



Análisis y Diseño de Sistemas

Tema: Introducción a UML

Urciuolo A.

UNPSJB – 2013

¿Qué es UML?

- UML = Unified Modeling Language
- Un lenguaje de propósito general para el modelado orientado a objetos
- Documento “OMG Unified Modeling Language Specification”
- UML combina notaciones provenientes desde:
 - Modelado Orientado a Objetos
 - Modelado de Datos
 - Modelado de Componentes
 - Modelado de Flujos de Trabajo (Workflows)

¿Qué es UML?

- Es un lenguaje, una notación, una sintaxis.
- NO DICE CÓMO DESARROLLAR EL PROCESO

Situación de Partida

- Diversos métodos y técnicas OO, con muchos aspectos en común pero utilizando distintas notaciones
- Inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc.
- Pugna entre distintos enfoques (y correspondientes gurús)



Establecer una notación estándar

Historia de UML

- Comenzó como el “Método Unificado”, con la participación de Grady Booch y Jim Rumbaugh. Se presentó en el OOPSLA'95
- El mismo año se unió Ivar Jacobson. Los “Tres Amigos” son socios en la compañía Rational Software. Herramienta CASE Rational Rose
- Es un mecanismo de comunicación gráfica para desarrolladores y clientes

Propósito de UML

- Brindar soporte de comunicación sobre el análisis y diseño del sistema a construir
- Brindar soporte al movimiento desde el dominio del problema del mundo real al dominio de la solución en la computadora.
 - Dos vistas del mismo sistema
 - Diagramas UML
 - Código Fuente

UML es un Lenguaje de Modelado

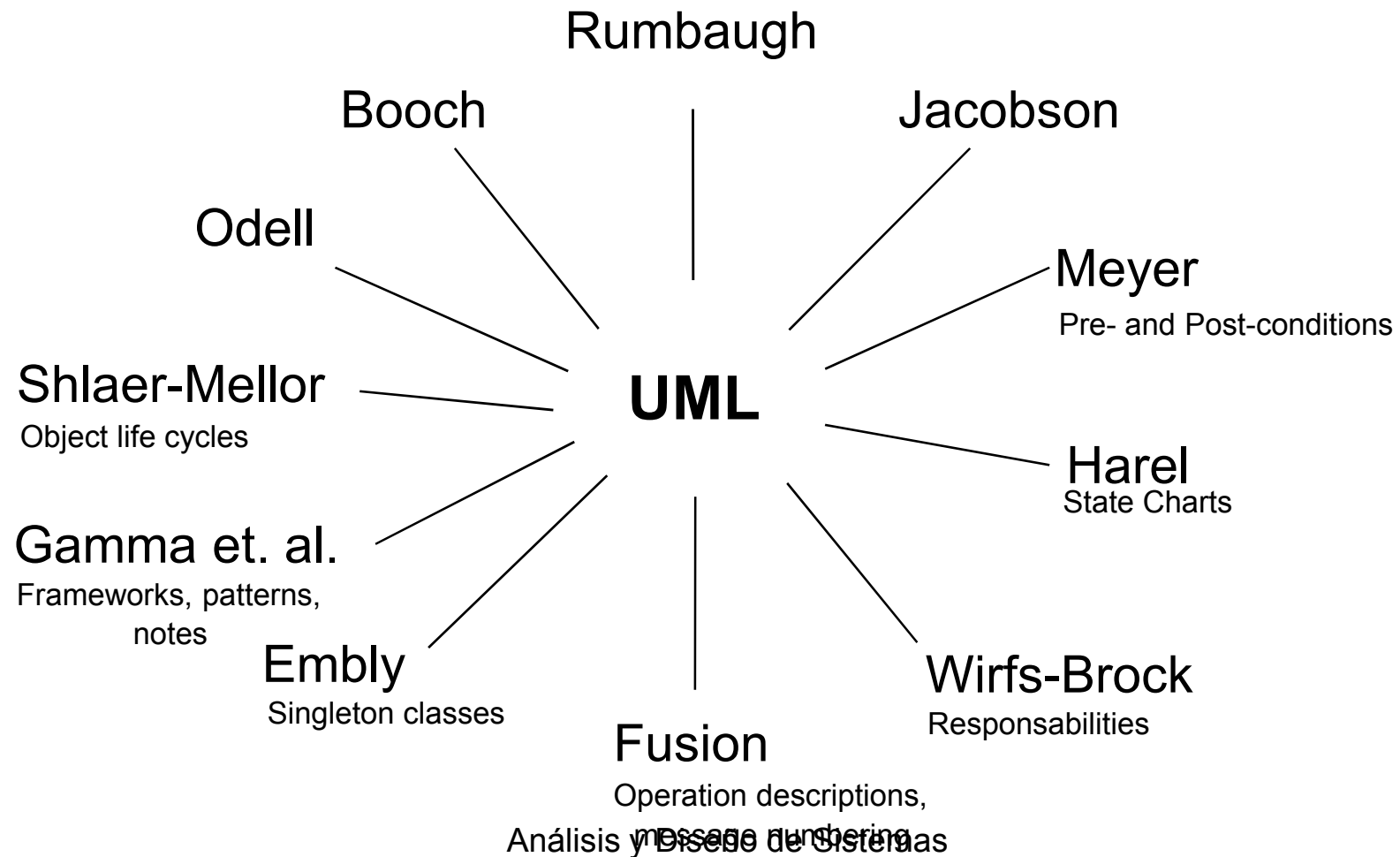
- UML
 - Es una notación gráfica que describe el análisis y diseño de software.
 - Tiene reglas para el dibujo de
 - clases
 - asociaciones entre objetos
 - asociaciones entre clases e interfaces
 - mensaje enviados entre objetos
- UML no es un proceso o metodología
- UML es un lenguaje para
 - visualizar
 - especificar
 - construir
 - documentar

Los artefactos de un sistema de software

Participantes en UML 1.0

- Rational Software
(Grady Booch, Jim Rumbaugh y Ivar Jacobson)
- Digital Equipment
- Hewlett-Packard
- i-Logix (David Harel)
- IBM
- ICON Computing
(Desmond D'Souza)
- Intellicorp and James Martin & co. (James Odell)
- MCI Systemhouse
- Microsoft
- ObjecTime
- Oracle Corp.
- Platinum Technology
- Sterling Software
- Taskon
- Texas Instruments
- Unisys

UML “aglutina” enfoques OO



Aspectos Novedosos

- Definición **semi-formal** del Metamodelo de UML
- Mecanismos de Extensión en UML:
 - *Stereotypes*
 - *Constraints*
 - *Tagged Values*

Permiten adaptar los elementos de modelado, asignándoles una semántica particular

Ventajas de utilización de UML

- Reunir los puntos fuertes de cada método
- Idear nuevas mejoras
- Proporcionar estabilidad al mercado
 - Proyectos basados en un lenguaje maduro
 - Aparición de potentes herramientas
- Eliminar confusión en los usuarios

Perspectivas de UML

- UML es el lenguaje de modelado orientado a objetos estándar predominante - Razones:
 - Participación de metodólogos influyentes
 - Participación de importantes empresas
 - Aceptación del OMG como notación estándar
- Evidencias:
 - Herramientas que proveen la notación UML
 - “Edición” de libros
 - Congresos, cursos, “camisetas”, etc.

Breve Tour por UML

Modelos y Diagramas

- Un modelo captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle: *Un modelo es una simplificación de la realidad.*
- Diagrama: una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo.

Modelos y Diagramas

- Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés
- El código fuente del sistema es el modelo más detallado de éste (y además es ejecutable). Sin embargo, se requieren otros modelos ...
- Cada modelo es completo desde su punto de vista del sistema. Existen relaciones de trazabilidad entre los diferentes modelos
- Los modelos se construyen usando Diagramas

Utilidad del modelado

- Los modelos:
 - **visualizan** cómo es o queremos que sea el sistema
 - **especifican** la estructura y comportamiento del sistema.
 - guían la **construcción** del sistema.
 - **documentan** las decisiones.

Modelos en UML

- Modelado de **Requerimientos**
 - Diagrama de Casos de Uso
- Modelado **Estructural**
 - Diagrama de Clases
- Modelado de **Comportamiento**
 - Diagramas de Interacción
 - Diagramas de Máquinas de Estado
- Modelado de flujos de **Actividades**
 - Diagramas de actividades
- Modelado de **Implementación**
 - Diagrama de Componentes
- Modelado de **Despliegue**
 - Diagramas de Despliegue



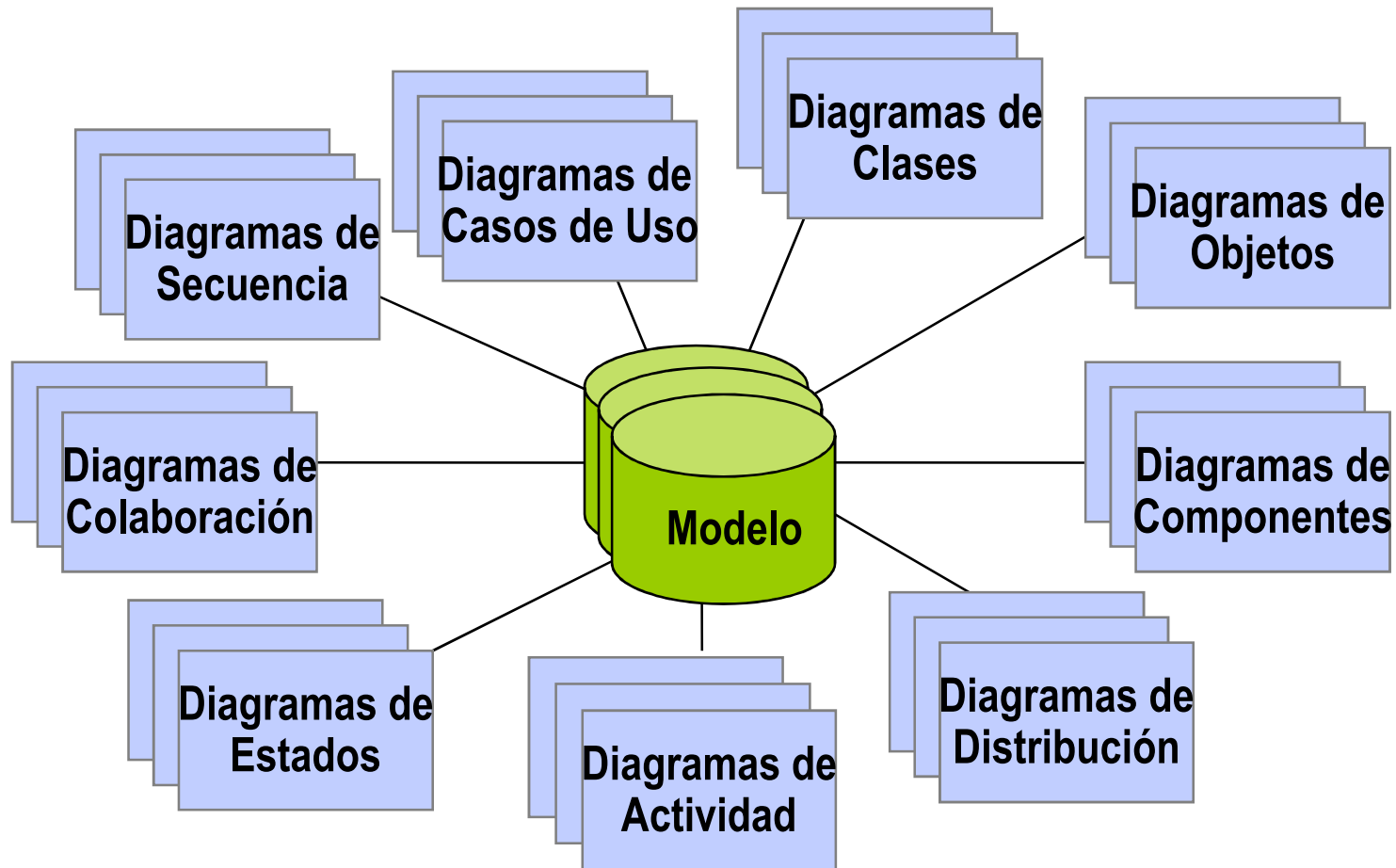
Diagramas
no son
modelos!!

Tipos de modelo

- ¿En qué **etapa del proceso** se usa? ¿Análisis o Diseño?
- ¿Cuál es su **grado de detalle**? ¿Abstracto o detallado?
- ¿**Qué sistema describe**? ¿Modelo de negocio o modelo software?
- ¿**Qué aspecto describe**? ¿Estructural o de comportamiento?
- ¿Es **específico o independiente de la plataforma**?
- ¿**A qué plataforma va dirigido**? EJB, JDBC, .NET, CORBA, etc.

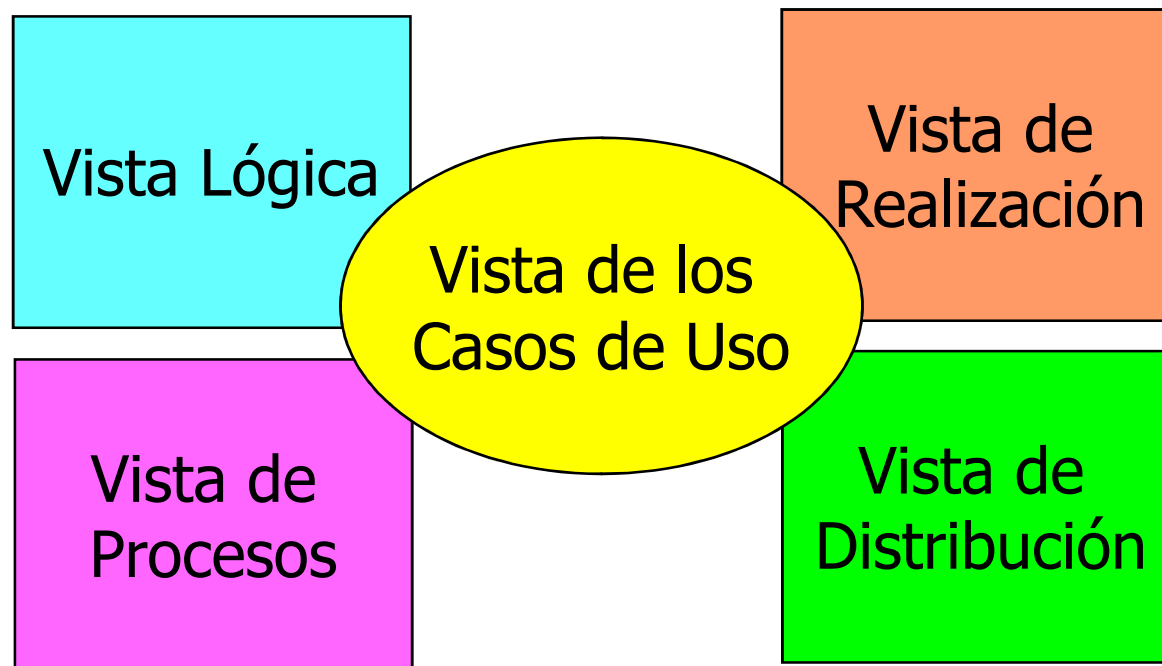
Diagramas de UML

Los diagramas expresan gráficamente partes de un modelo



Organización de Modelos

4+1 vistas de Kruchten (1995)



Este enfoque sigue el browser de Rational Rose

Análisis y Diseño de Sistemas

UML y el modelado

*UML es un lenguaje para **visualizar, especificar, construir y documentar** los artefactos (modelos) de un sistema que involucra una gran cantidad de software, desde una perspectiva **orientada a objetos**.*

- **UML es una notación**, no es un proceso
- Se han definido muchos procesos para UML.
 - *Rational* ha ideado *RUP*, el "*proceso unificado*".
- Utilizable para sistemas que no sean software

Organización de Modelos

Propuesta de Rational Unified Process (RUP)

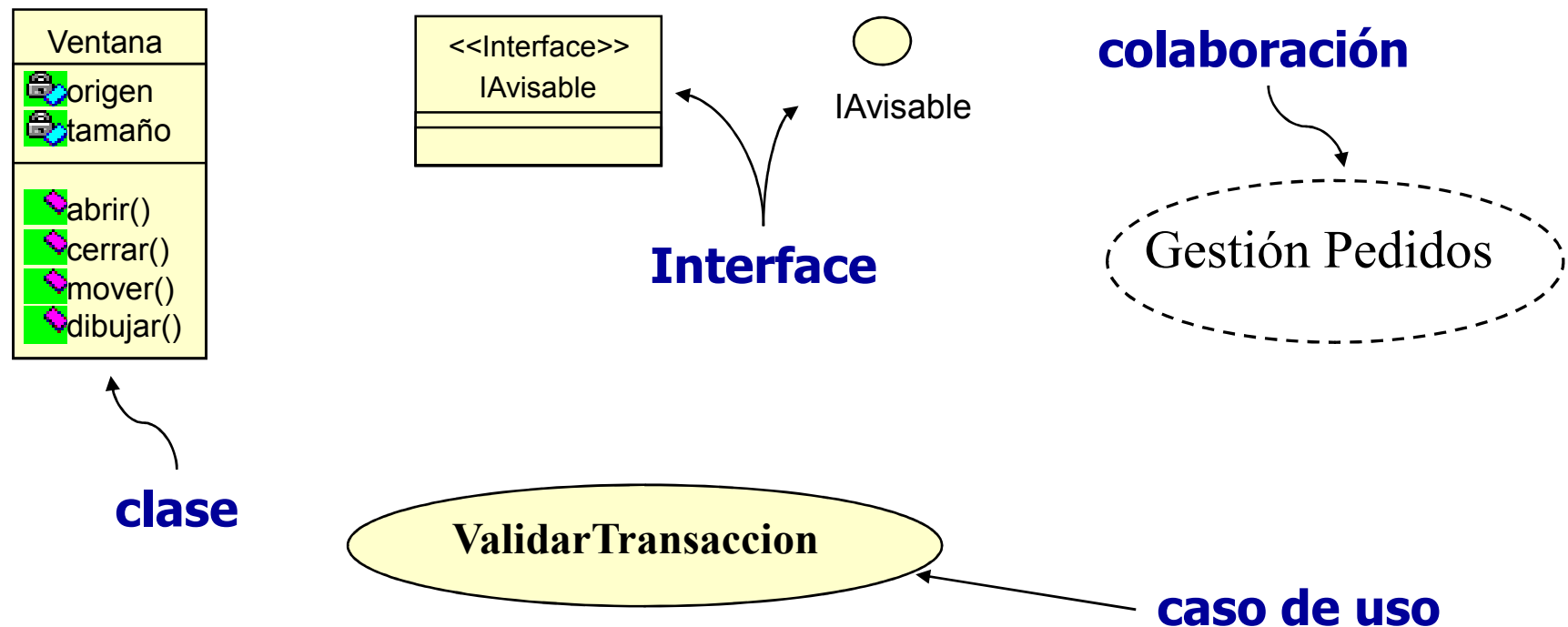
- Modelo de Negocio – Modelo de Dominio
- M. de Requerimientos
- M. de Análisis
- M. de Diseño
- M. de Despliegue
- M. de Datos
- M. de Implementación
- M. de Pruebas

Marco Conceptual de UML

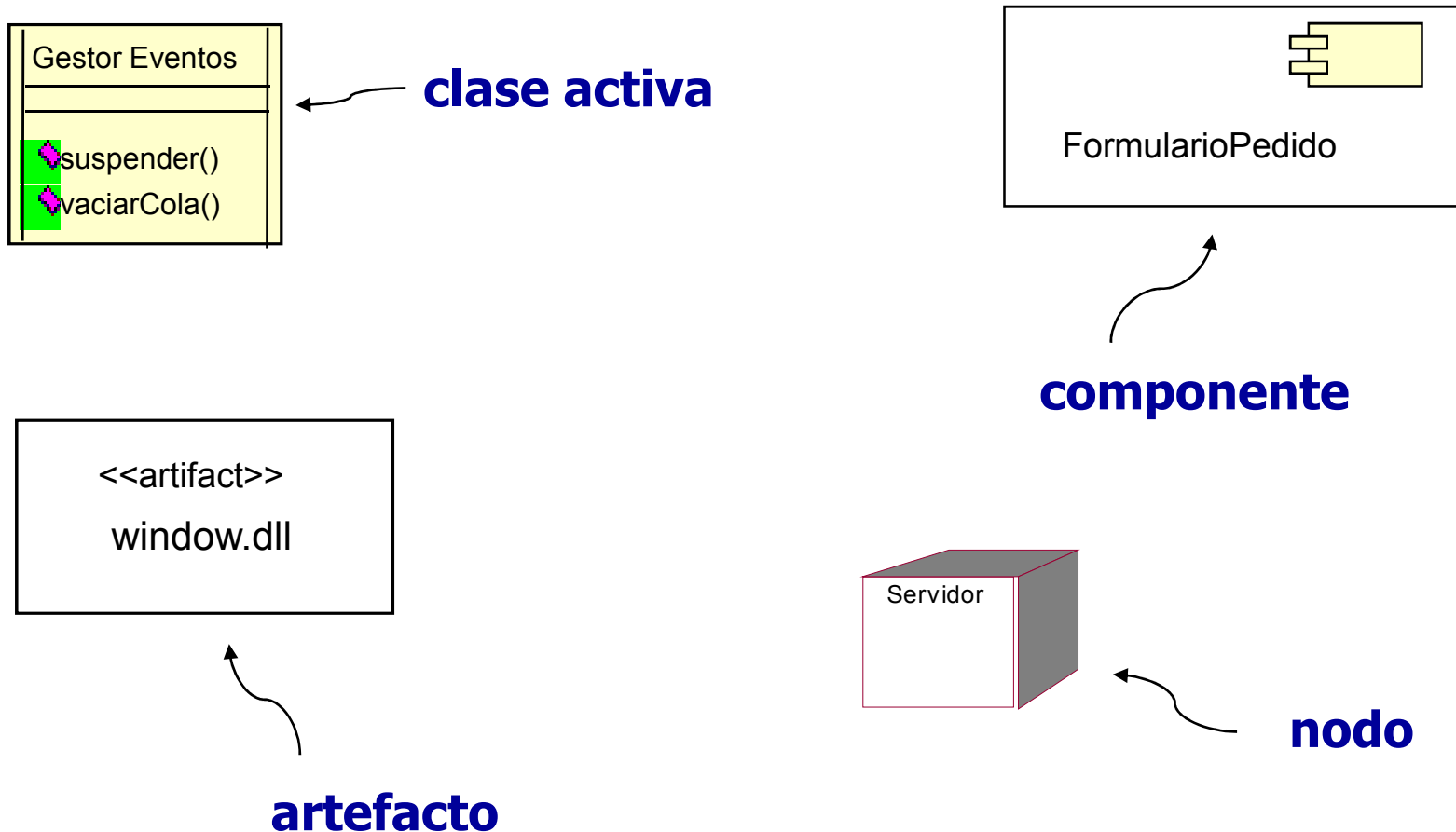
- **Bloques básicos de construcción**
 - Elementos
 - Estructurales, Comportamiento, Agrupación, Anotación
 - Relaciones
 - Diagramas
- **Reglas para combinar bloques**
 - Establecen qué es un *modelo bien formado*
- **Mecanismos comunes**
 - Especificaciones, Extensibilidad, Dicotomía clase-
instancia, Dicotomía interfaz-realización

Elementos Estructurales

Son las partes estáticas de un modelo.



Elementos Estructurales



Elementos de Comportamiento

Son las **partes dinámicas** de UML.

Interacción

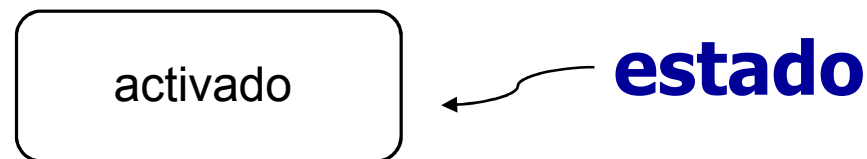
Conjunto de mensajes intercambiados entre un conjunto de objetos con un propósito particular.



Elementos de Comportamiento

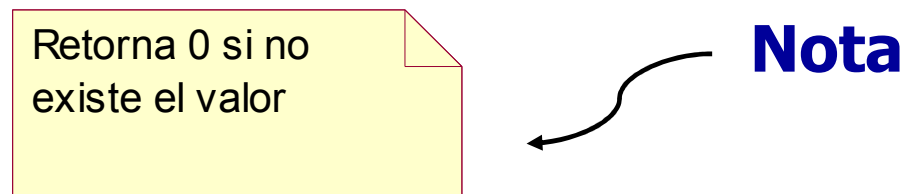
Máquina de estados

Secuencia de estados por los que pasa un objeto durante su vida en respuesta a eventos.



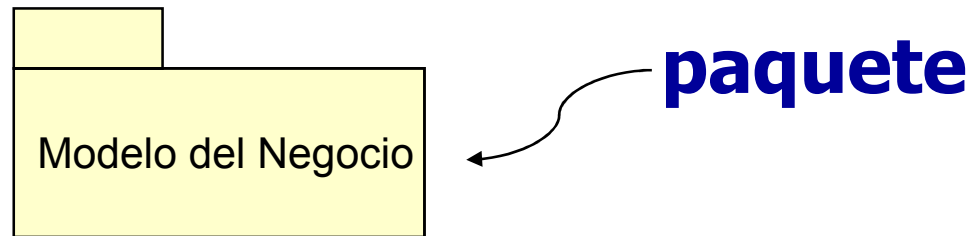
Elementos de Anotación

Son las **partes explicativas** de los modelos UML.



Elementos de Agrupación

Son las **partes de organización** de los modelos UML.

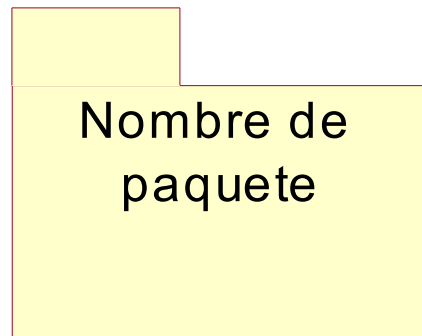


Un paquete incluye un conjunto de elementos de cualquier naturaleza.

Tiene una naturaleza conceptual.

Paquetes en UML

- Los paquetes ofrecen un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado

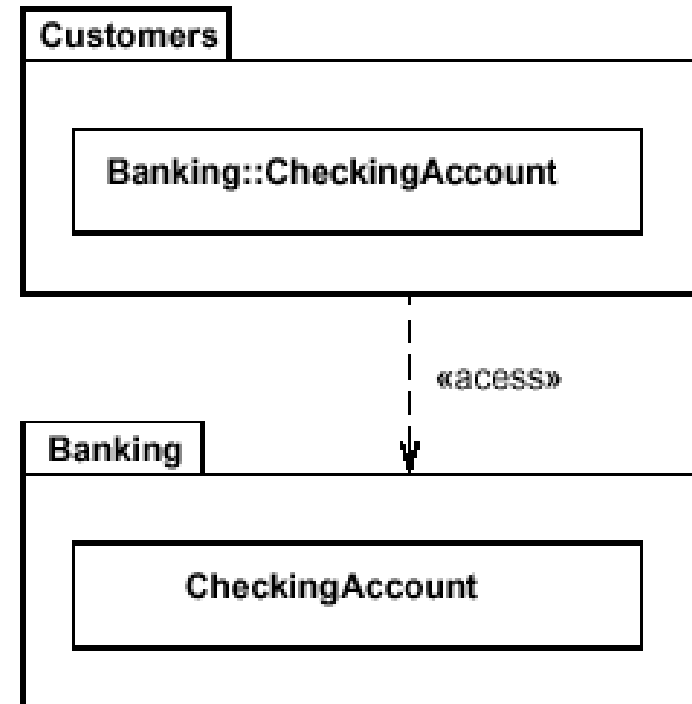


Paquetes en UML

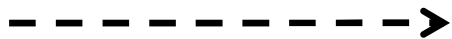
- Cada paquete corresponde a un submodelo (subsistema) del modelo (sistema)
- Un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a (está definido en) sólo un paquete
- Una clase de un paquete puede aparecer en otro paquete por la importación a través de una **relación de dependencia** entre paquetes

Paquetes en UML

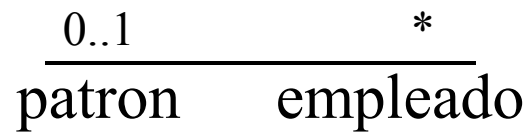
- Todas las clases no son necesariamente visibles desde el exterior del paquete, es decir, un paquete encapsula a la vez que agrupa
- El operador “::” permite designar una clase definida en un contexto distinto del actual



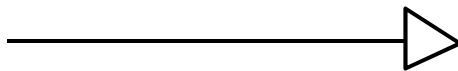
Relaciones



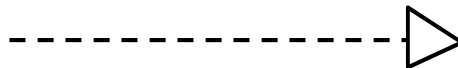
Dependencia



Asociación



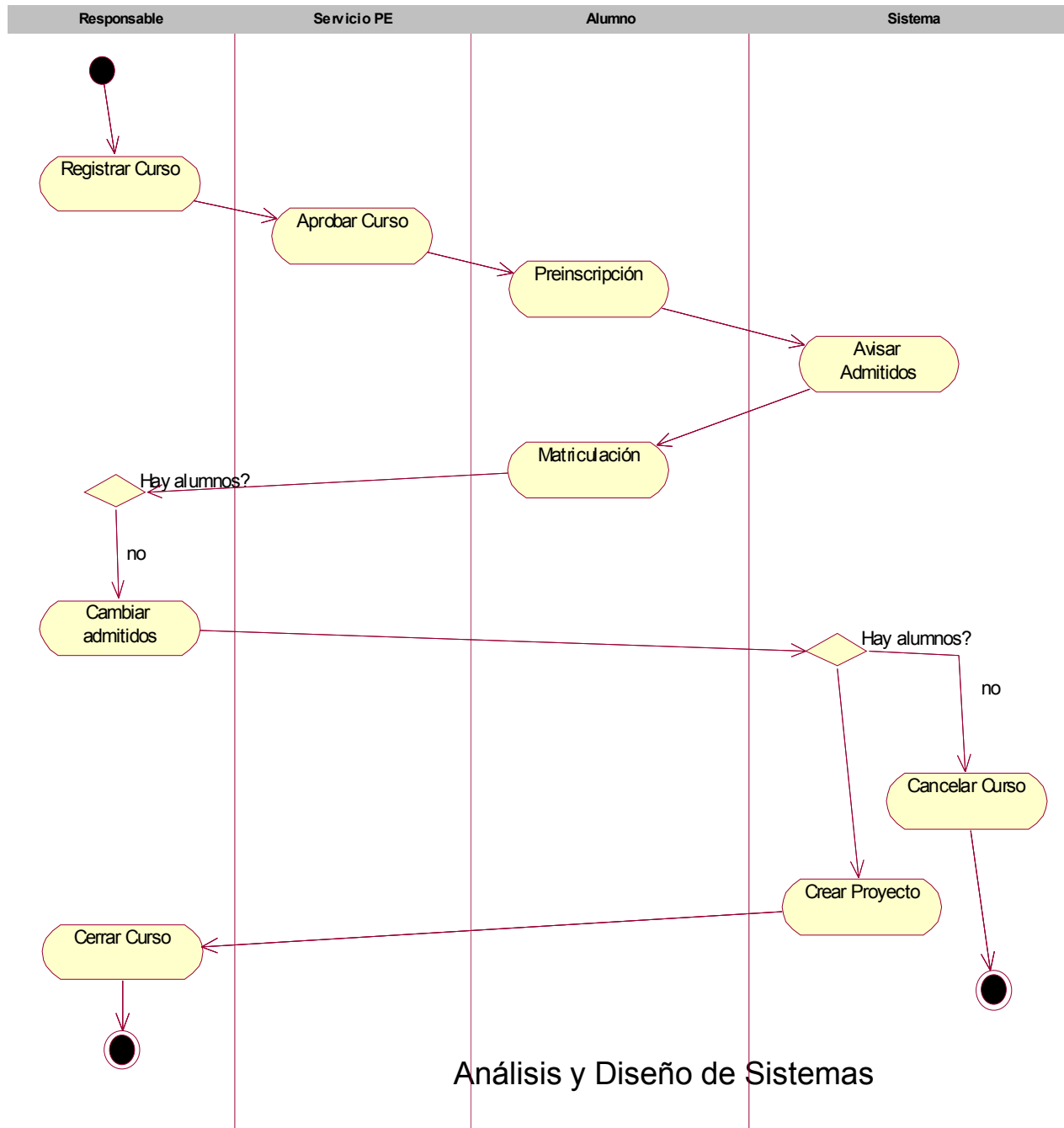
Generalización



Realización

Diagramas de UML

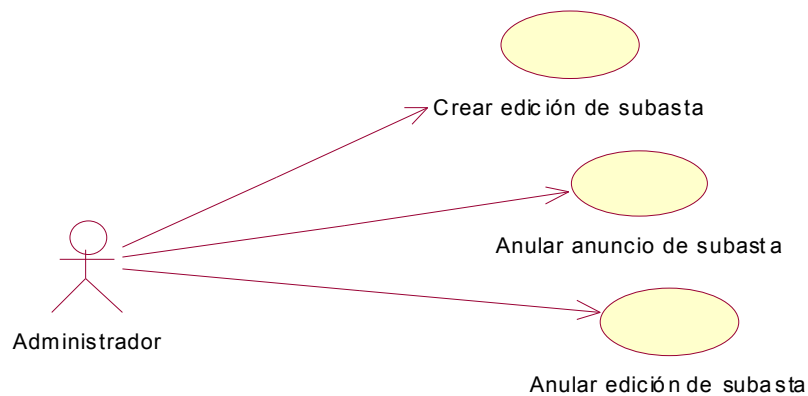
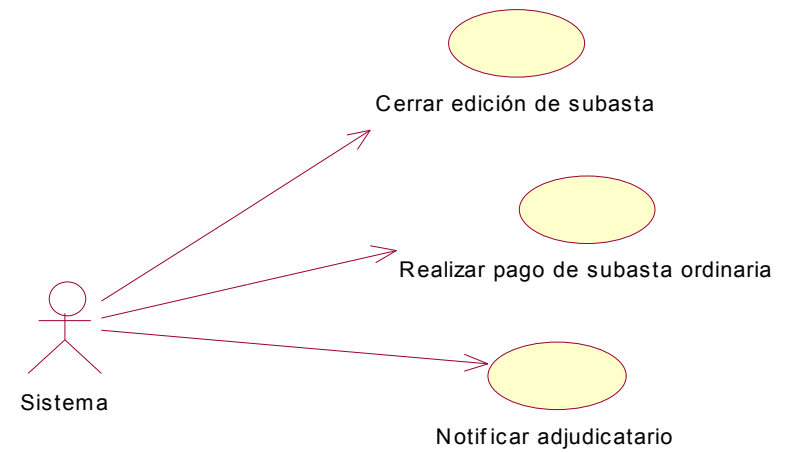
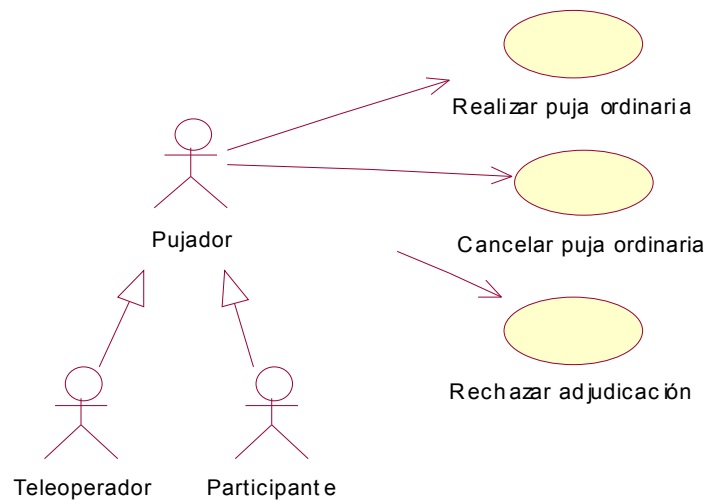
- Diagrama de Casos de Uso
- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de Secuencia
- Diagrama de Comunicación (UML 2.0)
- Diagrama de Estados
- Diagrama de Actividades
- Diagrama de Componentes
- Diagrama de Despliegue
- Diagrama de Estructura Compuesta (UML 2.0)
- Diagrama de Vista de interacción (UML 2.0)
- Diagrama de Paquetes (UML 2.0)
- Diagrama de Tiempos (UML 2.0)



Modelo del Negocio

Diagrama de actividad

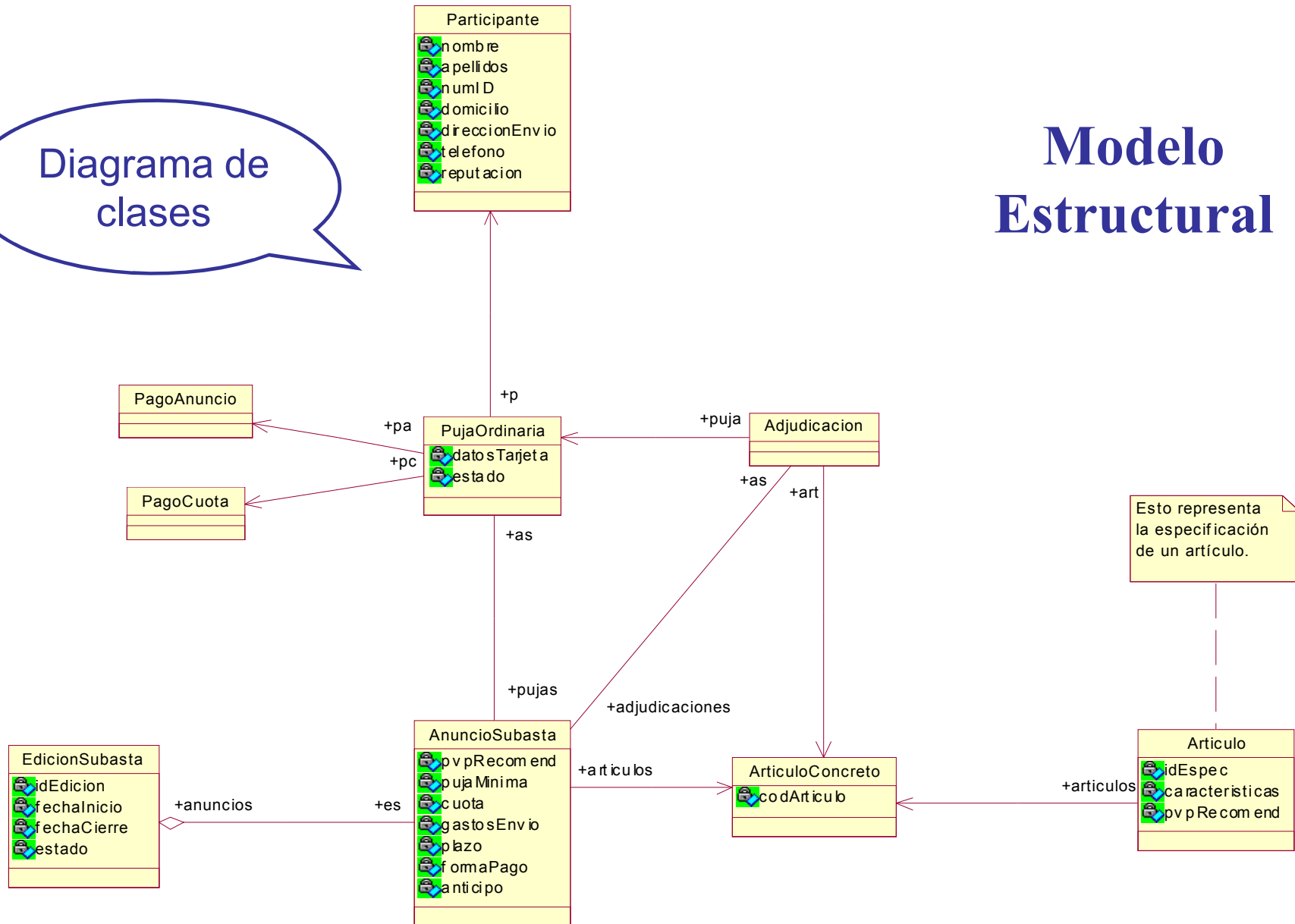
Modelo de Casos de Usos

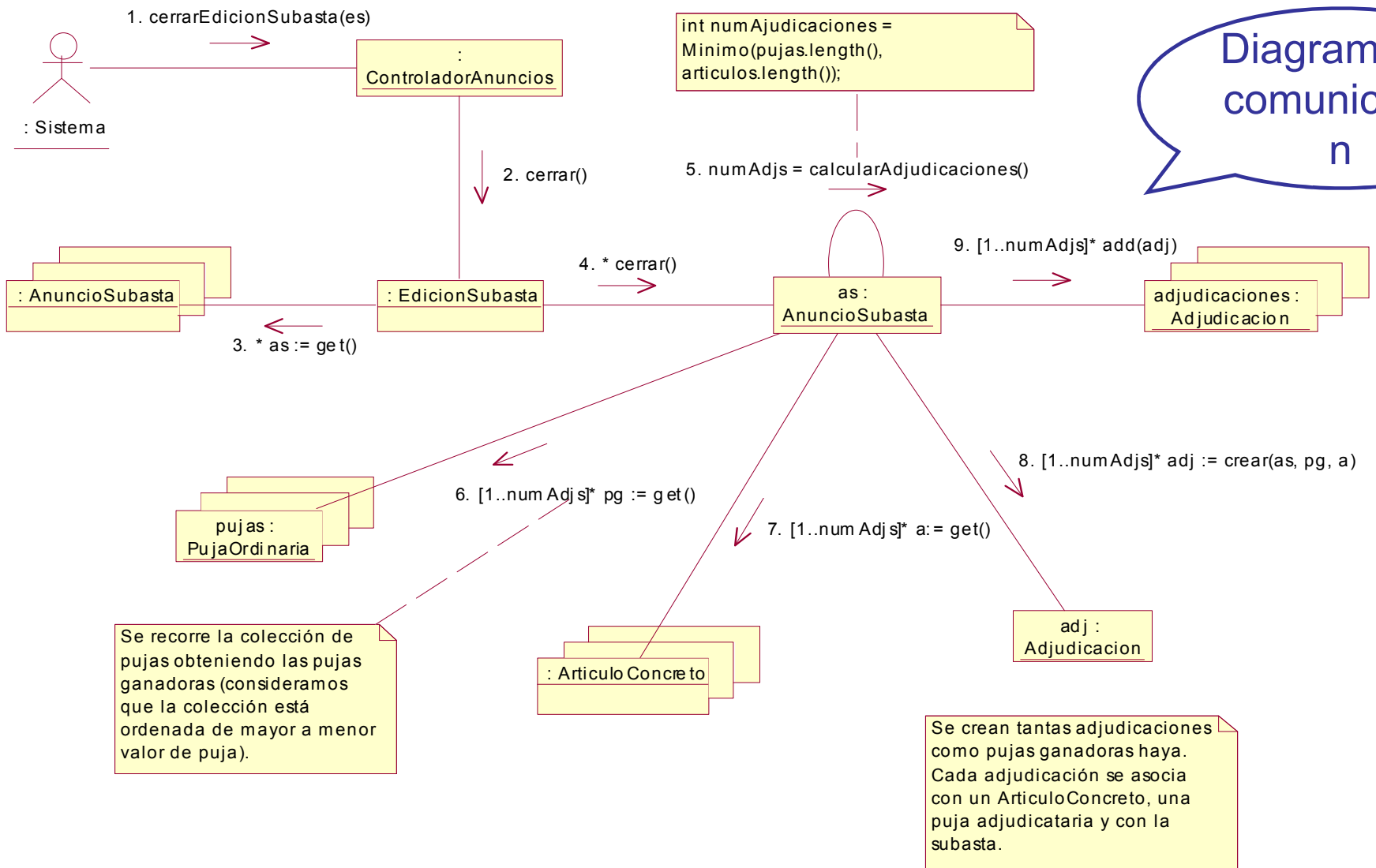


Diagramas de casos de uso

Diagrama de clases

Modelo Estructural

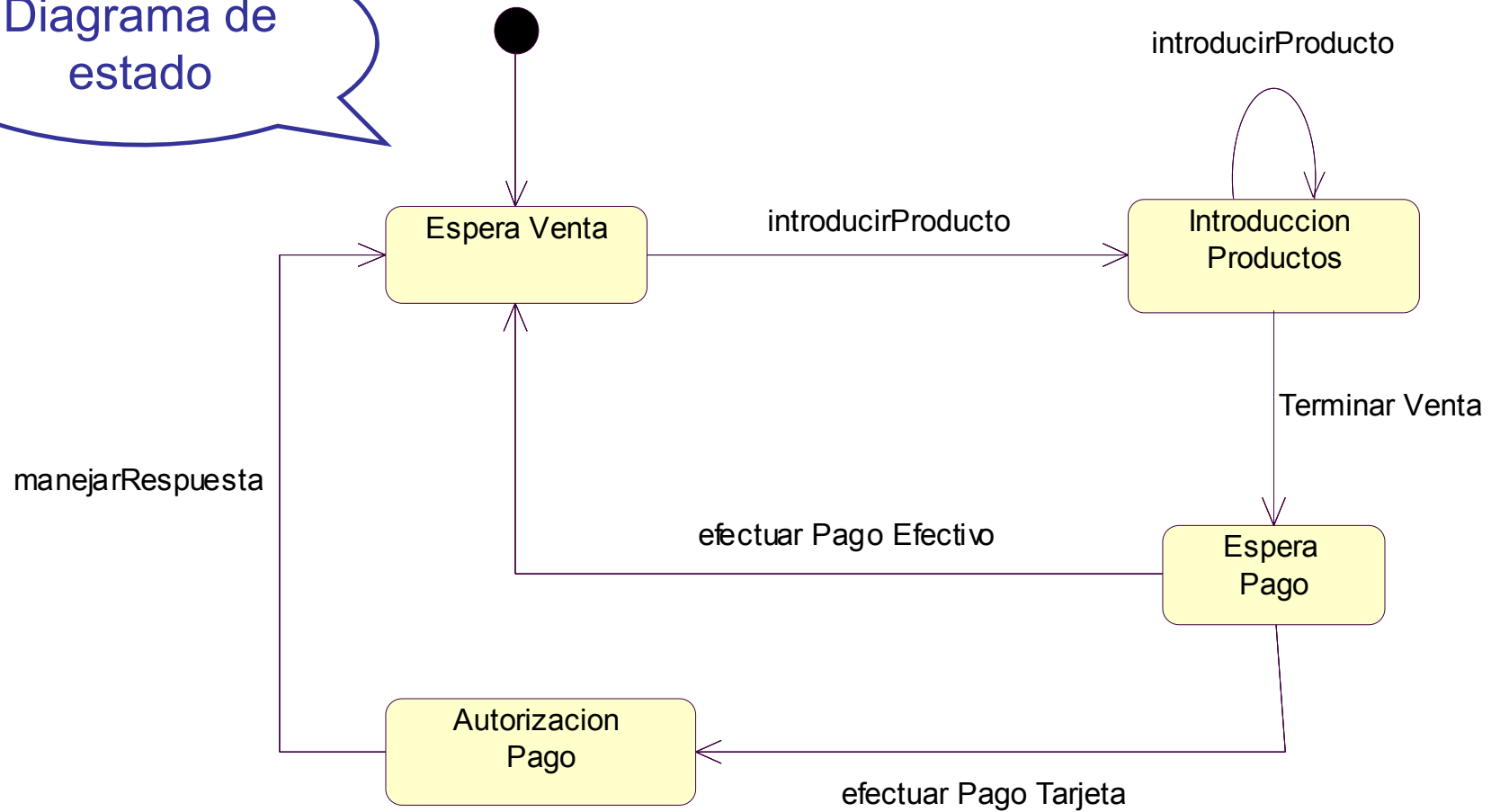




Modelo de Comportamiento

Análisis y Diseño de Sistemas

Diagrama de estado

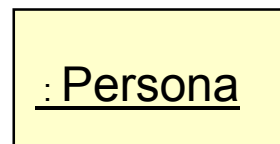
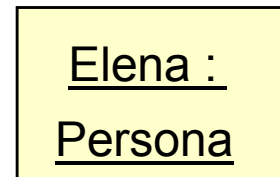
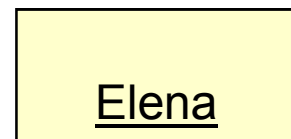
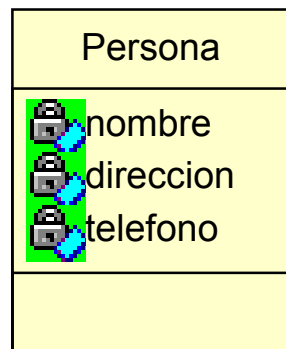


Modelo de Comportamiento

Análisis y Diseño de Sistemas

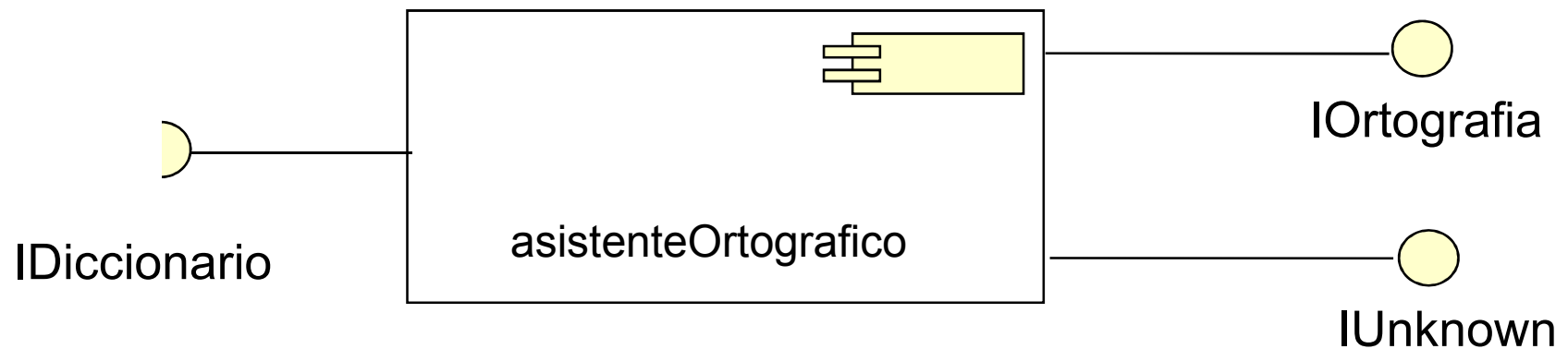
Mecanismos comunes de UML

- Dicotomía **clasificador** / *instancia*



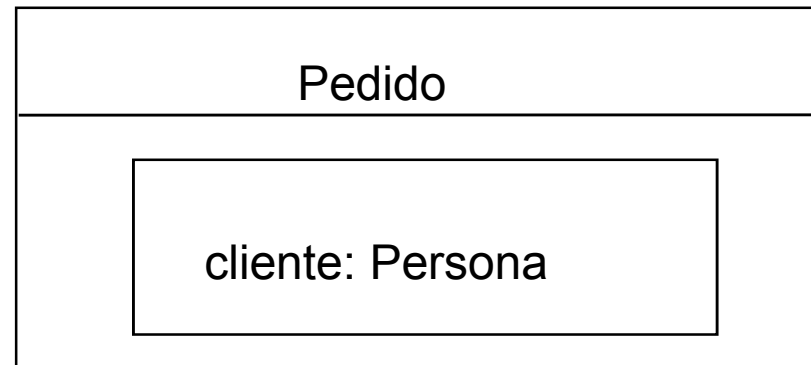
Mecanismos comunes de UML

- Dicotomía *interfaz* / *implementación*



Mecanismos comunes de UML

- Dicotomía *rol* / *tipo*

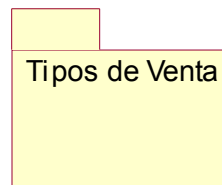
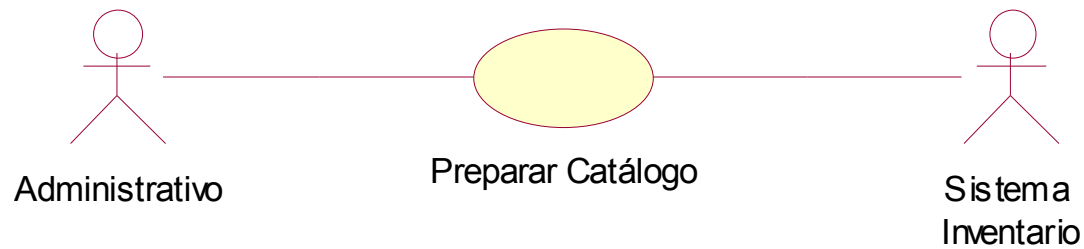
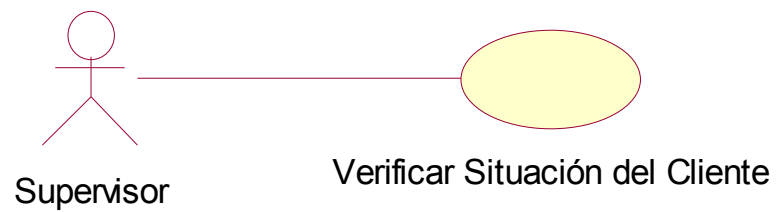


El **tipo** declara la clase de una entidad, por ejemplo un objeto o parámetro, y el **rol** describe el significado de la entidad en un determinado contexto, tal como una clase, componente o colaboración.

Diagrama de Casos de Uso

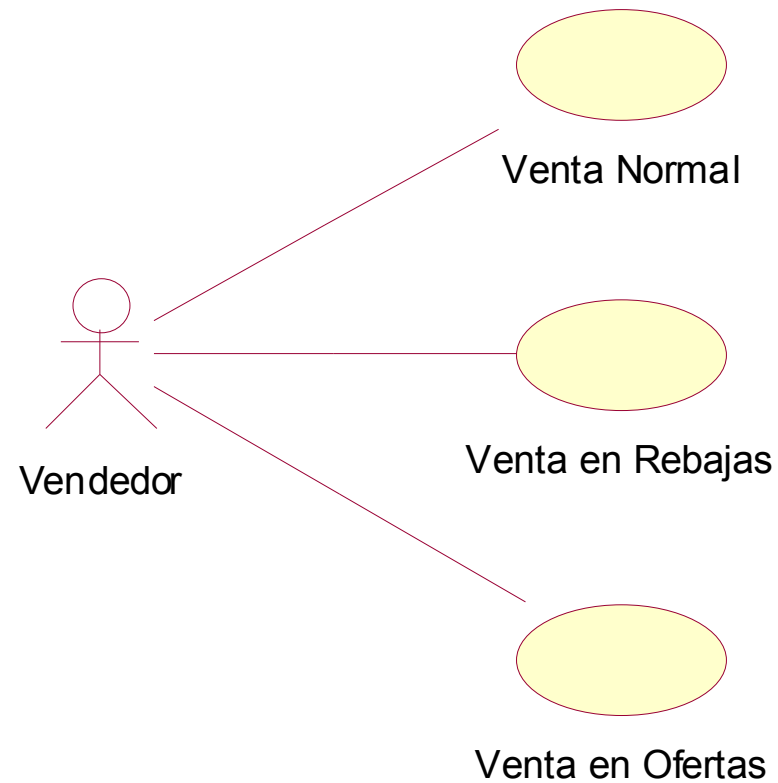
- Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje
- No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos

Ejemplos

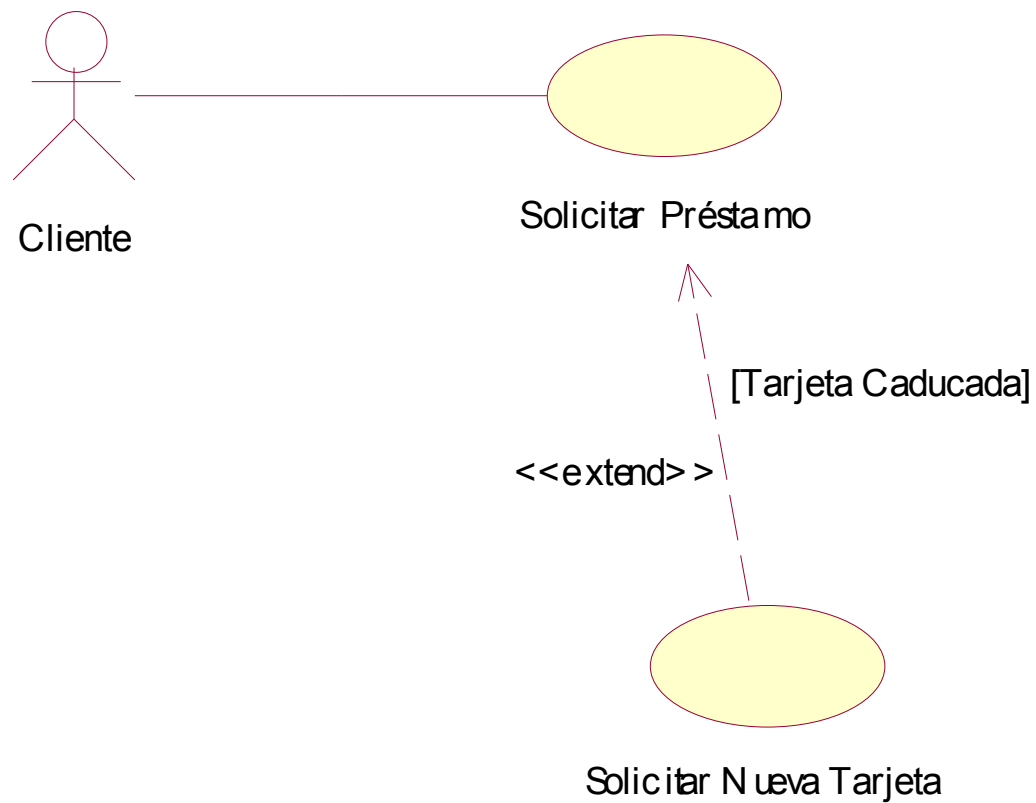


Ejemplos

En el paquete tipos de venta:



Ejemplos



Ejemplos

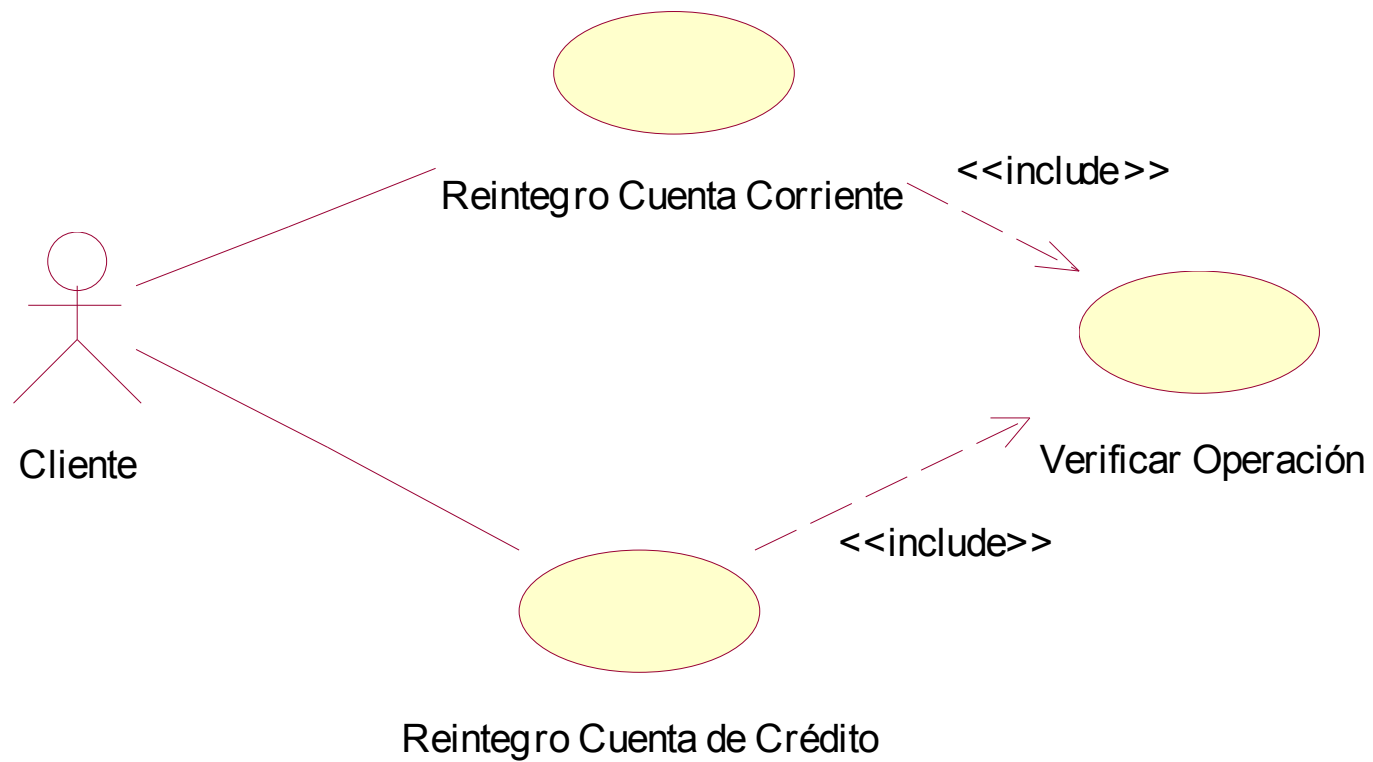


Diagrama de Secuencia

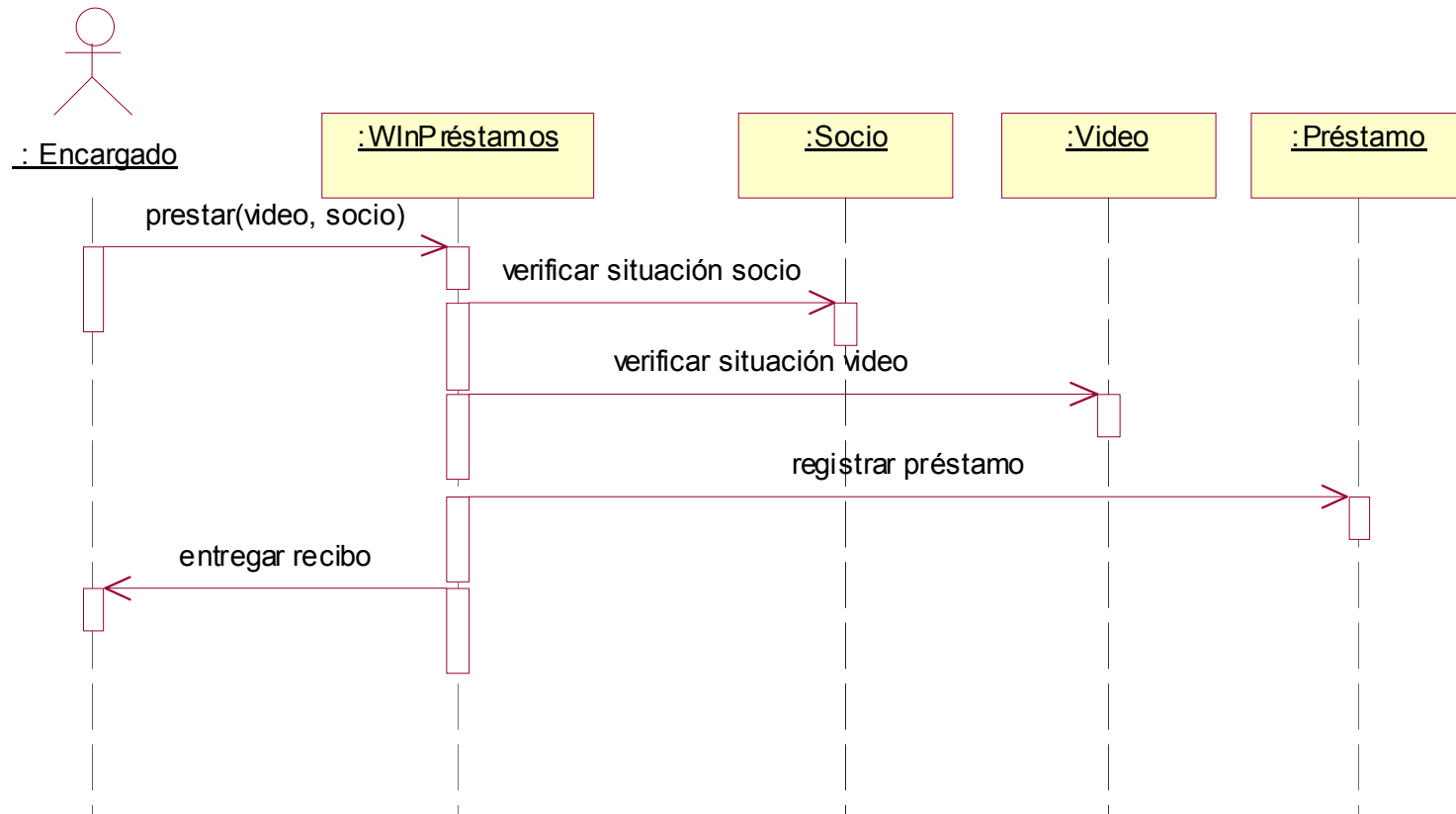


Diagrama de Comunicación

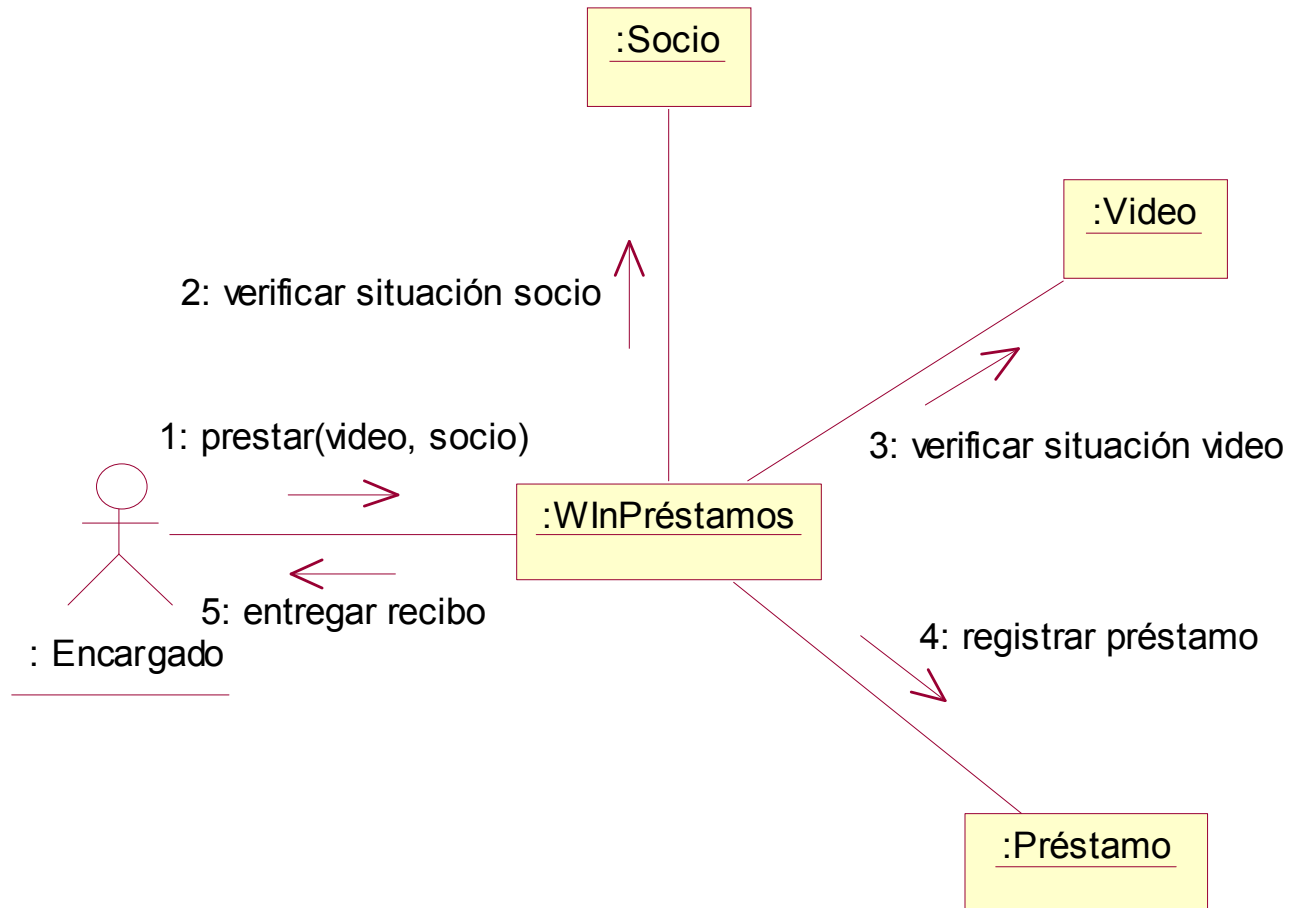
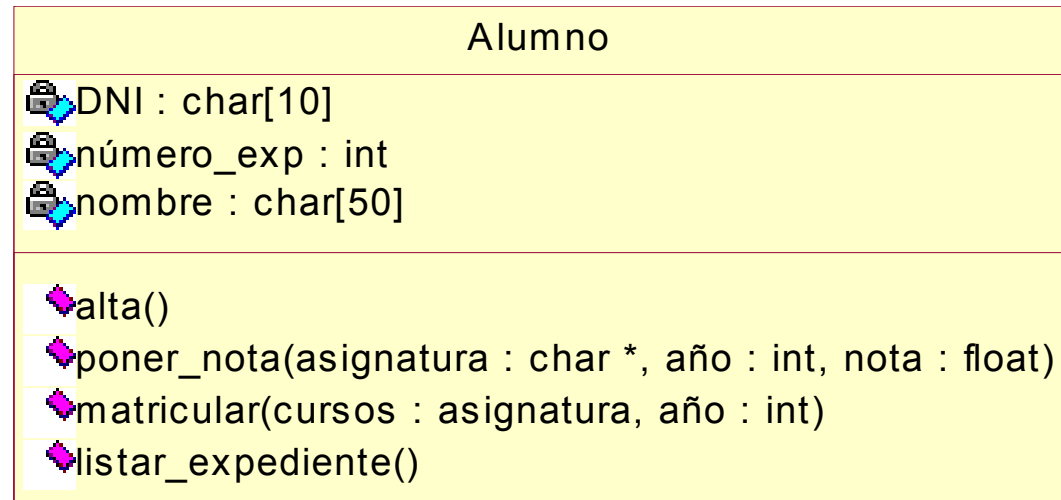


Diagrama de Clases

- El Diagrama de Clases es el diagrama principal para el análisis y diseño
- Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia
- La definición de clase incluye definiciones para atributos y operaciones
- El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones

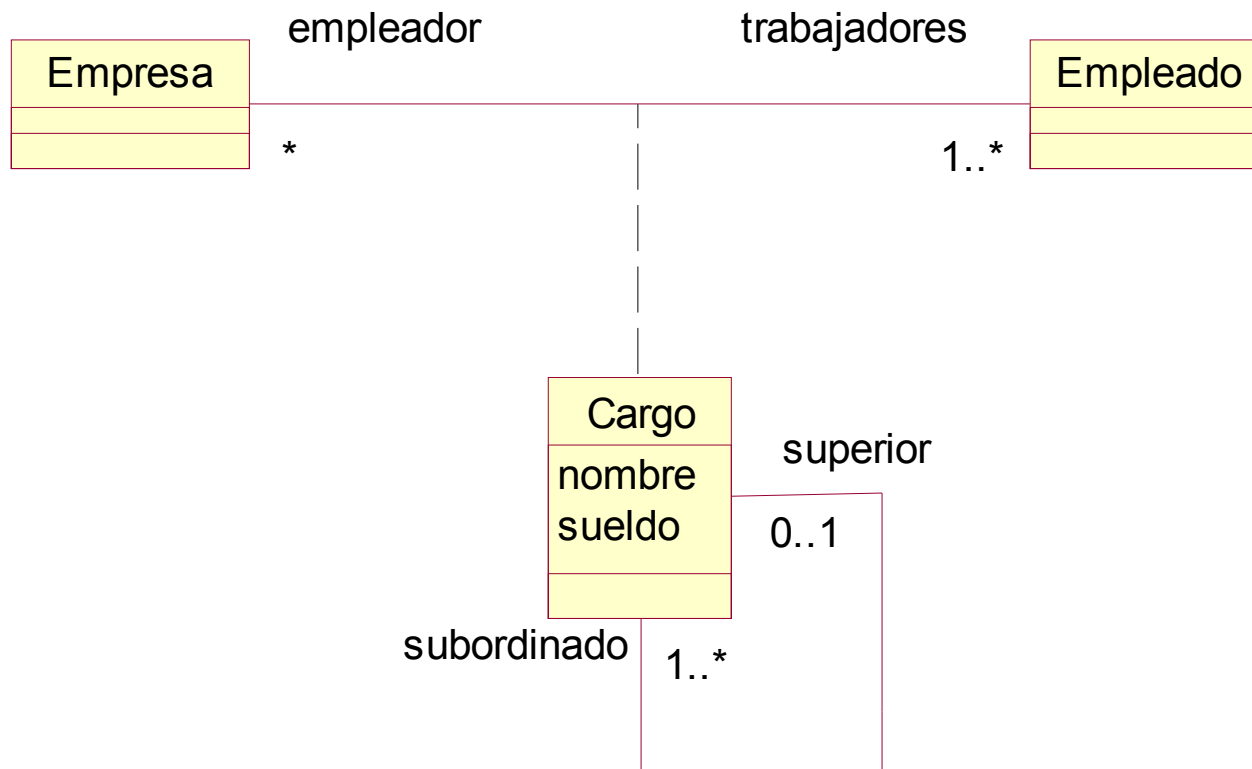
Ejemplos (Clase y Visibilidad)



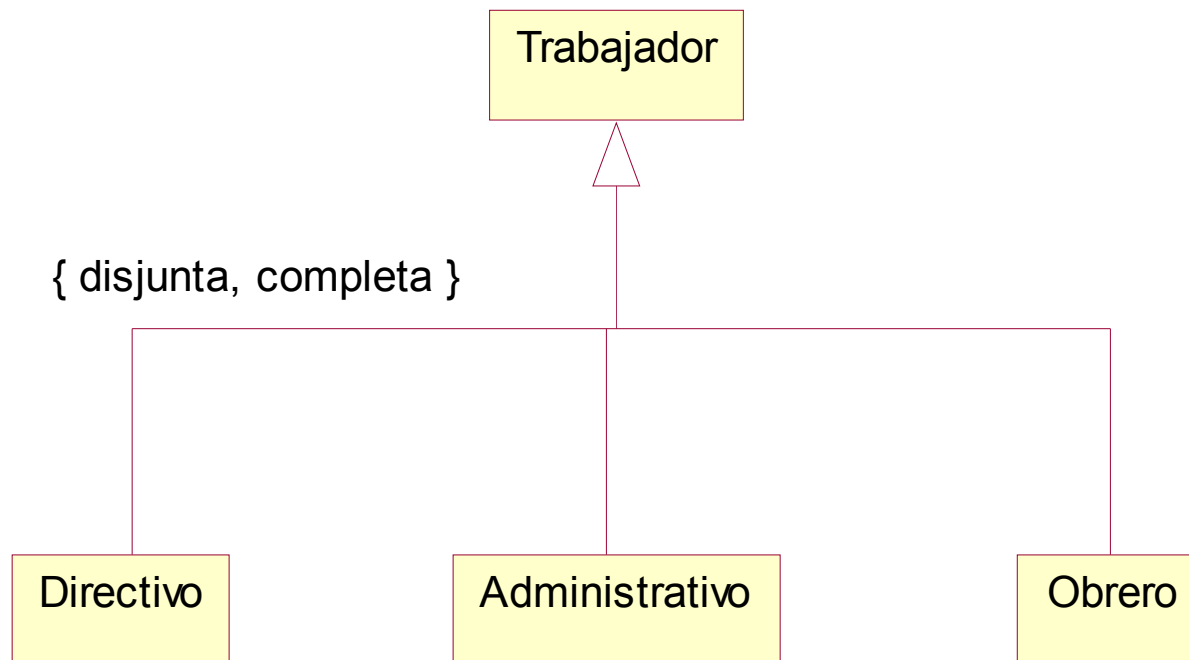
Ejemplos (Asociación)



Ejemplos (Clase Asociación)



Ejemplos (Generalización)



Ejemplos

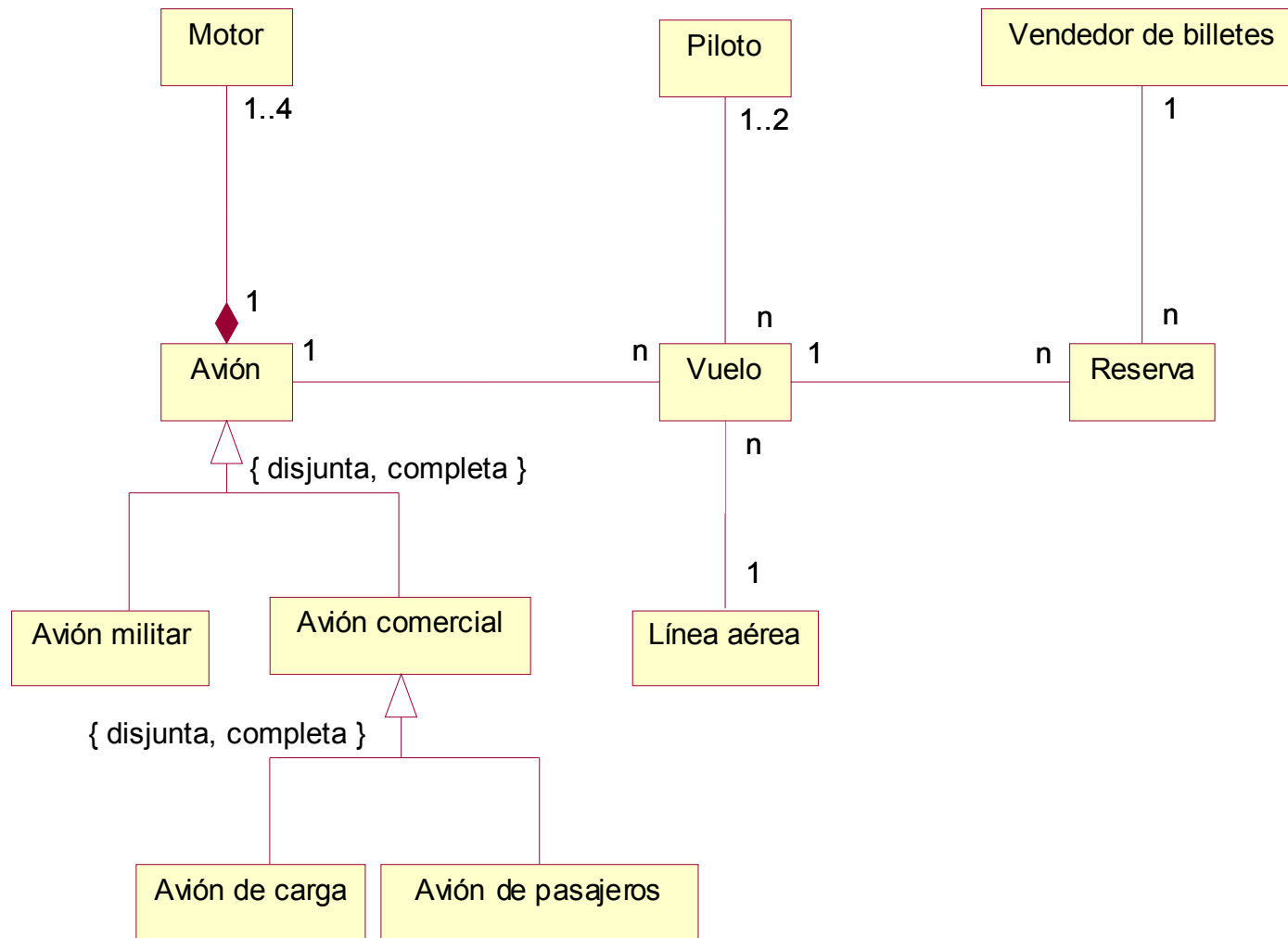


Diagrama de Máquinas de Estado

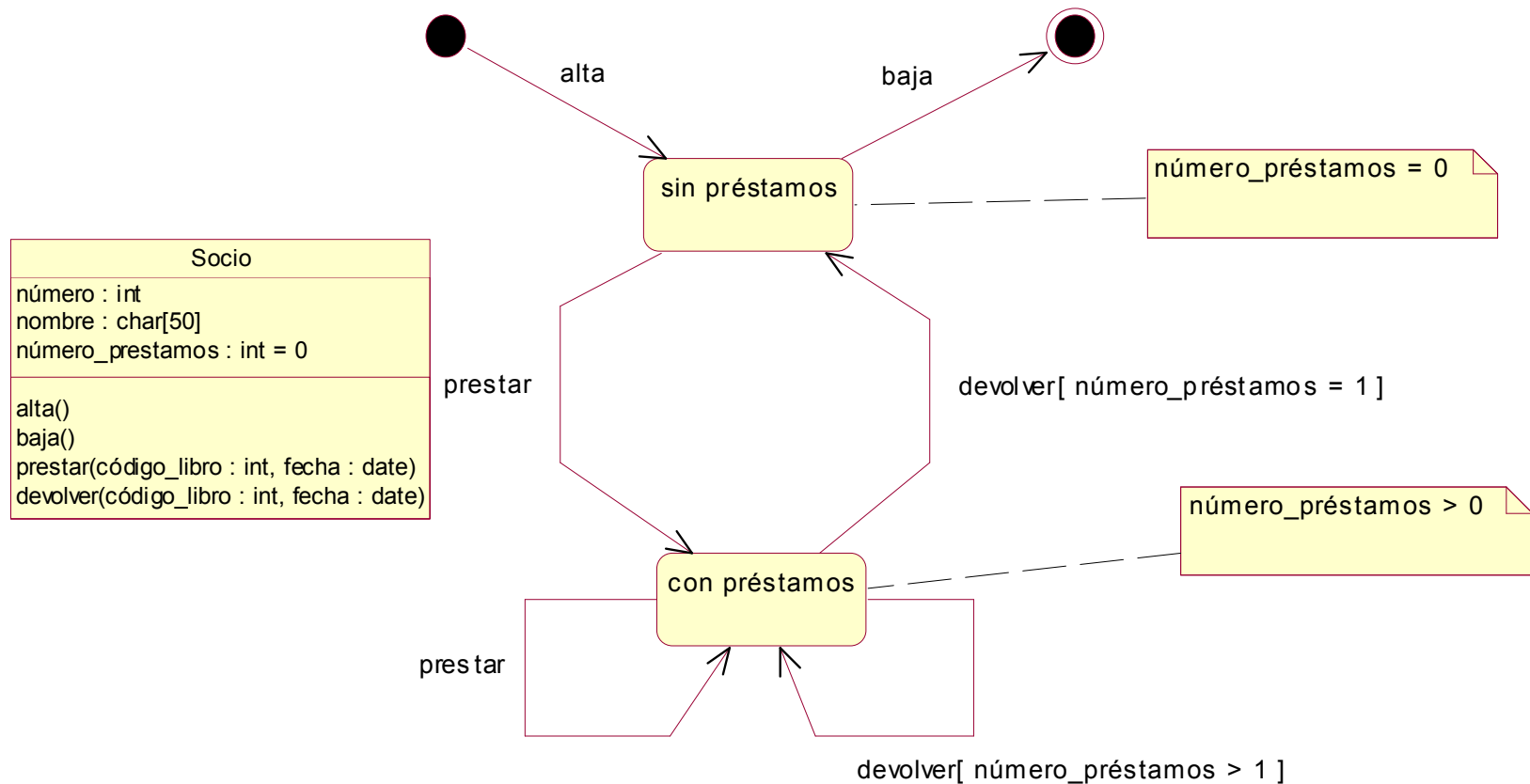


Diagrama de Actividad

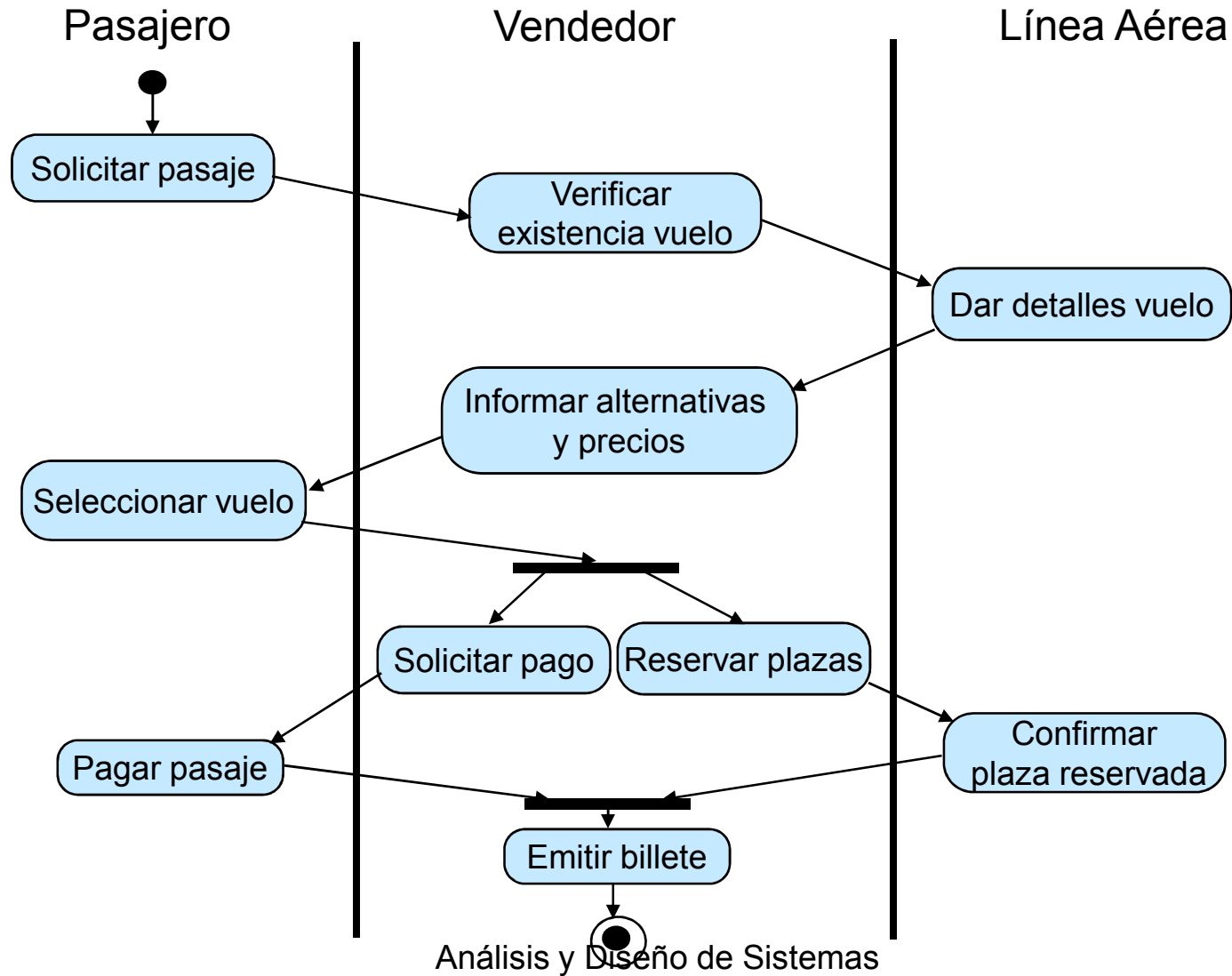


Diagrama Componentes

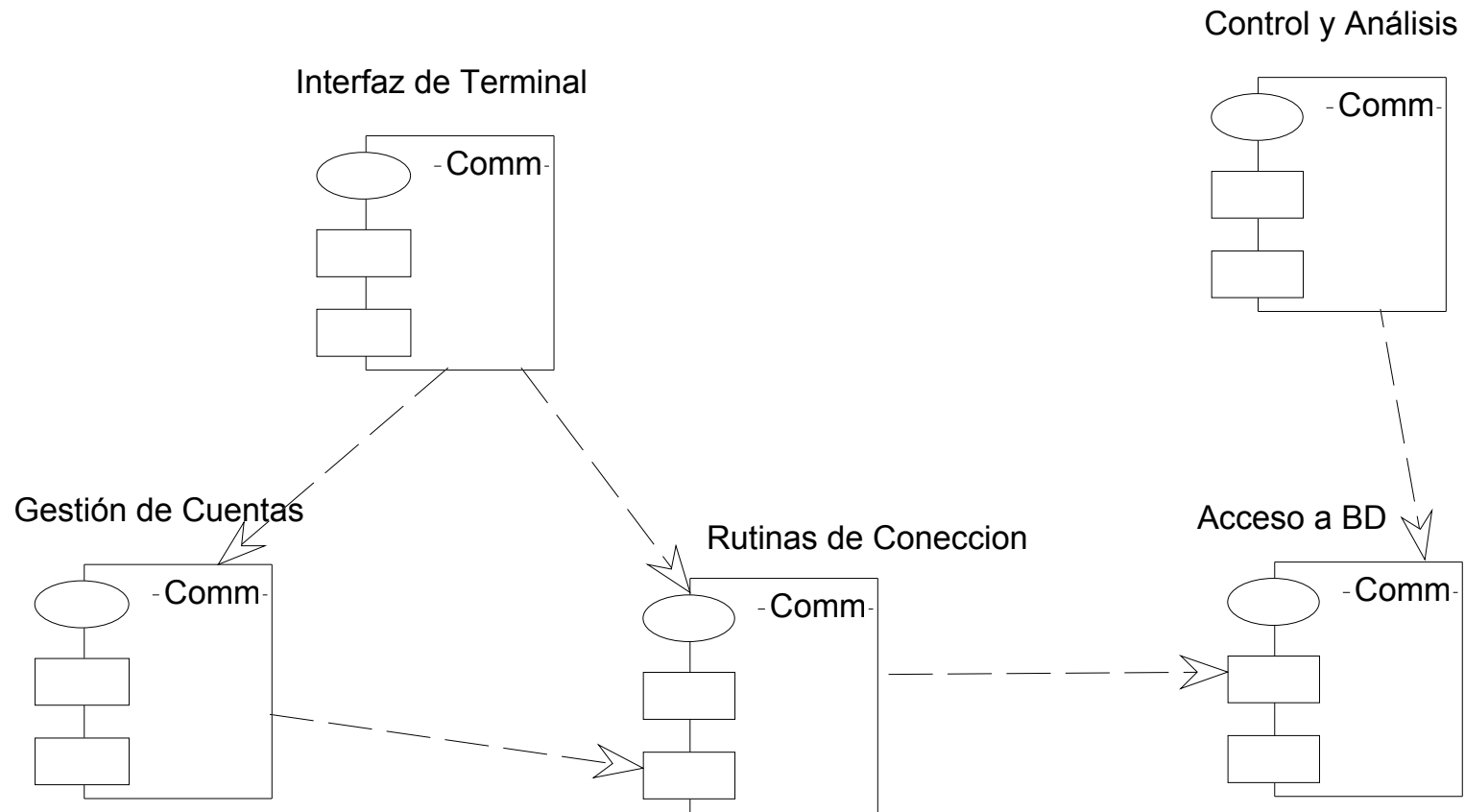
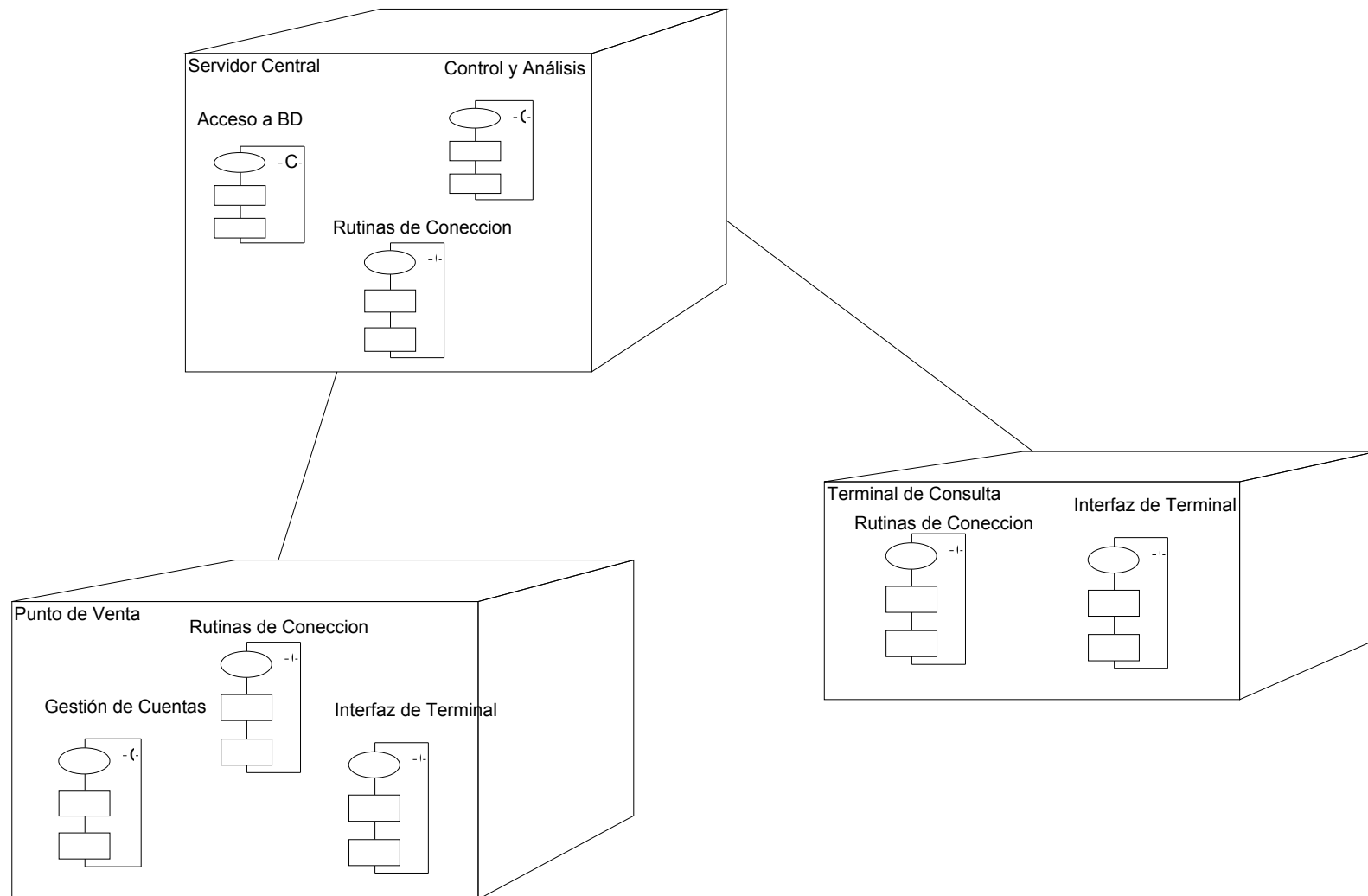


Diagrama de Despliegue



Resumen

- UML define una notación que se expresa como diagramas que sirven para representar modelos/subsistemas o partes de ellos
- *El 80 por ciento de la mayoría de los problemas pueden modelarse usando alrededor del 20 por ciento de UML-- Grady Booch*