

INGENIERÍA DE SOFTWARE I

Unidad V

**Tema: RUP Ágil.
Modelado de requerimientos
y de dominio**

UNTDF – 2020

Caso de Estudio: Sistema de Punto de Venta (PDV)

**El presente Sistema será utilizado como Ejemplo a
través del todo el Estudio del Proceso RUP Agil**

Caso de Estudio: Sistema de Punto de Venta (PDV)

- Un sistema PDV es una aplicación informática utilizada para registrar ventas y realizar pagos.
- Incluye: componentes hardware: computadora, lector de código de barras,... y software para ejecutar el sistema
- Interactúa con aplicaciones de servicio: servicio de cálculo de impuestos y control de inventario.

Ejemplo: Sistema de Punto de Venta (POS)

Punto de Venta para ventas al detal: POS (Point of Sale)

Uso principal:

 registrar ventas
 y manejar los pagos



Compuesto por:

 Computador,
 Lector de código de barra
 Software del Sistema

Ilustración extraída de:
Applying UML and patterns.
2º edición
C. Larman
Prentice Hall, 2002

:

RUP Agil. Workflows

- Fase de Inicio en RUP Agil
- Workflow de Requerimientos. Modelo de CU
- Fase de Elaboración. Iteración 1
- Workflow de Modelado de Negocio
 - Modelo del Dominio

RUP Agil. Dónde estamos???

Fase de Inicio

Definiendo pasos y artefactos del Inicio

Fase de Inicio en RUP Agil

- ▶ Esta fase es un paso inicial corto para establecer una visión común y el alcance básico del proyecto
- ▶ No es una fase de requerimientos (**confusión del pensamiento en cascada**)
- ▶ Incluye el análisis del 10% de CU y el análisis de los requerimientos no funcionales críticos
- ▶ Se prepara el ambiente de desarrollo de tal modo que la programación se desarrolla en la Fase de Elaboración

Artefactos de la Fase de Inicio

Artefactos importantes que se construyen:

- **Visión:** descripción del proyecto a alto nivel (gestión)
- **Modelo de casos de uso:** Requerimientos funcionales (esenciales!),
- **Especificación suplementaria:** explicitando requerimientos no funcionales
- **Glosario:** se necesita para una comunicación clara
- **Marco de desarrollo:** el proceso real utilizado y los artefactos a entregar. Descripción de pasos y artefactos de RUP adaptados al proyecto actual.
- **Otros:** lista de riesgos, prototipos, plan de iteración (elaboración01), estimaciones de la próxima fase del plan de desarrollo del SW, etc.

Fase de Inicio

Disciplina: Requerimientos

Modelo de Casos de Uso

Requerimientos

- ▶ ¿Qué deberá hacer el sistema?
- ▶ ¿En qué condiciones deberá hacerlo?
- ▶ ¿Qué cualidades o atributos deberá poseer el sistema?

Tipos y Categorías de Requerimientos

- ▶ En UP los requerimientos se clasifican de acuerdo al Modelo FURPS+ (Grady, 92) en:
 - **Funcionales y**

- **No Funcionales:**

Atributos de calidad:

Facilidad de Uso
Fiabilidad
Rendimiento
Soporte

+

Implementación
Interfaz
Operaciones
Empaquetamiento
Legales

Organización del Workflow de Requerimientos en Artefactos UP

- ▶ **Modelo de Casos de Uso:** Requerimientos funcionales
 - ▶ **Especificación suplementaria:** Requerimientos no funcionales, características, etc.
 - ▶ **Glosario:** Define términos significativos. Registra requerimientos relacionados a datos, tales como reglas de validación, valores aceptables, etc.
 - ▶ **Visión:** Sintetiza los requerimientos de alto nivel y el caso de Negocio del Proyecto
 - ▶ **Reglas de Negocio:** Describen requerimientos o políticas que trascienden el software (por ej., regulacion de impuestos)
-
- ▶ *Estos artefactos se construyen a través de varias fases e iteraciones, con el mayor esfuerzo en la Fase de Inicio*

Modelo de Casos de Uso

- ▶ Los requerimientos funcionales del sistema se estructuran de manera natural mediante Casos de Uso del sistema.
 - ▶ Los CU son documentos de texto, NO solo Diagramas.
 - ▶ El Modelo de CU provee un buen Diagrama del contexto del sistema y su entorno (casos de uso y actores).
-
- ▶ Actividades a realizar para el desarrollo del modelo:
 - Encontrar actores y casos de uso, Describir Casos de uso, describir el modelo completo.
 - Completar el Modelo con una lista de requerimientos no funcionales
 - Construir un prototipo de interfaz de usuario para cada Caso de Uso.

Importancia de Casos de Uso

- ▶ Los CU son valiosos porque enfatizan las metas y perspectiva del usuario
 - Hacemos las preguntas: Quién usa el sistema?, Cuáles son sus escenarios típicos de uso? Cuáles son sus objetivos?
- ▶ Los CU son vitales en RUP, por cuanto:
 - Se fomenta el desarrollo conducido por Casos de uso.
 - Son parte importante de la planificación iterativa, dado que el trabajo de una iteración se define eligiendo CU.
 - Las “realizaciones” de CU dirigen el análisis y el diseño.
- ▶ En RUP Agil el modelo de casos de uso comienza en la fase de inicio con el 10% de los CU escritos con detalle
 - Es decir que el modelo de CU comienza en la fase de inicio y se refina en iteraciones de la fase de Elaboración y siguientes

Modelo de Casos de Uso

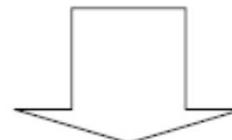
¿Qué hace el actor?



¡Usa el sistema !

El actor interactúa con el sistema.

Se realiza una secuencia específica de acciones



Escenario

Instancia de Caso de Uso:

Tipos de Casos de Uso

- ▶ Dependen de lo que se esté modelando (Negocio, Sistema, Subfunciones)

Objetivos Estratégicos → casos de uso del negocio

- Ejemplo: Evitar pérdidas

Objetivos de Usuario → casos de uso

- Ejemplo: Realizar Venta

Subfunciones → ¿casos de uso?

- Ejemplo: Iniciar sesión (*Login*)

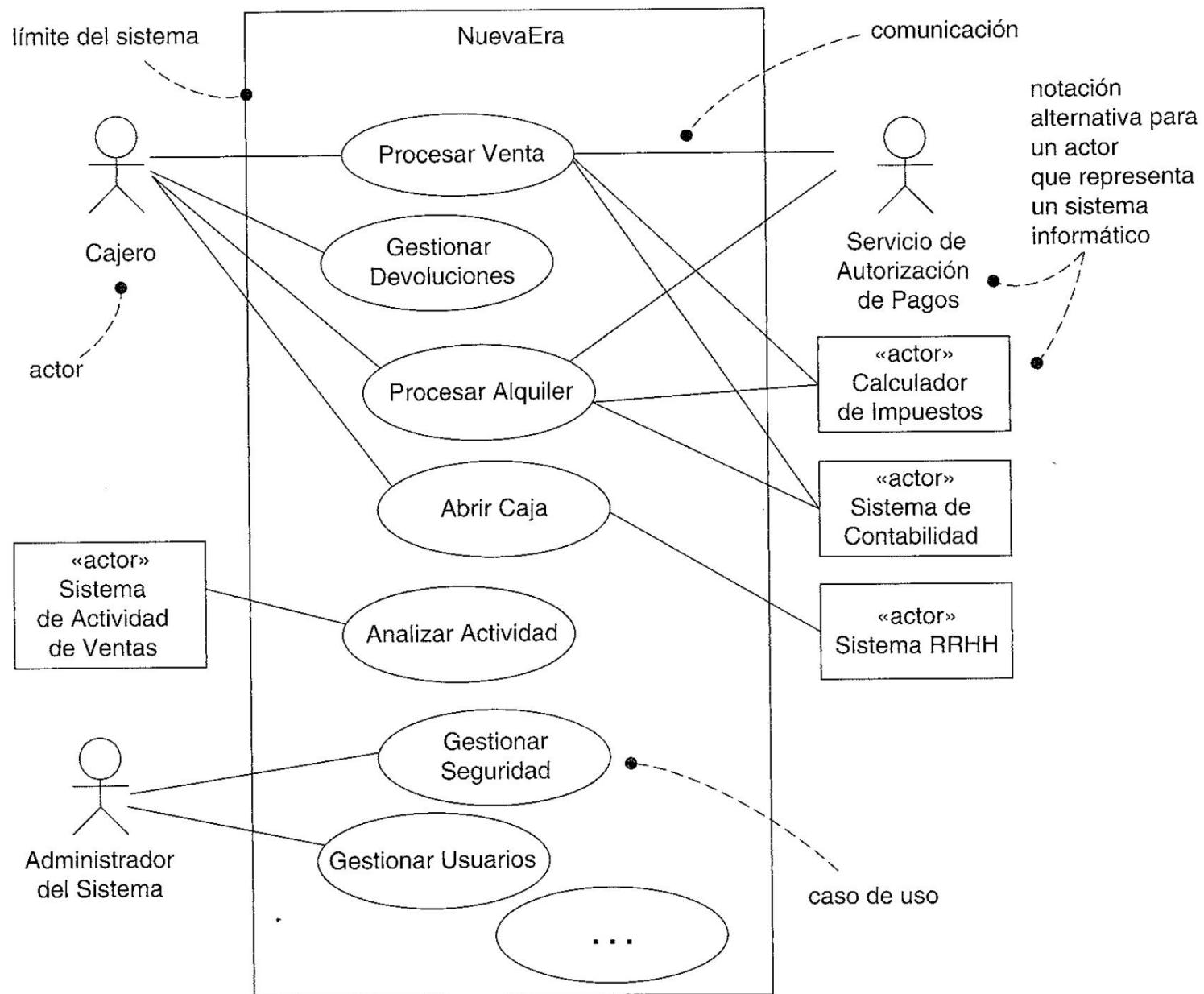
Tipos de Actores

- ▶ Primarios: tiene metas de usuario que alcanzan a través del uso de servicios del sistema
- ▶ De soporte: Provee servicios (por ej. Información) al sistema
- ▶ Pasivo: Presenta interés en el comportamiento del caso de uso, pero no es usuario directo ni de soporte.

Construir Diagrama de Casos de Uso

- ▶ Identificar Actores y sus Objetivos de usuario
- ▶ Definir un Caso de uso por cada objetivo de usuario
- ▶ Organizar Casos de Uso en grupos por Actor
- ▶ Identificar relaciones entre casos de uso (Comunicación, extensión, inclusión)
- ▶ Dibujar el diagrama completo.

Ejemplo Sistema PDV: Diagrama CU



Descripción de Casos de Uso

- ▶ Descripción en formato breve de cada Caso de uso
- ▶ Descripción en formato casual de Cada Caso de Uso
- ▶ Especificación de casos de uso (usando Plantilla).
 - Del escenario de éxito
 - De escenarios alternativos
- ▶ Construir Diagrama de actividades para modelar cada caso de uso.
 - Cada camino alternativo del Diagrama, será un Escenario diferente del CU.

Formatos de Descripción de CU

Formatos

- Breve

Un párrafo resumen correspondiente al escenario principal

- Casual

Formato informal, los escenarios se presentan en múltiples párrafos

- Completo

Formato elaborado. Todos los pasos y variaciones se describen en detalle; incluye secciones de soporte tales como postcondiciones ...

Ejemplo Sistema PDV:

CU en formato breve: Manejar devoluciones

Manejar devoluciones

Un cliente llega a la caja con productos para su devolución. El cajero usa el sistema POS para registrar cada producto

Ejemplo PDV:

CU en formato casual: Manejar devoluciones

Caso de Uso: Manejar Devoluciones

Principal Escenario de éxito

Un cliente llega a la caja con productos para su devolución. El cajero usa el sistema POS para registrar cada producto

Escenarios Alternativos

El cliente pagó con tarjeta de crédito y la transacción de reembolso es rechazada, se le informa al cliente y

.....

Ejemplo PDV:

Especificación de CU en formato completo

Se utiliza Plantilla de Especificación clásica o Variación dos columnas



- Actor Principal
- Personas involucradas e Intereses
- Precondiciones
- Postcondiciones
- Escenario Principal (Flujo Básico)
- Extensiones (Flujos Alternativos)
- Requisitos especiales
- Tecnología y Lista Variaciones de datos
- Frecuencia
- Cuestiones abiertas

Ejemplo PDV: CU en formato una columna

Escenario Principal de Éxito (Flujo Básico):

- 1) El Cliente llega al POS de salida con artículos a comprar
- 2) El Cajero inicia una nueva venta
- 3) El Cajero introduce la identificación del producto
- 4) El sistema registra cada línea de venta y presenta la descripción y el total acumulado

El Cajero repite los paso 3 y 4 hasta que se indique hecho

FORMATO DE UNA COLUMNA

Ejemplo PDV:

CU “Procesar Venta” en una columna

Resumen: Un cliente llega al TPV con un conjunto de artículos. El Cajero registra los artículos y se genera un ticket. El cliente paga en efectivo y recoge los artículos.

Actor Principal: Cajero

Personal Involucrado e Intereses:

- Cajero: quiere entradas precisas, rápida y sin errores de pago
- Compañía: quiere registrar transacciones y satisfacer clientes.
- ...

- **Precondición:** El cajero se identifica y autentica
- **Postcondiciones:** Se registra la venta. Se calcula el impuesto. Se actualiza contabilidad e inventario...

Ejemplo PDV:

CU “Procesar Venta” en una columna

Flujo Básico:

1. A: El cliente llega al TPV con los artículos.
2. A: El cajero inicia una nueva venta
3. A: El cajero introduce el identificador de cada artículo.
4. S: El sistema registra la línea de venta y presenta descripción del artículo, precio y suma parcial.

El Cajero repite los pasos 3 y 4 hasta que se indique.

5. S: El Sistema presenta el total
6. A: El Cajero le dice al Cliente el total a pagar
7. S: El Cliente paga y el sistema gestiona el pago.
8. S: El Sistema registra la venta completa y actualiza Inventario.
9. S: El Sistema presenta recibo

Ejemplo PDV:

CU “Procesar Venta” en una columna

Extensiones (Flujos Alternativos):

3a. Identificador no válido

1. El Sistema señala el error y rechaza la entrada

3-6a. El Cliente pide eliminar un artículo de la compra

1. El Cajero introduce identificador a eliminar
2. El sistema actualiza la suma

...

7a. Pago en efectivo

1. El Cajero introduce cantidad entregada por el cliente
2. El Sistema muestra cantidad a devolver

...

....

Ejemplo PDV: CU “Procesar Venta” en una columna

Requisitos especiales:

- Interfaz de usuario con pantalla táctil en un monitor de pantalla plana. El texto debe ser visible a un metro de distancia.
- Tiempo de respuesta para autorización de crédito de 30 sg. el 90% de las veces

...

Lista de Tecnología y Variaciones de Datos:

- El identificador podría ser cualquier esquema de código UPC, EAN,..
- La entrada de información de la tarjeta se realiza mediante un lector de tarjetas.

...

Cuestiones Pendientes:

- Explorar cuestiones de recuperación de accesos a servicios remotos
- ¿Qué adaptaciones son necesarias para diferentes negocios?

Ejemplo PDV: Variación dos columnas

Actor principal: ...

...como antes...

Escenario principal de éxito:

Acción del actor (o intención)

1. El Cliente llega a un terminal PDV con mercancías y/o servicios que comprar.
 2. El Cajero comienza una nueva venta.
 3. El Cajero introduce el identificador del artículo.
- El Cajero repite los pasos 3-4 hasta que se indique*
6. El Cajero le dice al Cliente el total y pide que le pague.
 7. El Cliente paga.

Responsabilidad del Sistema

4. Registra cada línea de la venta y presenta la descripción del artículo y la suma parcial.
5. El Sistema presenta el total con los impuestos calculados.
8. El Sistema gestiona el pago.
9. Registra la venta completa y envía la información de la venta y el pago al sistema de Contabilidad externo (para la contabilidad y las comisiones) y al sistema de Inventario (para actualizar el inventario). El Sistema presenta el recibo.

...

...

Ejemplo PDV: CU en la Fase de Inicio

En la Fase de Inicio no todos los CU se describen en formato completo. Sólo los más importantes (10-20%)

<i>Completo</i>	<i>Informal</i>	<i>Breve</i>
Procesar Venta	Procesar Alquiler	Abrir Caja
Gestionar Devoluciones	Analizar Actividad de Ventas	Cerrar Caja
	Gestionar Seguridad	Gestionar Usuarios
	...	Poner en Marcha
		Suspender Operación
		Gestionar Tablas del Sistema
		...

CU en la Fase de Elaboración

- ▶ En esta fase de múltiples iteraciones de duración fija se identifican y refinan la mayoría de los Requisitos
- ▶ En la primera iteración se implementan y se realizan pruebas sobre los Casos de Uso principales
- ▶ Al final de la Elaboración se habrán escrito en detalle alrededor del 80% de los CU

RUP Agil. Dónde estamos???

Fase de Elaboración Iteración I

*Definiendo pasos y artefactos del
Comienzo de la Fase de Elaboración*

Fase de Elaboración

- ▶ En esta fase comienza una serie de iteraciones durante las cuales:
 - Se programa y se prueba la arquitectura central del Software
 - Se descubren y normalizan la mayoría de los requerimientos
 - Se mitigan los mayores riesgos
- ▶ La Elaboración consiste de dos o más iteraciones. Cada iteración es acotada (tiene comienzo y fin)
- ▶ La Elaboración NO ES una fase de diseño o donde los modelos se desarrollan completos para la implementación.
- ▶ No se crean prototipos desecharables sino porciones del sistema final (arquitecturas ejecutables)

Artefactos de la Fase de Elaboración

- ▶ Modelo de Dominio
- ▶ Modelo de Diseño
- ▶ Documento de Arquitectura de Software
- ▶ Modelo de Datos
- ▶ Prototipos de Interfaz de usuario

Estos artefactos comienzan en esta fase.

No se incluyen en la lista los artefactos que se comenzaron en la Fase de Inicio.

Fase de Elaboración. Iteración 1

Disciplina: Modelado de Negocio

Modelo de Dominio

Modelo de Dominio. Objetivos:

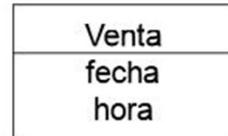
- ▶ Comprender y describir las clases más importantes dentro del dominio del problema (clases conceptuales).
- ▶ Contribuir:
 - a la comprensión de los requisitos del sistema que se desprenden de dicho contexto y
 - como fuente de inspiración para el diseño de objetos software.

Idea clave

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. *No* se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades.

Modelo de Dominio

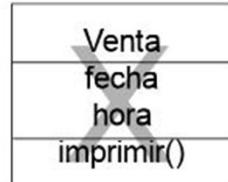
- ▶ Ilustra los conceptos más significativos del dominio.
- ▶ Establece el vocabulario del problema
- ▶ Junto al Modelo de requerimientos constituye el input para el Diseño.
- ▶ Captura los tipos más importantes de objetos en el contexto del sistema



Visualización de un concepto del mundo real del dominio de interés



Artefacto software; no forma parte del modelo del dominio



Clase software; no forma parte del modelo del dominio

Clases del Dominio

- ▶ Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema
 - Son **clases conceptuales** (perspectiva de Fowler) . No son **clases software**
 - Aparecen en tres formas típicas:
 - *Objetos de negocio* que representan cosas que se manipulan en el mismo, como pedidos, cuentas y contratos (o señales de sensores externos)
 - *Objetos del mundo real* y conceptos de los que el sistema debe hacer un seguimiento.
 - *Sucesos* que han ocurrido o ocurrirán, como la llegada de una avión, su salida,...

Perspectiva conceptual del DC

- ▶ En la Perspectiva conceptual del Diagrama de clases (según Fowler) se elaboran diagramas que reflejan los conceptos del dominio del problema, sin pensar en la solución del mismo:
 - Una clase representa un **concepto relevante** existente en el dominio del problema.
 - Un atributo representa una **propiedad relevante** propia de ese concepto
 - Las asociaciones representan **relaciones conceptuales** entre los conceptos

Clases conceptuales

Representan conceptos existentes en el dominio del problema



Cosas físicas:
Personas, Lugares,
Organizaciones,
Dispositivos, etc.

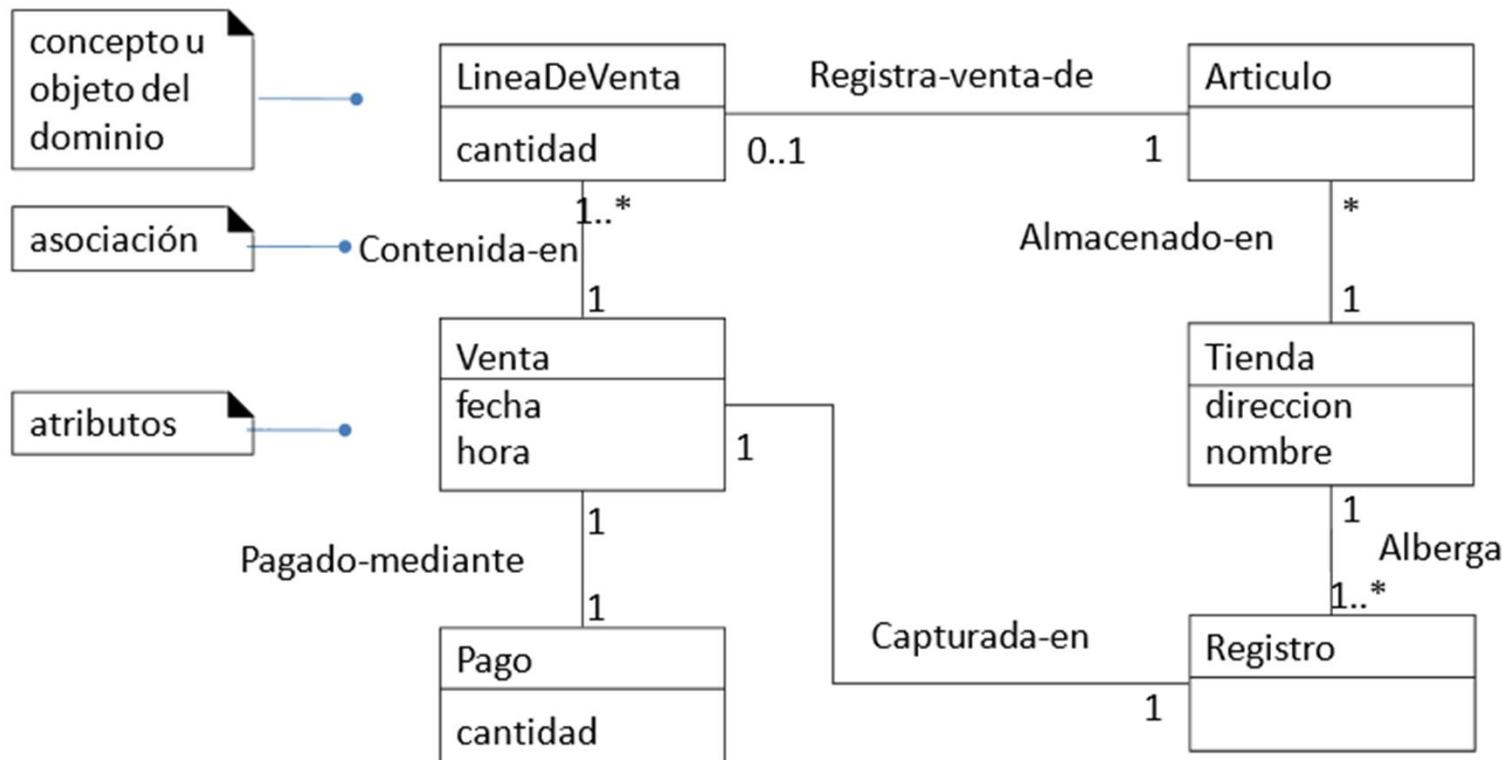
Cosas Lógicas:
Transacciones,
Reservas, Vuelos,
etc.

Modelo de Dominio en RUP

- ▶ RUP define un Modelo de Dominio como uno de los artefactos que podrían crearse durante la Disciplina de Modelado del Negocio (o del Contexto)
- ▶ Con UML el **Modelo de Dominio** se representa con un conjunto de **Diagramas de clases sin operaciones**, mostrando:
 - **Objetos del dominio** (**clases conceptuales**)
 - **Asociaciones** (con nombre) entre objetos del dominio
 - **Atributos de clases conceptuales**
- ▶ El **Modelo de Dominio** no se incentiva fuertemente en la Fase de Inicio. Se crea sobretodo durante las iteraciones de la fase de Elaboración.

Diagrama de clases del Dominio

▶ Ejemplo



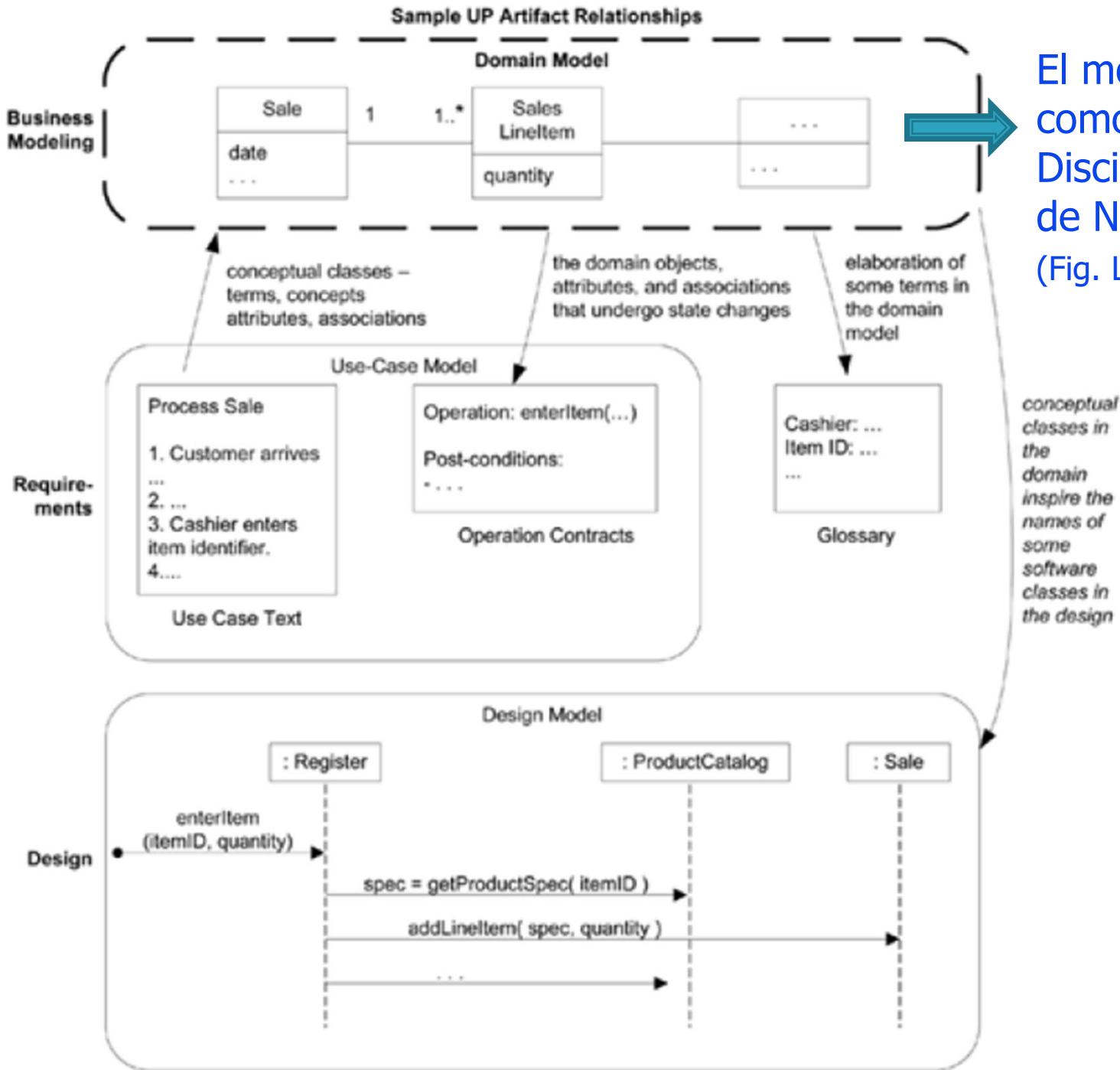
Uso del Modelo de Dominio en RUP

- ▶ Las clases del dominio y el glosario se utilizan en RUP para la construcción del Modelo de Diseño
 - Durante la realización de casos de uso
 - Para sugerir clases internas al sistema durante el análisis y el diseño

Qué es un Modelo de Dominio en UP?

- ▶ El modelo de dominio:
 - Puede considerarse *un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio*
 - Muestra clases conceptuales (ideas, cosas u objetos) de la realidad, NO MUESTRA artefactos (como Interfaces o BD) o clases Software
 - No debe confundirse con los objetos de la Capa de Dominio de una arquitectura de software
 - No es un Modelo de Datos

El modelo de dominio como artefacto de la Disciplina Modelado de Negocio en RUP (Fig. Larman, 2005)



Estrategias para identificar clases conceptuales

I. Utilizando lista de categorías de clases conceptuales

- Se comienza la creación de un modelo del dominio haciendo una lista de clases conceptuales candidatas.

II. Utilizando frases nominales

- Se identifican los nombres y frases nominales en las descripciones textuales del dominio, y se los considera como clases conceptuales o atributos candidatos

Estrategias para identificar clases conceptuales

I. Utilizando lista de categorías de clases conceptuales

<i>Categoría de clase conceptual</i>	<i>Ejemplos</i>
objetos tangibles o físicos	<i>Registro Avion</i>
especificaciones, diseños o descripciones de la cosas	<i>EspecificacionDelProducto DescripcionDelVuelo</i>
lugares	<i>Tienda</i>
transacciones	<i>Venta, Pago Reserva</i>
líneas de la transacción	<i>LíneaDeVenta</i>
roles de la gente	<i>Cajero Piloto</i>
contenedores de otras cosas	<i>Tienda, Lata Avion</i>
cosas en un contenedor	<i>Articulo Pasajero</i>
otros sistemas informáticos o electromecánicos externos al sistema	<i>SistemaAutorizadonPagoCredito ControlDeTraficoAereo</i>

Estrategias para identificar clases conceptuales

I. Utilizando lista de categorías de clases conceptuales (Cont.)

conceptos abstractos	<i>Ansia</i> <i>Acrofobia</i>
organizaciones	<i>DepartamentoDeVentas</i> <i>CompañiaAerea</i>
hechos	<i>Venta, Pago, Reunión</i> <i>Vuelo, Colisión, Aterrizaje</i>
procesos (normalmente <i>no</i> se representan como conceptos, pero podría ocurrir)	<i>VentaDeUnProducto</i> <i>ReservaUnAsiento</i>
reglas y políticas	<i>PoliticaDeReintegro</i> <i>PoliticaDeCancelación</i>
catálogos	<i>CatalogoDeProductos</i> <i>CatalogoDePiezas</i>
registros de finanzas, trabajo, contratos, cuestiones legales	<i>Recibo, LibroMayor, ContratoEmpleo</i> <i>RegistroMantenimiento</i>
instrumentos y servicios financieros	<i>LíneaDeCredito</i> <i>Stock</i>
manuales, documentos, artículos de referencia, libros	<i>ListaDeCambiosDePreciosDiarios</i> <i>Manual/Reparaciones</i>

Estrategias para identificar clases conceptuales

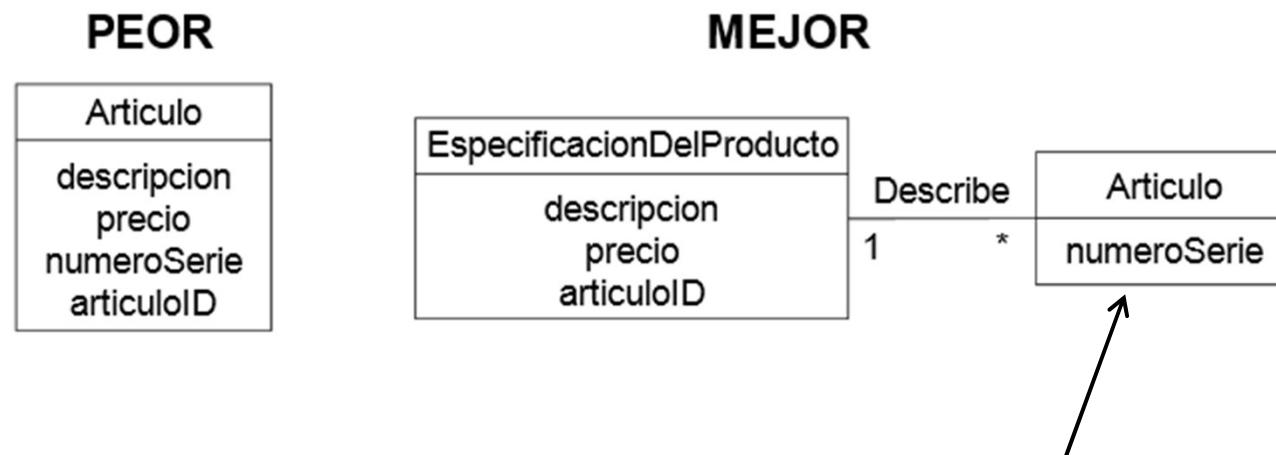
II. Mediante la identificación de clases nominales

- ▶ Identificar nombres y frases nominales (sustantivos) en las descripciones textuales de un dominio y considerarlos como clases conceptuales o atributos candidatos
- ▶ Los casos de uso constituyen una fuente (se subrayan sustantivos en la especificación completa).
- ▶ Algunas frases nominales subrayadas pueden ser clases y otras pueden ser atributos de clases.
- ▶ Se recomienda combinar esta técnica con la anterior

Estrategias para identificar clases conceptuales

Necesidad de clases conceptuales de Especificación

- En un Modelo de dominio es típico establecer que una *EspecificaciónDeX DescribeUnX*
- Los objetos de descripción o especificación están fuertemente relacionados con las cosas que describen (artículos, servicios)
- La necesidad de clases conceptuales de especificación es habitual en los dominios donde se requiere una descripción de la cosa que se fabrica, que es distinta de la cosa en sí misma, (por ej. ventas y productos).



Serían las instancias inventariadas de la Tienda

Estrategias para identificar clases conceptuales

Cuándo utilizar clases conceptuales de Especificación

- Se necesita la descripción de un artículo o servicio independiente de la existencia actual de algún ejemplo de ellos.
- Se pierde información que se debe mantener, cuando se eliminan instancias de lo que se describe (por ejemplo, Artículo)
- Se reduce información redundante o duplicada

Avanzando con el ejemplo PDV

I. Clases conceptuales candidatas para el Dominio de Ventas

- A partir de las estrategias vistas, se define la lista:

Registro

EspecificaciónDelProducto

Artículo

LíneaDeVenta

Tienda

Cajero

Venta

Cliente

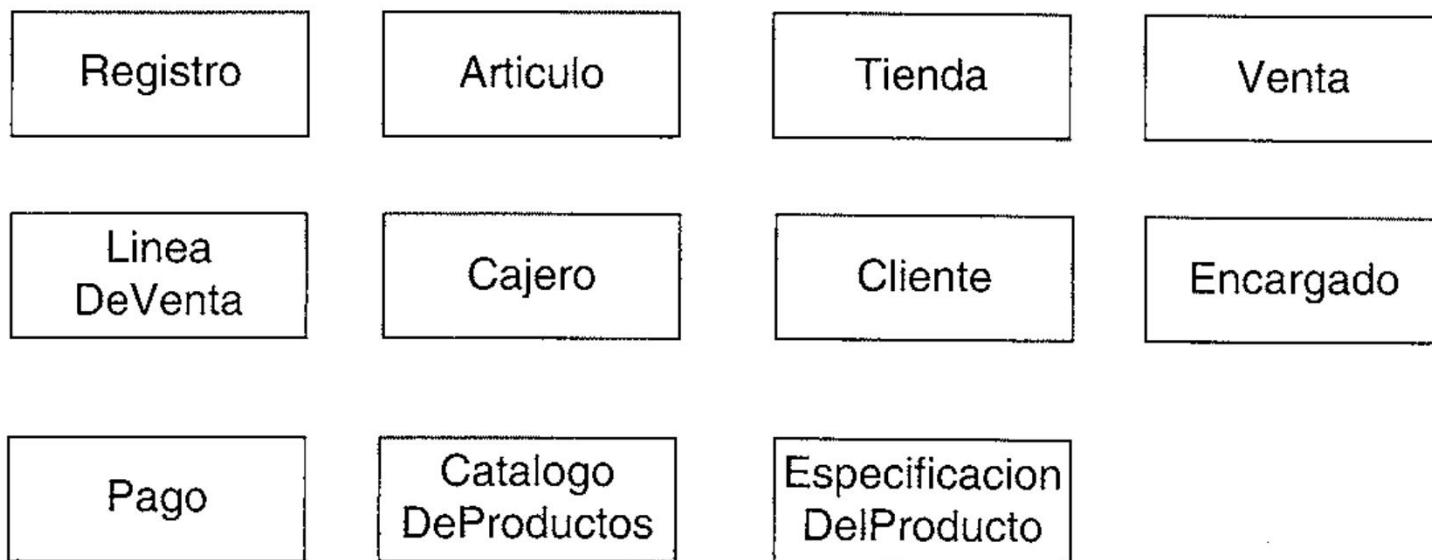
Pago

Encargado

CatálogoDeProductos

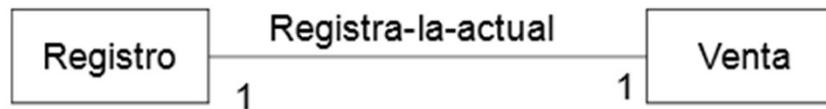
Avanzando con el ejemplo PDV

Representación gráfica de las Clases conceptuales candidatas



Asociaciones

- ▶ Representan relaciones entre conceptos del Dominio
- ▶ Se representan como una línea entre clases con un nombre de asociación
- ▶ La asociación es inherentemente bidireccional. Este recorrido es puramente abstracto en este modelo
- ▶ Los extremos de la asociación pueden contener una expresión de multiplicidad.



Identificación de Asociaciones

- ▶ Asociaciones de alta prioridad son:
 - A es parte lógica de B
 - A es parte física de B
 - A está contenido lógicamente en B
 - A está contenido físicamente en B
 - A es una descripción de B
 - A conoce (o se registra en) B
- ▶ En esta etapa es más importante identificar clases conceptuales que asociaciones.

Guías para Asociaciones

- ▶ Es conveniente centrarse en aquellas asociaciones para las que se necesita conservar el conocimiento de la relación durante algún tiempo ("necesito-conocer")
- ▶ Demasiadas asociaciones tienden a confundir un modelo del dominio.
- ▶ Se debe evitar mostrar asociaciones redundantes o derivadas.

Categoría	Ejemplos
A es una parte física de B	Cajon-Registro (o más concretamente, TPDV) Ala-Avion
A es una parte lógica de B	LíneaDeVenta- Venta EtapaVuelo-RutaVuelo
A está contenido físicamente en B	Registro-Tienda, Artículo-Estanteria Pasajero-Avion
A está contenido lógicamente en B	DescripcionDelArticulo-Catalogo Vuelo-PlanificacionVuelo
A es una descripción de B	DescripcionDelArticulo-Articulo DescripcionDelVuelo- Vuelo
A es un línea de una transacción o informe de B	LíneaDeVenta-Venta TrabajoMantenimiento-RegistroDe- Mantenimiento
A se conoce/registra/recoge/informa/captura en B	Venta-Registro Reserva-ListaPasajeros
A es miembro de B	Cajero- Tienda Piloto-CompañiaAerea
A es una subunidad organizativa de B	Departamento- Tienda Mantenimiento-CompañiaAerea
A utiliza o gestiona B	Cajero-Registro Piloto-Avion
A se comunica con B	Cliente-Cajero AgenteDeReservas-Pasajero
A está relacionado con una transacción B	Cliente-Pago Pasajero-Billete
A es una transacción relacionada con otra transacción B	Pago-Venta Reserva-Cancelacion
A es propiedad de B	Registro- Tienda Avion-CompañiaAerea
A es un evento relacionado con B	Venta-Cliente, Venta- Tienda Salida-Vuelo

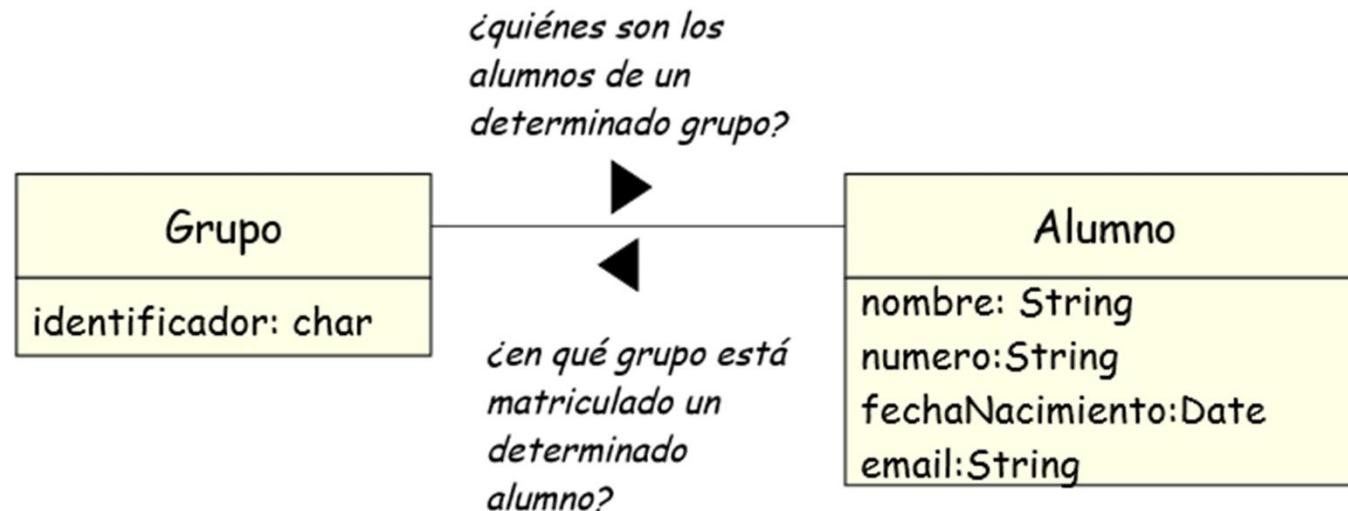
Lista de Asociaciones comunes (Larman, 2003)

Semántica de la Asociación

- Se pueden usar adornos para expresar semántica.



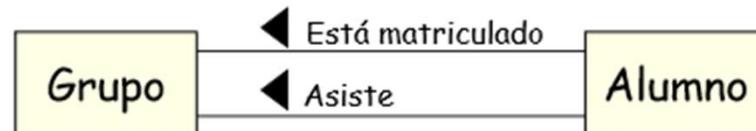
- La dirección por defecto para la lectura de los nombres de las asociaciones es de izquierda a derecha o de arriba abajo.



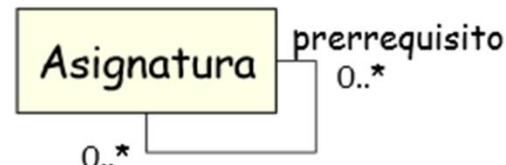
Asociaciones múltiples y reflexivas

Pueden existir:

Más de una asociación entre dos conceptos.



Asociaciones de un concepto consigo mismo.



Avanzando con el ejemplo PDV

► Aplicación de lista de Categoría de Asociaciones

- A es parte física de B
 - TPV-Caja
- A es parte lógica de B
 - LineaVenta-Venta
- A está físicamente contenida en B
 - Item-Tienda, TPV-Tienda
- A está lógicamente contenida en B
 - EspecificacionProducto-CatalogoProductos
- A es una descripción de B
 - EspecificacionProducto-Item

Avanzando con el ejemplo PDV

► Aplicación de lista de Categoría de Asociaciones

- A es una línea de una transacción o informe B
 - LineaVenta-Venta
- A es conocido/registrado/informado/capturado en B
 - Venta-TPV
- A es miembro de B
 - Cajero-Tienda
- A usa o maneja B
 - Cajero-TPV, Gerente-TPV
- A comunica con B
 - Cliente-Cajero
- A está relacionado con una transacción B
 - Cliente-Pago, Cajero-Pago
- A es el siguiente a B
 - LineaVenta-LineaVenta
- A es propiedad de B
 - TPV-Tienda

Avanzando con el ejemplo PDV

- ▶ Relaciones evidentes (necesito-conocer).

Registro Registra Venta

Para conocer la venta actual, generar el total, imprimir recibo

Venta Pagada-mediante Pago

Para conocer si se ha pagado la venta, relacionar la cantidad entregada con el total de la venta, e imprimir un recibo

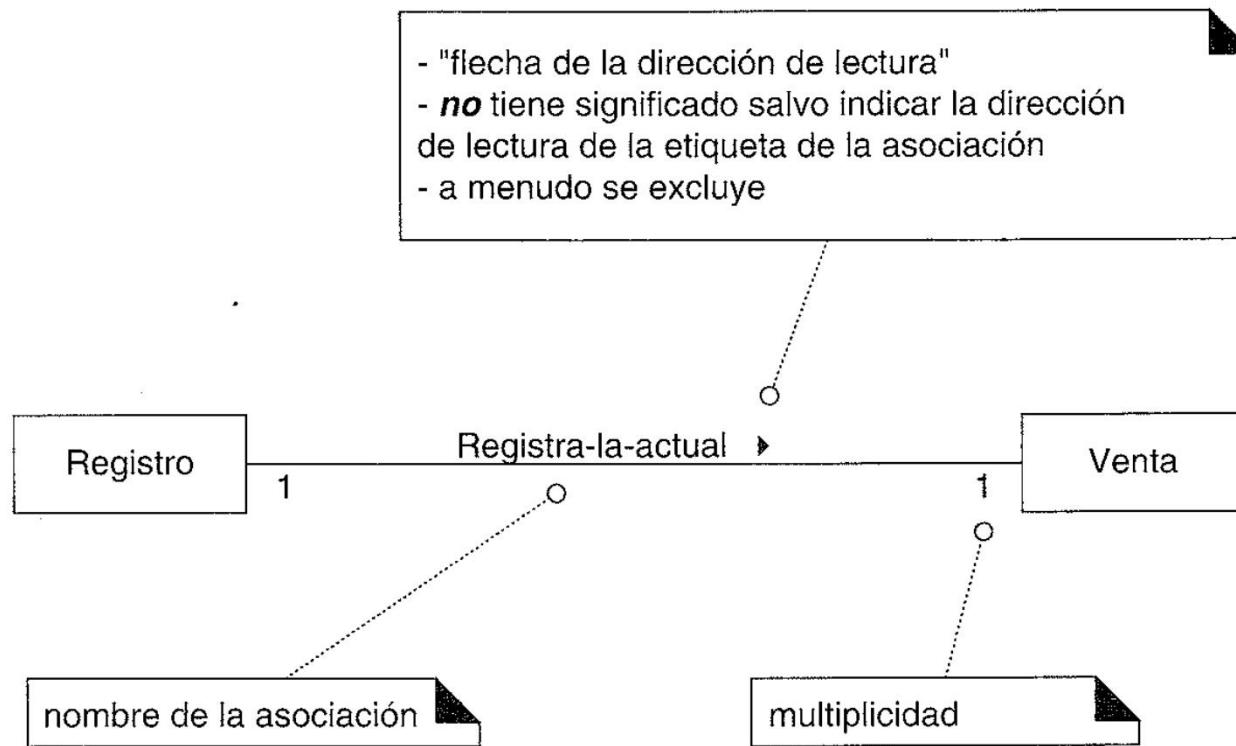
*CatalogoDeProductos Registra
EspecificacionDelProducto*

Para recuperar una *Especificacion Del Producto* a partir de un articuloID

- ▶ Se desecharon las asociaciones que no son necesarias

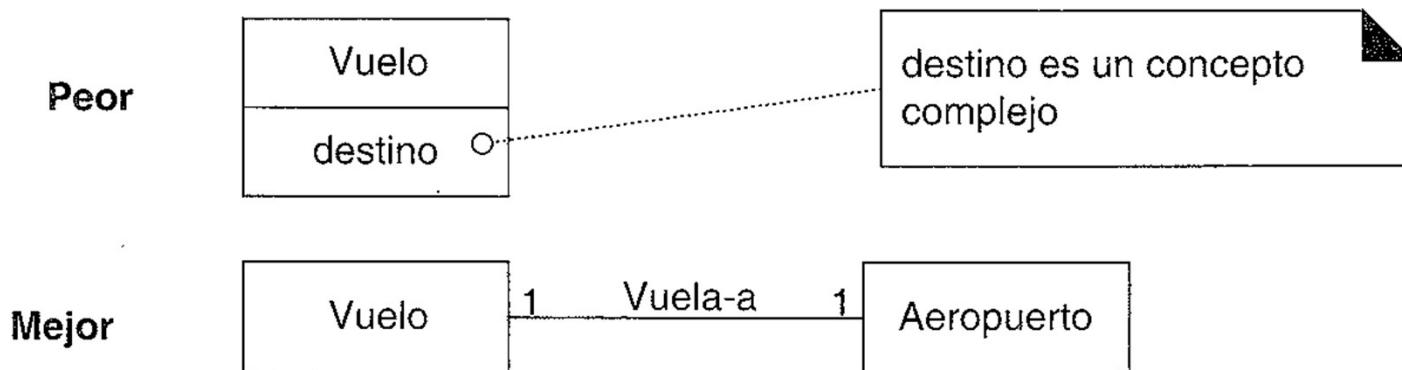
Avanzando con el ejemplo PDV

Se añaden asociaciones de la siguiente forma:



Atributos

- ▶ Se identifican atributos de las clases conceptuales necesarios para satisfacer los requisitos de registrar información de los actuales escenarios en estudio.
- ▶ Algunos atributos deben representarse como asociaciones
 - La mayoría de los atributos simples son los que se conocen como tipos de datos primitivos (números, chars, ...)
 - El tipo de un atributo, no debería ser un concepto de dominio complejo, como Venta o Aeropuerto, sino simples (o tipos de datos).



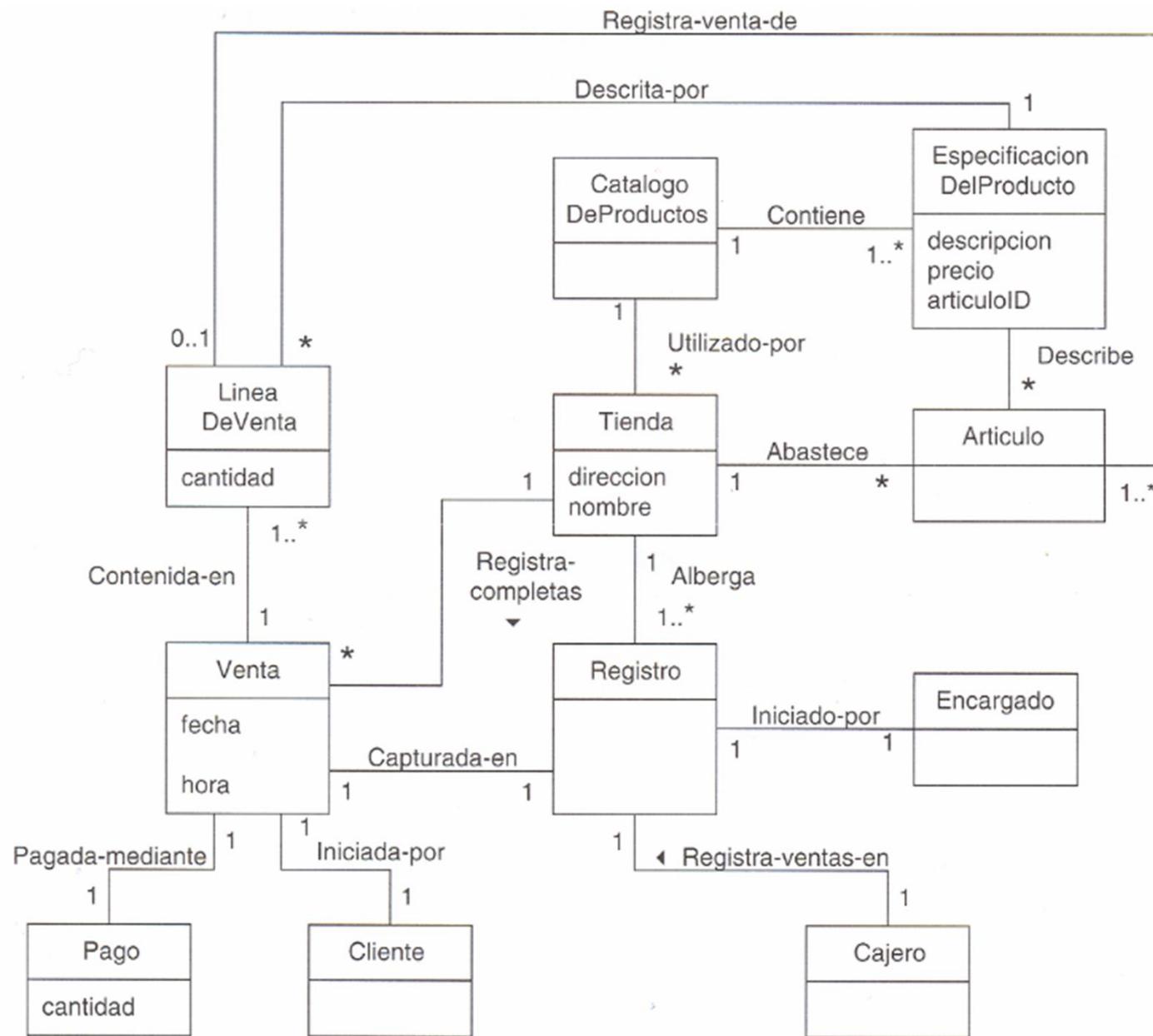
Las clases conceptuales deben relacionarse mediante una asociación, no mediante un atributo

Avanzando con el ejemplo PDV

► Atributos necesarios para la realización de ProcesarVenta

CONCEPTO	ATRIBUTO
Pago	<i>cantidad</i> : Se debe capturar una cantidad (también conocida como "cantidad entregada") para determinar si se proporciona el pago suficiente y calcular el cambio.
Especificacion Del Producto	<i>descripcion</i> : Para mostrar la descripción en una pantalla o recibo. <i>id</i> : Para buscar una <i>EspecificacionDelProducto</i> , dado un articuloID introducido, es necesario relacionadas con un <i>id</i> . <i>precio</i> : Para calcular el total de la venta y mostrar el precio de la línea de venta.
Venta	<i>fecha, hora</i> : Un recibo es un informe en papel de una venta. Normalmente muestra la fecha y la hora de la venta.
LíneaDeVenta:	<i>cantidad</i> : Para registrar la cantidad introducida, cuando hay más de un artículo en una línea de venta (por ejemplo, cinco paquetes de tofu).
Tienda tienda.	<i>direccion, nombre</i> : El recibo requiere el nombre y la dirección de la

Modelo de Dominio parcial del ejemplo PDV



Referencias

- “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process”. Craig Larman. Prentice-Hall, Inc. 2003 ISBN 0-13-092569
- Writing effective use cases. Cockburn. Addison Wesley, 2000