

SOFTWARE INGENIERIA

Modelo de análisis



Caso de uso y uso

Diagramas de casos

- Cuándo: análisis de requisitos
- Diagramas de casos de uso: actores, casos de uso, asociación, límite del sistema
- Caso de uso: nombre, actor, descripción (flujo principal, flujo alternativo, relaciones con otros casos de uso)
- Relaciones de casos de uso: **incluir**, **extender**, generalización

Mapa vial

- Clase UML y diagramas de secuencia
- Romper
- Ideas salvajes y locas

Dr. Juan Saldaña

Análisis Modelo

- El modelo de análisis describe la estructura del sistema o aplicación que está modelando. Consiste en diagramas de clase y diagramas de secuencia que describen la implementación lógica de los requisitos funcionales que identificó en el modelo de caso de uso.

Modelo de análisis

- El modelo de análisis identifica las **principales clases** en el sistema y contiene un conjunto de realizaciones de casos de uso que describen **cómo se construirá el sistema. Los diagramas de clase describen la estructura estática del sistema** mediante el uso de estereotipos para modelar las partes funcionales del sistema. **Los diagramas de secuencia realizan los casos de uso al describir el flujo de eventos** en los casos de uso cuando se ejecutan. Estas realizaciones de casos de uso modelan cómo las partes del sistema interactúan dentro del contexto de un caso de uso específico.

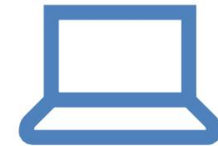


diagrama de clases UML

diagrama de clases UML



Cuándo usar

análisis de requisitos y diseño de arquitectura

representar clases y relaciones de clases



Elementos clave

clase

atributos

operaciones

relaciones entre clases

Diagrama de clases UML: clase y objeto

- “ Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica”. (Guía de usuario de UML, 1999)
- Perspectiva OOD: Una clase representa un concepto de solución. (Piense en clases de Java/C++)
- El objeto es una instancia de la clase.

Clase en UML

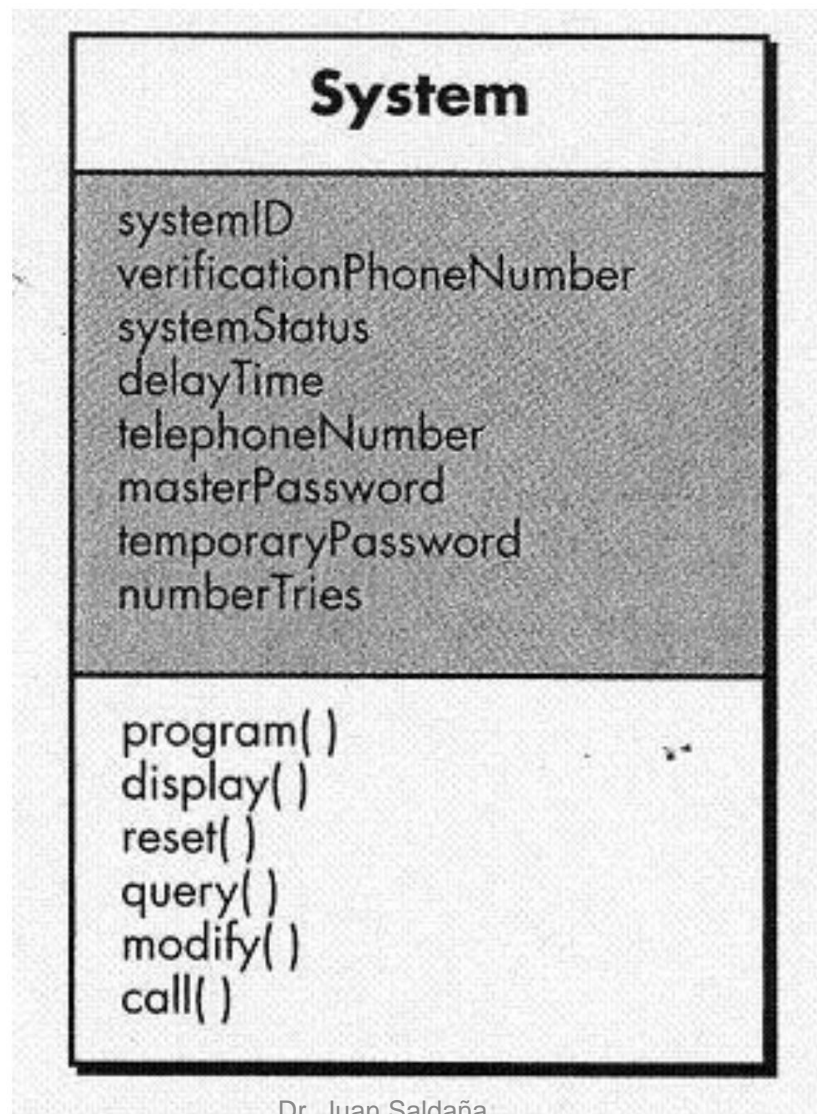
- Una clase encapsula estado (*atributo*) y comportamiento (*operaciones*).

- Cada atributo tiene un *tipo*. •

Cada operación tiene una *firma*. • El

nombre de la clase es la única información obligatoria.

Clase en UML



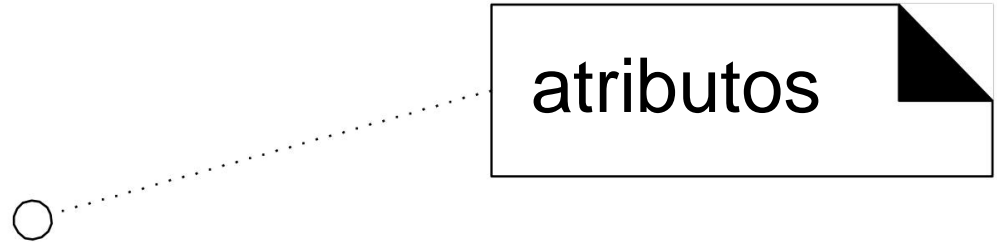
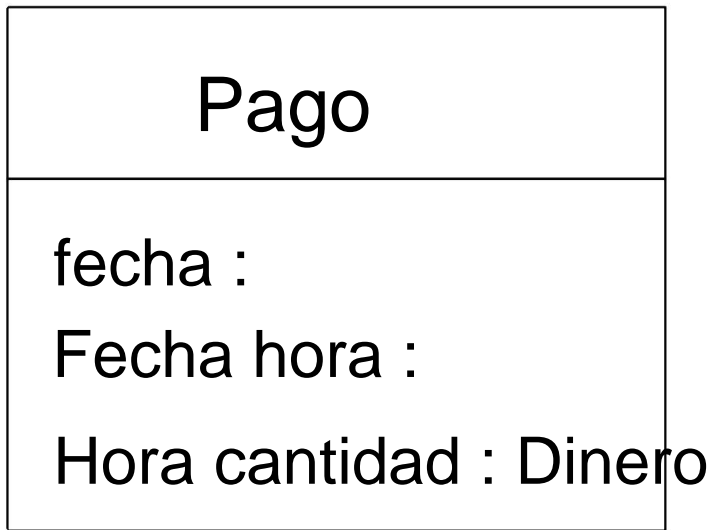
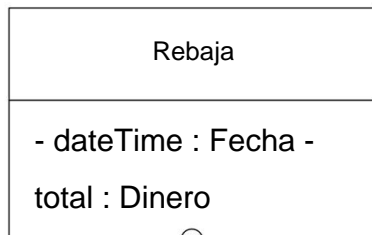


Diagrama de clases UML – atributos

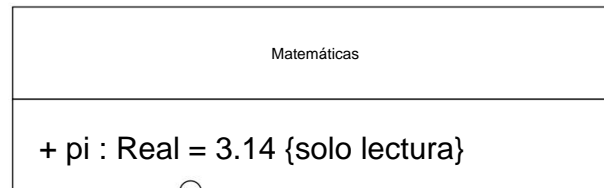
- atributo: una propiedad con nombre
- cada objeto tiene un valor
- tipos primitivos "simples"

Sintaxis para atributos en UML

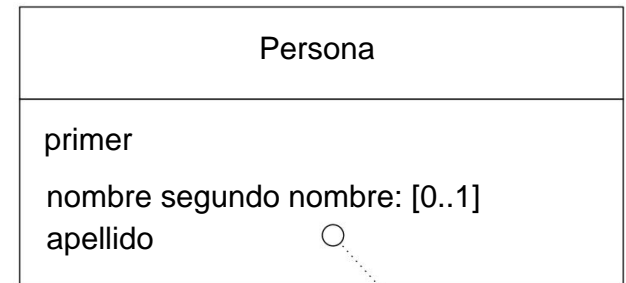
[visibilidad] nombre [multiplicidad] [: tipo] [= valor inicial]



Atributos de
visibilidad privada



Atributo de solo lectura de
visibilidad pública con inicialización

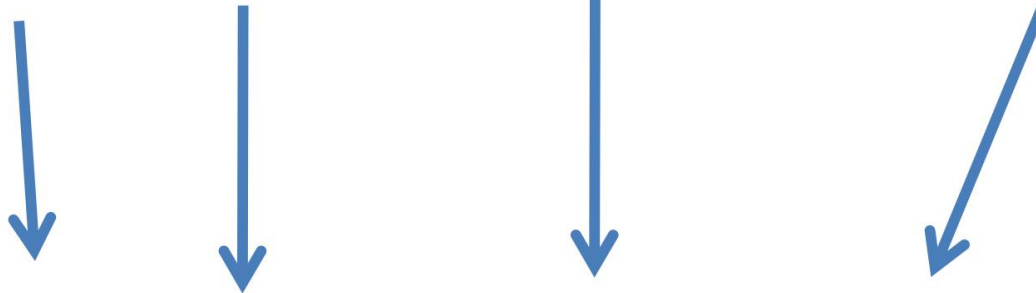


Valor opcional

Operaciones:

Sintaxis

[visibilidad] nombre [(lista de parámetros)] [: tipo de retorno]

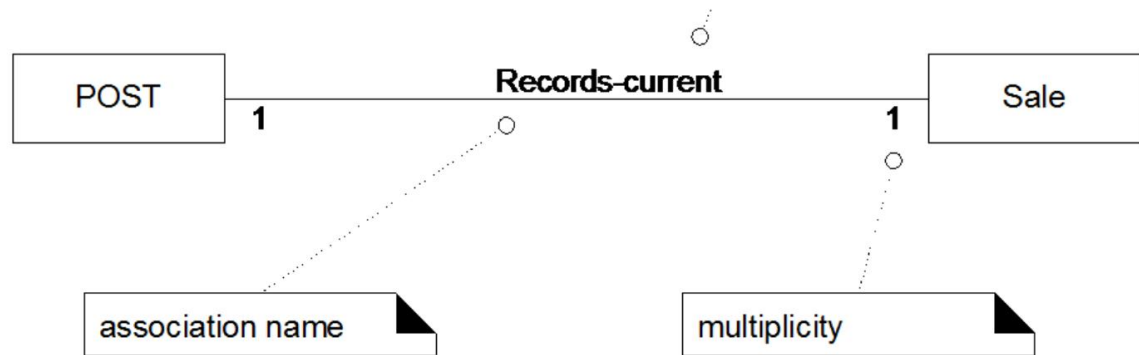


+ balanceEn (fecha: Fecha) : Dinero

Clase Relaciones

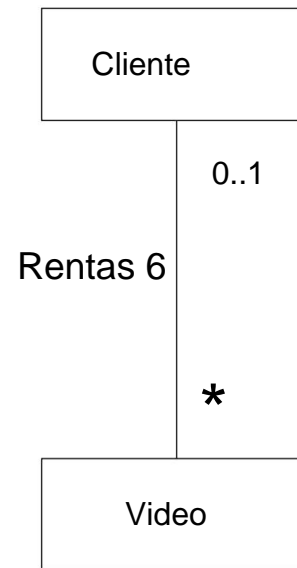
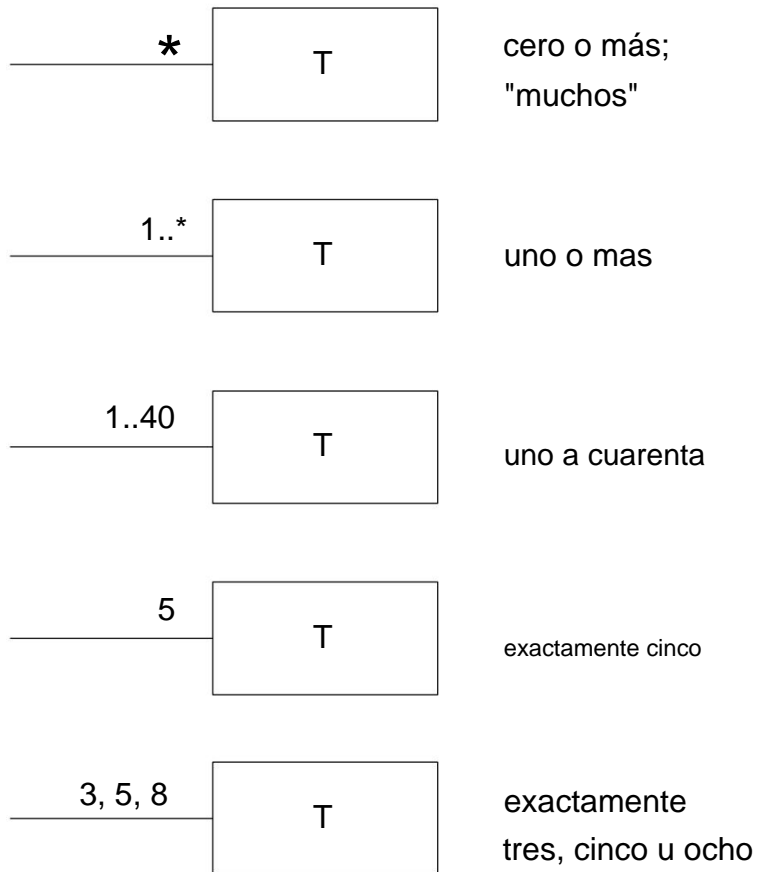
- Asociación: relación
estática compartida entre
los objetos de dos clases
 - Agregación
 - Composición
- Generalización:
estructuras de nivel de clase de
generalización/especialización

Asociación



- Línea sólida desde la clase de origen hasta la clase de destino (la flecha es opcional)

Multiplicidad



Una instancia de un Cliente puede estar alquilando cero o más Videos.

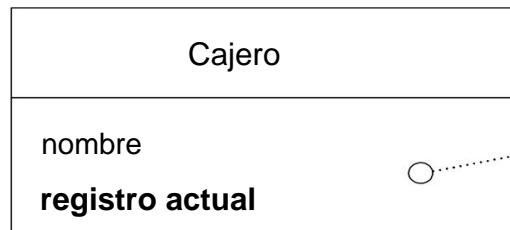
Una instancia de un Video puede ser alquilada por cero o un Cliente.

Asociación VS Atributo

No suele poner la asociación como un atributo en la clase.

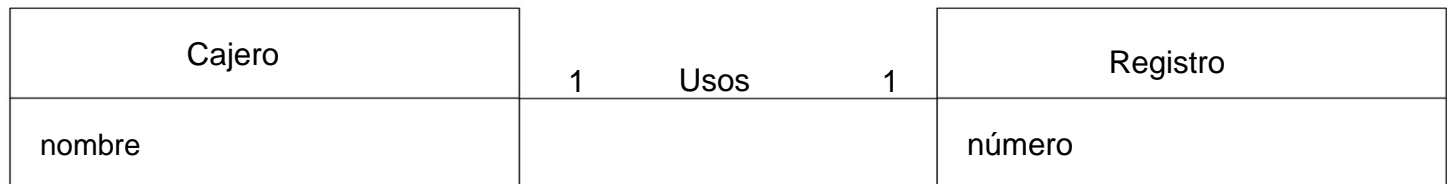
Las conexiones con otros conceptos deben representarse como asociaciones, no como atributos.

Peor



no es un atributo de "tipo de datos"

Mejor



Agregación

- Si la asociación transmite la información de que un objeto es parte de otro objeto ("tiene-a"), pero sus tiempos de vida son independientes (podrían existir de forma independiente).
- La agregación es unidireccional, a diferencia de las asociaciones en las que las clases tienen el mismo estatus
- Hay un contenedor y uno o más objetos contenidos.
 - Por ejemplo, podemos decir que “un Departamento contiene un conjunto de Empleados”, o que “una Facultad contiene un conjunto de Profesores”.

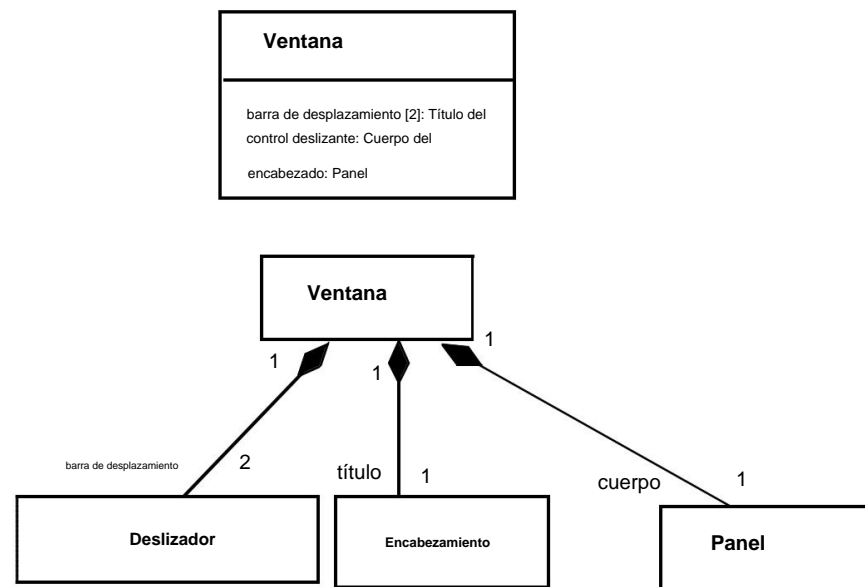
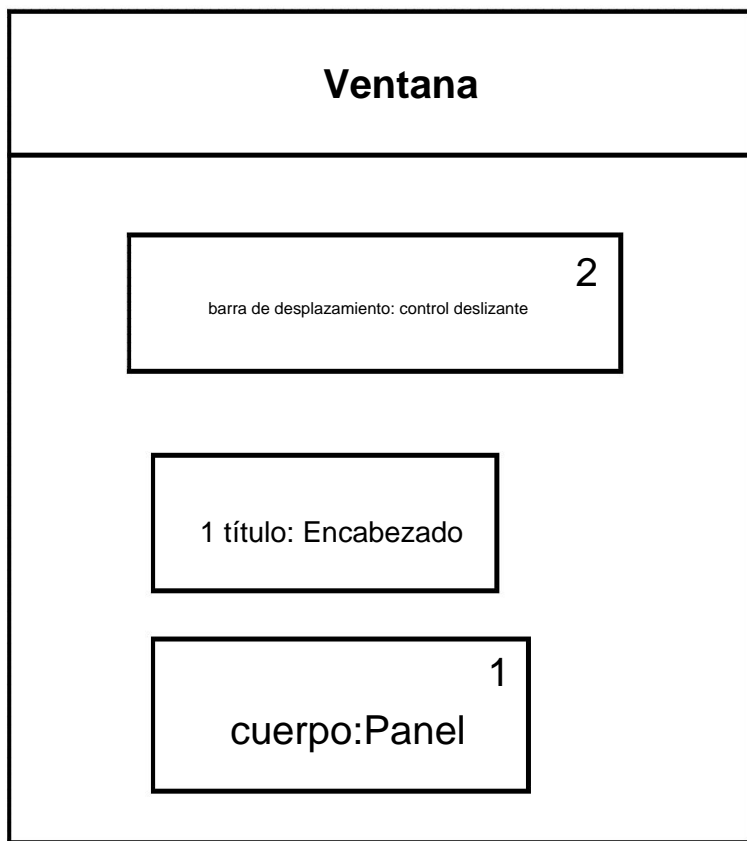


Composición

Cuando un objeto está contenido en otro objeto, y puede existir solo mientras exista el contenedor y solo existe para el beneficio del contenedor.

- Una agregación es una forma especial de asociación; composición es una forma más fuerte de agregación.
- la multiplicidad en el extremo entero debe ser 1 o 0..1
- Tanto la agregación como la composición son una jerarquía parte-todo.

Composición



Asociaciones de modelos

- Si un objeto es parte de otro objeto y no existe una dependencia de existencia entre los objetos, modele como una agregación
- Si la parte es existencialmente dependiente del todo, modelo como composición
- Si existe una relación conceptual entre dos objetos pares, modele como una asociación general

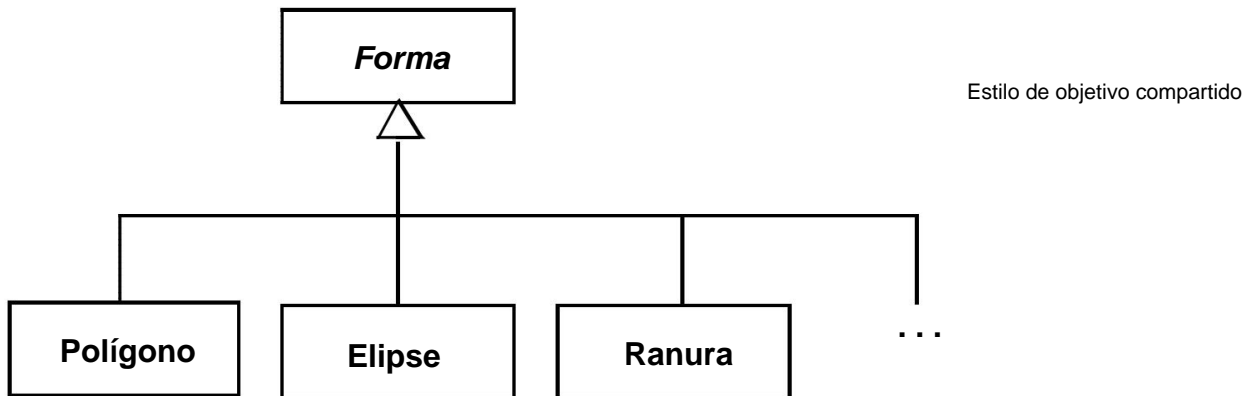
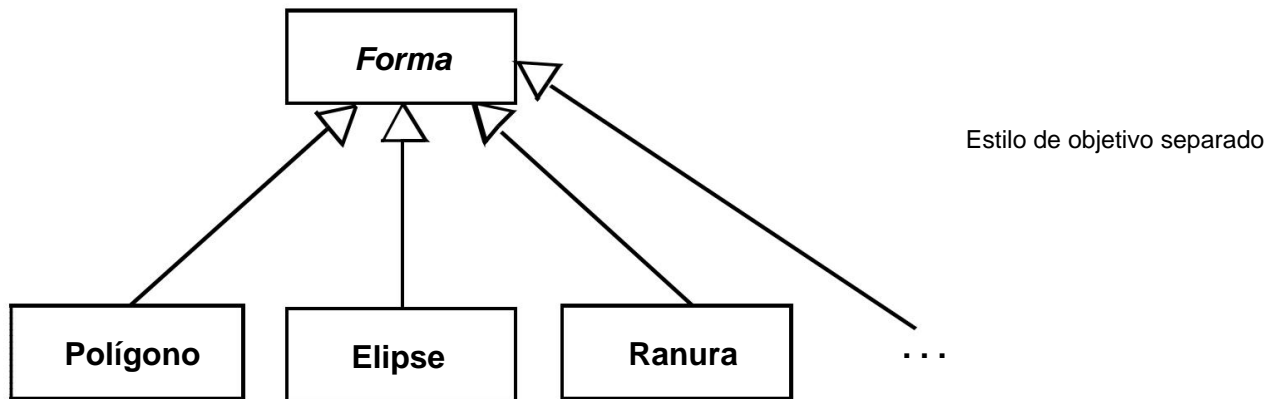
Generalización

- Una generalización (o especialización) es una relación entre un concepto general y sus especializaciones.
 - Ejemplo: *Ingeniero Mecánico* e *Ingeniero Aeronáutico* son especializaciones de *Ingeniero*
- Las relaciones de generalización denotan herencia entre clases.
- Las clases secundarias heredan los atributos y operaciones de la clase padre.

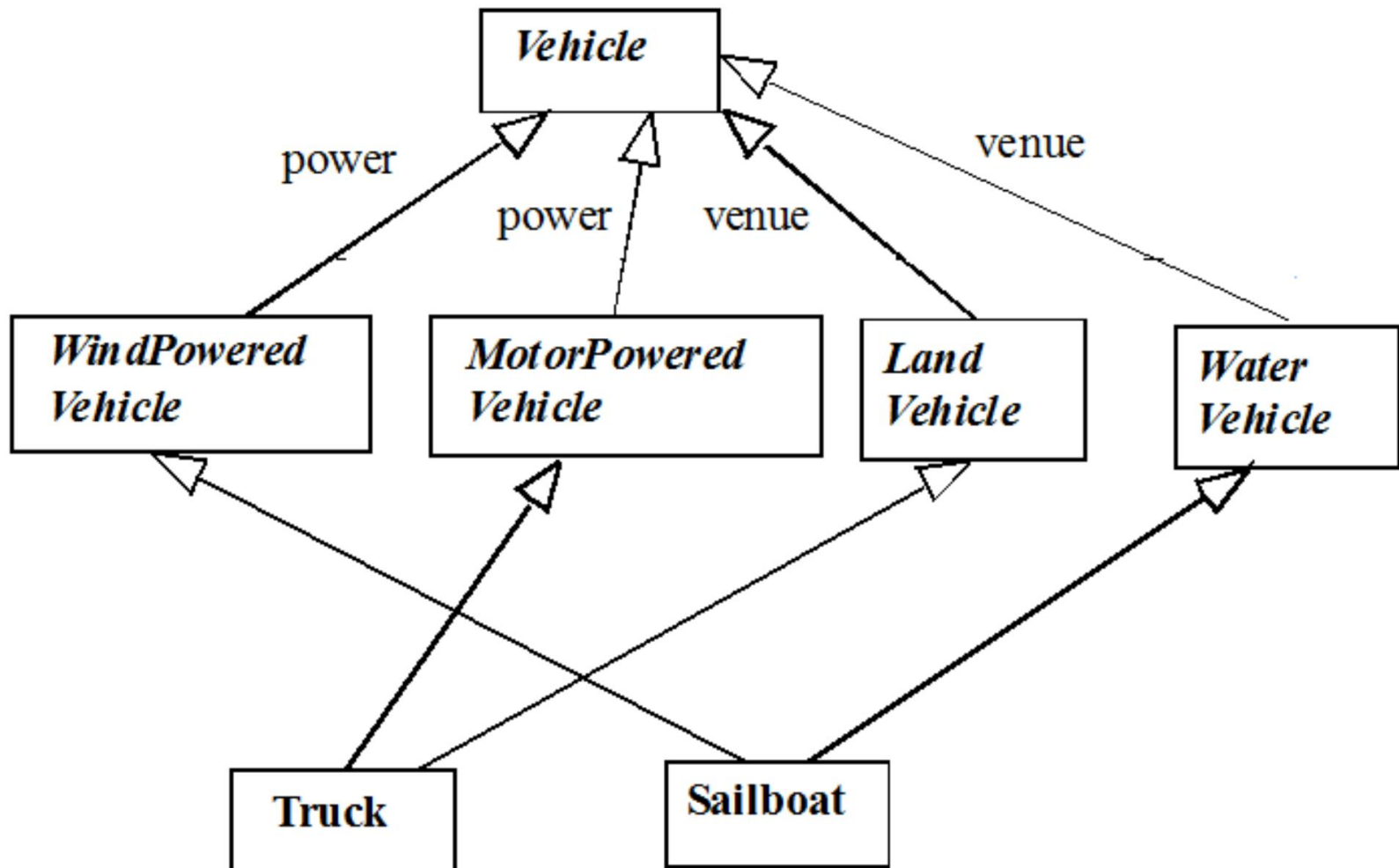
Representación de generalizaciones

- La generalización se presenta como una línea continua dirigida con una gran punta de flecha abierta.
 - La punta de flecha apunta hacia la generalización
- Se puede usar un discriminador para identificar el naturaleza de las especializaciones

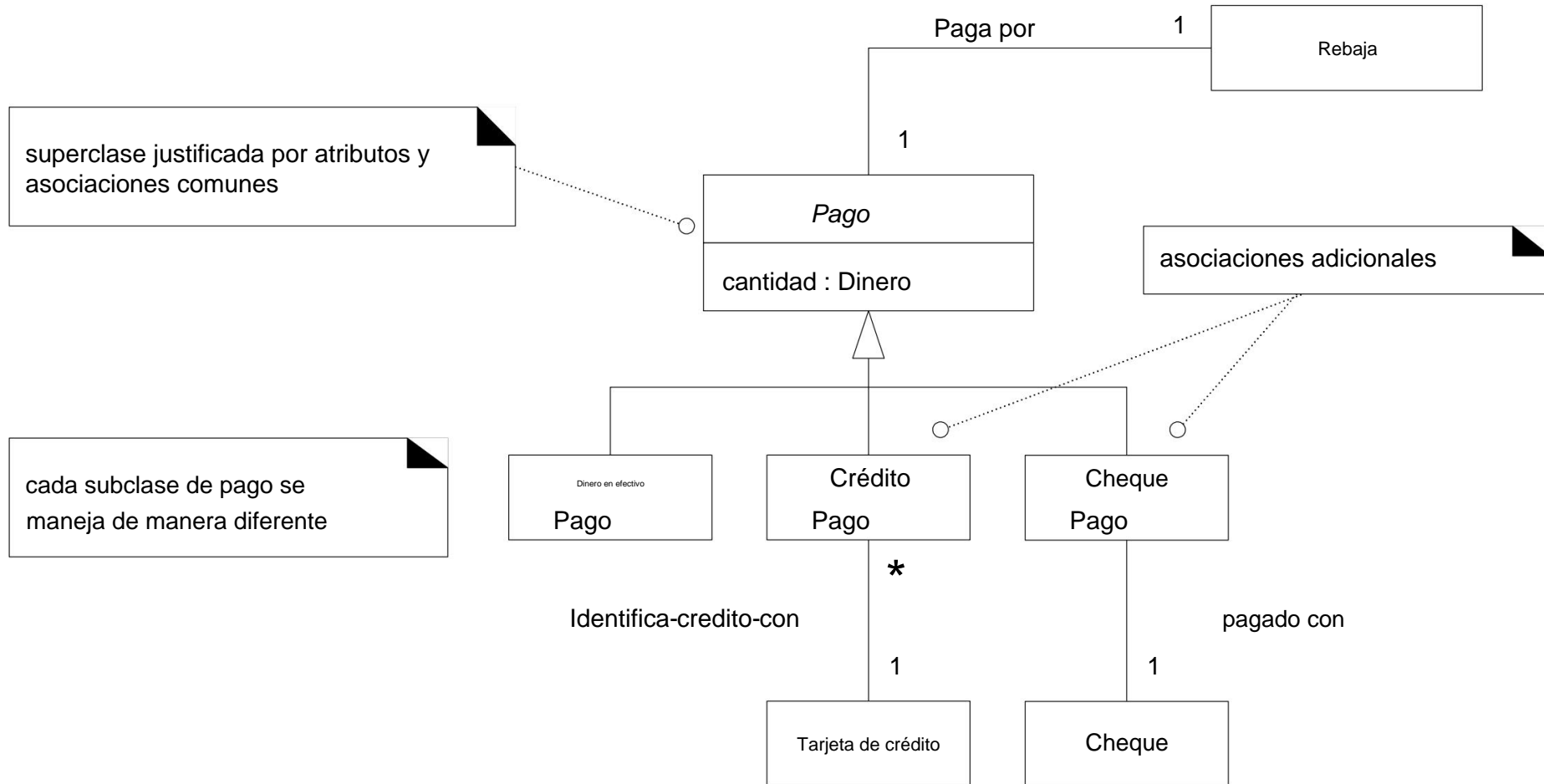
Generalización



Generalización



Herencia de asociaciones



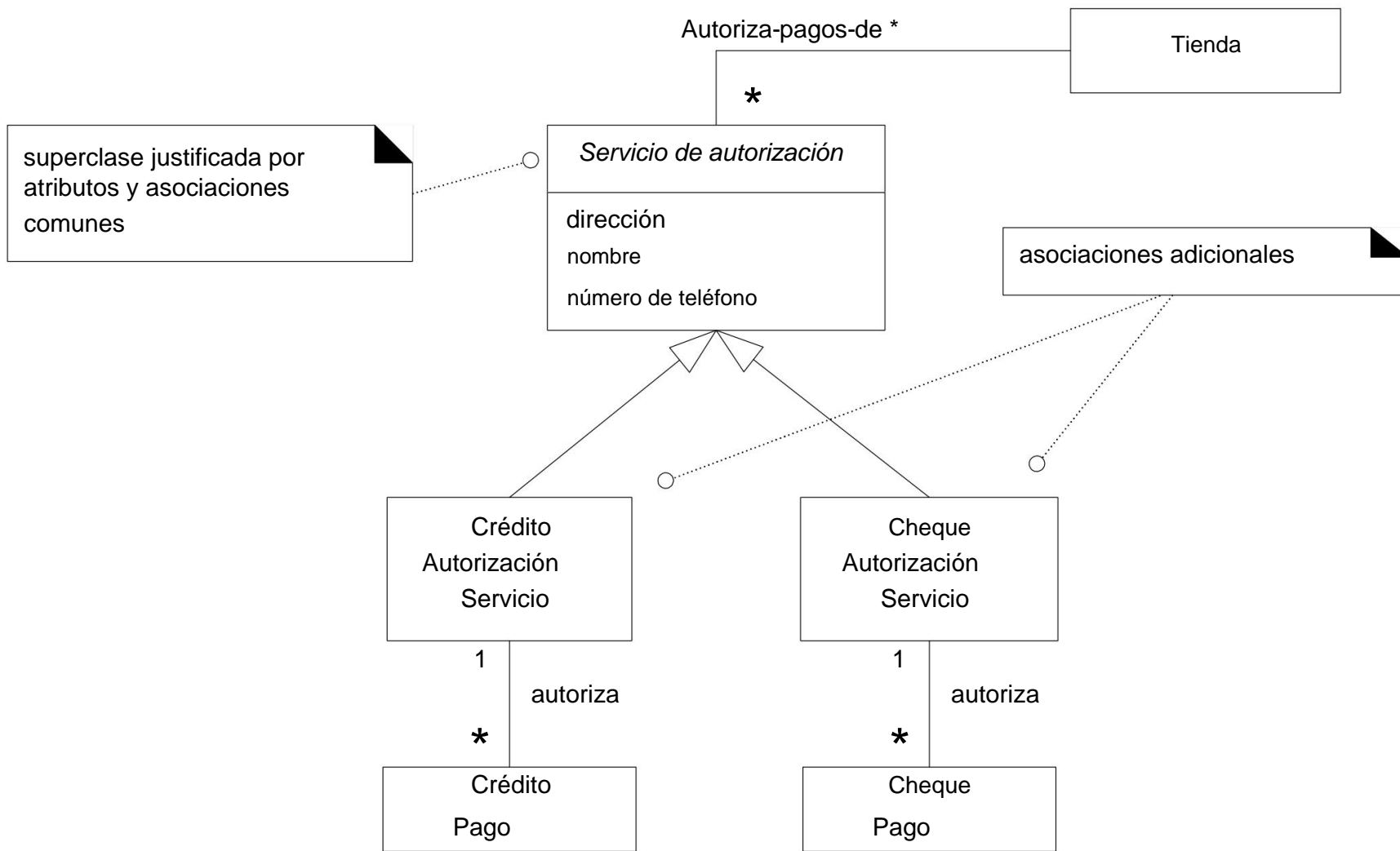
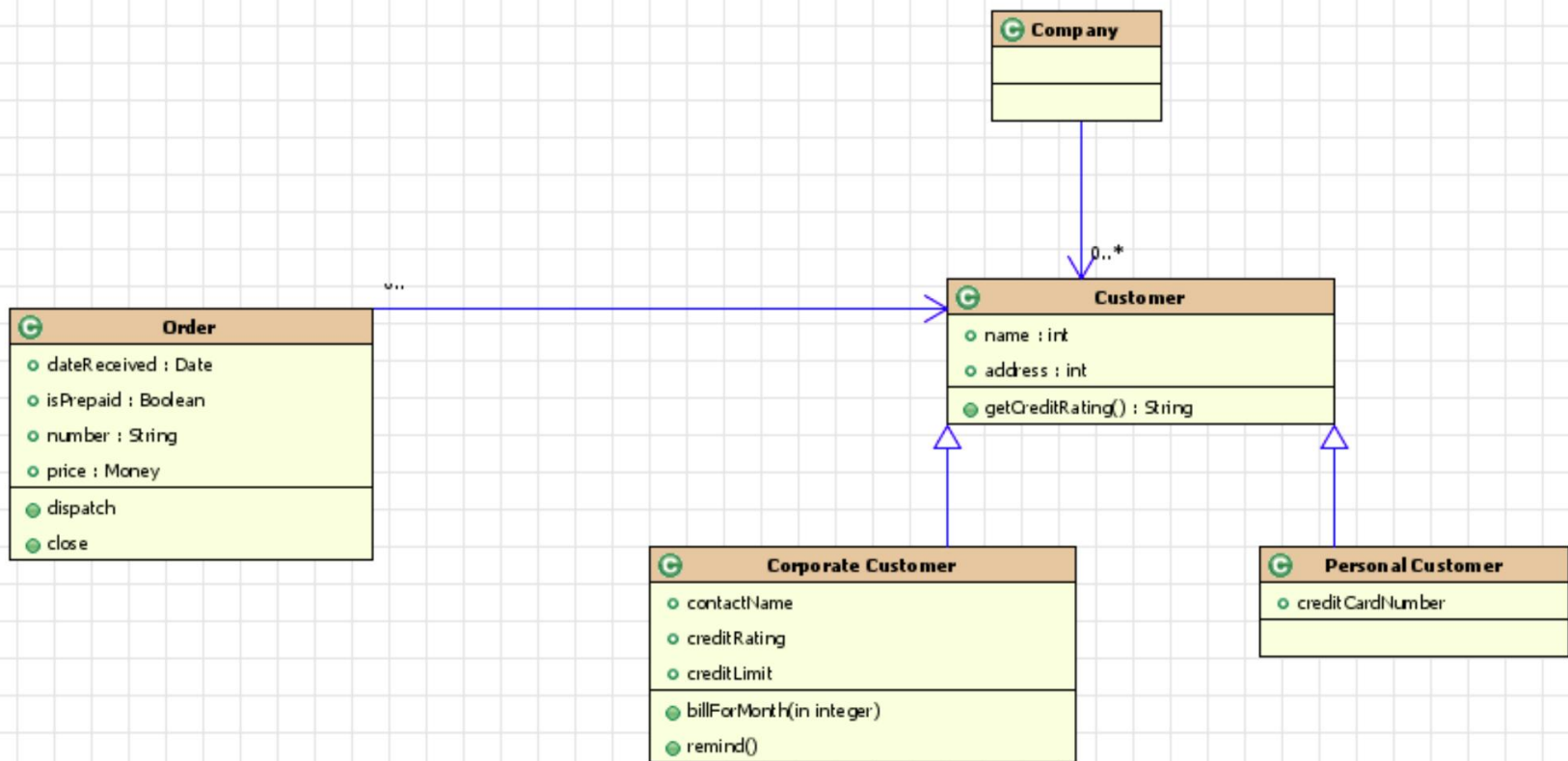
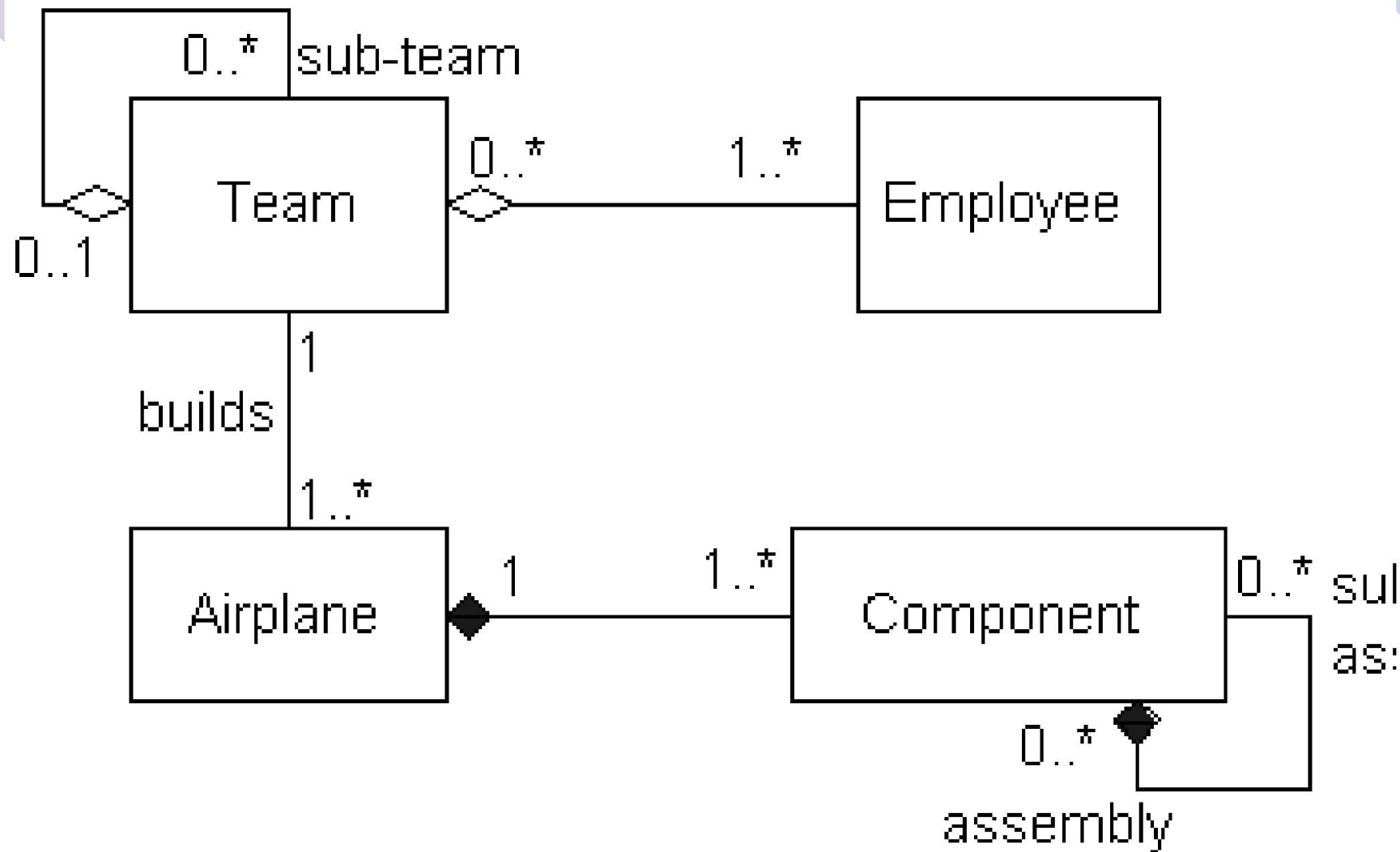


Diagrama de clase de construcción

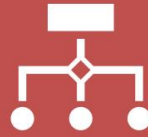
La empresa XYZ tiene dos tipos de clientes, clientes corporativos y clientes personales.

Todos los clientes pueden hacer pedidos. Cada pedido es realizado por un cliente.

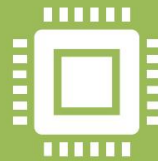




Interfaces (clase abstracta)



Una *interfaz* es una colección de operaciones con nombre que representan los servicios proporcionados por una clase o componente.



Las interfaces *no* tienen atributos.

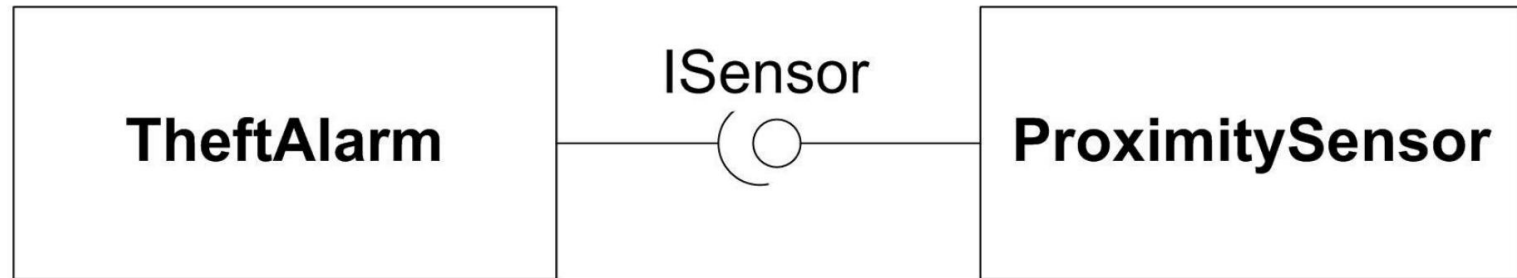
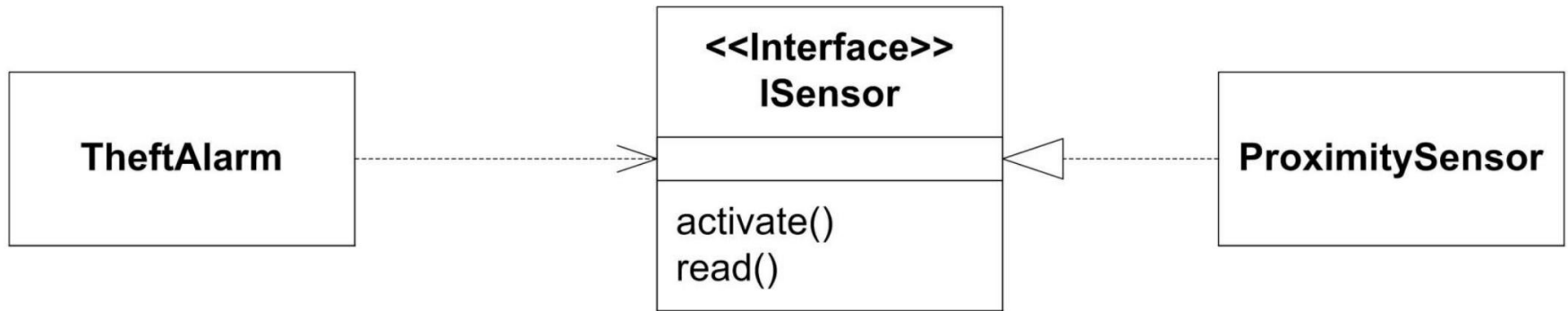


Las interfaces *no* proporcionan implementaciones para sus operaciones.

Relaciones

- Relaciones de interfaz
 - Asociaciones (deben estar dirigidas A la interfaz)
 - Generalizaciones
- Una clase o componente puede realizar una interfaz
 - La clase/componente proporciona la implementación de los servicios especificados por la interfaz
 - Una clase/componente puede realizar una o más interfaces

Representando interfaces



Modelos de comportamiento

- Los modelos de clase describen elementos conceptuales y sus relaciones, el comportamiento no es el enfoque principal de dicho modelo
- Modelos de comportamiento a nivel de requisito
 - Modelos de interacción: describen las interacciones entre los actores y el sistema – diagrama de secuencia
 - Contratos: Especifique las operaciones invocadas por los actores.

diagrama de secuencia U

Diagrama de secuencia

Un diagrama de secuencia muestra la secuencia de mensajes intercambiados por el conjunto de objetos que realizan una determinada tarea.

Se utiliza durante el análisis de requisitos • Para
refinar las descripciones de casos de uso • Para
buscar objetos adicionales

Se utiliza durante el diseño del sistema para refinar las interfaces del subsistema.

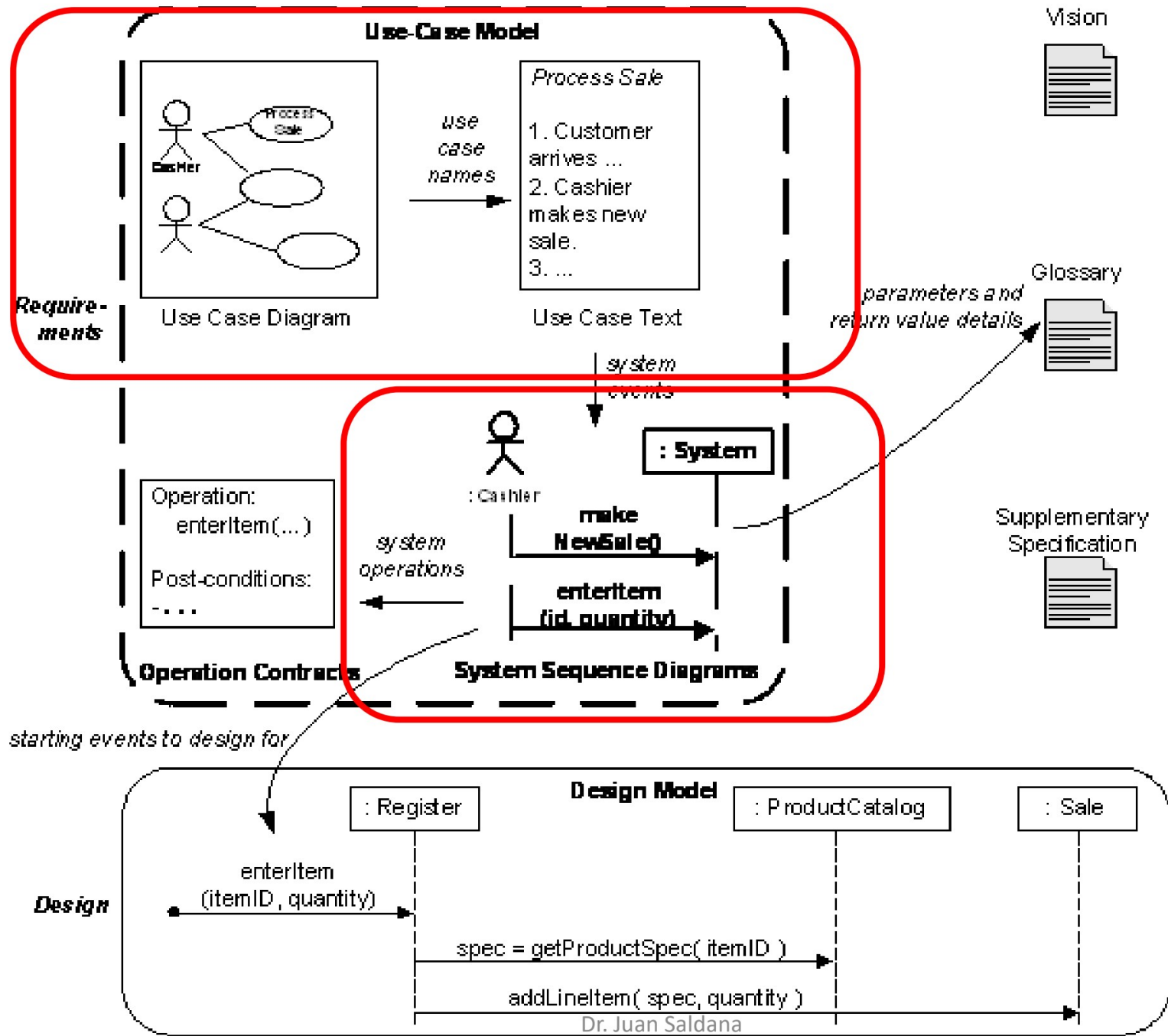


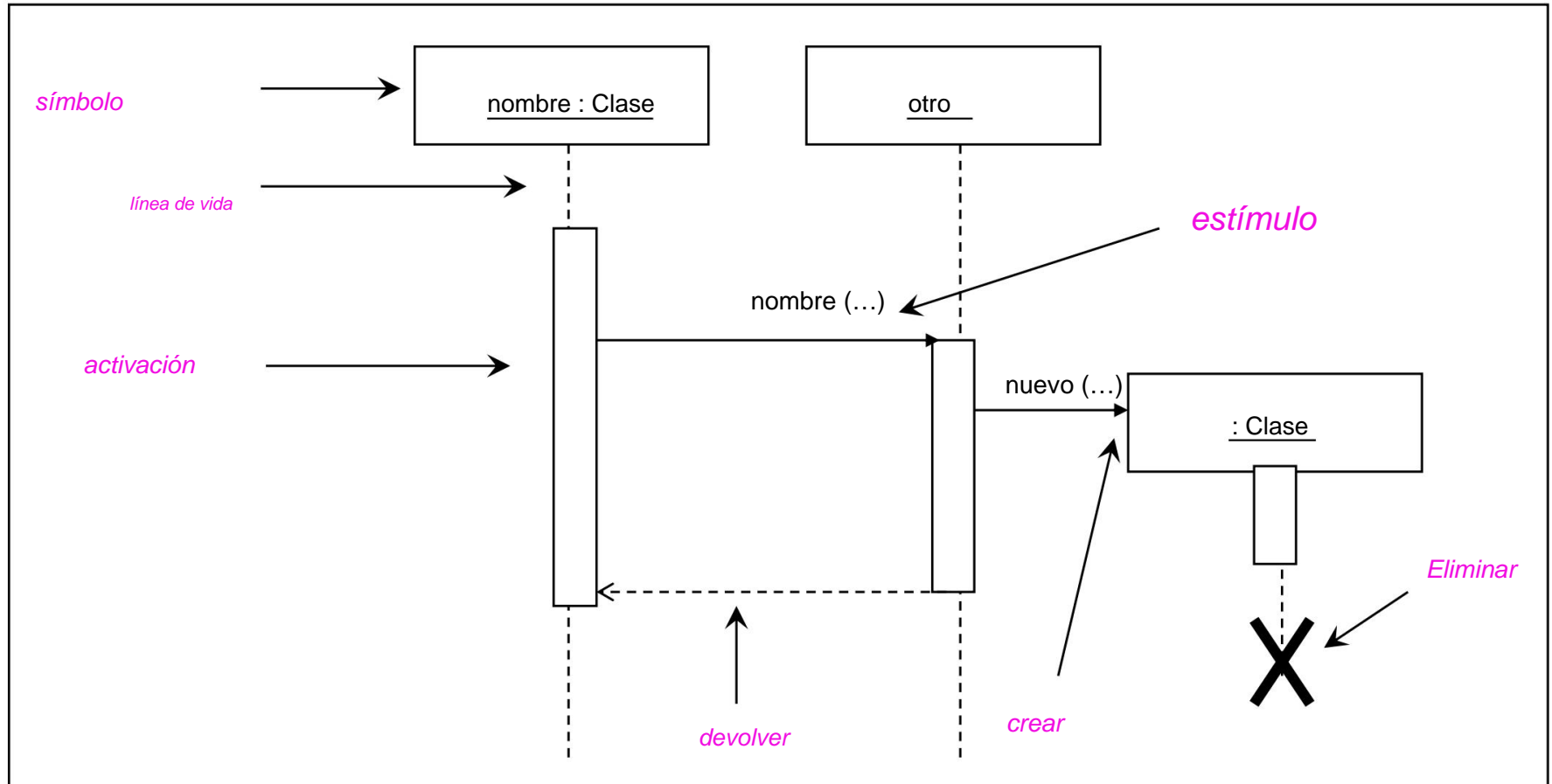
Diagrama de secuencia

1. **Las clases** están representadas por columnas (la primera columna es el actor que inicia el caso de uso)
2. **Los mensajes** están representados por flechas
3. **Las activaciones** de una operación se representan mediante rectángulos estrechos

Diagrama de secuencia

- Los **objetos** están dispuestos horizontalmente a lo largo del diagrama.
- Un **actor** que inicia la interacción a menudo se muestra a la izquierda.
- La **dimensión vertical representa el tiempo**.
- Se adjunta una línea vertical, llamada línea de **vida**, a cada objeto o actor.
- La línea de vida se convierte en un cuadro ancho, llamado *cuadro de activación* durante el período de **activación en vivo**.
- Un **mensaje** se representa como una flecha entre los cuadros de activación del emisor y del receptor.
 - Un mensaje está etiquetado y puede tener una lista de argumentos y un valor de retorno.

Diagrama de secuencia resumen de construcciones básicas



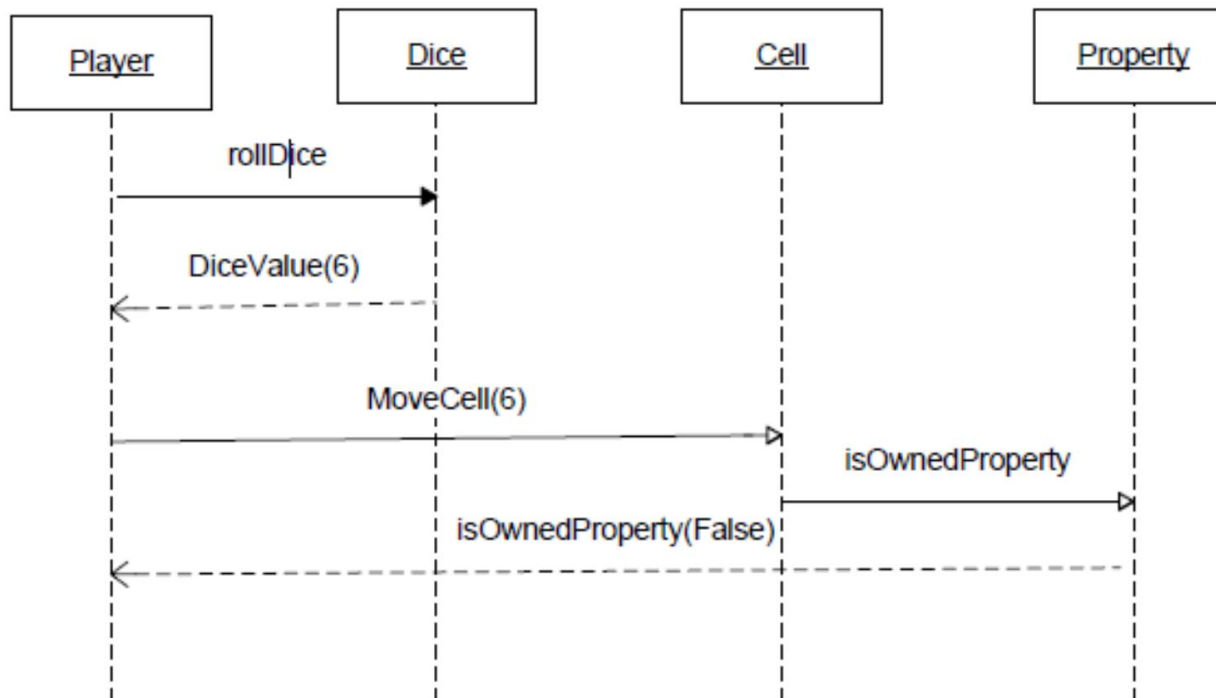
Ejemplo

- Un jugador tira los dados y obtiene un 6. El jugador mueve 6 celdas. El jugador aterriza en una celda que es una propiedad sin dueño . El turno del jugador es terminado.

Ejemplo

Un **jugador** tira los **dados** y obtiene un 6. El jugador mueve 6 **celdas**. El jugador aterriza en una celda que es una propiedad sin dueño . El turno del jugador ha terminado.

Not all nouns become objects such as "turn"



Condicional Lógica

- Si el jugador aterriza en una celda que es una propiedad sin dueño, el turno del jugador es terminado
- Si el jugador aterriza en una celda que es de su propiedad, el jugador debe pagar el alquiler al dueño de la propiedad
- Entonces, el turno del jugador es terminado.

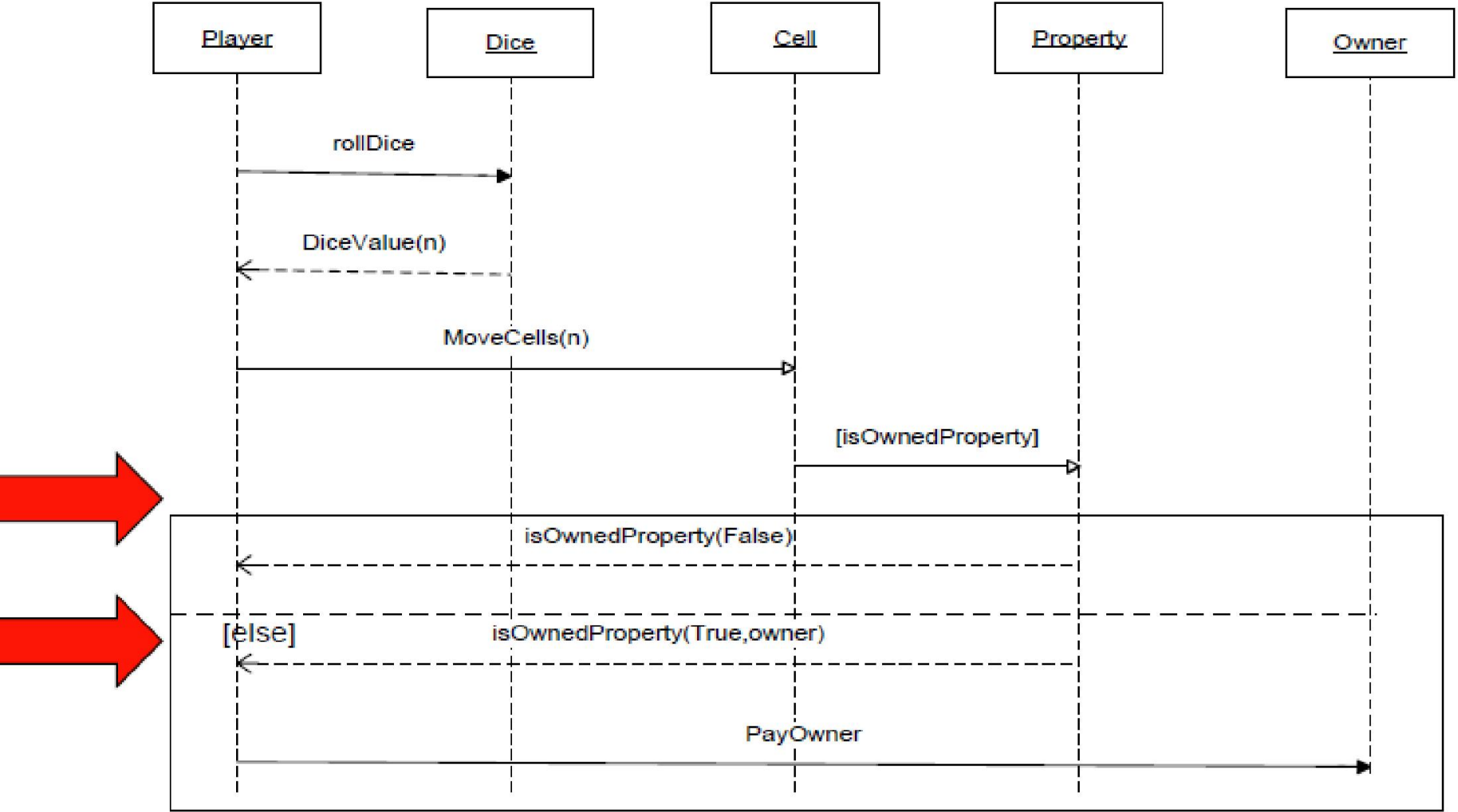
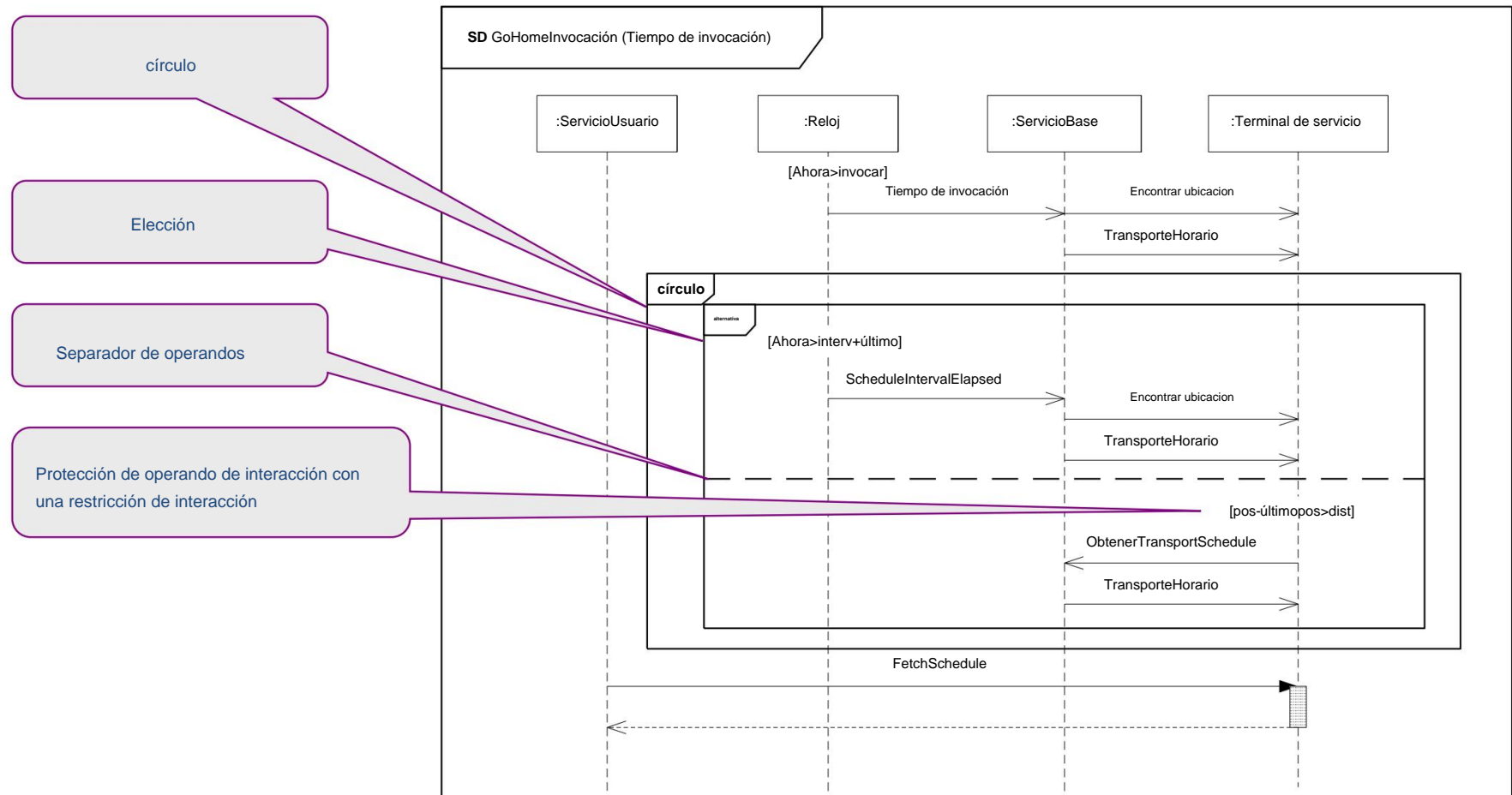


Diagrama de secuencia más complicado



Sesiones de ideas salvajes y locas



Código abierto VS código cerrado

Referencia

Buenos enlaces para información de diagramas de secuencia.

<http://www.websequencediagrams.com/> [http://](http://www.websequencediagrams.com/examples.html)

[www.websequencediagrams.com/examples.h](http://www.websequencediagrams.com/examples.html)

[tml](http://www.websequencediagrams.com/examples.html)