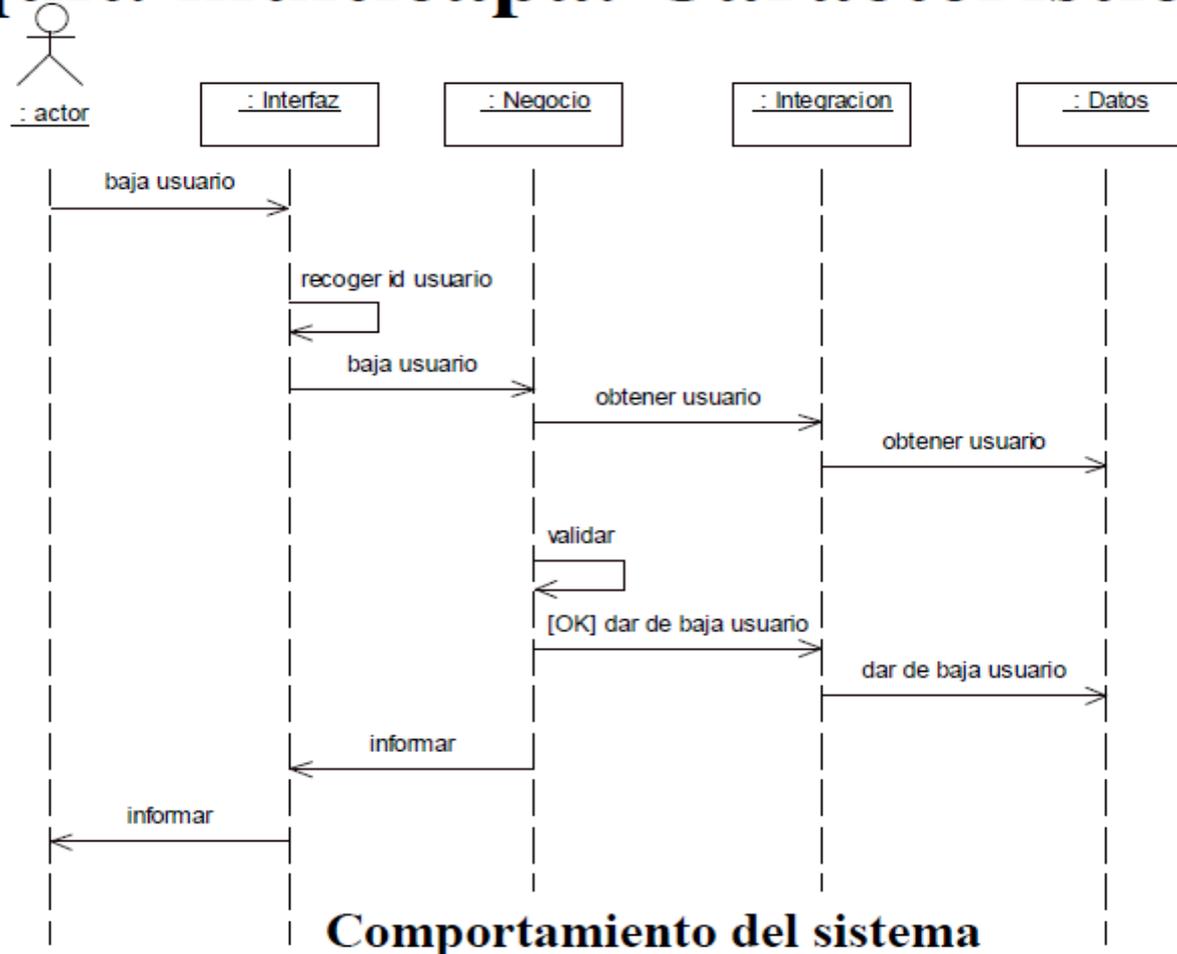
# Arquitectura multicapa

## Características



**Clases del sistema**

# Arquit. multicapa. Características



## Arquitectura multicapa

### Características

- Nótese que estas son capas lógicas
- Otra cosa son las capas *físicas*
- Así, la capa de presentación web y la lógica del negocio podrían estar en la misma máquina o en máquinas distintas

## Arquitectura multicapa

### Características

- Ventajas
  - Se puede modificar cualquier capa sin afectar a las demás
  - ¿Simplicidad fáctica?
- Inconvenientes
  - Mayor complejidad arquitectónica

## Arquitectura multicapa

### Características

- Ventajas:
  - Integración y reusabilidad
  - Encapsulación
  - Distribución
  - Particionamiento
  - Escalabilidad
  - Mejora del rendimiento
  - Mejora de la fiabilidad

# Arquitectura multicapa

## Características

- Manejabilidad
- Incremento en la consistencia y flexibilidad
- Soporte para múltiples clientes
- Desarrollo independiente
- Desarrollo rápido
- Empaquetamiento
- Configurabilidad

# Arquitectura multicapa

## Características

- Inconvenientes:
  - Posible pérdida de rendimiento
  - Riesgos de seguridad
  - Gestión de componentes