



Arquitectura de Software

Elementos de descripción arquitectónica

UML y arquitectura

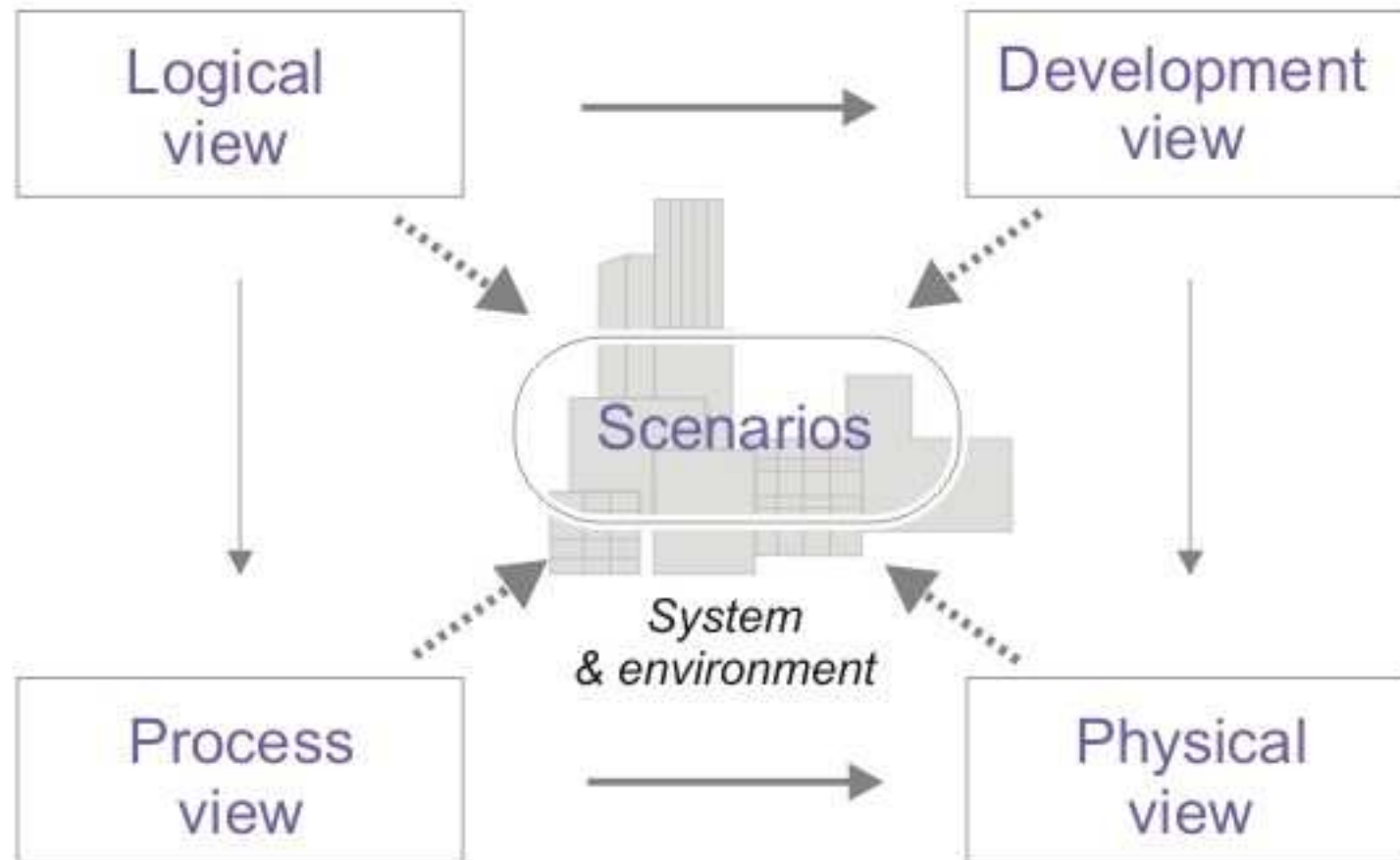
Definiciones

- Modelo del sistema
 - Es una representación abstracta del sistema creada para comprender su estructura y operación.
- Descripción arquitectónica
 - Es un documento, producto u otro artefacto utilizado para registrar o comunicar la arquitectura del sistema.
(IEEE Architecture Working Group)
- Vista
 - Representación de un conjunto cohesivo de elementos arquitectónicos dirigidos a un determinado grupo de accionistas de la arquitectura del sistema.

Modelo 4+1

- Modelo desarrollado por Philippe Kruchten (1995) para describir arquitecturas de software basado en múltiples vistas concurrentes.
- Se presenta a partir de 4 vistas que describe la arquitectura presentando aspectos bien definidos de la misma y las decisiones tomadas en el proceso de diseño arquitectónico.
- Adicionalmente, esas 4 vistas son ilustradas a partir de un conjunto de casos de uso representativos que constituyen la quinta vista del modelo.

Modelo 4+1



Modelo 4+1

- Vista **Lógica**, de **Diseño** o de **Módulos**
 - Describe el modelo desde los elementos de diseño y las colaboraciones entre ellos.
 - Descompone el sistema en un conjunto de abstracciones tomadas generalmente del dominio del problema en la forma de objetos, clases o módulos.
 - Permite identificar mecanismos y elementos de diseño comunes a las partes del sistema.
 - Explora principios de diseño como abstracción, encapsulamiento y herencia.

Modelo 4+1

- Vista **Desarrollo** o de **Componentes**
 - Describe la organización estática en su ambiente de desarrollo.
 - Presenta como se “empaquetan” los elementos presentados en la vista lógica a partir de librerías, DLLs o subsistemas: Componentes.
 - Presenta además como se relacionan esos componentes en tiempo de ejecución a partir de interfaces, puertos, protocolo: Conectores.

Modelo 4+1

- **Vista Procesos**

- Describe aspectos de sincronización, concurrencia, distribución, integridad, tolerancia a fallos.
- Describe mecanismos para cumplir con requerimientos no funcionales como rendimiento o disponibilidad.
- Ubica los elementos de la vista lógica en su ambiente de ejecución: En que proceso o hilo de control se ejecuta determinada funcionalidad encapsulada en un objeto.

Modelo 4+1

- Vista **Física** o de **Despliegue**
 - Describe las relaciones entre el software y el hardware y aspectos de distribución.
 - Se ocupa fundamentalmente de requerimientos no funcionales tales como disponibilidad, confiabilidad, rendimiento y escalabilidad.
 - Ubica los elementos presentados en la vista de componentes sobre la infraestructura del sistema: hardware.

Modelo 4+1

- Vista de **Casos de Uso o Escenarios**
 - Describe el comportamiento del sistema, basado en los principales casos de uso, visto desde la perspectiva de los usuarios, analistas y testers.
 - Integra las otras vistas ilustrándolas a partir de escenarios concretos representativos.
 - Esta vista en muchos casos es redundante con lo presentado en otras vistas, sin embargo suele ser útil para descubrir determinados elementos arquitectónicos mientras se diseñan las otras vistas y como mecanismo de validación luego que el diseño arquitectónico se ha completado.

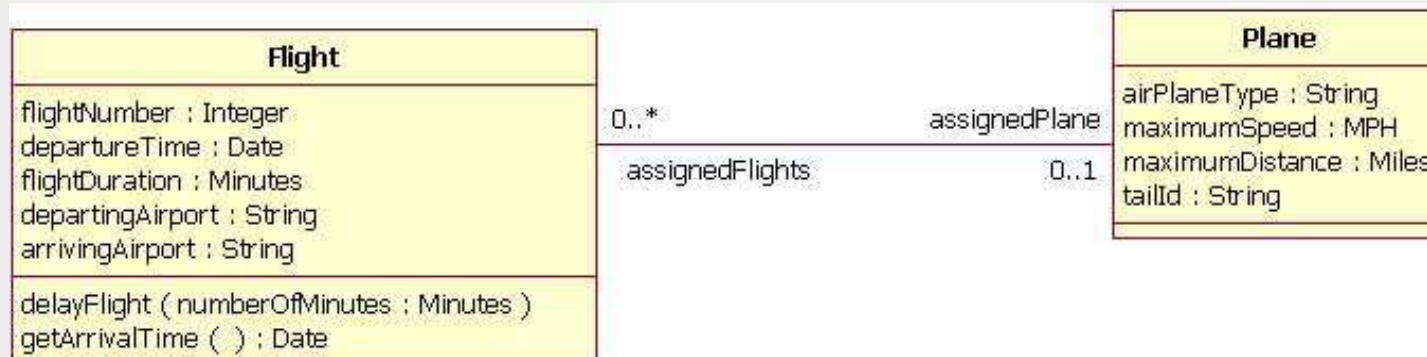
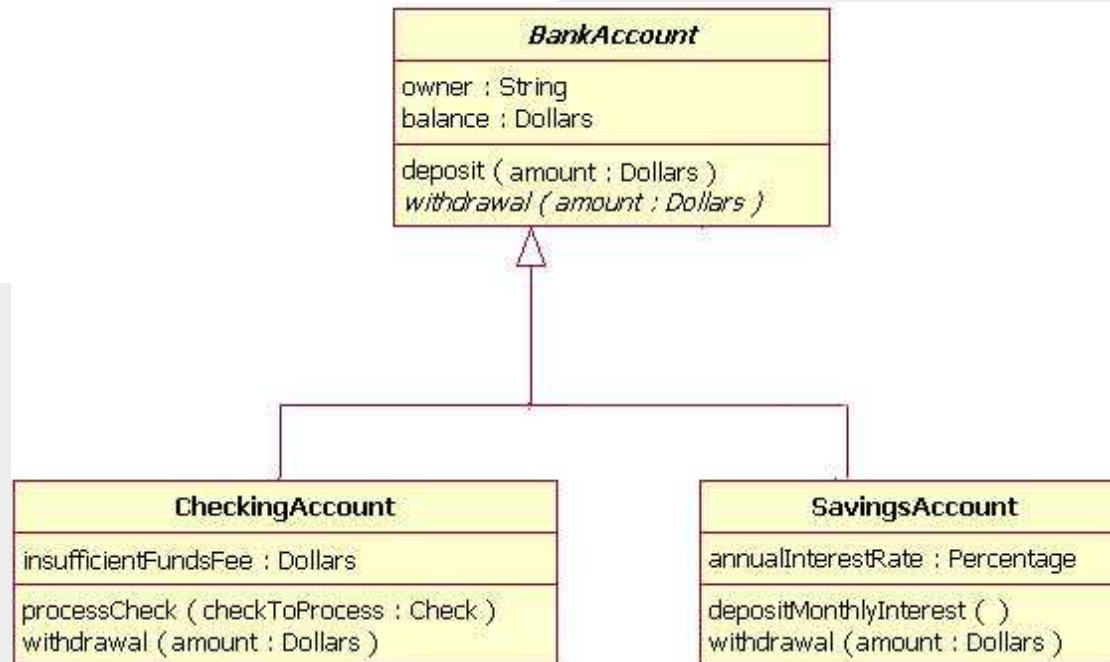
UML (Unified Modeling Language)

- Creado en 1996 por Booch, Rumbaugh, Jacobson de Rational Software Corp.
- Lenguaje estándar para el modelado de sistemas de software.
- UML es un lenguaje de modelado de carácter general que incluye una notación grafica utilizada para crear modelos abstractos de un sistema.
- UML es útil para especificar, visualizar, construir y documentar sistemas de software.
- Mapeable con las construcciones de los lenguajes orientados a objetos.

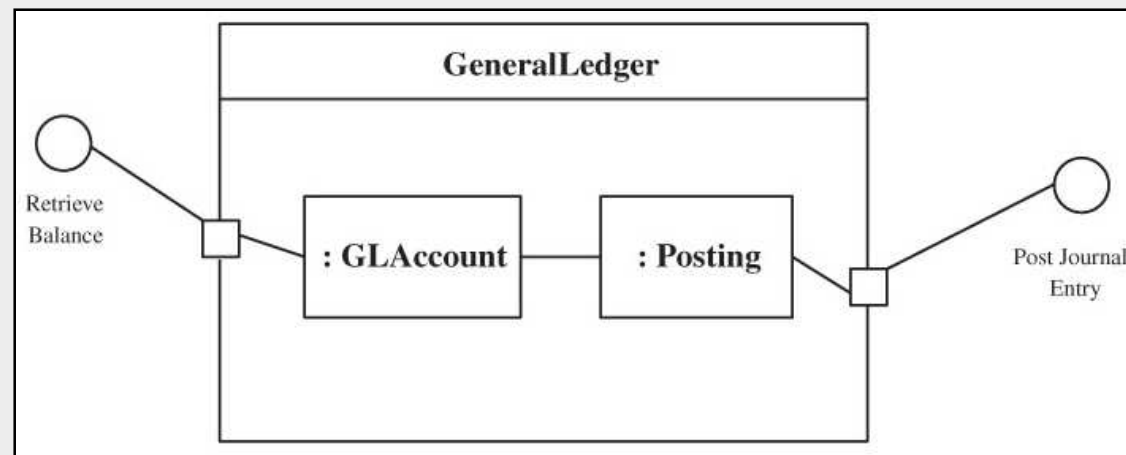
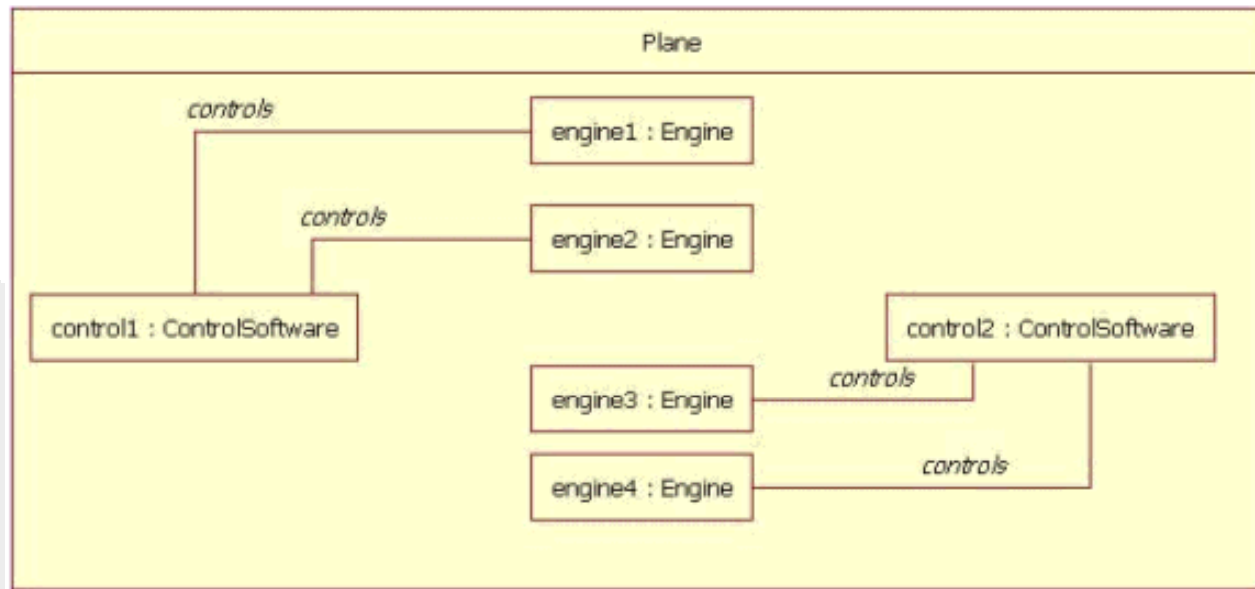
Vista Lógica

- **Elementos**
 - Clases
 - Objetos
 - Interfaces
 - Paquetes
 - Colaboraciones
- **Relaciones**
 - Asociaciones
 - Dependencia
 - Generalización
 - Realización

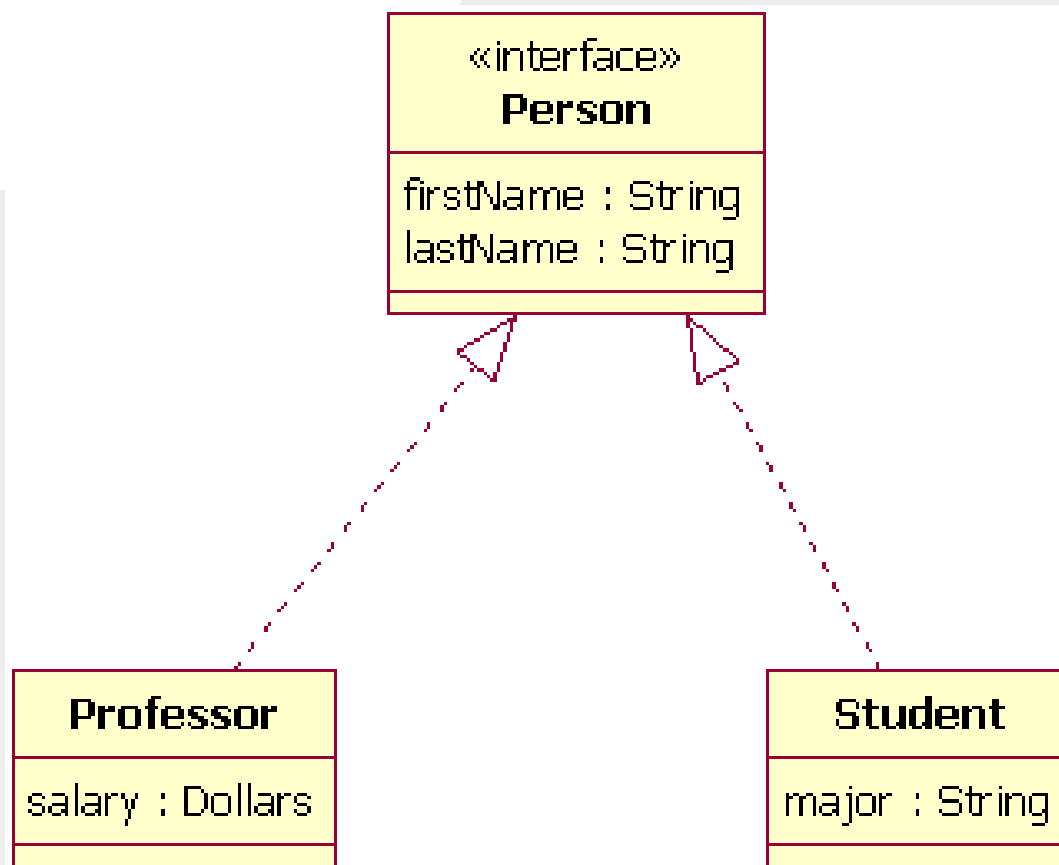
Vista Lógica



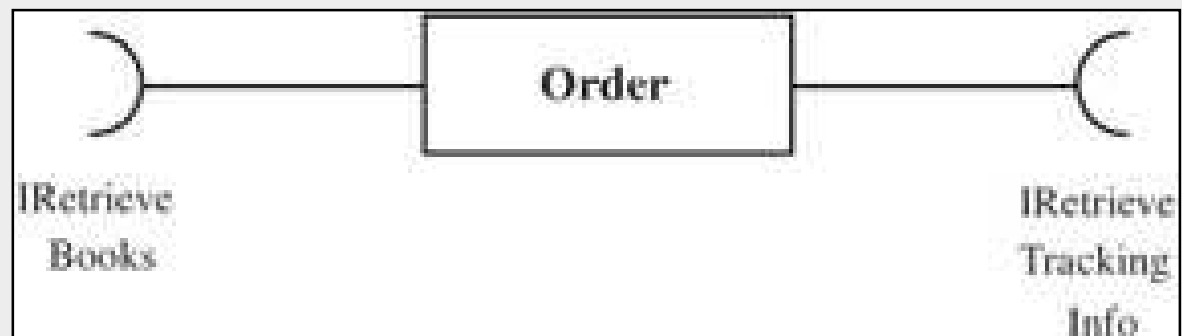
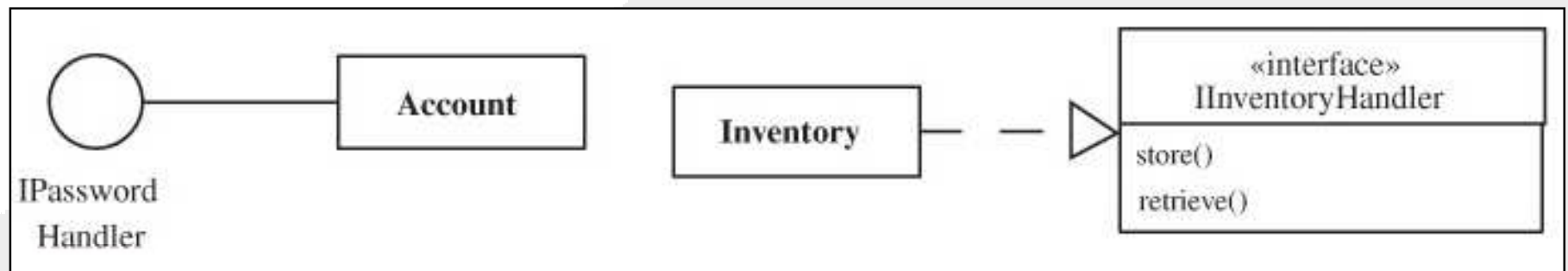
Vista Lógica



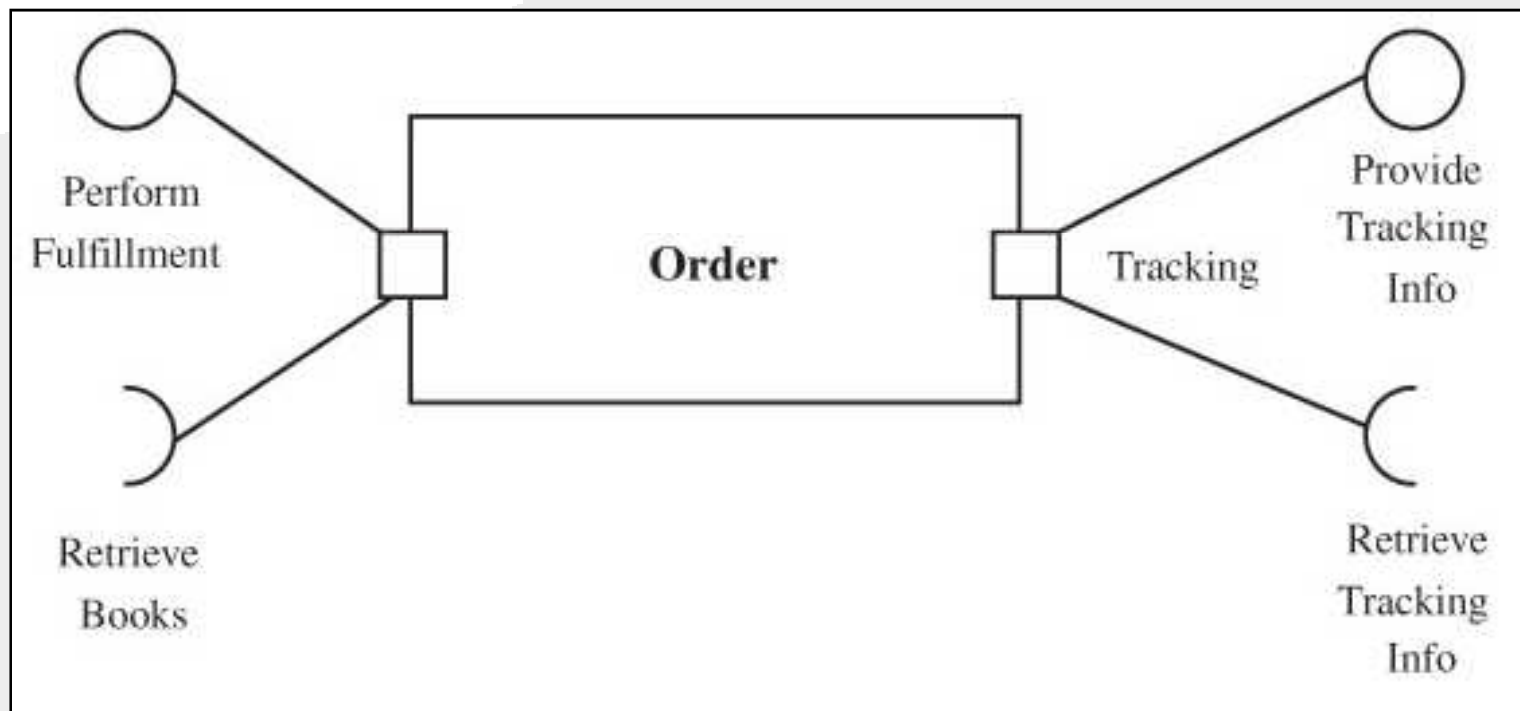
Vista Lógica



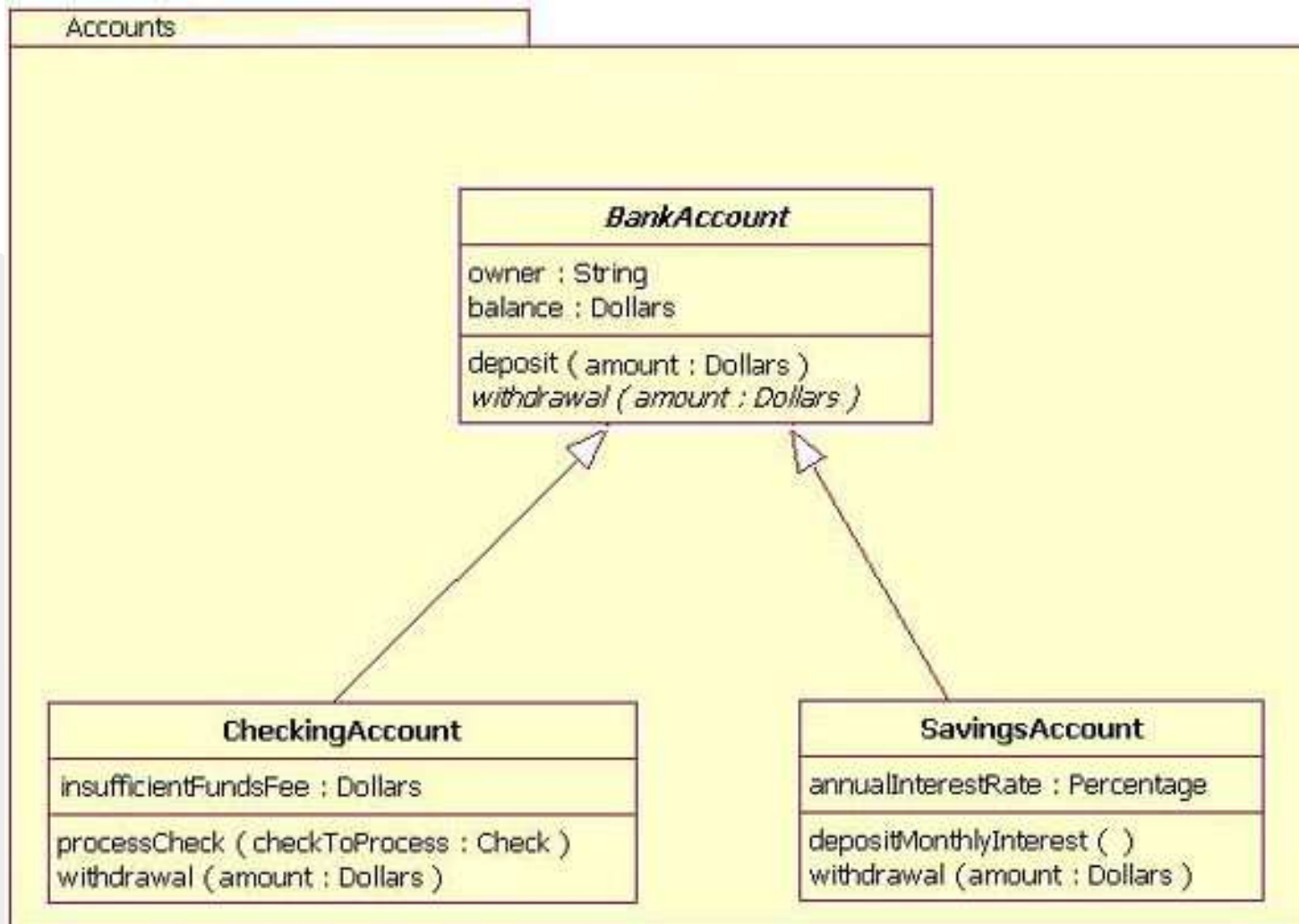
Vista Lógica



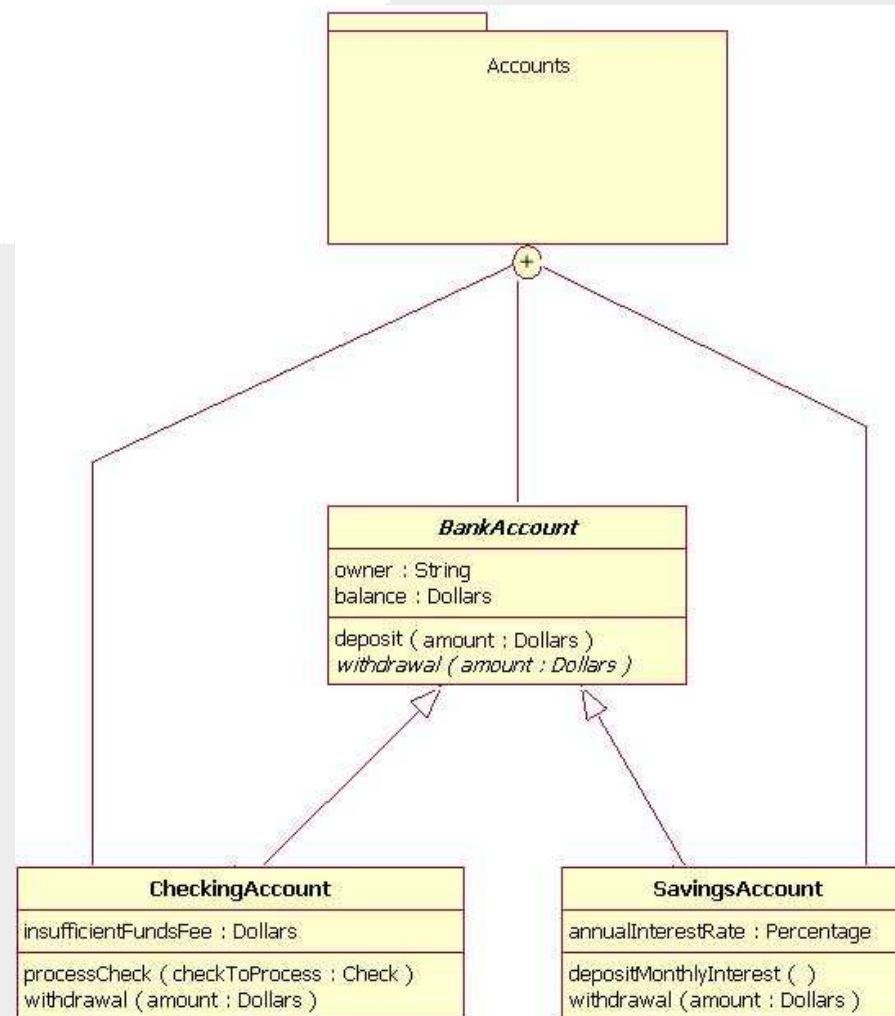
Vista Lógica



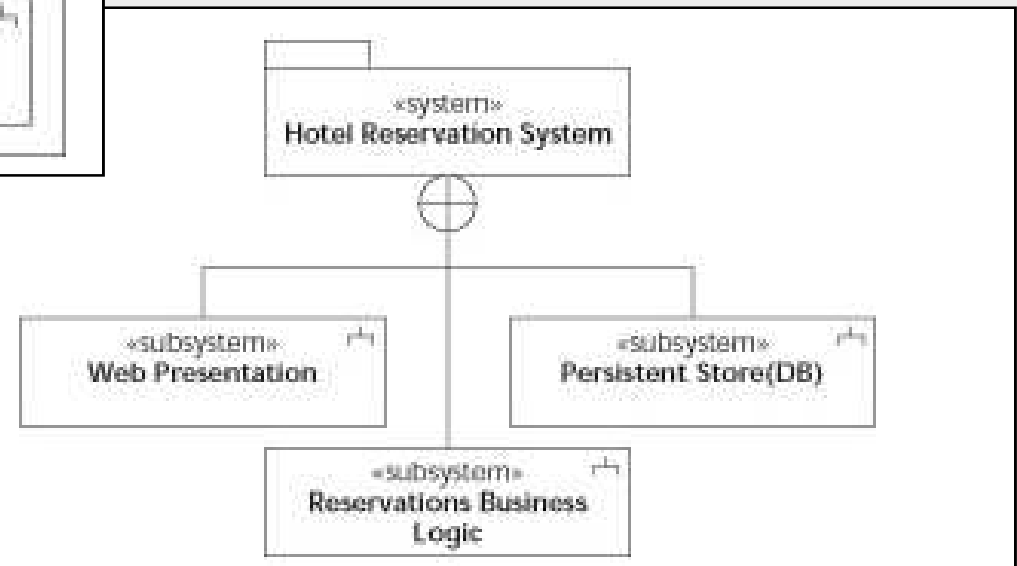
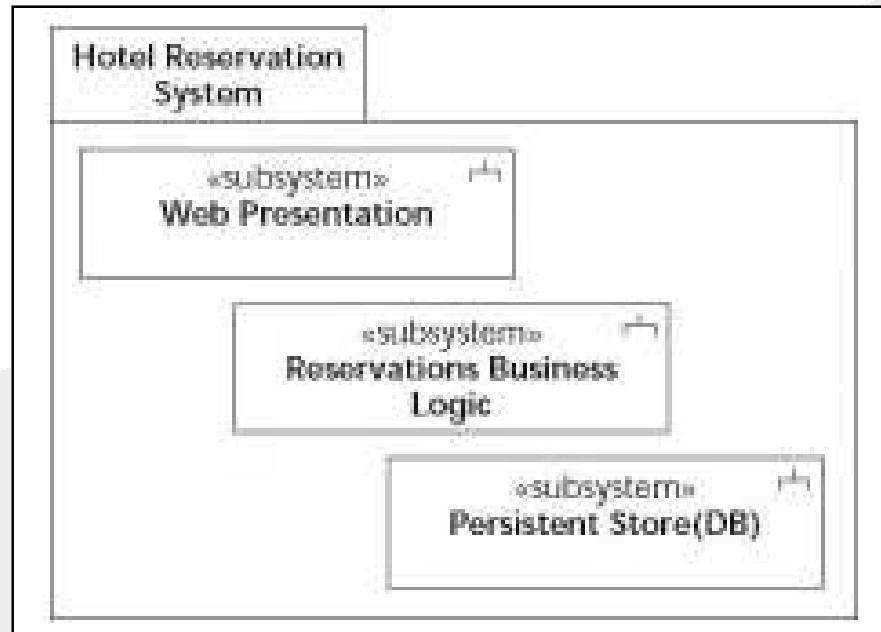
Vista Lógica



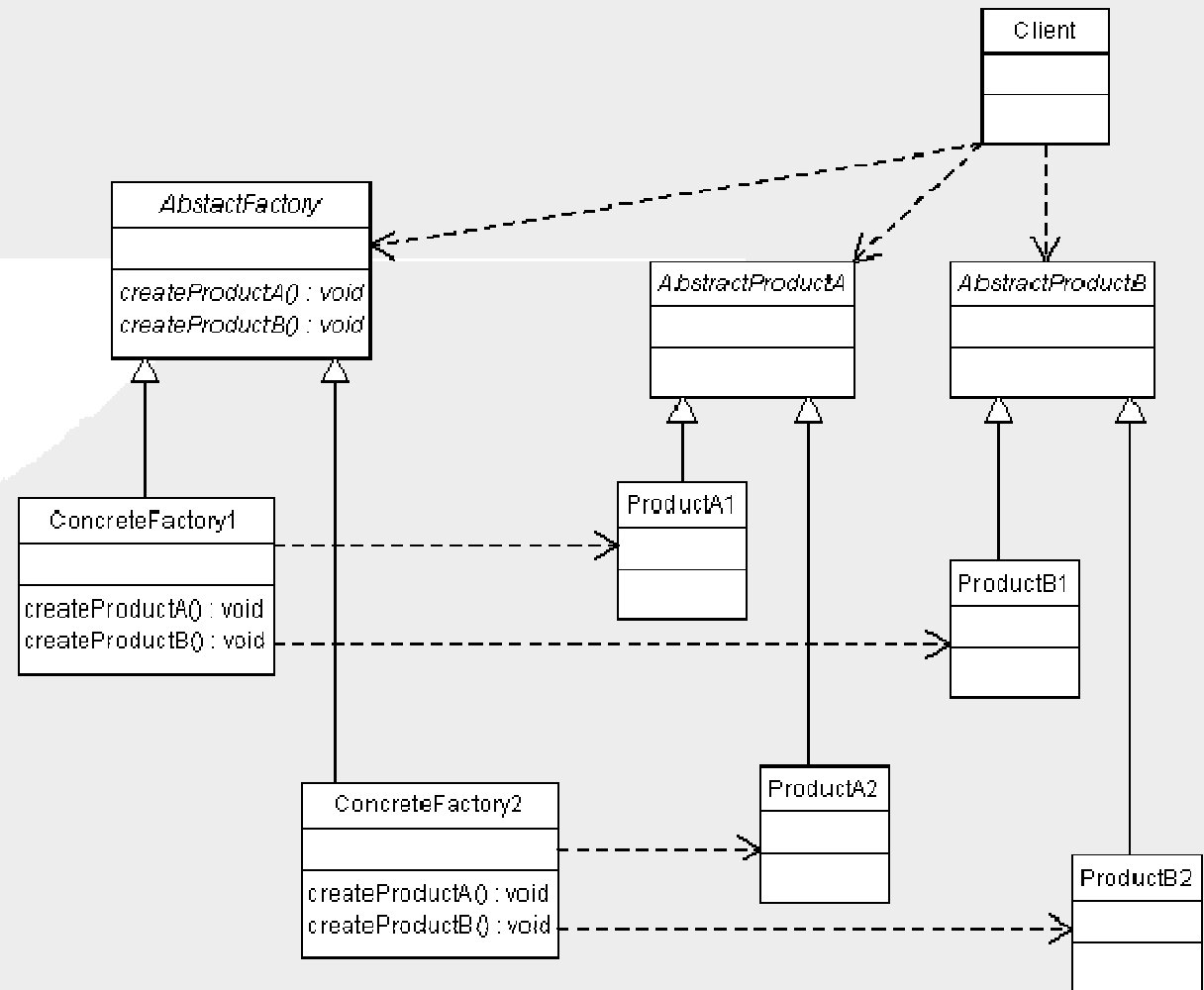
Vista Lógica



Vista Lógica

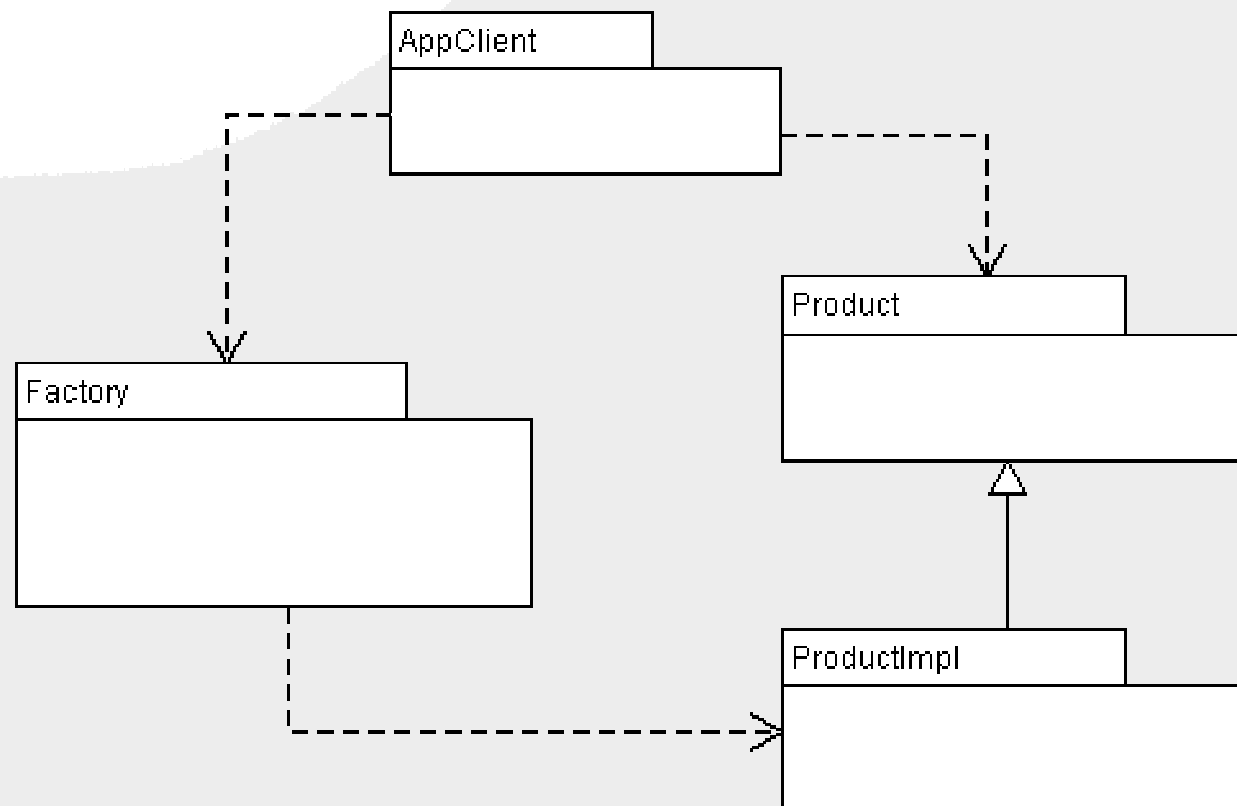


Vista Lógica



A partir del diagrama de clases de la figura, bosqueje un diagrama de paquetes con las relaciones correspondientes teniendo en cuenta que la fábrica pertenece al paquete **Factory**, que **AbstractProductA** y **AbstractProductB** pertenecen al paquete **Product** mientras que sus derivados pertenecen al paquete **ProductImpl** y finalmente la entidad **Client** pertenece al paquete **AppCliente**

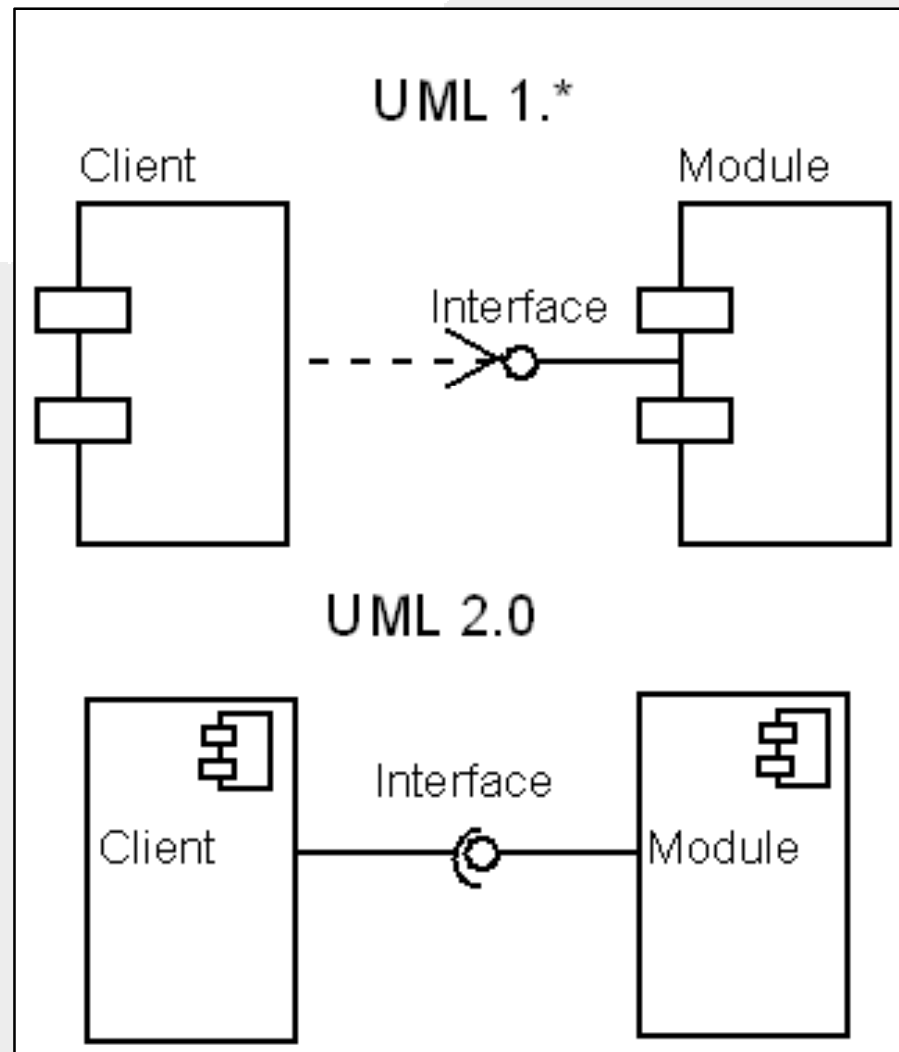
Vista Lógica



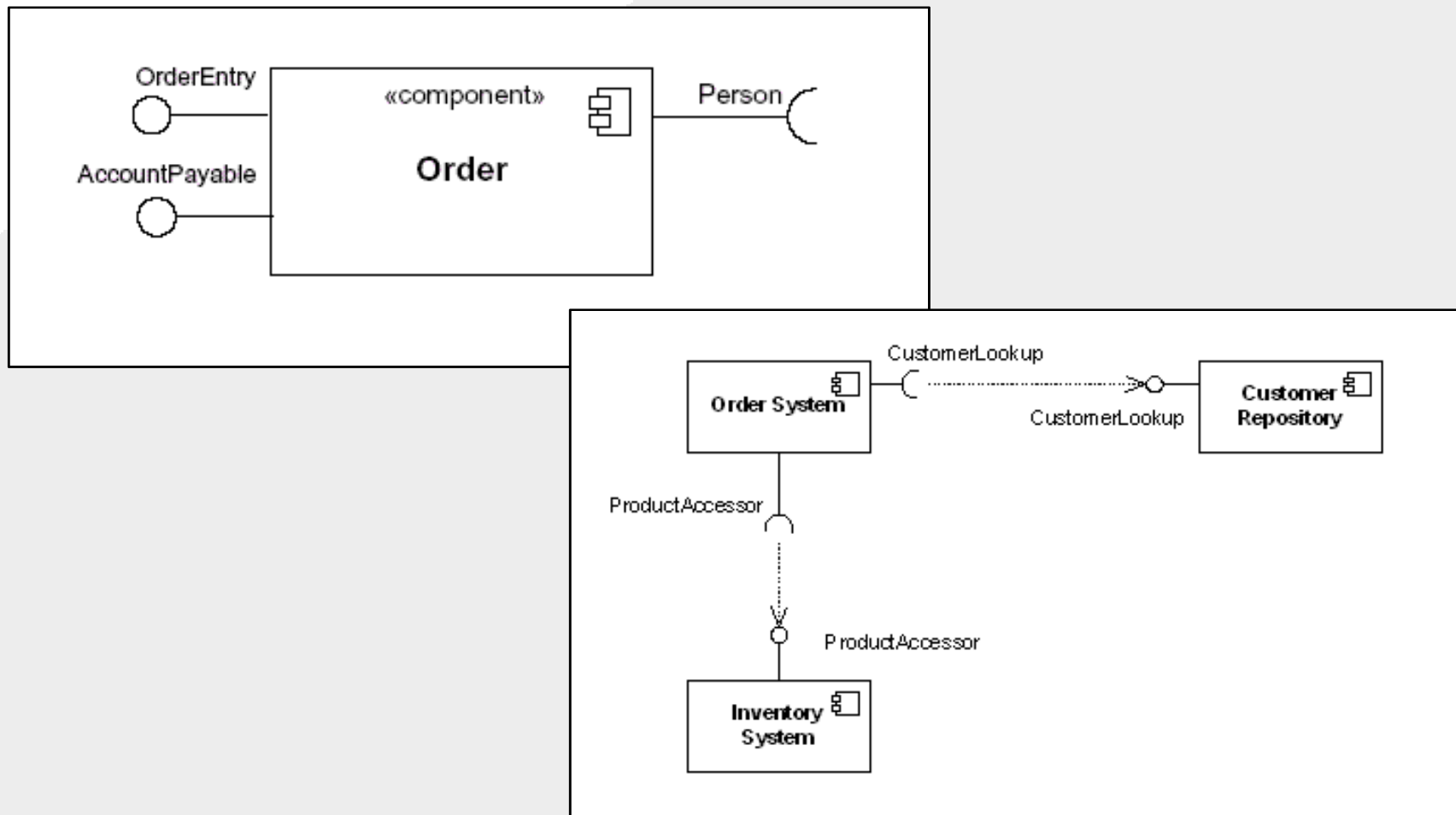
Vista Componentes

- **Elementos**
 - Componentes
 - Interfaces
 - Puertos
- **Relaciones**
 - Dependencia
 - Generalización
 - Realización
 - Dependencia (<<delegate>>)

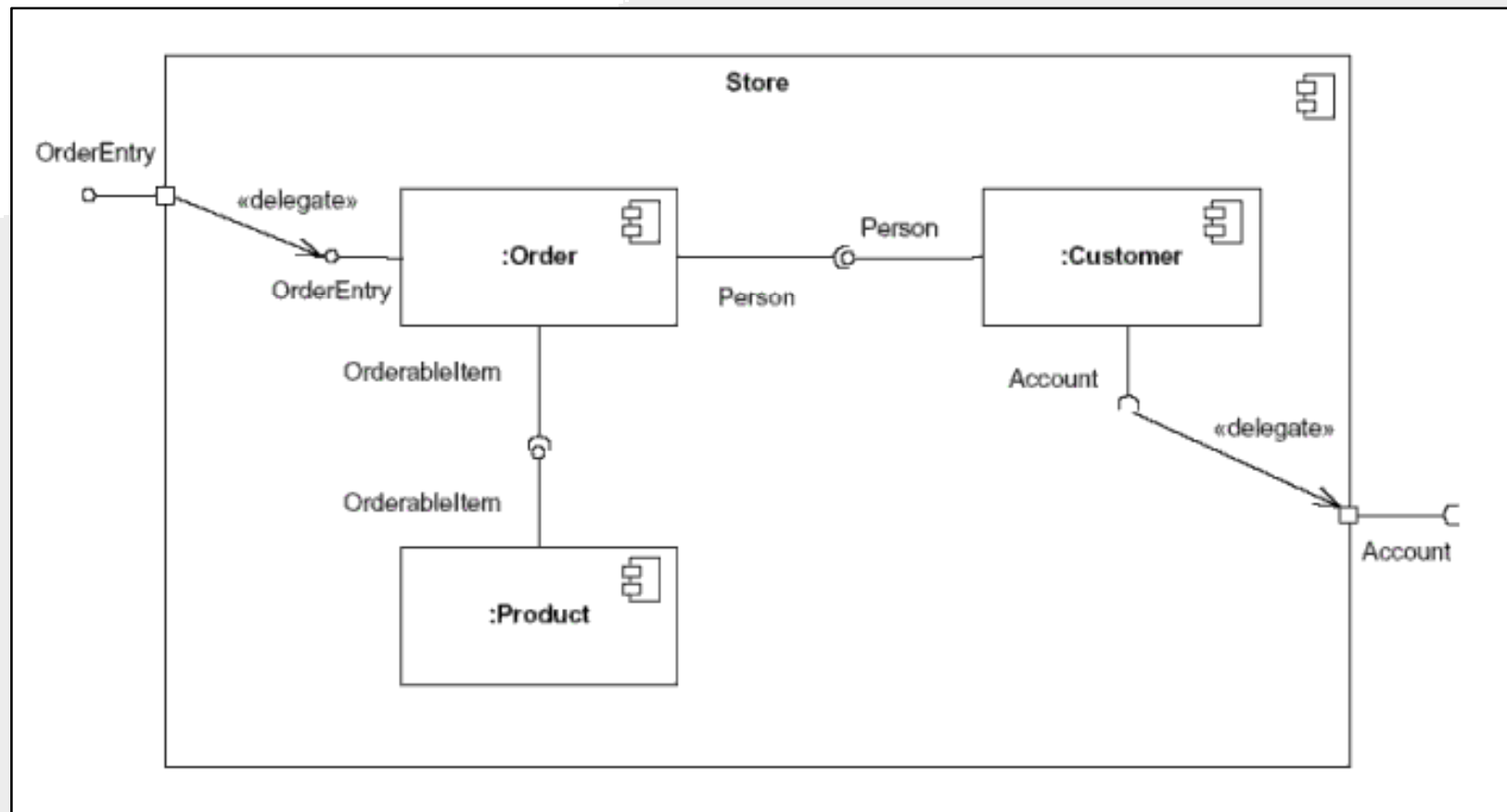
Vista Componentes



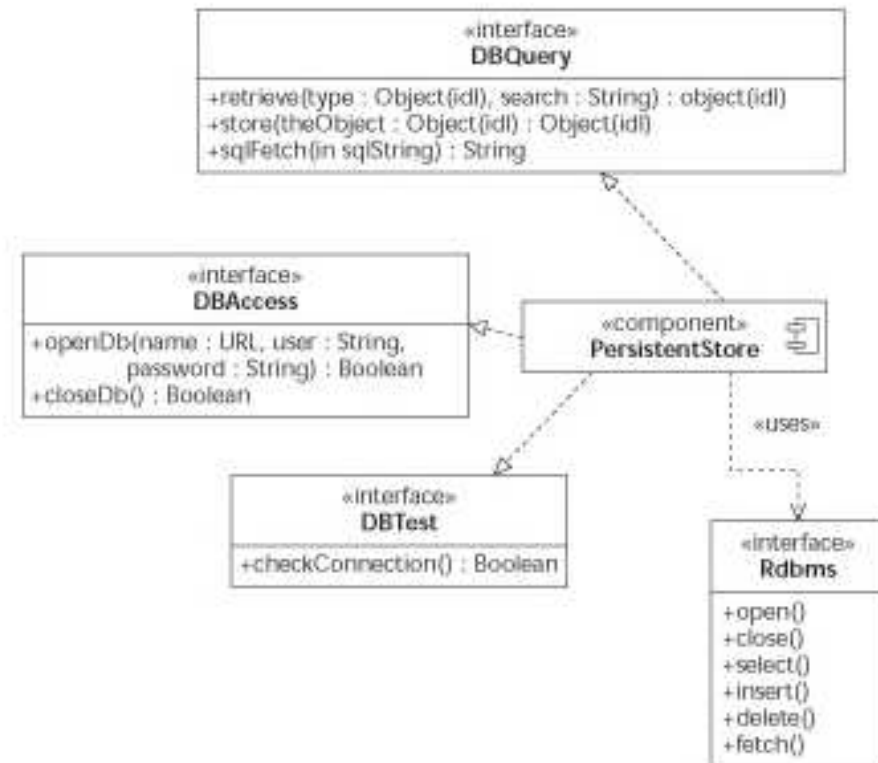
Vista Componentes



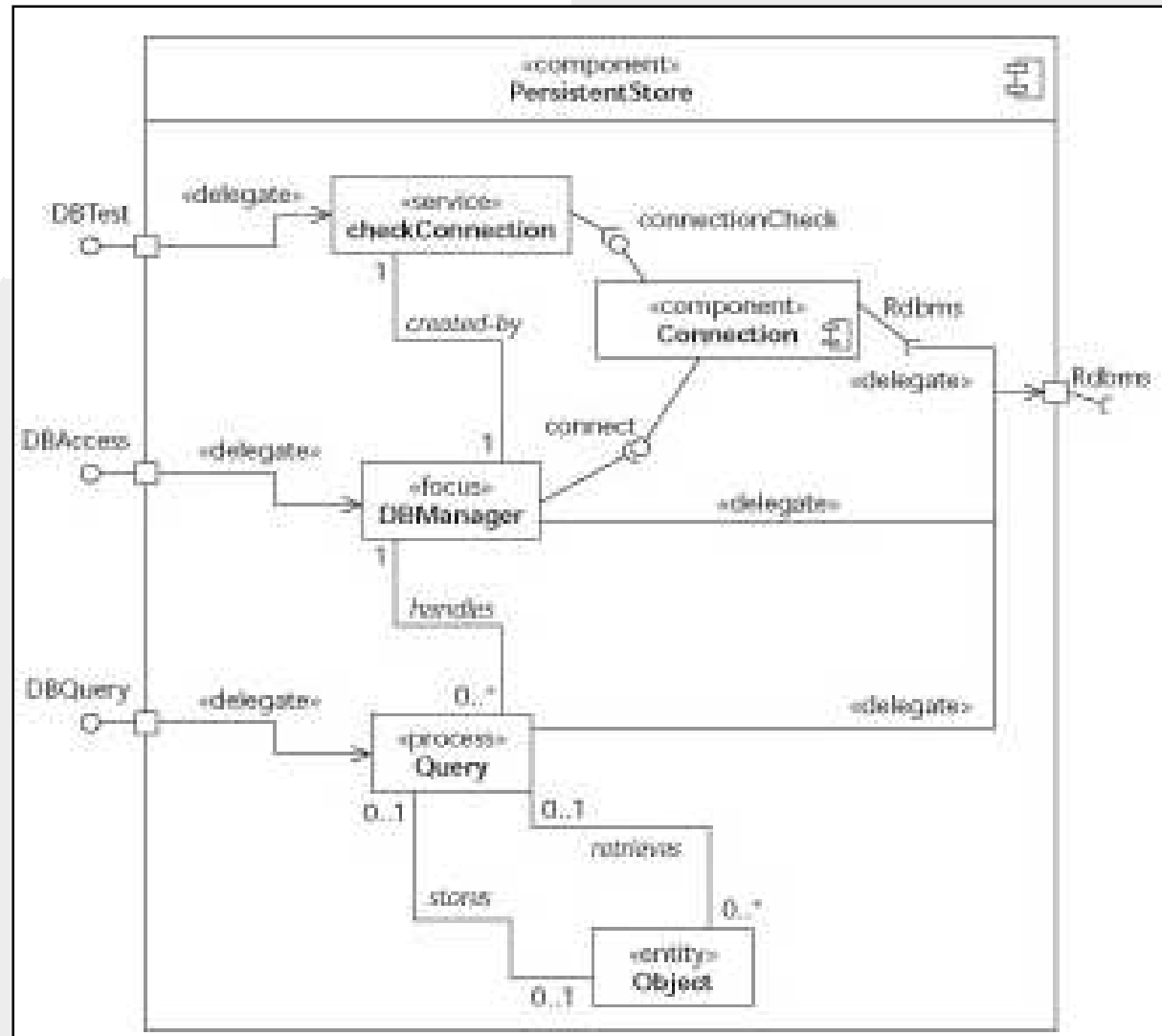
Vista Componentes



Vista Componentes



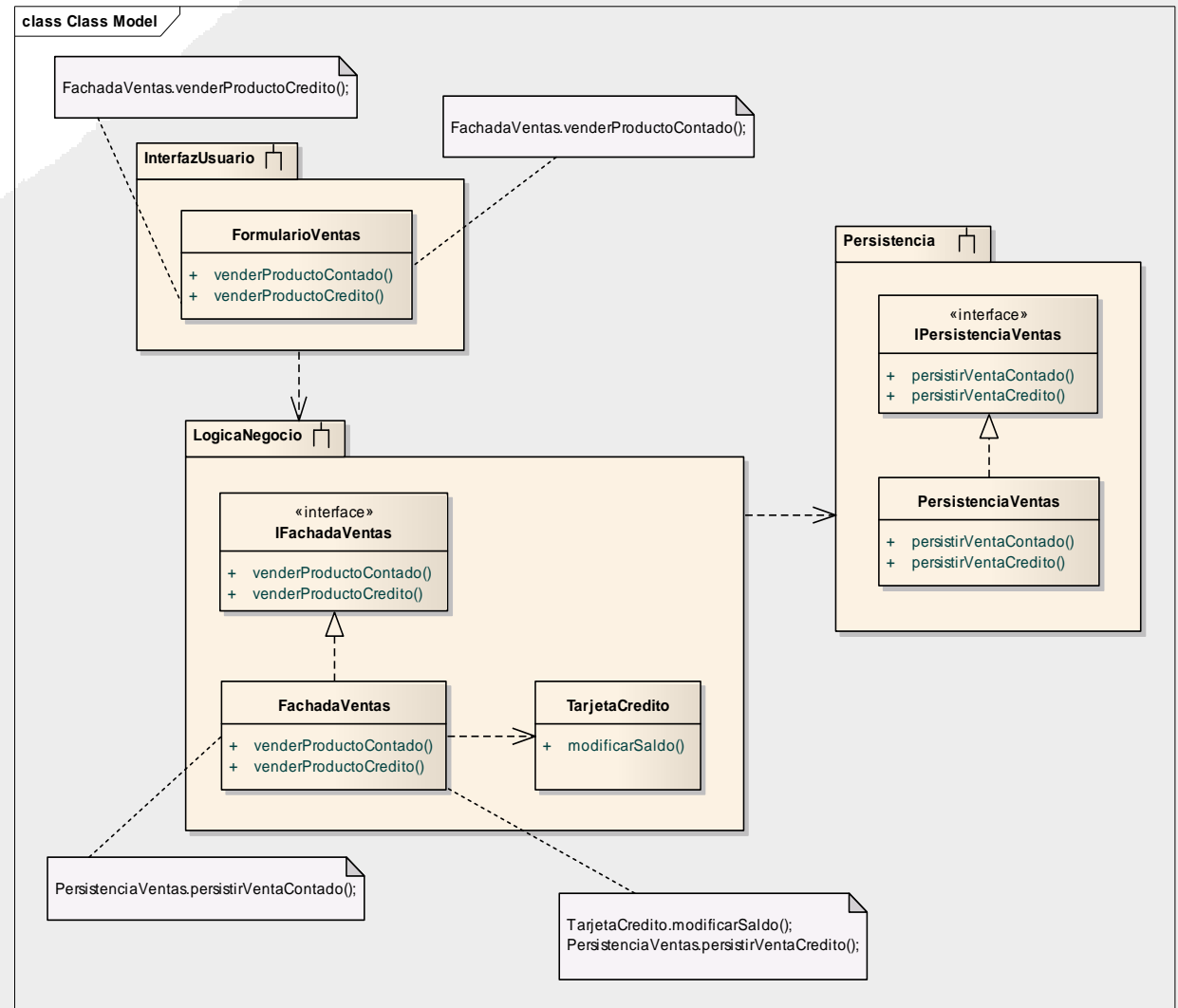
Vista Componentes



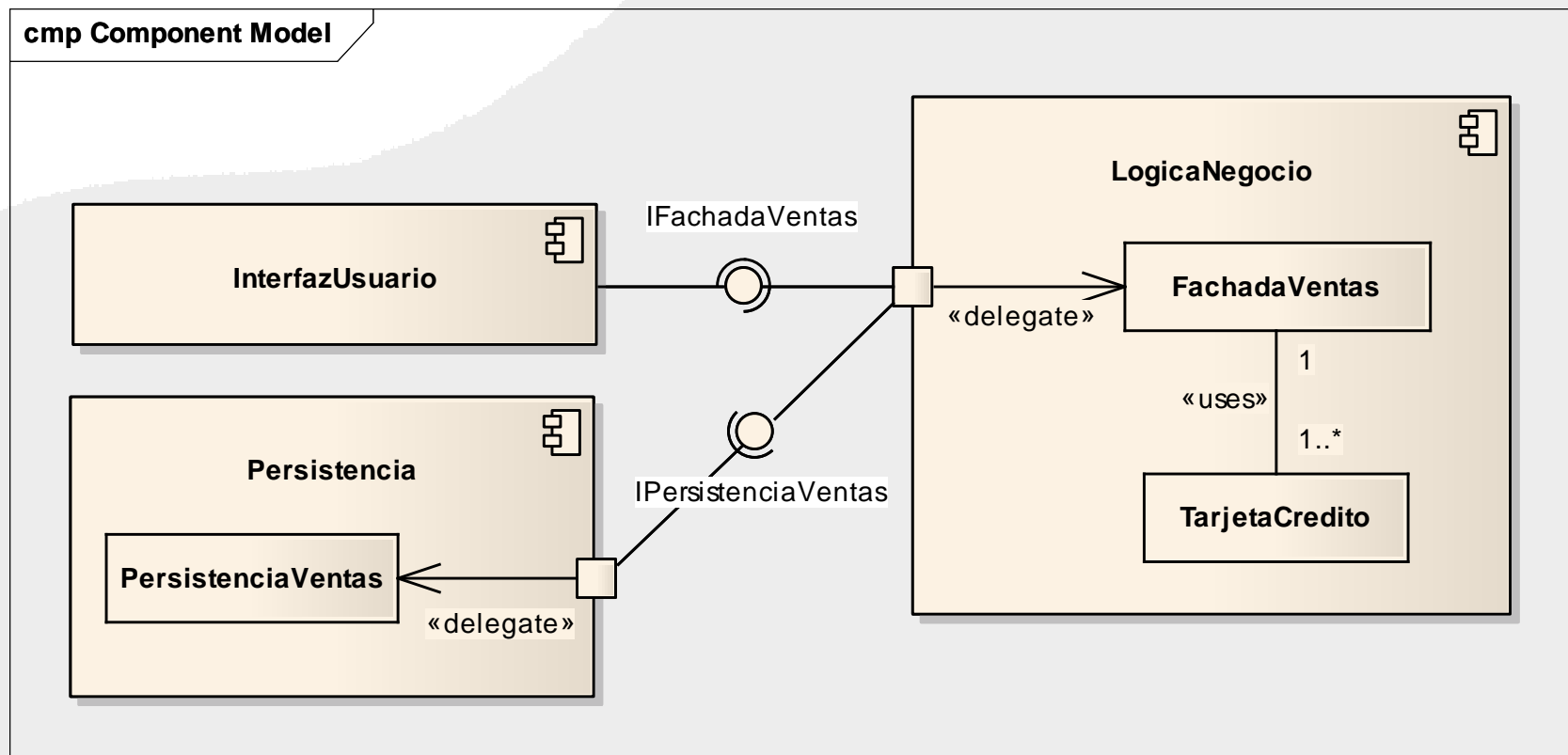
Ejercicio...

A partir del siguiente diagrama de paquetes diseñe un diagrama de componentes UML asumiendo que cada subsistema se corresponde con un único componente incluyendo:

- Componentes
- Interfaces requeridas y provistas
- Estructura interna de los componentes que se corresponden con los paquetes de LogicaNegocio y Persistencia.



...solución



Puertos

- Proveen un nivel adicional de información sobre la interacción entre un componente y su entorno (Opcionales).
- Se utilizan para:
 - Relacionar una interfaz provista o requerida de un componente con su vista interna (partes) mediante la asociación <<delegate>>.
 - Agrupar lógica o físicamente un conjunto de interfaces a partir de un nombre o a partir de sus características de interacción (unidireccional/bidireccional).

Vista Procesos

- **Elementos**

- Clases (estereotipos) <<process>> <<thread>>
- Objetos (estereotipos)
- Estados (diagramas de maquinas de estado)
- Tareas (diagramas de actividad)

- **Relaciones**

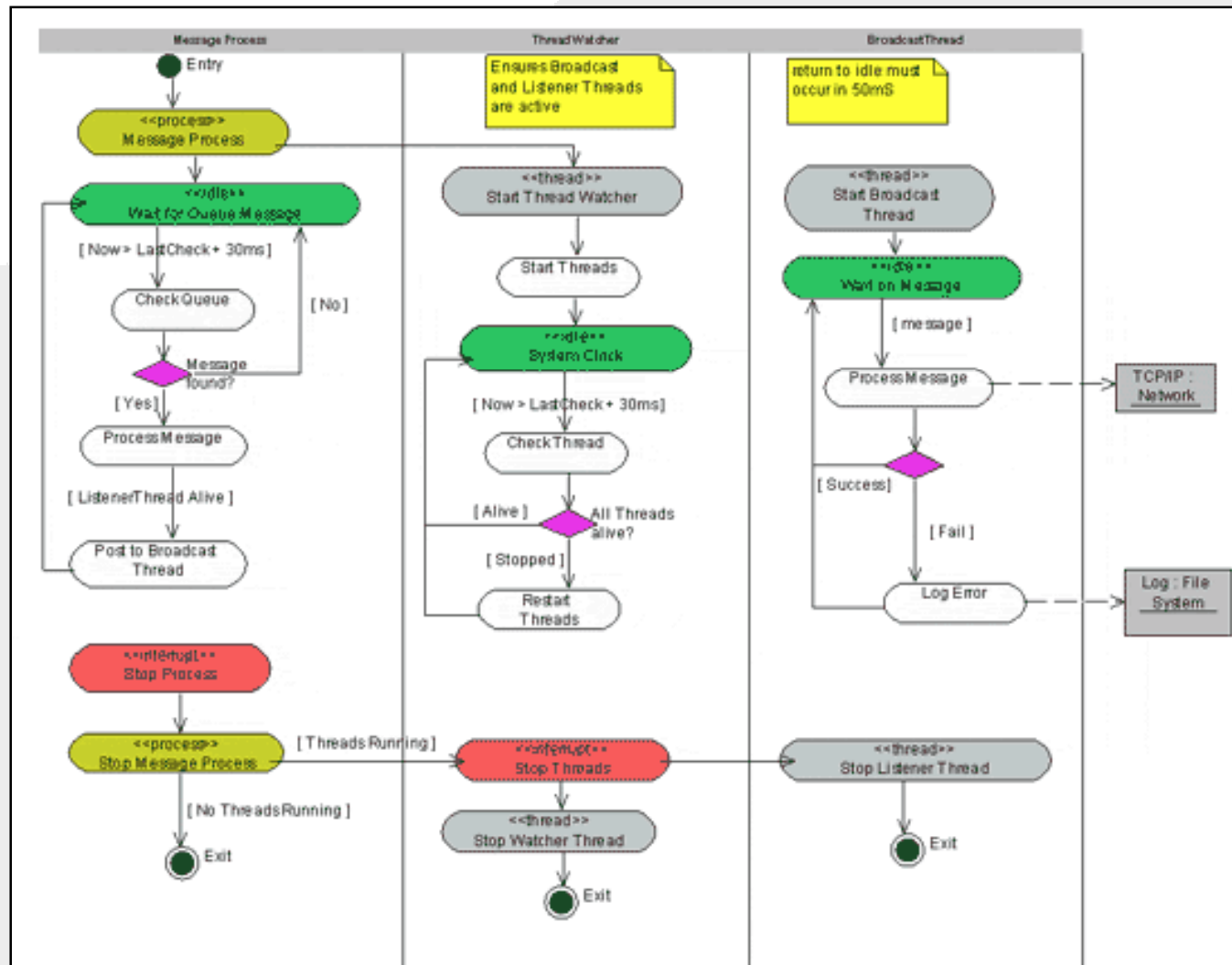
- Asociaciones
- Dependencia
- Generalización

Vista Procesos

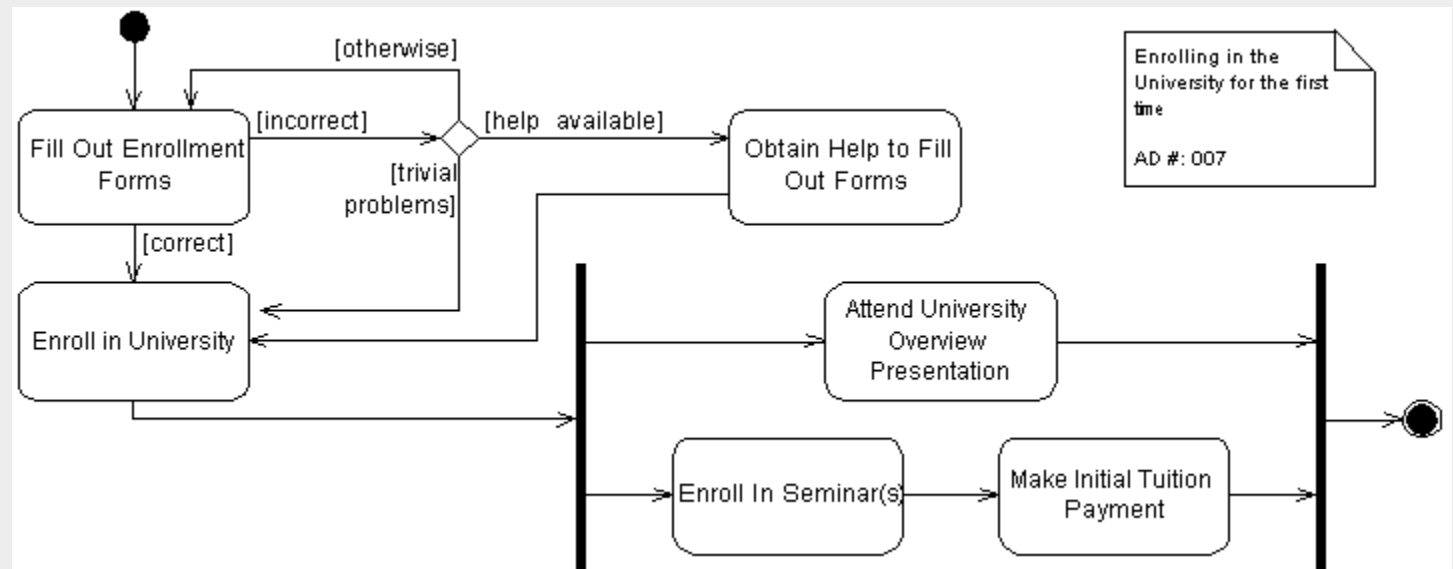
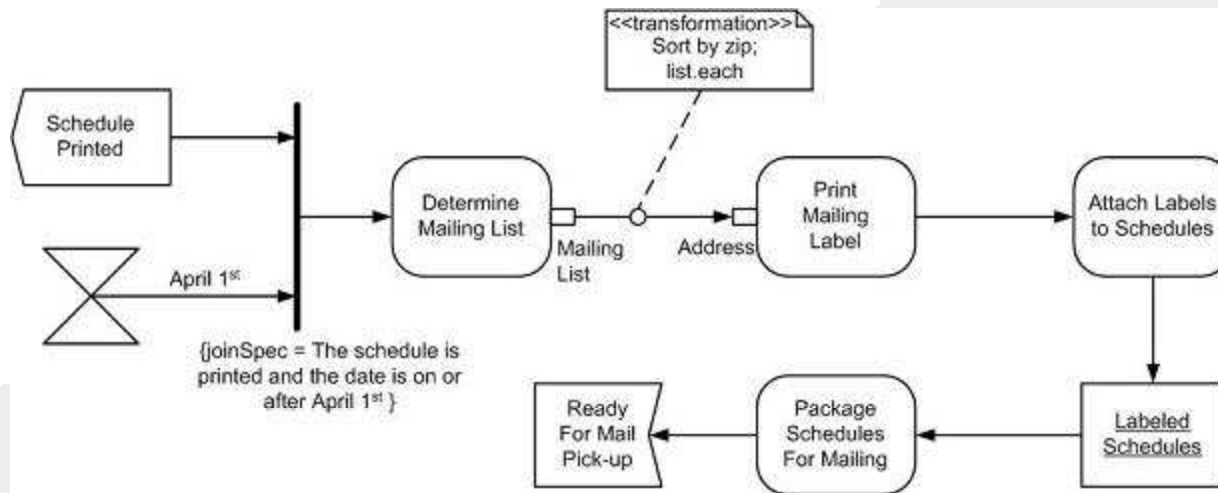
| |
|--|
| <<Process>> Manejador De Comunicaciones |
| |

| |
|---------------------------------|
| <<Thread>> StreamAudioPlayer |
| |
| |

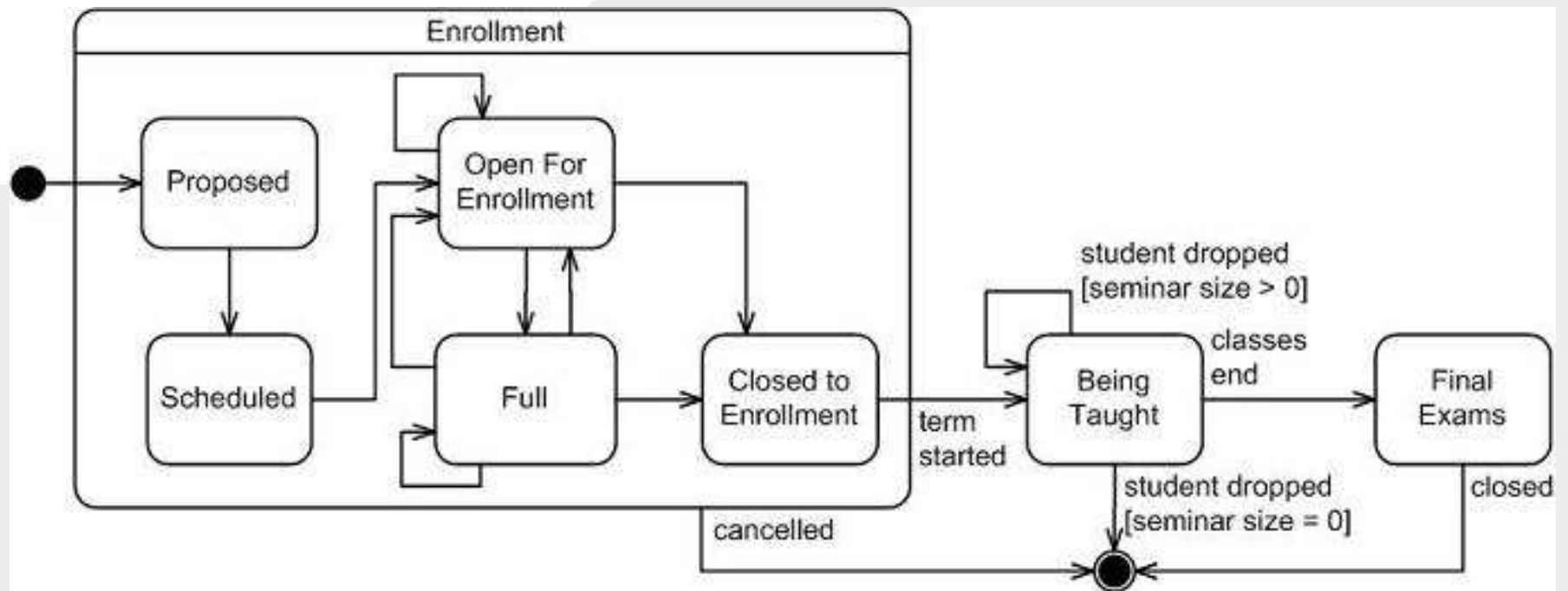
Vista Procesos



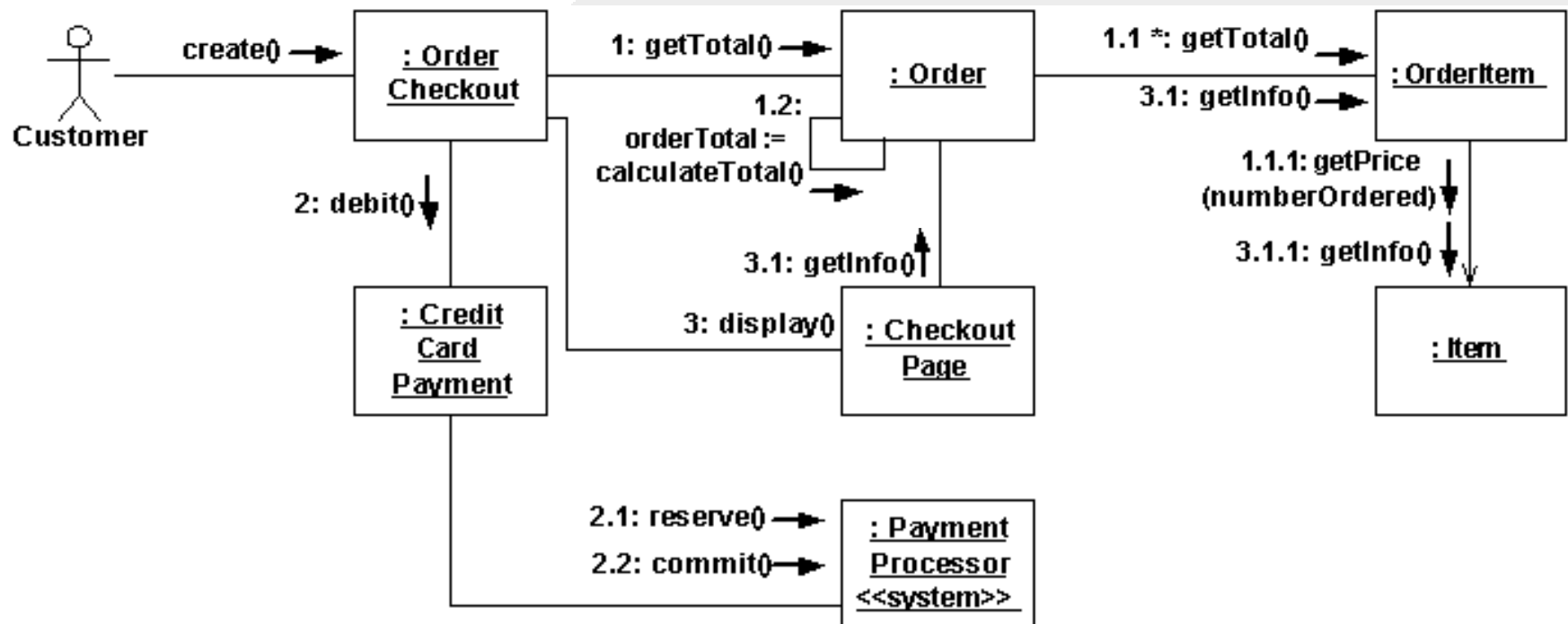
Vista Procesos



Vista Procesos

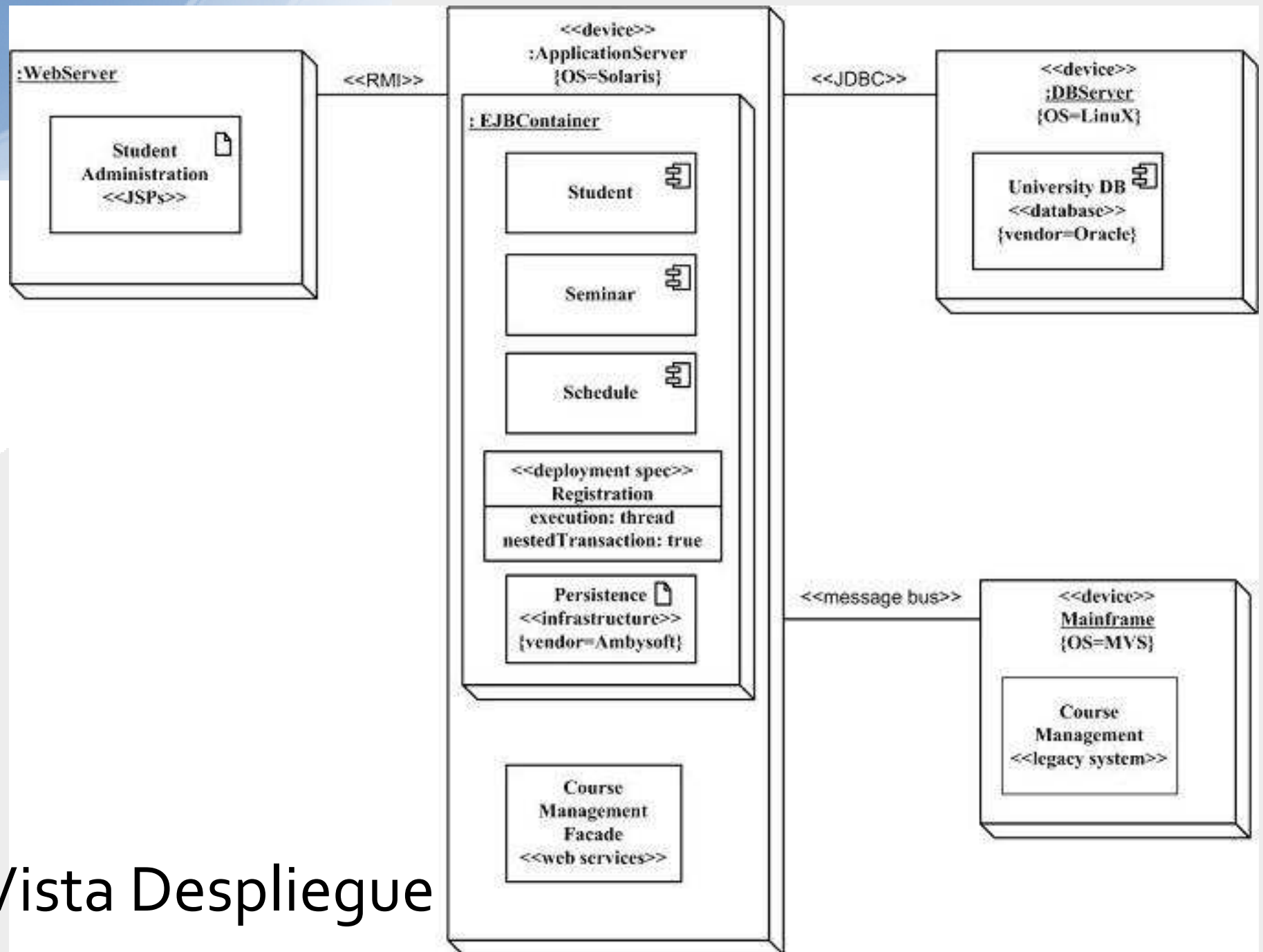


Vista Procesos



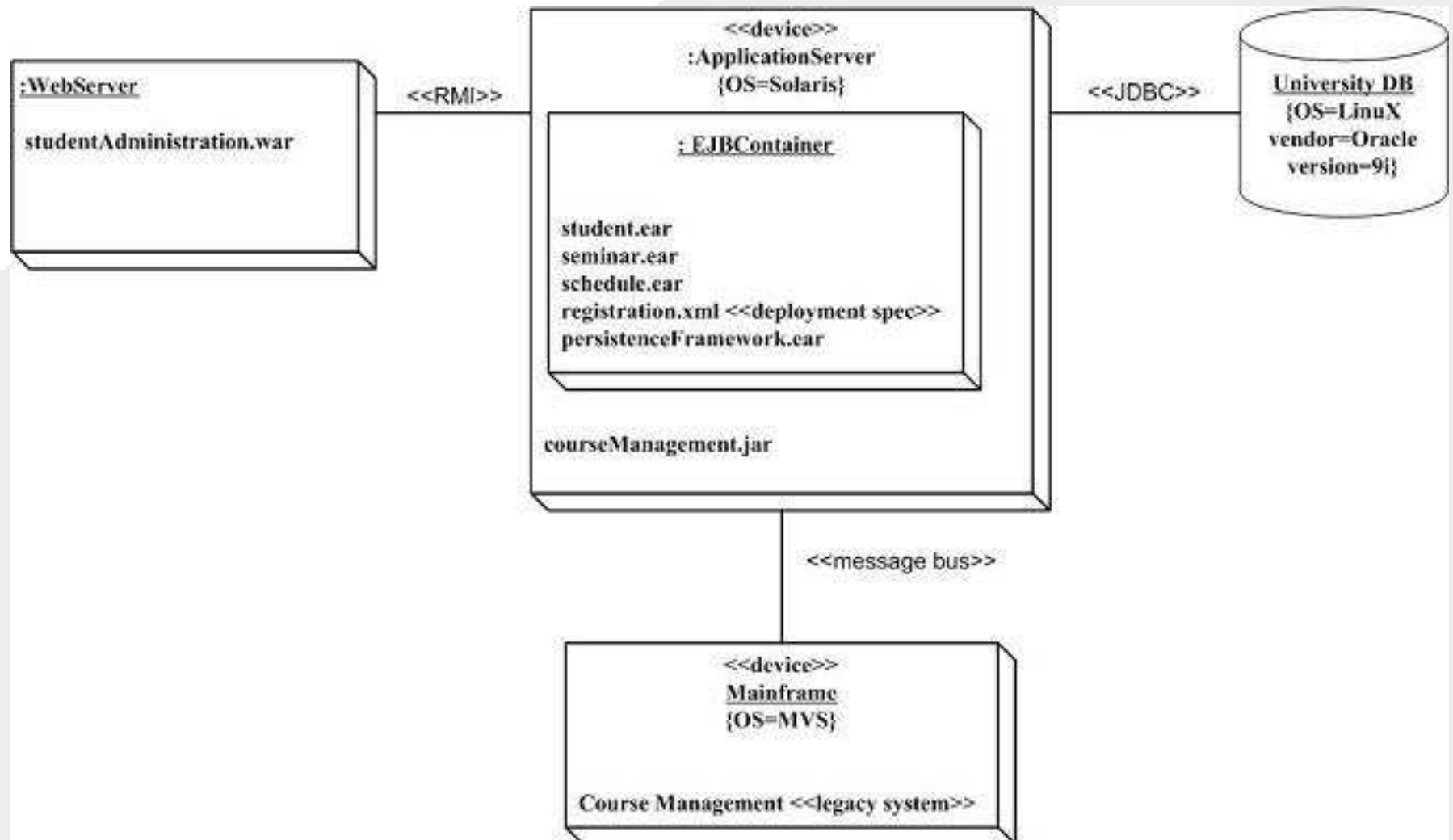
Vista Despliegue

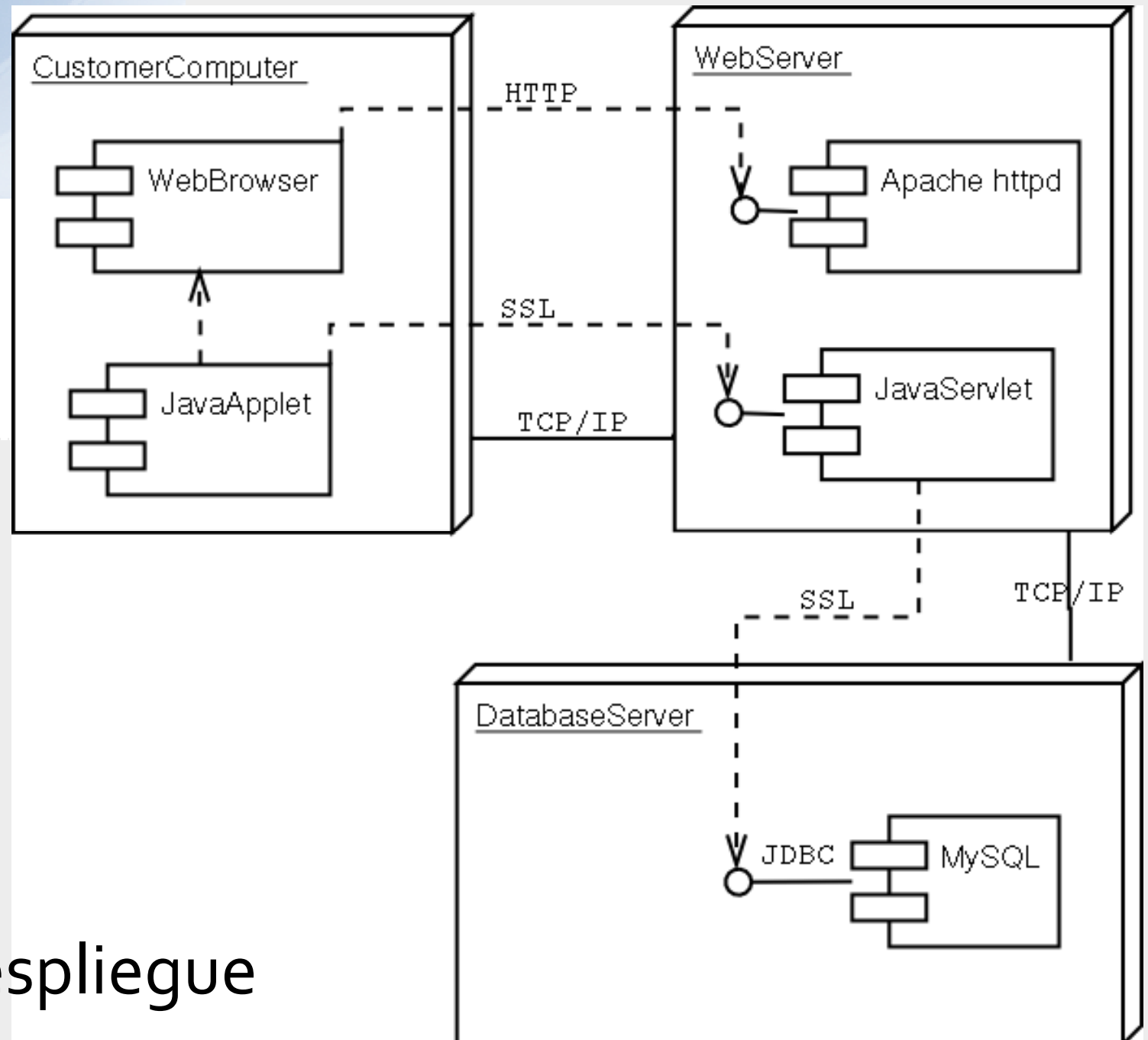
- **Elementos**
 - Nodos
- **Relaciones**
 - Asociaciones
 - Dependencia
 - Generalización



Vista Despliegue

Vista Despliegue



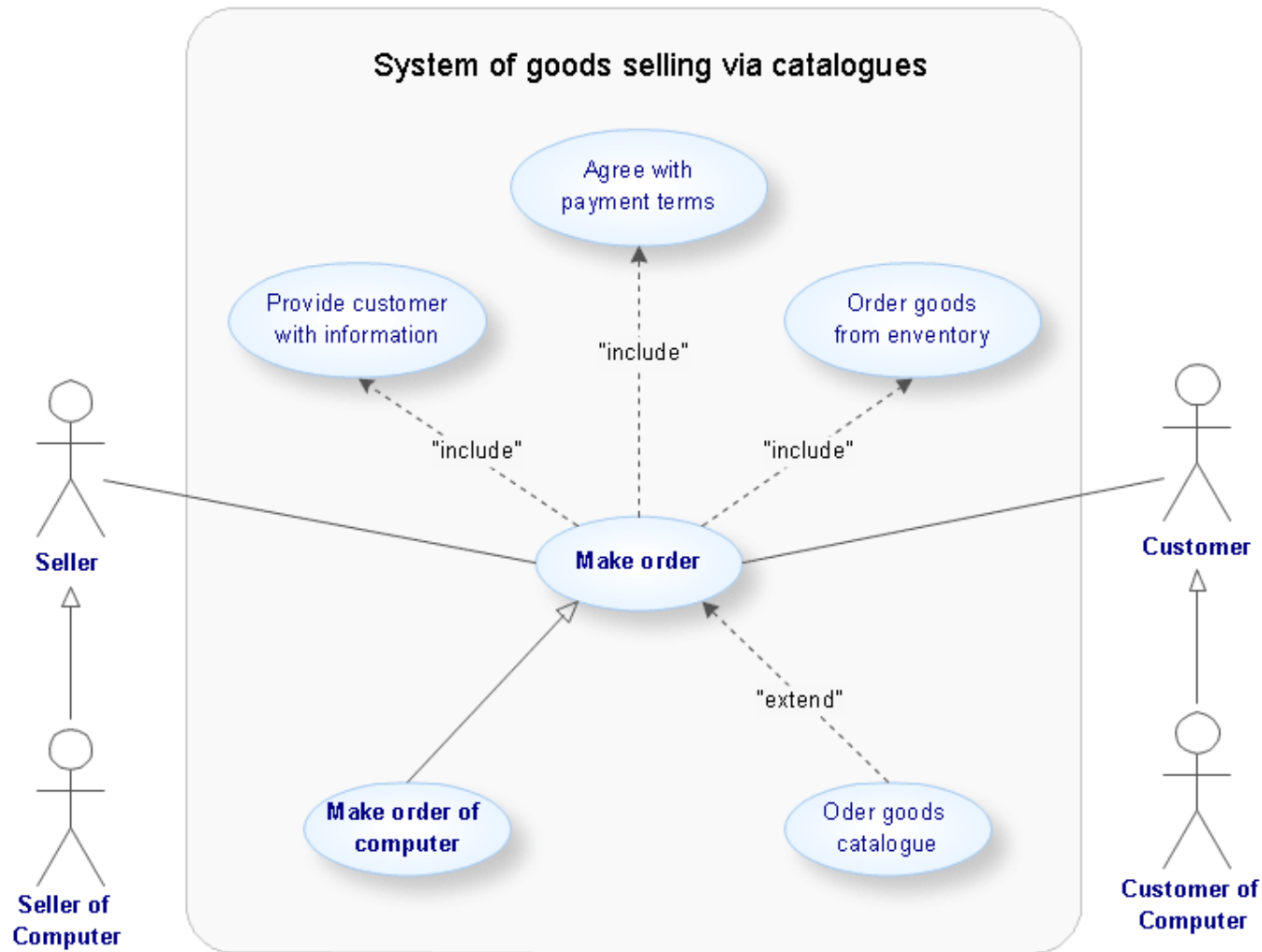


Vista Despliegue

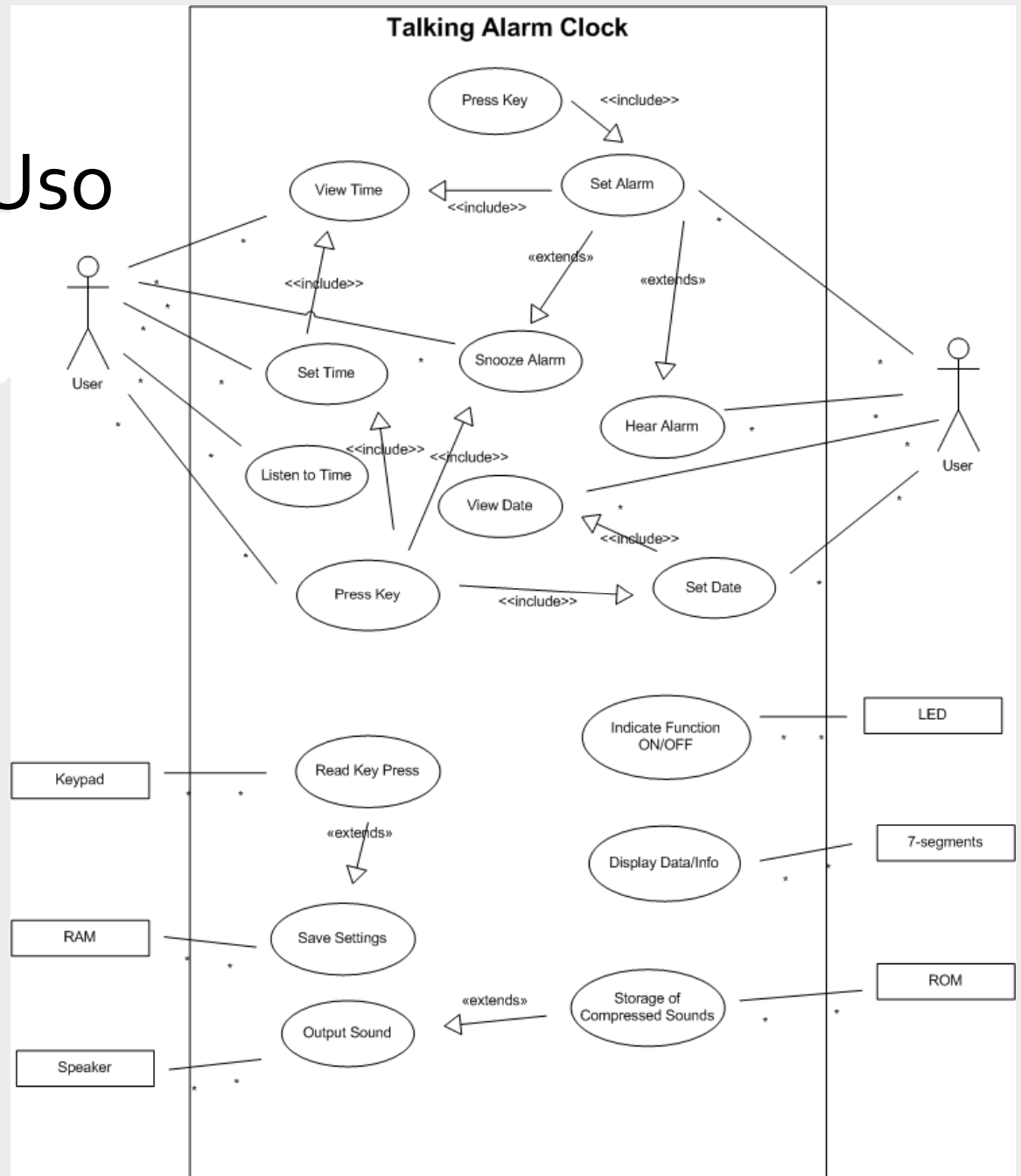
Vista Casos de Uso

- **Elementos**
 - Actores
 - Casos de uso
- **Relaciones**
 - Asociaciones
 - Dependencia
 - Generalización

Vista Casos de Uso



Vista Casos de Uso



Vista Casos de Uso

