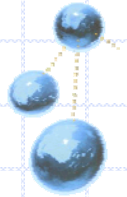




CC68J APLICACIONES EMPRESARIALES CON JEE



# PATRONES ARQUITECTURALES

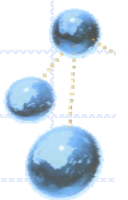
N-Capas, MVC & SOA

Profesores:

■ Andrés Farías

# Objetivos

- Entender las ventajas de una arquitectura N-capas.
- Entender el valor del patrón MVC.
- Entender la problemática empresarial respecto a las arquitecturas de TI.
- Entender los beneficios de una arquitectura orientada a servicios.



*¿Qué es esto de los patrones?*

# INTRODUCCIÓN GENERAL



# Patrones

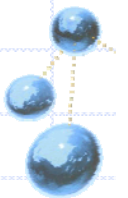
## De diseño y arquitecturales

- Un patrón es un problema-solución (plantilla de soluciones en un contexto de objetos)
  - ✓ Facilitan la reusabilidad de arquitecturas software (no de código)
- Las descripciones de patrones son una especie de “patentes de software”
- “Gang of Four” (GoF). Libro más representativo:
  - ✓ “Design patterns, elements of reusable object-oriented software”. Gamma, Helm, Johnson, Vlissides, 1995, Addison Wesley.



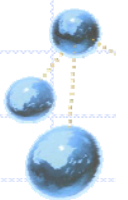
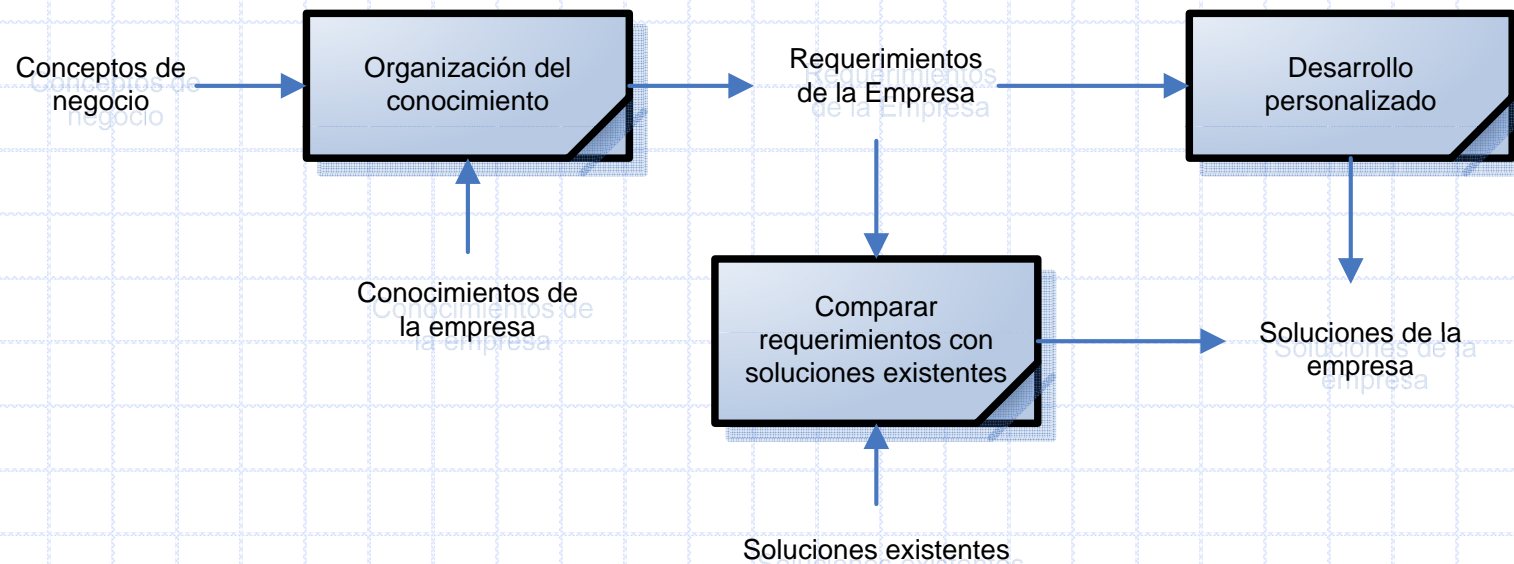
# ¿Qué es una Arquitectura?

- La **arquitectura** de un sistema Informático es la estructura o las estructuras de ese sistema, y comprende las componentes del software, las Componentes de la Plataforma (HW, Redes), sus **propiedades** y las relaciones entre las mismas.



# Arquitectura de TI

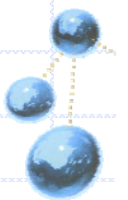
## La visión de la organización



# Arquitectura de TI

## El negocio... cuenta!

- La *comprensión* de los productos y procesos de negocio que apoyan la producción de éstos, es una etapa fundamental y necesaria para dar los principales lineamientos de la arquitectura.
- Comprender los productos y procesos de una organización implica:
  - ✓ Identificar los *procesos de negocio* que permiten que el Negocio de la organización se lleve a cabo
  - ✓ Identificar *requerimientos fundamentales* de los procesos de negocio.
  - ✓ Identificar cada elemento participante en cada proceso de negocio: *Actores, Recursos, Situaciones*, etc.





# Arquitectura de TI

## Propiedades de una arquitectura

- Las propiedades observables:
  - ✓ Desempeño,
  - ✓ Seguridad,
  - ✓ Disponibilidad,
  - ✓ Funcionalidad,
  - ✓ Usabilidad
- Las propiedades no observables:
  - ✓ Modificabilidad,
  - ✓ Portabilidad,
  - ✓ Reusabilidad,
  - ✓ Integrabilidad,
  - ✓ Testeabilidad
- Una arquitectura no puede satisfacer todas las propiedades deseadas: la definición debe considerar las propiedades más importantes.





# Propiedades de una Arquitectura

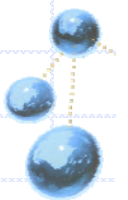
## Observables: Desempeño y seguridad

- Desempeño:

- ✓ *Tiempo que requiere el sistema para responder a un evento o estímulo, o bien el número de eventos procesados en un intervalo de tiempo.*

- Seguridad:

- ✓ *Medida de la capacidad del sistema para resistir intentos de uso y negación de servicios a usuarios no autorizados sin restar servicios a los usuarios autorizados.*

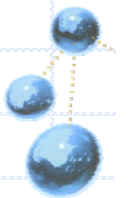


# Propiedades de una Arquitectura

## Observables: Disponibilidad

- *Proporción del tiempo que el sistema está en ejecución.*
- *Se mide como el tiempo entre fallas o la rapidez en que el sistema puede reiniciar la operación cuando ocurre una falla.*

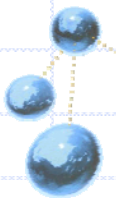
$$\alpha = \frac{\text{tiempo\_promedio\_entre\_fallas}}{\text{tiempo\_promedio\_entre\_fallas} + \text{tiempo\_medio\_reparación}}$$



# Propiedades de una Arquitectura

## Observables: Funcionalidad

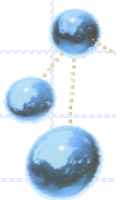
- *Habilidad de un sistema para hacer la tarea para la cual fue creado.*
- *Todas las partes del sistema deben coordinarse para logra el objetivo común:*
  - ✓ *asignar la responsabilidad a cada componente*
  - ✓ *cada cual debe saber el momento en que debe ejecutar su responsabilidad*
- *La funcionalidad es ortogonal a la estructura, y de hecho muchas estructuras diferentes pueden lograr la misma funcionalidad.*



# Propiedades de una Arquitectura

## Observables: Usabilidad

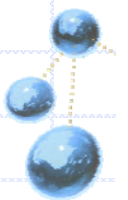
- *Aprendibililidad* - ¿Cuán rápido y fácil es para un usuario el aprender a usar la interfaz del sistema?
- *Eficiencia* - ¿El sistema responde con la rapidez apropiada a las exigencias del usuario?
- *Recordabilidad* - ¿Pueden los usuarios recordar cómo usar el sistema entre dos sesiones de uso?
- *Propenso a errores* - ¿El sistema anticipa y previene los errores comunes de los usuarios?
- *Manejo de errores* - ¿El sistema ayuda a los usuarios a recuperarse de los errores?
- *Satisfacción* - ¿El sistema facilita la tarea de los usuarios?



# Arquitectura de Negocio

## Integración organizacional

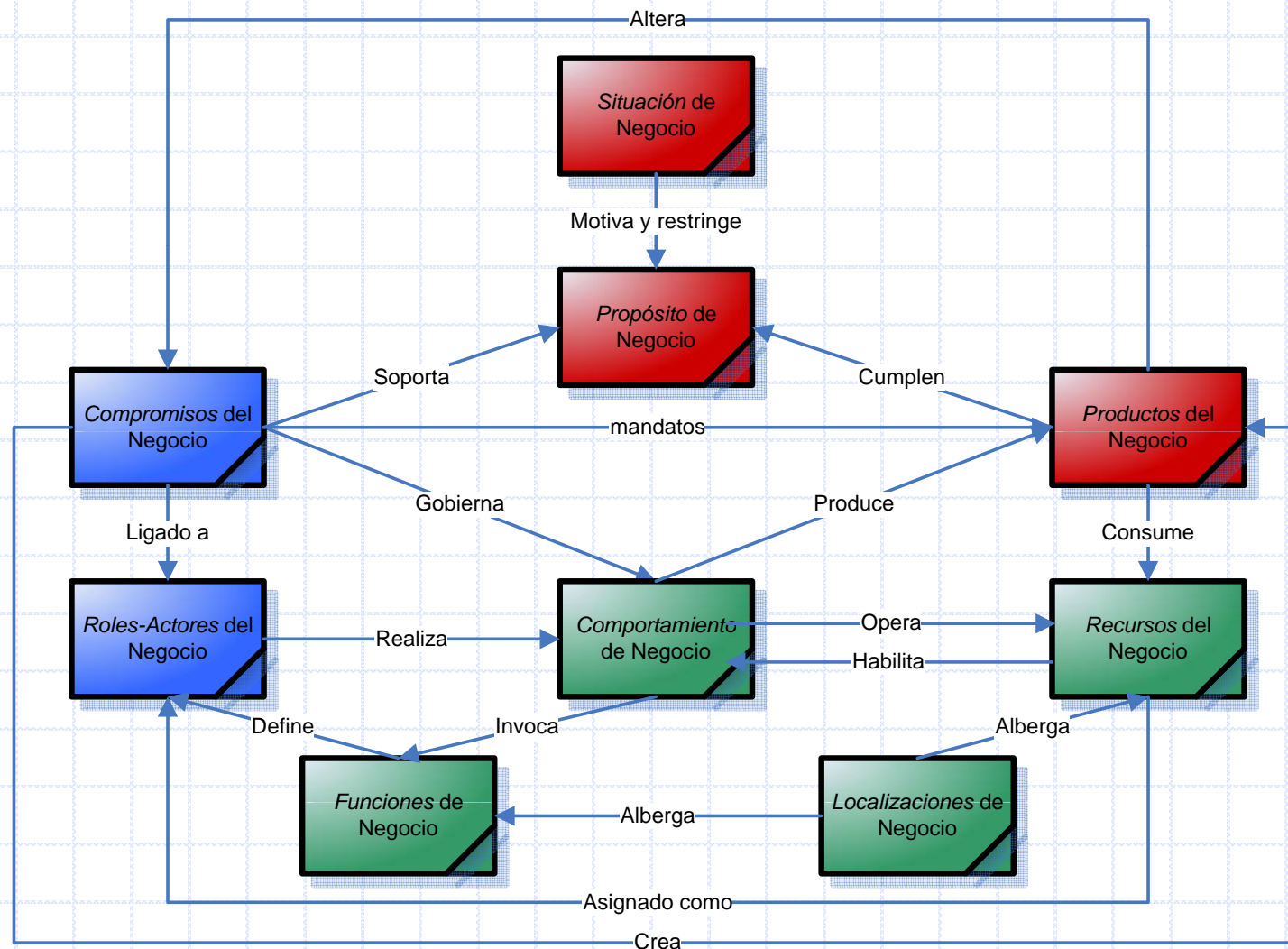
- Construimos un modelo de la **arquitectura del negocio** con el fin de proveer los elementos necesarios para la especificación de un sistema de **tecnología de información** (TI).
- Conceptos Arquitecturales capturan:
  - ✓ Requerimientos del negocio
  - ✓ Limites del negocio
  - ✓ Producto y Servicios que entrega el negocio.
- **Relación con la arquitectura de IT:** Definimos la manera en que los elementos de la arquitectura del negocio serán representados en la arquitectura de TI.





# Arquitectura de Negocio

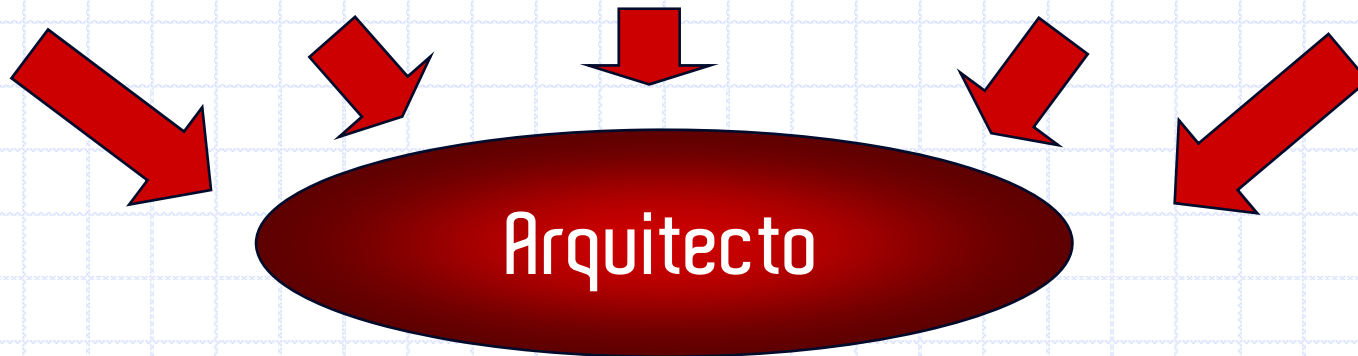
## Elementos para su descripción



# Arquitectura de Negocio

## Elementos para su descripción

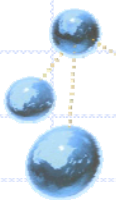
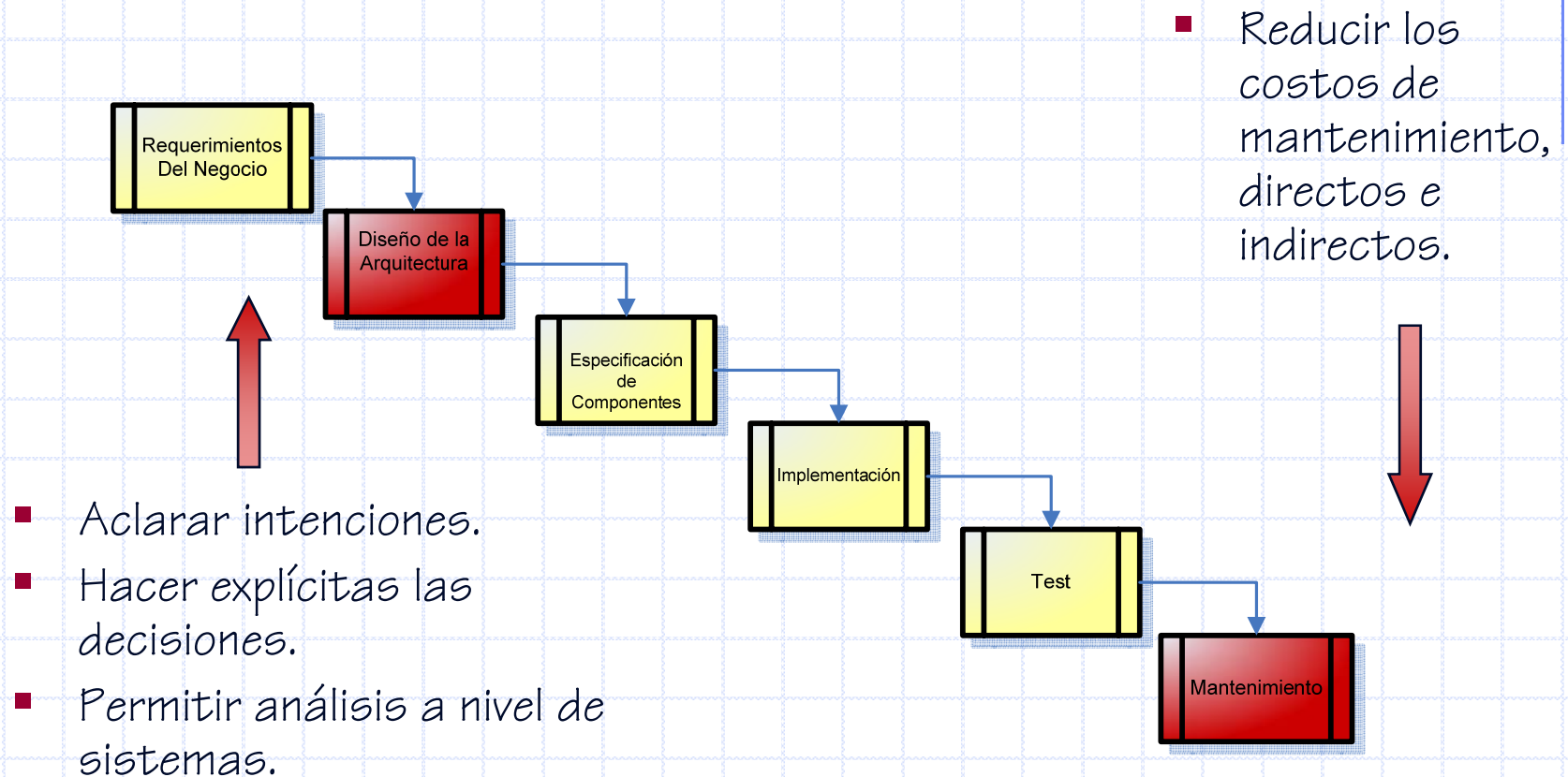
Constructor del Software	Comercial	Usuario Final	Mantenedores	Cliente
<ul style="list-style-type: none"><li>✚ Bajos costos,</li><li>✚ Mantener el Foco de los equipo de desarrollo</li><li>✚ Eficiencia y Control</li></ul>	<ul style="list-style-type: none"><li>✚ Elementos atractivos.</li><li>✚ Corto tiempo para poner en el mercado,</li><li>✚ Bajos costos,</li><li>✚ Comparable con productos de la competencia.</li></ul>	<ul style="list-style-type: none"><li>✚ Comportamiento apropiado.</li><li>✚ Desempeño,</li><li>✚ Seguridad,</li><li>✚ Confiabilidad.</li></ul>	<ul style="list-style-type: none"><li>✚ Mantenibilidad.</li></ul>	<ul style="list-style-type: none"><li>✚ Bajos costos.</li><li>✚ Entrega a tiempo</li></ul>





# Diseño de la Arquitectura

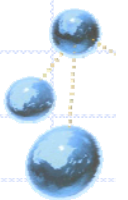
## El proceso



# Diseño de la Arquitectura

## El proceso

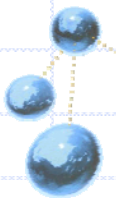
- La arquitectura debe ser producida por un equipo pequeño de Arquitectos de TI y del Negocio.
- El arquitecto debe definir los requisitos técnicos de los sistema y una construir una lista priorizada de propiedades cualitativas que el software debe satisfacer
- La arquitectura debe estar bien documentada usando una *notación acordada* que todos los interesados puedan comprender con poco esfuerzo.
- Se debe facilitar la arquitectura a los directores del negocio, los cuales deben estar involucrados activamente en su revisión.



# Diseño de la Arquitectura

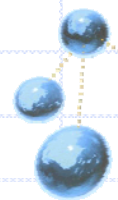
## El proceso

- La arquitectura debe ser analizada para comprobar sus propiedades cuantitativas y cualitativas tempranamente en el proceso de diseño
- La arquitectura debe permitir crear una estructura (esqueleto) de sistema donde se reflejen todas las vías de comunicación pero con mínima funcionalidad.
- El diseño de la arquitectura debe dar como resultado un conjunto específico de áreas críticas de consumo de recursos, cuya resolución estará claramente especificada, definida y mantenida.

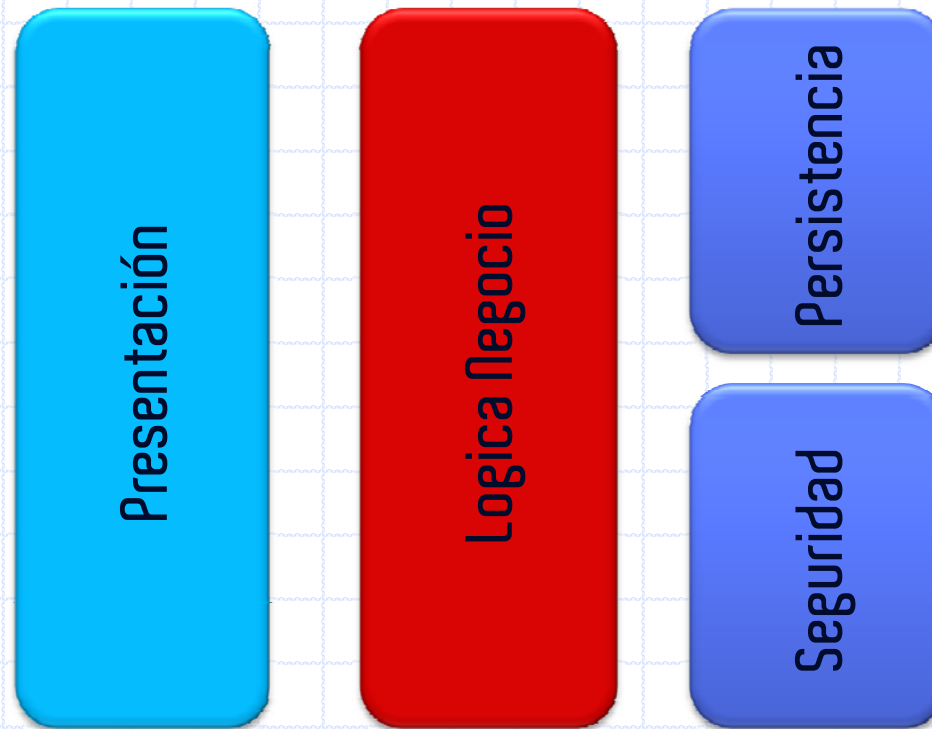


Separación de las aguas en TI

# ARQUITECTURA N-CAPAS



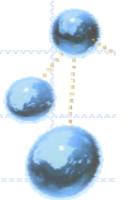
# Patrón Arquitectural N-Capas / 3-Capas



# Patrones Arquitecturales

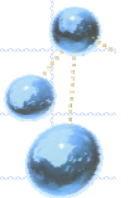
## N-Capas: Ventajas

- Clara separación de las funciones de control de la interfaz y presentación de datos con la lógica de la aplicación
- Reusabilidad de componentes
- Independencia de la interfaz del cliente y la arquitectura de datos
- Mejores posibilidades de balancear la carga
- Uso de protocolos abiertos.



Solucionando una arquitectura N-capas

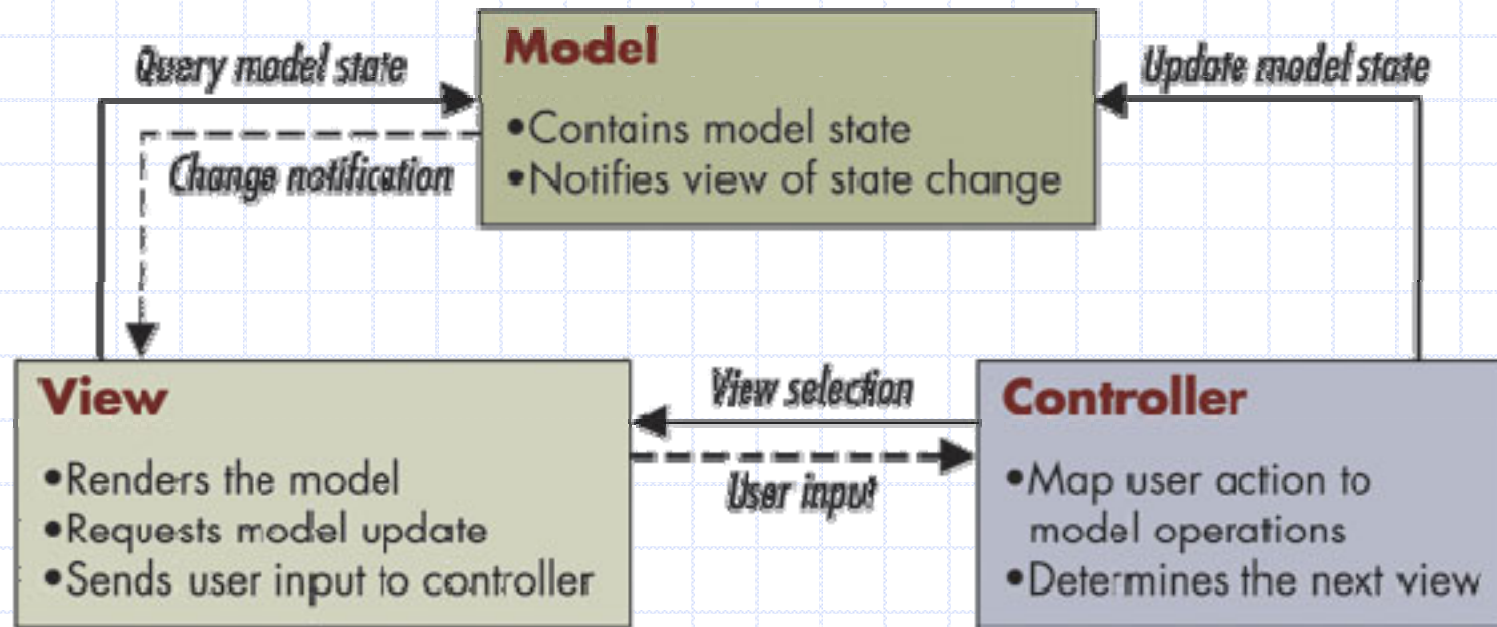
# ARQUITECTURA MODELO VISTA CONTROLADOR





# Modelo MVC

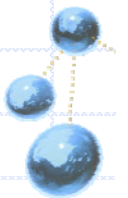
## Uista Arquitectural



# Patrones Arquitecturales

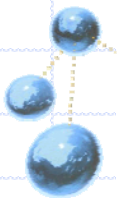
## Model-View-Controller: Ventajas

- Se puede distribuir el esfuerzo de desarrollo hasta cierto punto, tal que los cambios de implementación en una parte de la aplicación Web no requieran cambios en otros.
- Se puede prototipar más fácilmente sus trabajos.
- Facilita la migración de aplicaciones legadas.
- El diseño MVC tiene una estructura organizada que mejora la escalabilidad de soporte (construir aplicaciones más grandes) y fácil de modificar y mantener (debido a la clara separación de tareas).



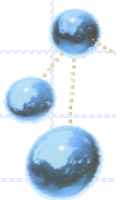
Separación de las aguas a nivel corporativo

# SOA: SERVICE-ORIENTED ARCHITECTURE



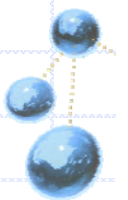
# SOA: ¿Qué es SOA?

- SOA (*Service-Oriented Architecture*) es un estilo arquitectural de diseño que guía todos los aspectos de la creación y uso de *servicios de negocio* a lo largo de su ciclo de vida (desde la concepción hasta el término).
- Un SOA es también una manera de definir y construir una *infraestructura de TI* para permitir que diferentes aplicaciones intercambien datos y participen en procesos de negocio, sin importar el OS o lenguajes de programación subyacentes de las aplicaciones.



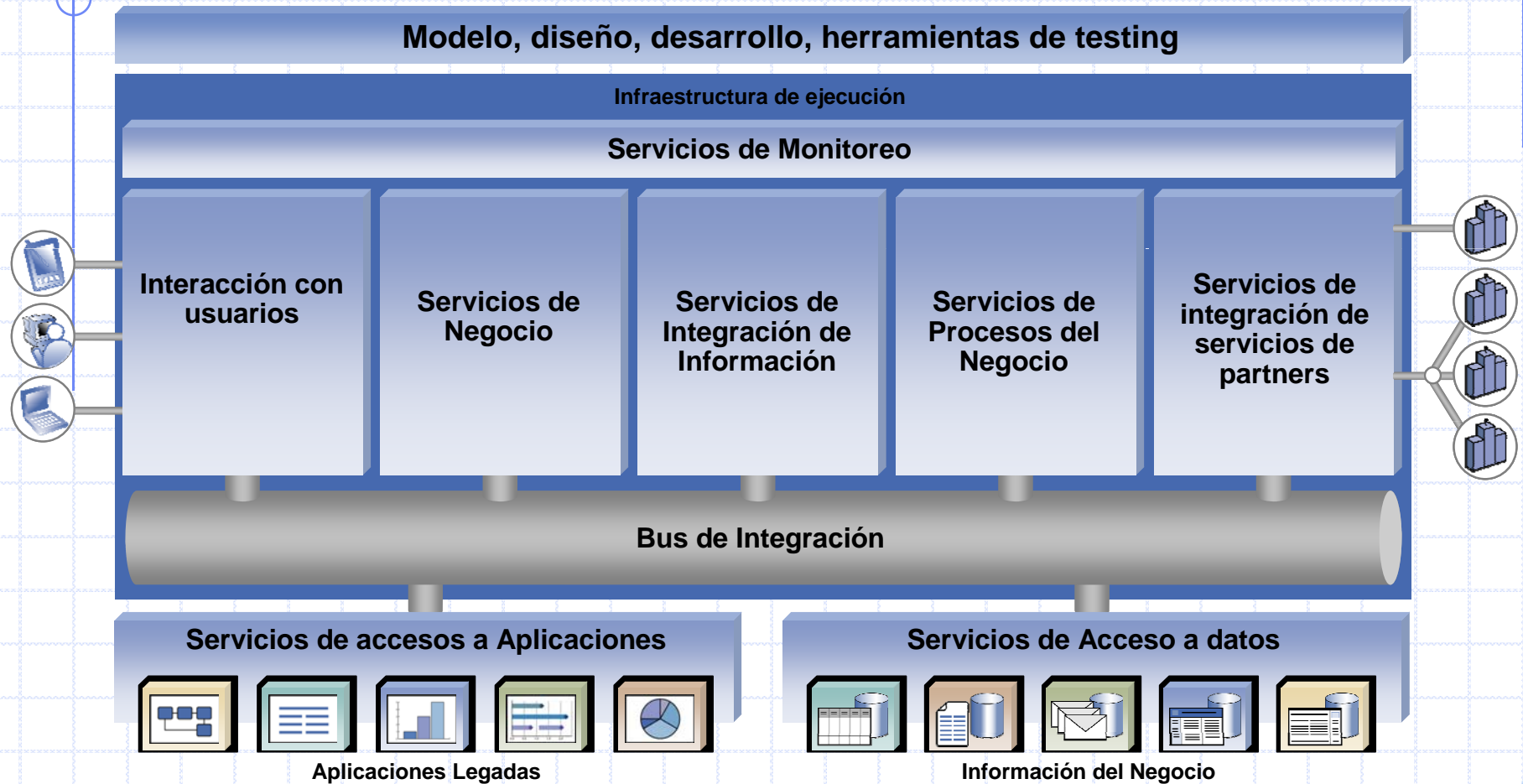
# SOA: Conceptualizando

- ¿Qué es un Servicio?
  - ✓ Un *Servicio de Negocio* es una unidad independiente de trabajo que se realiza repetidamente para apoyar las actividades de procesos de negocio.
- En el ámbito Informático
  - ✓ Un servicio es un lugar en la red que tiene una descripción, legible por una máquina, de los mensajes que recibe y opcionalmente los que retorna.
  - ✓ Un servicio se define en términos de patrón de intercambio de mensajes que soporta.



# Arquitectura SOA Coopeuch

## Infraestructura tecnológica





# Plataforma tecnológica SOA

## Rol del Bus de Integración

### Mediación de Servicios

- ✓ Ruteo y transformación de mensajes y eventos.

### Servicios de Eventos

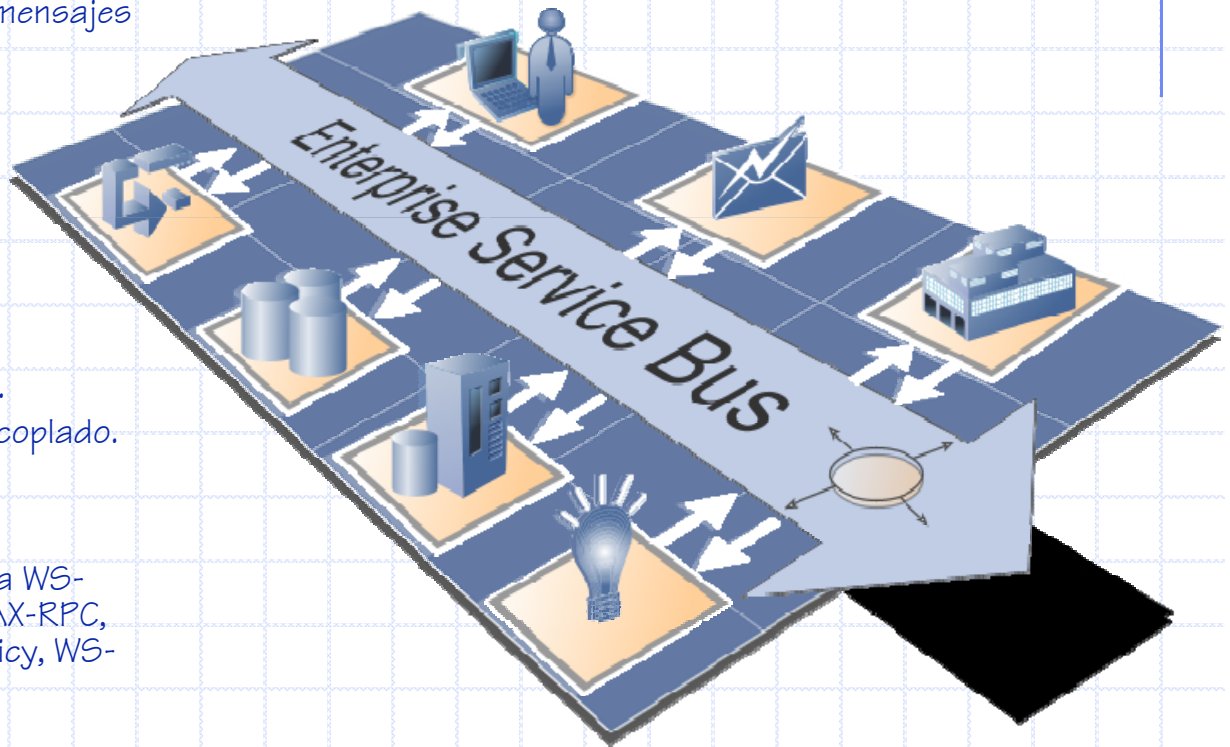
- ✓ Publicación y suscripción.

### Servicios de Transporte.

- ✓ Síncrono / Asíncrono.
- ✓ Persistente/No-Persistente.
- ✓ Poco acoplado/Altamente acoplado.

### Base estándar

- ✓ HTTP/HTTPS con opción para WS-Reliable Messaging, JMS, JAX-RPC, SOAP, WS-Security, WS-Policy, WS-Addressing.

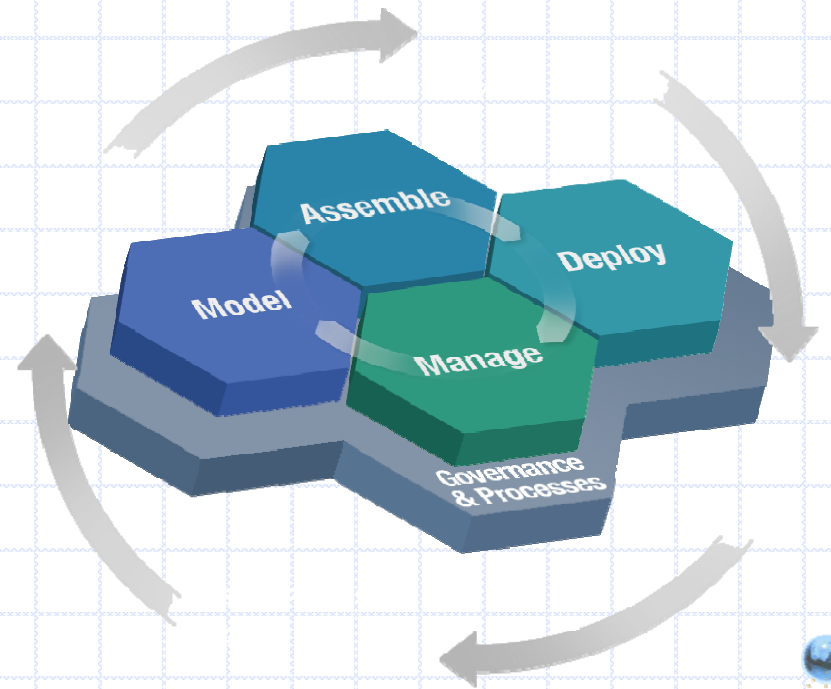




# Arquitectura orientada a Servicios

## Ciclo de vida

- La operación de la arquitectura orientada a servicios está definida por el siguiente ciclo de vida:
  - **Modelamiento:** Identificar los procesos de negocio importantes y los servicios que los apoyan.
  - **Ensamblaje:** Se orquestan los servicios para construir el proceso.
  - **Puesta en marcha:** Se pone en marcha el proceso apoyado por una interacción con el usuario.
  - **Gestión:** Se monitorea el desempeño de los servicios y del proceso (desde el punto de vista del negocio).



PREGUNTAS?

