

# ANÁLISIS: EL MODELO DEL DOMINIO

## INDICE

1	INTRODUCCIÓN .....	3
2	DIAGRAMA DE CLASES .....	3
2.1	¿QUÉ ES EL DIAGRAMA DE CLASES? .....	3
2.2	NOTACIÓN BÁSICA DEL DIAGRAMA DE CLASES DEL LENGUAJE DE MODELADO UNIFICADO .....	3
2.2.1	Clases .....	3
2.2.2	Asociaciones .....	5
2.2.3	Visibilidad .....	14
2.2.4	Ejemplos .....	15
3	DIAGRAMA DE OBJETOS .....	17
4	MODELO DEL DOMINIO .....	20
4.1	¿ QUÉ ES EL MODELO DEL DOMINIO ? .....	20
4.2	CONSEJOS PARA REALIZAR UN BUEN MODELO DEL DOMINIO .....	22
4.2.1	Pasos para la elaboración del Modelo del Dominio .....	22
4.2.2	Identificación de Conceptos .....	23
4.2.2.1	Categorías de Conceptos Típicos .....	23
4.2.2.2	Análisis lingüístico de Casos de Uso y Requisitos .....	24
4.2.2.3	Visión del Cartógrafo .....	25
4.2.3	Clases Conceptuales de Especificación o de Descripción .....	25
4.2.4	Atributos .....	26
4.2.5	Asociaciones en el Modelo del Dominio .....	28

# 1 INTRODUCCIÓN

En este documento se presenta el **Modelo del Dominio**, que en ocasiones también se denomina **Modelo Conceptual** o **Modelo del Negocio**<sup>1</sup>, y que se genera, típicamente, como fruto de la actividad de Análisis durante la fase de Elaboración del Proceso Unificado de Desarrollo Software. El Modelo del Dominio se representa mediante un Diagrama de Clases del Lenguaje de Modelado Unificado (UML, Unified Modelling Language) con bajo nivel de detalle y en él se modelan los **conceptos del dominio** mediante las denominadas **clases conceptuales**, que no deben confundirse con las **clases del diseño** que son las que se mostrarán en el **Diagrama de Clases del Diseño** y que deberán ser ya **clases software**, es decir, elementos propuestos para la solución software y, por lo tanto, clases que finalmente se codifiquen. Puesto que para la representación del Modelo del Dominio se utiliza un Diagrama de Clases, en este documento también se presenta dicho artefacto UML. El documento se completa con unos breves apuntes sobre los Diagramas de Objetos que son un caso especial de los Diagramas de Clases que muestran instancias específicas de las clases involucradas en algún momento temporal escogido.

## 2 DIAGRAMA DE CLASES

### 2.1 ¿QUÉ ES EL DIAGRAMA DE CLASES?

El Diagrama de Clases es un Diagrama de Estructura Estática propuesto por el Lenguaje de Modelado Unificado que permite mostrar las clases de un sistema y sus relaciones, y que está íntimamente relacionado tanto con el Modelo del Dominio, también denominado Modelo Conceptual o Modelo del Negocio, realizado como parte de la actividad de Análisis, como con el Diagrama de Clases del Diseño, realizado como parte de la actividad de Diseño, puesto que se emplea para la representación de ambos, aunque con diferente nivel de detalle.

### 2.2 NOTACIÓN BÁSICA DEL DIAGRAMA DE CLASES DEL LENGUAJE DE MODELADO UNIFICADO

#### 2.2.1 Clases

Una clase se representa mediante una caja con tres compartimentos: En el superior se muestra el nombre de la clase, en el intermedio los atributos y en el inferior los métodos. La información de una clase puede representarse con diferente nivel de detalle e incluso de forma esquemática, con los atributos y métodos suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase, o apareciendo los rectángulos intermedio e inferior vacíos. En la Figura 1 se observa como una misma clase puede representarse con distinto nivel de detalle según interese, y según la Fase y Actividad del Proceso Unificado de Desarrollo de Software en la que se esté. No obstante, por lo general, en el Modelo del

---

<sup>1</sup> Como ya es sabido, el término Negocio es más restrictivo que el término Dominio, no obstante, en ocasiones se intercambian y se utilizan como si fueran sinónimos, teniendo este hecho su origen en la traducción literal del término *Business* utilizado en lengua inglesa.

Dominio se utilizará la representación para las clases con un nivel de detalle intermedio tal y como se hace en la Figura 2, es decir, especificando el nombre de la clase y el nombre de sus atributos, y opcionalmente su tipo de dato, para referirse al concepto del universo del problema representado y a sus propiedades o características.

Debe tenerse además en cuenta que las clases o métodos abstractos aparecerán especificados en cursiva o precedidos por la palabra reservada `<<abstract>>`<sup>2</sup> aunque es preciso señalar que la representación de la característica de abstracción no es relevante en el Modelo del Dominio en el que además habitualmente no se mostrarán los métodos de las clases sean abstractos o no.

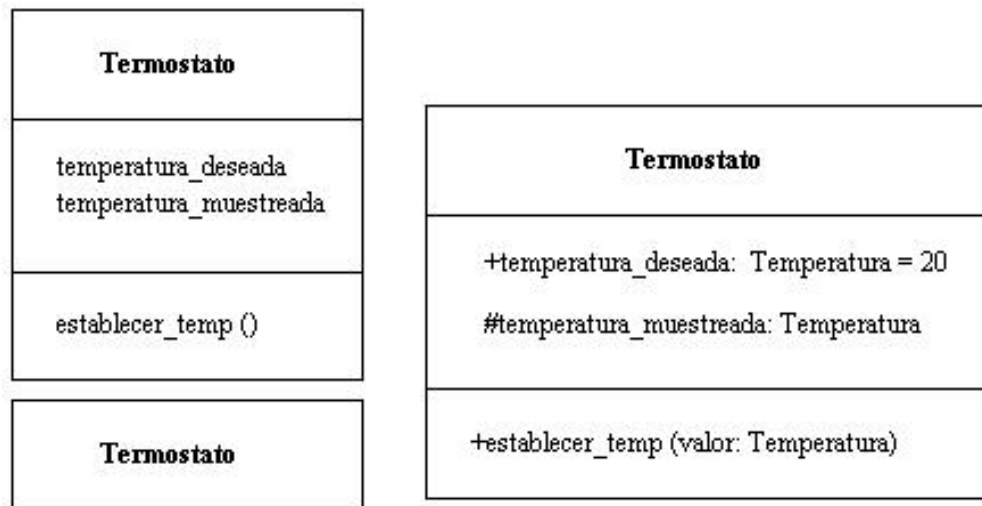


Figura 1: Representación de una clase, con diferente nivel de detalle, en la notación del Lenguaje de Modelado Unificado.

<sup>2</sup> Las clases abstractas definen e implementan comportamientos, pero no de una forma completa, de manera que otros programadores pueden hacer uso de estas clases detallando esos comportamientos mediante subclases o clases derivadas especializadas. Las clases abstractas no pueden ser instanciadas, sino que únicamente sirven para ser utilizadas como superclases o clases base de otras clases. Su propósito es, por tanto, servir de modelo para la creación de clases derivadas o subclases cuya implementación dependa de las características particulares de cada una de ellas.

Los métodos abstractos son aquellos que se declaran, pero para los que no se implementa su cuerpo y, por lo tanto, las clases que deriven o hereden de la clase base o superclase abstracta que los contenga deberán implementar sus detalles. Debe tenerse en cuenta que, si se declara algún método como abstracto, entonces, la clase debe declararse obligatoriamente también como abstracta. O, dicho de otro modo, cuando se declara una clase como abstracta, esta deberá tener como mínimo un método declarado como abstracto.

Véase más información sobre las clases y métodos abstractos en el tutorial de Orientación a Objetos.

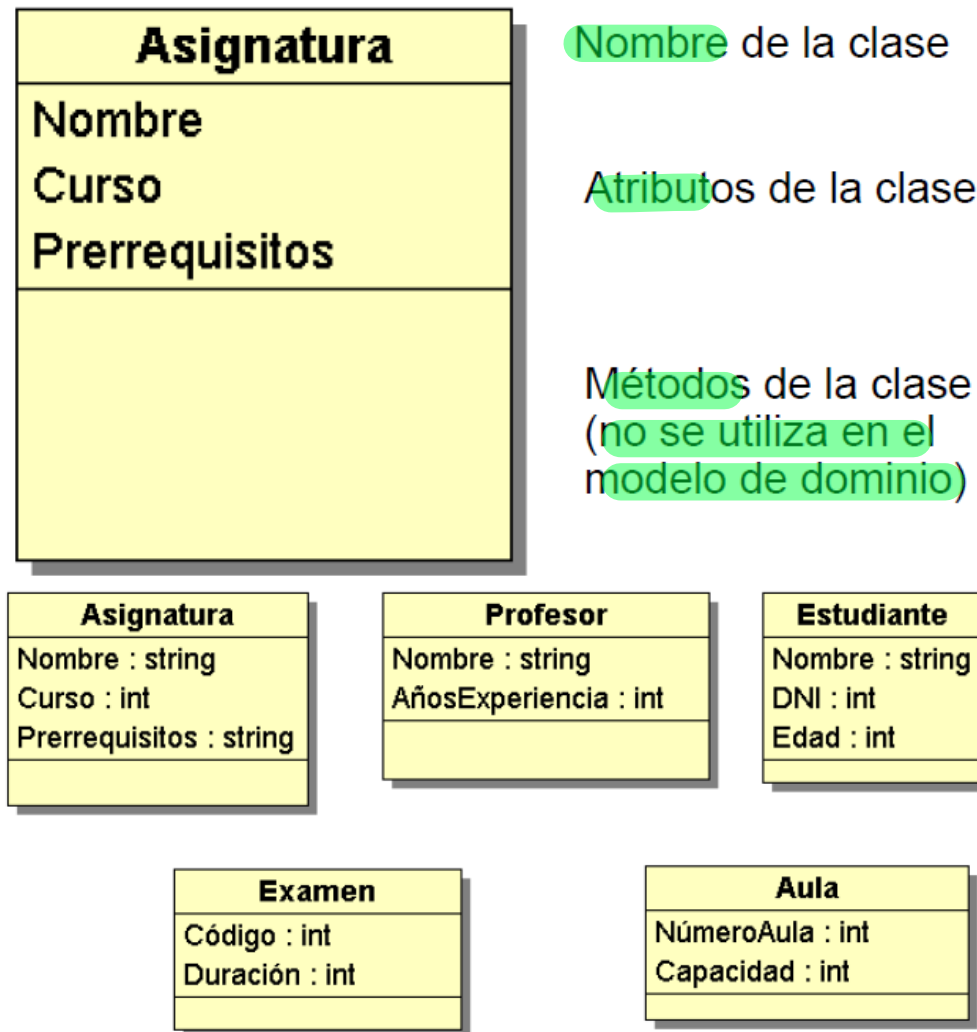


Figura 2: Representación de diferentes clases con el nivel de detalle con el que típicamente se mostrará en el Modelo del Dominio.<sup>3</sup>

### 2.2.2 Asociaciones

Una asociación entre dos clases se representa mediante una línea sólida que une ambas clases. Entre dos clases podría existir más de una asociación. La línea sólida que representa una asociación puede ir acompañada de una serie de elementos que expresan características particulares de la asociación. De entre dichos elementos, cabe destacar los siguientes:

Nombre de la Asociación:

El nombre de la asociación hace referencia a su propósito. Es opcional y, de presentarse, se mostrará como un texto próximo a la línea sólida que dibuja la asociación. Su formato será

<sup>3</sup> Los tipos de datos de los atributos o propiedades de las clases conceptuales podrían también aparecer especificados mediante tipos de datos genéricos como texto, número, etc., en vez de utilizando tipos de datos específicos de algún lenguaje de programación.

NombreClase-FraseVerbal-NombreClase, donde los NombreClase corresponderían a los nombres de las clases entre las que se especifique la asociación.

En el ejemplo de la Figura 3 la asociación se leería como *Director manda sobre Empleado* ya que, por defecto, la dirección de lectura de los nombres de las asociaciones es de izquierda a derecha y de arriba a abajo.

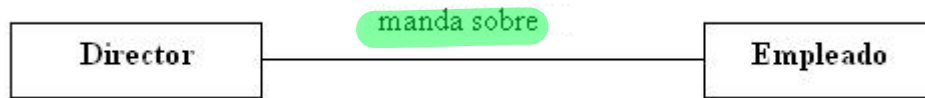


Figura 3: Representación de una Asociación entre clases, en la notación del Lenguaje de Modelado Unificado: Nombre de la Asociación (I).

Si, por algún motivo, no fuera posible escoger nombres para las asociaciones de manera que su lectura pueda realizarse en las direcciones indicadas como direcciones de lectura por defecto, o si bien se quisiera indicar la dirección de lectura de los nombres de las asociaciones de manera explícita, puede añadirse un triángulo sólido junto al nombre de la asociación apuntando en la dirección de lectura, tal y como se muestra en la Figura 4, sin que esto signifique ningún tipo de direccionalidad para la asociación entre ambas clases ya que, las asociaciones en un Modelo del Dominio son intrínsecamente bidireccionales, es decir, que la clase *Director* se relaciona con la clase *Empleado*, y viceversa, en igualdad de condiciones.

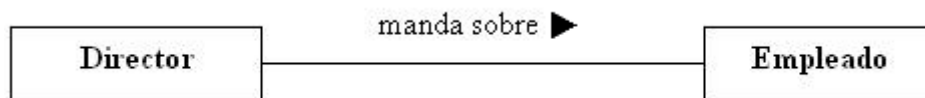


Figura 4: Representación de una Asociación entre clases, en la notación del Lenguaje de Modelado Unificado: Nombre de la Asociación (II).

Los nombres de las asociaciones normalmente se incluyen con el objetivo de mejorar la legibilidad. Sin embargo, en ocasiones pueden hacer que la información que se presente en el diagrama sea excesiva, con el consiguiente riesgo de saturación. En ese caso se podría suprimir el nombre de las asociaciones consideradas como suficientemente conocidas.

Además, debe tenerse en cuenta que en ciertos tipos de asociaciones como agregación, herencia, etc., sobre las que se hablará más adelante, no se especifica nunca su nombre, por estar su propósito claramente delimitado.

#### Roles de la Asociación:

El término Roles hace referencia a los extremos de las asociaciones los cuales pueden estar caracterizados por diversos elementos. Así, de manera alternativa a un único nombre en la asociación pueden indicarse sendos nombres para sus Roles en los extremos de la línea sólida que la dibuja, de forma que la lectura del propósito de la misma pueda realizarse en ambos sentidos si esto se considera conveniente.

Así, en la Figura 5, la lectura que se realizaría sería *Director manda Empleado* y *Empleado es mandado por Director*.

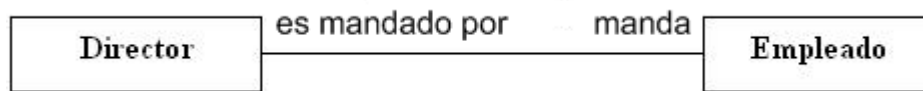


Figura 5: Representación de una Asociación entre clases, en la notación del Lenguaje de Modelado Unificado: Roles de la Asociación (I).

Igualmente, los nombres de los Roles podrían indicarse junto con el nombre de la asociación y tener el objetivo de indicar el papel que cada clase juega en dicha asociación tal y como se hace en la Figura 6.

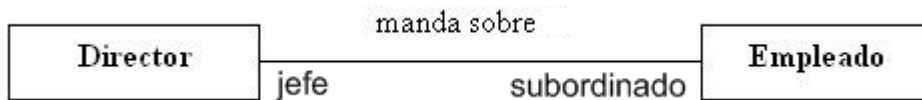


Figura 6: Representación de una Asociación entre clases, en la notación del Lenguaje de Modelado Unificado: Roles de la Asociación (II).

#### Multiplicidad o Cardinalidad:

La multiplicidad es una restricción que caracteriza a una asociación limitando el número de instancias que pueden intervenir en la misma de cada una de las clases vinculadas mediante dicha asociación. Esto implica que es posible especificar cantidades o rangos de cantidades en las relaciones establecidas, los cuales pueden expresarse de alguna de las siguientes formas [ver Figura 7]:

- Mediante un número fijo: 1.
- Mediante un conjunto de números fijos: { 2, 5, 9, 15 }.
- Mediante un intervalo de valores: 2 . . 5.
- Mediante un intervalo de valores en el cual uno de los extremos es un asterisco, lo cual significa que se trata de un intervalo abierto, así por ejemplo, 2 . . \* significa dos o más, 1 . . \* significa uno o más, 0 . . \* significa cero o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3 . . 5, 7, 15 . . \*.
- Con un asterisco \*, o con 0 . . \*, lo cual indica que puede tomar cualquier valor, es decir, cero o más.

Debe tenerse en cuenta además que, si no se especifica, la cardinalidad de una asociación se considera que es 1.

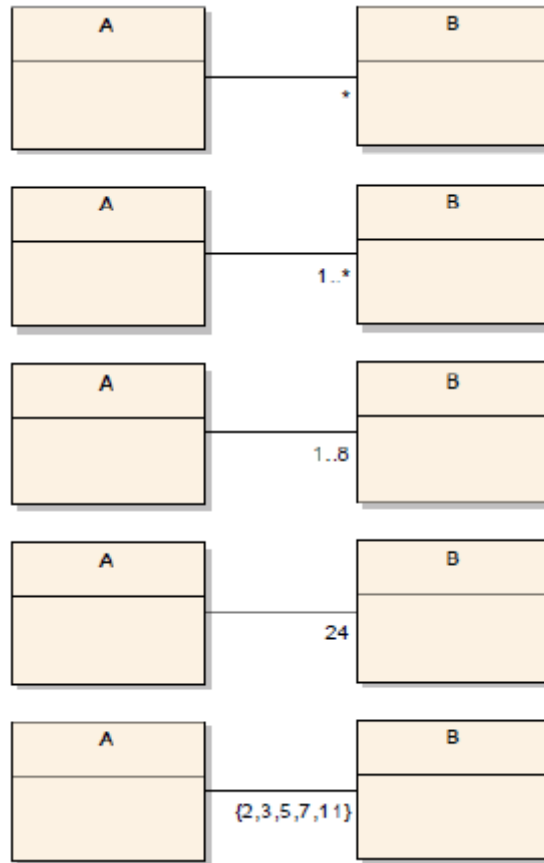


Figura 7: Diferentes formas de especificar la multiplicidad en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado.

La cardinalidad en el extremo de una asociación puede tomar cualquier valor lo que implica que las relaciones entre clases pueden ser de uno a uno, de uno a varios, de varios a varios, etc. En la Figura 8 se presenta un ejemplo de una asociación en la que se ha agregado la correspondiente información sobre su cardinalidad. La asociación del ejemplo presentado debe leerse como que *cero o más Nietos* viven con *de cero hasta cuatro Abuelos*, lo cual deja abiertas múltiples posibilidades.

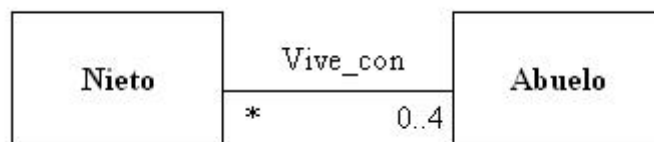


Figura 8: Especificación de la multiplicidad en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado.

### Clases de Asociación:

En las diferentes asociaciones identificadas, especialmente en aquellas cuya cardinalidad será de varios a varios, es importante reflexionar sobre si la asociación entre ambas clases es directa o si bien existe alguna clase intermedia.



Así, en el ejemplo de la Figura 9 la vinculación entre un *Cliente* y el *Banco* se establece a través de la existencia de una *Cuenta* del cliente en dicho banco, y no existiría de no tener el cliente una cuenta en dicho banco. Igualmente, nótese que se presentan dos notaciones alternativas para mostrar las asociaciones identificadas utilizándose en la segunda lo que se denomina una clase de asociación que es asociación y clase simultáneamente, aunque algunos expertos consideran esta notación<sup>4</sup> confusa y recomiendan evitarla, especialmente en la Actividad de Análisis.

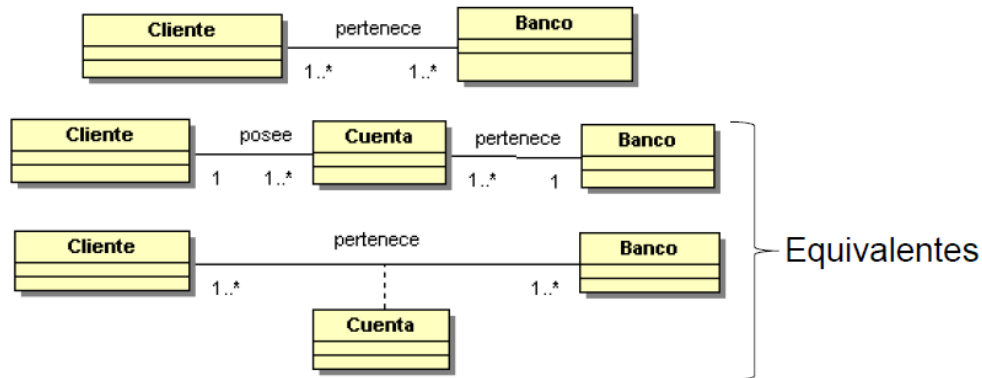


Figura 9: Especificación de una Clase de Asociación en la notación del Lenguaje de Modelado Unificado.

#### Dirección:

Como ya se ha comentado, en los Diagramas de Clases utilizados para representar Modelos del Dominio se considerará que las asociaciones entre las clases conceptuales o conceptos representados son bidireccionales.

#### Agregación vs Composición:

Existen dos posibilidades para modelar objetos complejos que son la Composición y la Agregación.

La Composición es un tipo de relación estática en la que el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del objeto que lo incluye ya que el objeto base, típicamente se construye a partir del objeto incluido, es decir, se trataría de una relación del tipo parte/todo, de forma que ninguna parte puede existir sin el todo. Por tanto, en la Composición se considera que tanto el todo como las partes tienen el mismo ciclo de vida y que un objeto podrá pertenecer solamente a una Composición.

En el ejemplo de la Figura 10, una mano tiene dedos, en concreto, de uno a cinco dedos, y, cuando se destruye la mano, los dedos que la componen quedan también destruidos.

En el ejemplo de la Figura 11, un libro se compone de capítulos, en concreto, se compone de uno o más capítulos.

<sup>4</sup> En los textos ingleses en ocasiones esta notación se denomina *hanging basket notation*.

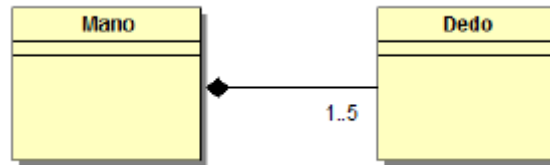


Figura 10: Ejemplo de una asociación entre clases de tipo Composición, en la notación del Lenguaje de Modelado Unificado (I).

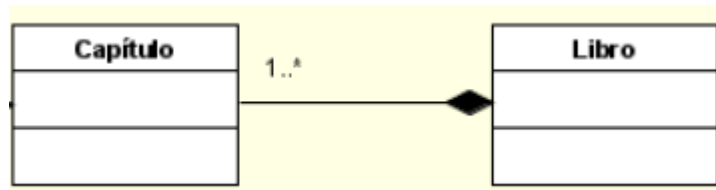


Figura 11: Ejemplo de una asociación entre clases de tipo Composición, en la notación del Lenguaje de Modelado Unificado (II).

La Agregación es un tipo de relación dinámica, en la que el tiempo de vida del objeto incluido es independiente del tiempo de vida del objeto que lo incluye. En este tipo de relación el vínculo entre ambos objetos es, por ejemplo, que el objeto base utiliza al incluido para su funcionamiento.

Según el diagrama mostrado como ejemplo de la Figura 12 un ordenador tiene CPU – una o más –, tarjeta de vídeo – una o más – y puede tener o no tarjeta de sonido – e incluso más de una –, componentes todos ellos que utiliza para su funcionamiento. Sin embargo, que un ordenador se destruya no implica que todos sus componentes también lo hagan.

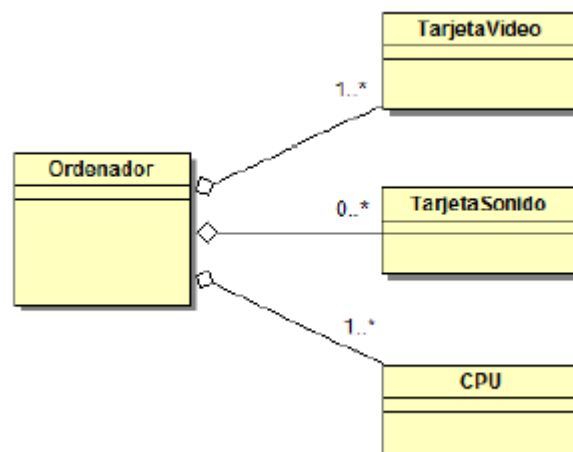


Figura 12: Ejemplo de una asociación entre clases de tipo Agregación, en la notación del Lenguaje de Modelado Unificado (I).

En el ejemplo de la Figura 13 un equipo tiene jugadores – uno o más y no todos los jugadores tienen por qué pertenecer a algún equipo –, pero si el equipo se disuelve, cada jugador podrá unirse a otro equipo si así lo desea.

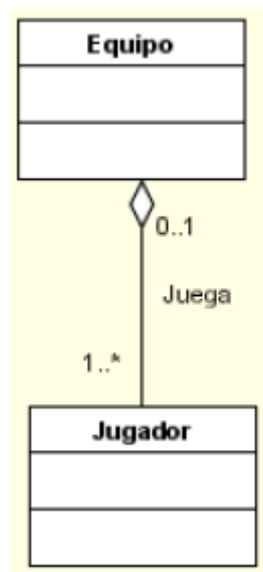
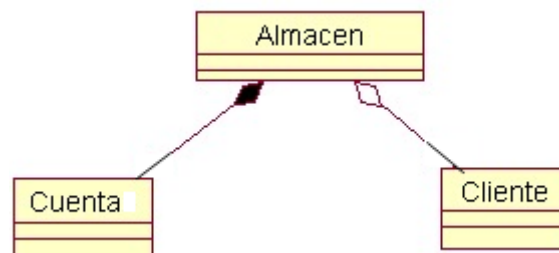


Figura 13: Ejemplo de una asociación entre clases de tipo Agregación, en la notación del Lenguaje de Modelado Unificado (II).<sup>5</sup>

En resumen, en una relación de Agregación, un objeto A está compuesto por objetos B, pero los objetos B son autónomos y pueden existir más allá del objeto A. Mientras que una relación de Composición, un objeto A está compuesto por objetos B y los objetos B no pueden existir si no es como parte de A.

Tal y como puede observarse en la Figura 10, Figura 11, Figura 12 y Figura 13 cuando se representan asociaciones por Composición o Agregación, se dibujará un rombo sólido o transparente junto al objeto base respectivamente, es decir, en aquel objeto que representa el todo y que se compone de, o agrega, a los otros objetos involucrados en la asociación.

Otro ejemplo podría ser aquel en el que existe una relación de la clase Almacen, de Composición con la clase Cuenta y de Agregación con la clase Cliente. Si esto es así, cuando se destruyera un objeto base Almacen también se destruiría la Cuenta asociada, sin embargo, no se vería afectado el Cliente vinculado. [ver Figura 14]



<sup>5</sup> Nótese que en este ejemplo podría tener sentido una multiplicidad 0..2 en la clase Equipo para indicar que uno o más jugadores pudieran pertenecer a hasta dos equipos, por ejemplo, uno de fútbol y otro de baloncesto. Todo dependerá de la realidad que se esté intentando modelar.

Figura 14: Ejemplo de una asociación entre clases de tipo Agregación-Composición, en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado. (I)

Un último ejemplo podría ser aquel en el que existe una relación de la clase `AddressBook` - Libro de Direcciones -, de Composición con la clase `Contact` - Contacto -<sup>6</sup> y con la clase `ContactGroup` - Grupo de Contactos -<sup>7</sup>. E igualmente una relación de Agregación entre la clase `ContactGroup` y la clase `Contact`<sup>8</sup>. Si esto es así, cuando se destruyera un Libro de Direcciones también se destruirían los Contactos y los Grupos de Contactos que dicho Libro de Direcciones tuviera. Sin embargo, si se destruye un Grupo de Contactos no se verían afectados los Contactos que lo forman, que, además, podrían también estar formando parte de otros Grupos de Contactos. [ver Figura 15]

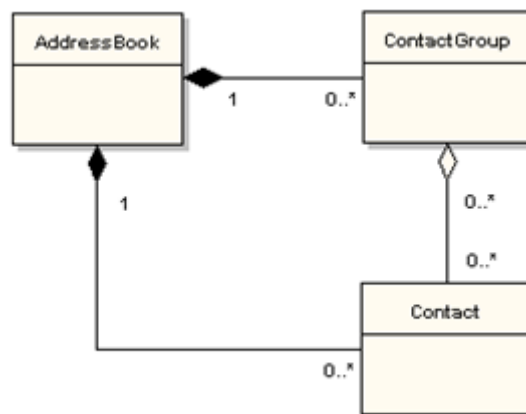


Figura 15: Ejemplo de una asociación entre clases de tipo Agregación-Composición, en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado. (II)

Nótese que por la definición de ambos conceptos, Composición y Agregación, en una Composición la multiplicidad del extremo del compuesto no puede ser mayor que uno, mientras que en una Agregación la multiplicidad del extremo del compuesto sí podría ser mayor que uno tal y como puede observarse en alguno de los ejemplos propuestos [ver Figura 13 y Figura 15].

Finalmente, es importante señalar que hay autores que no distinguen entre ambos conceptos Agregación y Composición.

#### Herencia:<sup>9</sup>

La relación de herencia se representa mediante un triángulo vacío en el extremo de la relación que corresponde a la clase más general, clase padre, clase base o superclase.

<sup>6</sup> Un Libro de Direcciones contiene ninguno, uno o varios Contactos.

<sup>7</sup> Un Libro de Direcciones contiene ninguno, uno o varios Grupos de Contactos.

<sup>8</sup> Un Grupo de Contactos contiene ninguno, uno o varios Contactos, no todos los Contactos tienen por qué pertenecer a un Grupo de Contactos y algunos Contactos pueden pertenecer a varios Grupos de Contactos.

<sup>9</sup> Véase más información sobre el concepto de Herencia en el tutorial de Orientación a Objetos.

Un ejemplo es el que puede verse en la Figura 16. Se han definido las clases `Bicicleta` y `BicicletaCarreras`. La clase `Bicicleta` es la clase base de la clase `BicicletaCarreras` con la que mantiene una relación de herencia.

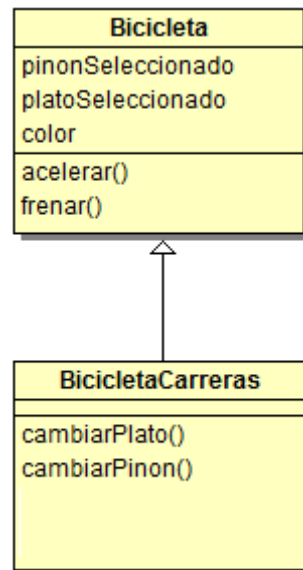


Figura 16: Ejemplo de especificación de la Herencia, en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado (UML).

Otro ejemplo es el que puede verse en la Figura 17. Se han definido las clases `Persona`, `Profesor` y `Estudiante`. La clase `Persona` es la clase base de las clases `Profesor` y `Estudiante` con las que mantiene una relación de herencia y recoge las características generales comunes de ambas clases.

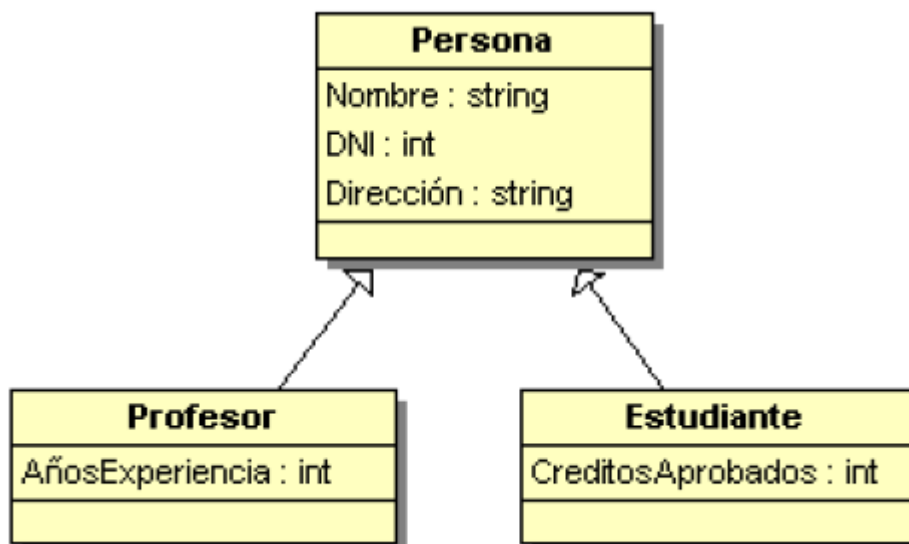


Figura 17: Ejemplo de especificación de la Herencia, en una asociación entre clases, en la notación del Lenguaje de Modelado Unificado (UML).

En el contexto del Modelo del Dominio, la intencionalidad de una relación de herencia es, simplemente, el mostrar una relación del tipo generalización-especialización identificada entre dos o más clases conceptuales.

### 2.2.3 Visibilidad

Esta característica permite definir, para los atributos y métodos de las diferentes clases involucradas en una asociación, su grado de comunicación y visibilidad entre ellos y para con su entorno. Los diferentes grados que se establecen son los siguientes:<sup>10</sup>

- (-) Privado. Indica que el atributo o método será invisible a otras clases, o dicho de otra forma, solo será accesible desde dentro de la propia clase. En el caso de los atributos esto supone que solo pueden ser accedidos por los métodos de la propia clase, y en el caso de los métodos que solo estarán accesibles para otros métodos de la propia clase.
- (#) Protegido. Indica que el atributo o método no será accesible desde fuera de la clase, aunque sí podrá ser accedido por métodos tanto de la propia clase como de las subclases que de esta se deriven. Es decir, que estos atributos o métodos serán visibles para la propia clase y para sus clases derivadas.
- (~) Paquete. Indica que el atributo o método no será accesible desde fuera de la clase, aunque sí podrá ser accedido por métodos tanto de la propia clase como de las clases que formen parte del mismo paquete. Es decir, que estos atributos o métodos serán visibles para la propia clase y para clases dentro del mismo paquete.<sup>11</sup>

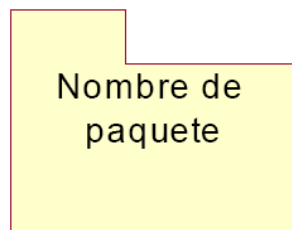


Figura 18: Representación visual de un paquete, en la notación del Lenguaje de Modelado Unificado.

- (+) Público. Indica que el atributo o método será visible tanto dentro como fuera de la clase, es decir, será completamente visible y, por lo tanto, accesible desde cualquier parte.

Finalmente cabe señalar que el grado de visibilidad no es relevante en la representación del Modelo del Dominio.

<sup>10</sup> Véase más información sobre los grados de Visibilidad de los atributos y métodos de las clases software en el tutorial de Orientación a Objetos.

<sup>11</sup> Los paquetes agruparían clases e interfaces que tienen alguna vinculación conceptual, es decir, que un paquete podría corresponder a un submodelo (subsistema) del modelo (sistema). Véase más información sobre el concepto de Clase e Interfaz en el tutorial de Orientación a Objetos.

Privado	Invisible a otras clases
Protegido	Visible a clases derivadas
Paquete	Visible dentro del mismo paquete
Público	Visible desde cualquier clase

Tabla 1: Grados de visibilidad para los atributos y métodos de clases involucradas en asociaciones.

## 2.2.4 Ejemplos

En la Figura 19, la Figura 20 y la Figura 21 se presentan ejemplos de Diagramas de Clases en los que se puede observar, de forma conjunta, diversos elementos de la notación en el Lenguaje de Modelado Unificado a utilizar en este tipo de diagramas para la representación de diferentes características como la herencia o la abstracción.

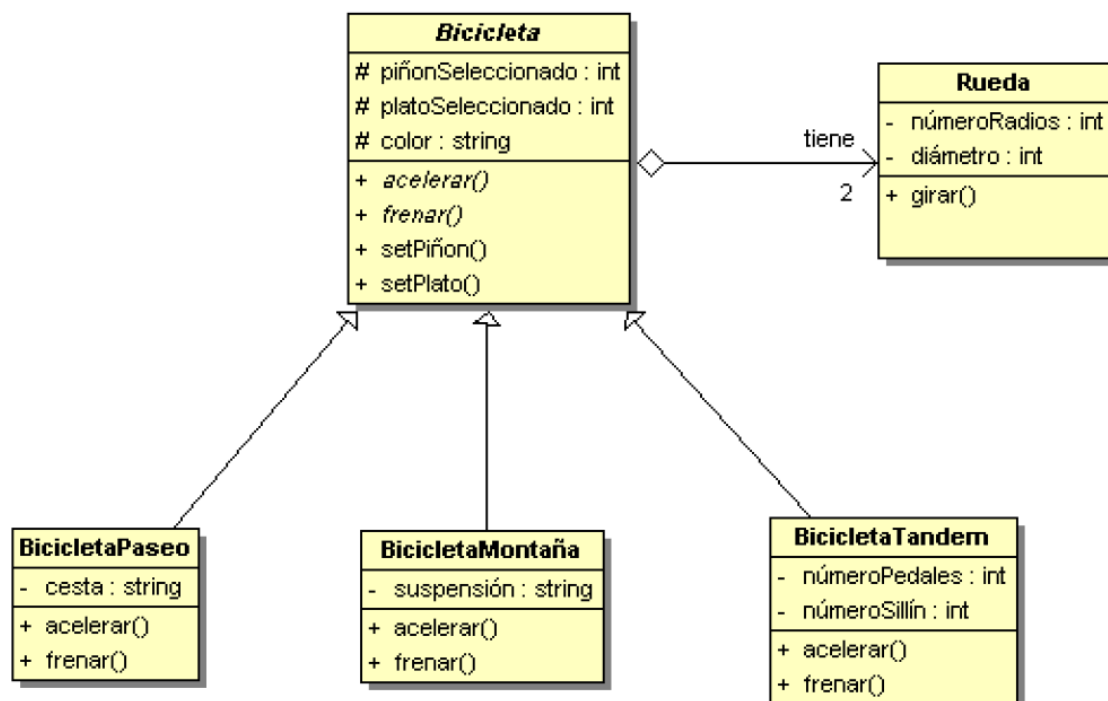


Figura 19: Ejemplo de Diagrama de Clases (I).

En el ejemplo de la Figura 19 se han definido las clases `Bicicleta`, `BicicletaMontaña`, `BicicletaPaseo` y `BicicletaTandem` entre las que existe una relación de herencia siendo la **clase abstracta `Bicicleta` la clase base o superclase del resto**

de clases. Además del diagrama se desprende que cada bicicleta tendrá dos ruedas.<sup>12</sup> Nótese además que la relación entre las clases *Bicicleta* y *Rueda* se ha modelado como una relación de Agregación.

El nivel de detalle usado para especificar las diferentes clases incluye los atributos y su tipo de dato y su grado de visibilidad, así como los métodos y su grado de visibilidad, indicando además si alguno de ellos como el método `acelerar()` de la clase *Bicicleta* es abstracto.

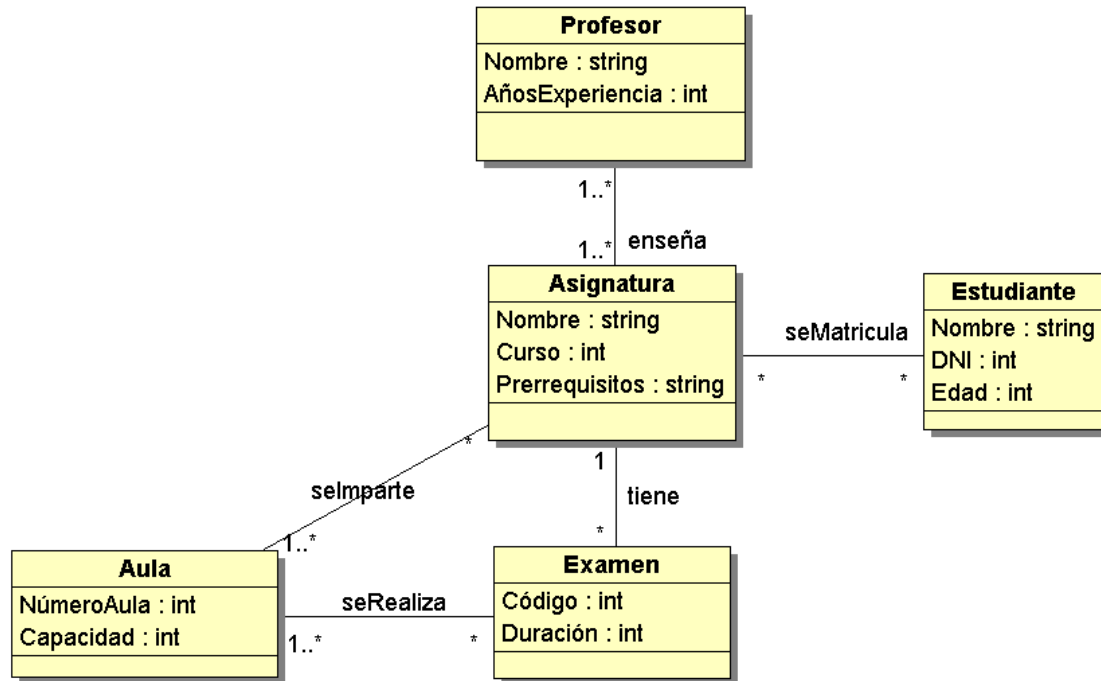


Figura 20: Ejemplo de Diagrama de Clases (II).

En el ejemplo de la Figura 20 se han definido las clases *Profesor*, *Asignatura*, *Estudiante*, *Examen* y *Aula*. Además, para cada asociación establecida entre clases se indica el propósito de la asociación y la multiplicidad de los roles de la asociación. El nivel de detalle usado para especificar las diferentes clases incluye los atributos y su tipo de dato.

Un último ejemplo es el que se presenta en la Figura 21 en el que el nivel de detalle usado en su especificación correspondería al nivel de detalle que se propone utilizar cuando un Diagrama de Clases se utilice para representar un Modelo del Dominio.

En este ejemplo una *Venta* se compone de una o más *Líneas de Productos*, está vinculada a un *Pago* y es registrada por un TPDV<sup>13</sup>. Además, cada *Línea de Producto* registra la venta de un *Producto*, aunque no todos los *Productos* tienen por qué venderse en cada *Venta*. Finalmente, los *Productos* se almacenan en la *Tienda* la cual contiene uno o más TPDV.

<sup>12</sup> La punta de flecha junto a la clase *Rueda* indica que la clase software *Bicicleta* debería tener un atributo de tipo *Rueda* modelado como una colección de dos elementos, para cada que cada *Bicicleta* pudiera mantener la visibilidad a sus dos *Ruedas*.

<sup>13</sup> Las siglas TPDV corresponden a Terminal de Punto de Venta.



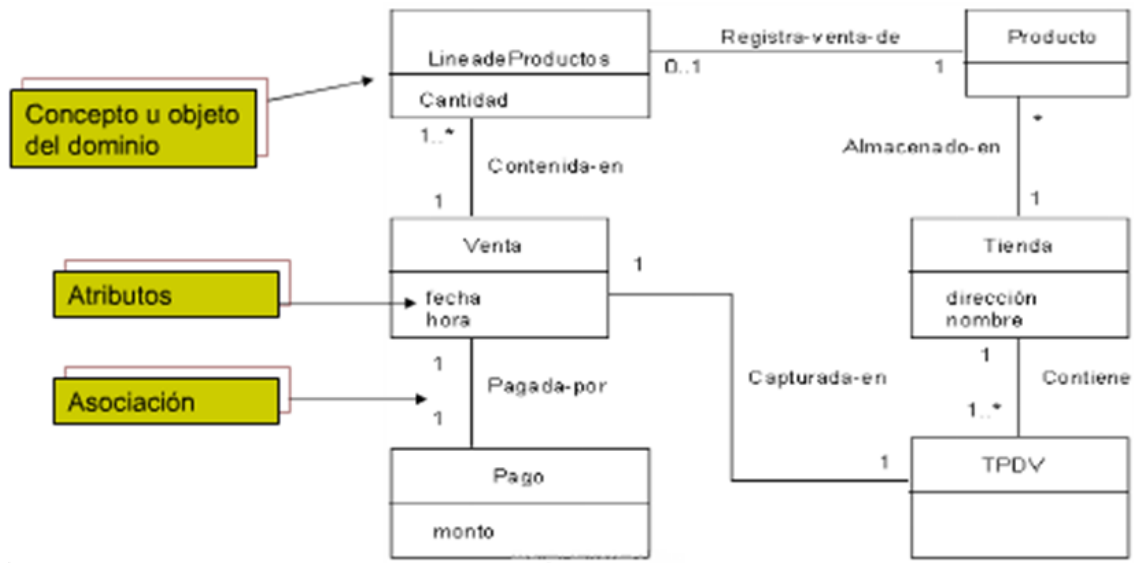


Figura 21: Ejemplo de Diagrama de Clases (III).

### 3 DIAGRAMA DE OBJETOS

El Diagrama de Objetos es un caso especial del Diagrama de Clases lo que implica que igualmente forma parte de la vista estática del sistema. El Diagrama de Objetos se caracteriza por mostrar las instancias específicas de las clases involucradas en el Diagrama de Clases en un momento temporal determinado, por lo que previamente debe haberse decidido qué situación se desea representar. El Diagrama de Objetos es por tanto un diagrama que modela las instancias de las clases mostradas en el Diagrama de Clases y que muestra dichas instancias y cómo se relacionan entre ellas, permitiendo proporcionar una visión de casos reales que ayudará a comprender mejor el funcionamiento del sistema. Es decir, que un Diagrama de Objetos presenta los objetos y sus relaciones, pero en un momento concreto del sistema. Los Diagramas de Objetos resultan de utilidad para entender los Diagramas de Clases. Además, debe tenerse en cuenta que los Diagramas de Objetos no reflejarán la multiplicidad en las relaciones representadas.

En la Figura 22 se muestra un Diagrama de Clases en la parte superior de la figura, junto con un Diagrama de Objetos vinculado en la parte inferior de la figura.

Tal y como puede observarse, en el Diagrama de Objetos lo que se muestran son instancias nombradas presentándose el nombre de la instancia en cuestión, seguido de dos puntos (:) y del nombre de la clase a la que pertenece, como, por ejemplo, **Orbea: Bicicleta** que hace referencia a la instancia **Orbea** de la clase **Bicicleta**, o **Delantera: Rueda** que hace referencia a la instancia **Delantera** de la clase **Rueda**.

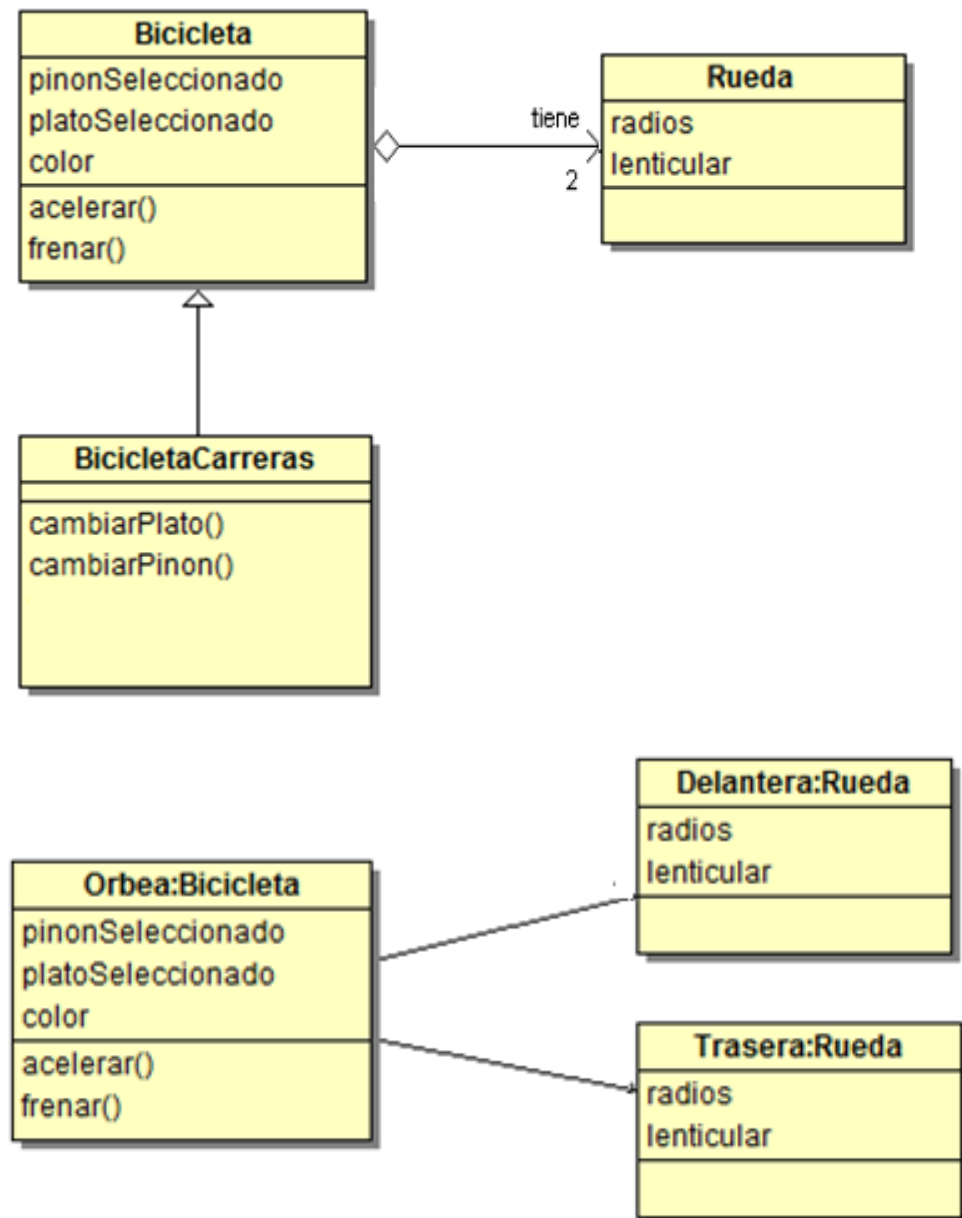


Figura 22: Ejemplo de Diagrama de Clases y Diagrama de Objetos (I).

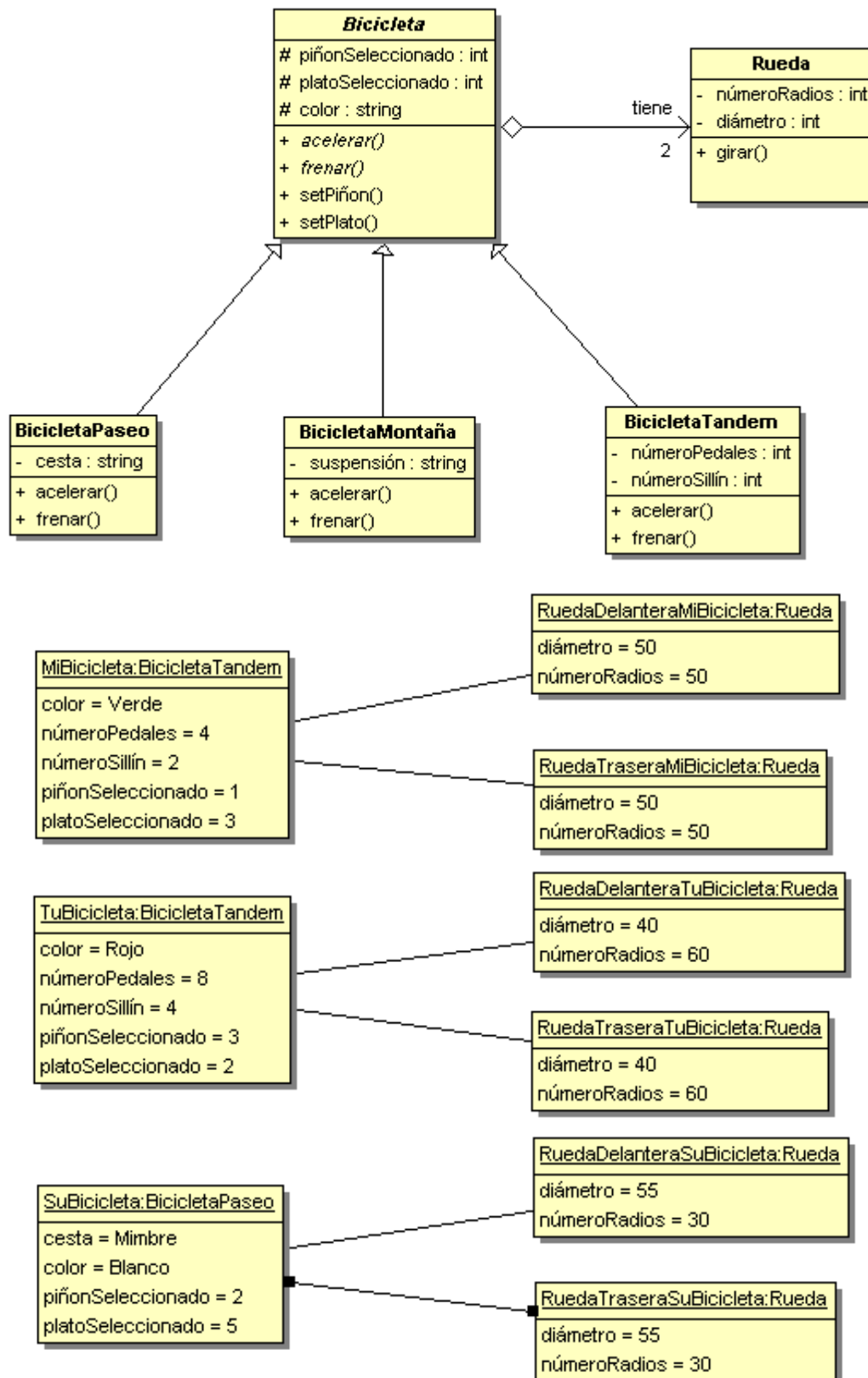


Figura 23: Ejemplo de Diagrama de Clases y Diagrama de Objetos (II).

En la Figura 23 se muestra nuevamente un Diagrama de Clases en la parte superior de la figura, junto con un Diagrama de Objetos vinculado en la parte inferior de la figura. El ejemplo es similar al de la Figura 22, pero se ha representado con un nivel de detalle superior y se han indicado asimismo en el Diagrama de Objetos los valores que toman los diferentes atributos miembro<sup>14</sup> de las diferentes instancias de las clases representadas. Igualmente, en el Diagrama de Objetos lo que se muestran son instancias nombradas como, por ejemplo, `MiBicicleta: BicicletaTandem` que hace referencia a la instancia `MiBicicleta` de la clase `BicicletaTandem`.

## 4 MODELO DEL DOMINIO

### 4.1 ¿QUÉ ES EL MODELO DEL DOMINIO ?

Como parte de la Actividad de Análisis, típicamente durante la Fase de Elaboración del Proceso Unificado de Desarrollo de Software se investigará sobre los conceptos relacionados con el subconjunto de Casos de Uso que se esté contemplando, intentando llegar a una buena comprensión del problema bajo estudio por parte del equipo de desarrollo, sin entrar, no obstante, a abordar cómo será la solución en cuanto a detalles de su implementación. Una parte de la investigación sobre el dominio o negocio del problema consistirá por tanto en identificar los conceptos que lo conforman. Para representar estos conceptos se utilizará un Modelo del Dominio también denominado Conceptual o Modelo del Negocio.

Los Modelos del Dominio se representan haciendo uso de Diagramas de Clases que, como ya se ha comentado, son Diagramas que forman parte de la Estructura Estática del Lenguaje de Modelado Unificado y que también se utilizarán para representar los Diagramas de Clases del Diseño. Los Modelos del Dominio y los Diagramas de Clases del Diseño son distintos desde un punto de vista conceptual, ya que mientras que los primeros modelan elementos del dominio o del negocio, los segundos presentan los elementos que formarán parte de la solución software. Ambos comparten, sin embargo, como cabe esperar, la notación para los elementos que los forman (clases) y las relaciones que existen entre los mismos (asociaciones) puesto que, tal y como ya se ha comentado, se representan haciendo uso del mismo artefacto UML. Se podría hablar en este sentido de clases conceptuales y de clases del diseño. Las primeras se corresponderían con la esencia del universo del problema, mientras que la segunda categoría surgiría de la solución lógica planteada durante la Actividad de Diseño. Las clases conceptuales podrían considerarse en términos de tres elementos que serían su símbolo, su intención y su extensión. El símbolo sería la palabra o imágenes que representan una clase conceptual, la intención sería la definición de la clase conceptual y la extensión sería el conjunto de ejemplos a los que se aplica la clase conceptual [ver Figura 24].

<sup>14</sup> Los atributos miembro son atributos vinculados a las instancias u objetos de una clase pudiendo tomar un valor diferente para cada instancia u objeto. Véase más información sobre el concepto de atributos miembro en el tutorial de Orientación a Objetos.

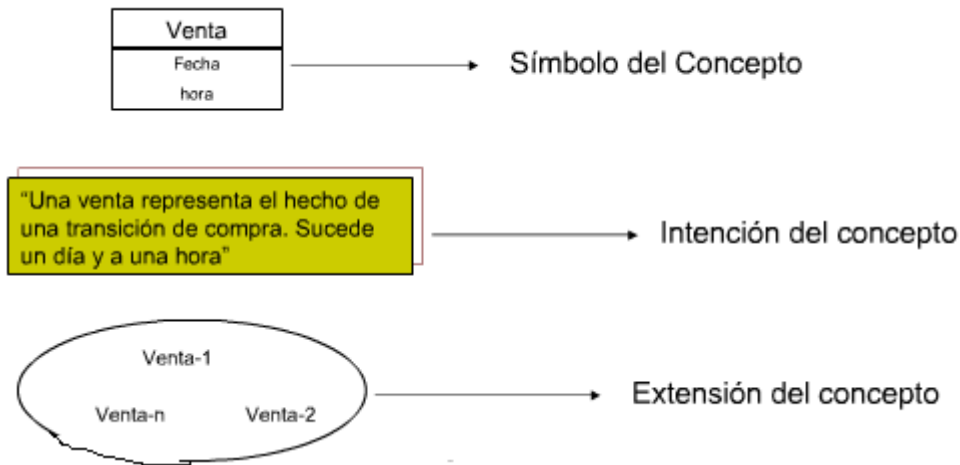


Figura 24: Ejemplo de los elementos de la clase conceptual para el evento de una transacción.

Como en el Modelo del Dominio no se trabajaría con clases software, se evitará incluir elementos como la Interfaz Gráfica de Usuario (GUI, Graphical User Interface), las Bases de Datos, etc., salvo que el dominio a modelar estuviera relacionado con alguna aplicación del ámbito del software, y se tratará de alcanzar una representación visual abstracta mediante las clases conceptuales del dominio o negocio al que corresponde el problema bajo estudio. Por este motivo, tampoco tendrían cabida en un Modelo del Dominio los métodos, ya que harían referencia a las responsabilidades de las clases conceptuales, mientras que en este momento lo relevante son sus propiedades.

El Modelo del Dominio persigue, por consiguiente, una representación de conceptos del mundo real, no de componentes software, y tiene por objetivo mejorar la comprensión del problema. Además será un diagrama que se compartirá con el cliente. Por tanto, a la hora de incluir conceptos en el Modelo del Dominio, es mejor crear un modelo con muchos conceptos que lo contrario, corriendo el peligro de olvidar algún concepto que pudiera ser importante.

Finalmente señalar que aunque, tal y como ya se ha comentado, el Modelo del Dominio y el Diagrama de Clases del Diseño son diferentes, puesto que persiguen objetivos distintos, y se representan mediante un Diagrama de Clases del Lenguaje de Modelado Unificado con diferente nivel de detalle, que contiene clases conceptuales o conceptos y clases del diseño o clases software respectivamente, sí puede afirmarse que el Modelo del Dominio servirá de inspiración para el Modelo del Diseño, que incluirá el Diagrama de Clases del Diseño, y que, aunque no deberá existir una equivalencia una a una entre clases conceptuales y clases del diseño o clases software, sí debería el diagrama realizado durante la Actividad de Análisis – Modelo del Dominio – inspirar el que más tarde se realizará como parte de la Actividad de Diseño – Diagrama de Clases del Diseño –.

En la Figura 25 se presenta un ejemplo de un Modelo del Dominio. En dicho Modelo del Dominio, en primer lugar, se han identificado los conceptos relevantes para el dominio relacionado con la gestión de un videoclub como Cliente, VideoClub, CarnetSocio o Catálogo. Posteriormente se ha decidido qué características se considera necesario registrar para cada concepto identificado, como por ejemplo las características de dirección, nombre y teléfono para el concepto VideoClub. Finalmente se han representado las asociaciones existentes entre dichos conceptos, y sus características relevantes en el contexto de un Modelo del Dominio como puede ser su multiplicidad. Algunas de las relaciones mostradas son el hecho de que un VideoClub posee un Catálogo o el hecho de que un Cliente tiene un Carnet de Socio.

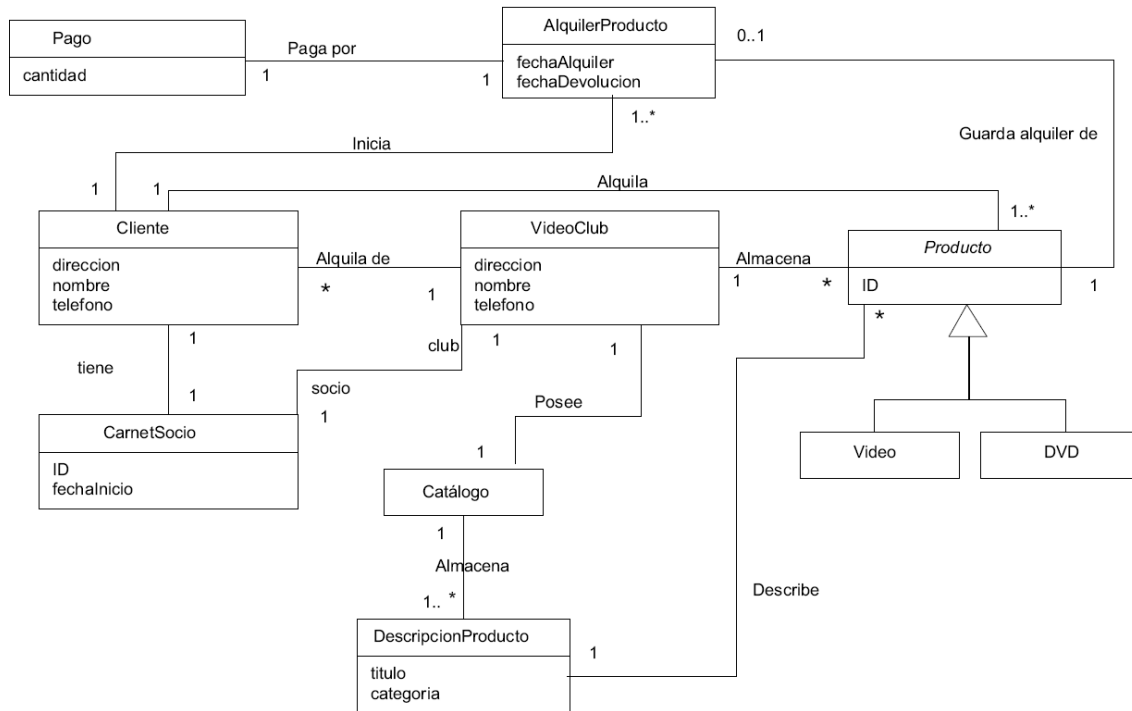


Figura 25: Ejemplo de un **Modelo del Dominio**, **Modelo del Negocio** o **Modelo Conceptual** representado mediante un Diagrama de Clases en la notación del Lenguaje de Modelado Unificado.<sup>15</sup>

## 4.2 CONSEJOS PARA REALIZAR UN BUEN MODELO DEL DOMINIO

### 4.2.1 Pasos para la elaboración del Modelo del Dominio

Para crear el Modelo del Dominio se deben seguir los pasos que a continuación se describen:

- Hacer una lista de conceptos candidatos recurriendo a la lista sugerida de categorías de conceptos [ver Tabla 2] así como al análisis nominal de los Requisitos y Casos de Uso en consideración [ver sección 4.2.2 Identificación de Conceptos].
- Representarlos en un Diagrama de Clases del Lenguaje de Modelado Unificado.
- Añadir los atributos necesarios para contener todas las características o información que se necesite registrar de cada concepto.

<sup>15</sup> En este Modelo del Dominio puede observarse un ejemplo del uso de Clases de Especificación o de Descripción ya que la clase *DescripcionProducto* es una clase que se usa para especificar o describir la clase *Producto*. Véase más información en la sección del presente documento dedicada a este tema de forma específica.

Nótese también que el nombre de la clase *Producto* se muestra en cursiva para indicar que dicha clase es abstracta, información que, por otra parte, no sería imprescindible haber proporcionado en el Modelo del Dominio.

- Añadir las asociaciones necesarias para ilustrar las relaciones entre conceptos que sea necesario conocer, así como las características de sus roles<sup>16</sup>, en especial su multiplicidad, que se consideren relevantes.

## 4.2.2 Identificación de Conceptos

### 4.2.2.1 Categorías de Conceptos Típicos

Tal y como se ha comentado, el objetivo del Modelo del Dominio es hacer una representación de conceptos del mundo real, en concreto de los conceptos del dominio o del negocio bajo estudio, para conseguir mejorar la comprensión del problema. Para identificar los conceptos intervinientes es conveniente tomar como base los Casos de Uso involucrados, los Requisitos especificados, así como el conocimiento general que se tenga acerca del dominio o negocio del problema. En la siguiente tabla se muestran algunas categorías de conceptos típicos que podrían contemplarse en el Modelo del Dominio, junto con ejemplos pertenecientes al dominio o negocio de la gestión de supermercados y al dominio o negocio de la gestión de la reserva y compra de billetes de avión. [ver Tabla 2]

Tipo de Concepto	Ejemplos pertenecientes al dominio o negocio de la gestión de supermercados	Ejemplos pertenecientes al dominio o negocio de la gestión de la reserva y compra de billetes de avión
Objetos físicos o tangibles	Terminal de Caja	Avión
Especificaciones, diseños o descripciones	Especificación de Producto	Descripción de Vuelo
Lugares	Supermercado	Aeropuerto
Transacciones	Venta	Pago, Reserva
Líneas de la transacción	Artículo de Venta	Vuelo Vendido o Reservado
Roles de personas	Cajero	Piloto, Pasajero
Contenedores de otros elementos	Supermercado, Cesta	Avión
Elementos en un contenedor	Artículo	Pasajero
Otros ordenadores o sistemas informáticos o electromecánicos externos al sistema en cuestión	Sistema de Autorización de Tarjetas de Crédito	Sistema Controlador de Tráfico Aéreo
Conceptos abstractos	Hambre	Acrofobia <sup>17</sup>
Organizaciones	Departamento de Ventas	Compañía Aérea
Eventos, Hechos, Procesos	Venta, Robo, Reposición de	Vuelo, Accidente, Aterrizaje

<sup>16</sup> Como ya es sabido, el término roles hace referencia a los extremos de las asociaciones que pueden estar caracterizados por diversos elementos como su nombre o su multiplicidad.

<sup>17</sup> La acrofobia se define como el miedo a las alturas.

	Artículos	
Reglas y políticas	Política de Devoluciones	Política de Cancelaciones
Catálogos	Catálogo de Productos	Catálogo de Vuelos
Archivos, Registros... financieros, de trabajo, de contratos, de asuntos legales...	Recibo, Contrato de Empleo	Registro de Reservas, Registro de Cancelaciones
Instrumentos, Servicios... financieros	Línea de Crédito, Stock	Línea de Crédito
Documentos, Manuales, Libros...	Manual del Empleado	Normas de Seguridad
[...]	[...]	[...]

Tabla 2: Categorías de conceptos típicos junto con ejemplos pertenecientes al dominio o negocio de la gestión de supermercados y al de la gestión de la reserva y compra de billetes de avión.

#### 4.2.2.2 Análisis lingüístico de Casos de Uso y Requisitos

Otro consejo para identificar los conceptos que formarán parte del Modelo del Dominio consiste en buscar sustantivos en la descripción textual de los Casos de Uso, fundamentalmente en los diferentes escenarios descritos y en especial en el escenario principal de éxito, y en los Requisitos enunciados dentro o fuera de los Casos de Uso, es decir, en realizar un *análisis lingüístico de los nombres y/o frases nominales* existentes en ellos y considerarlos como clases conceptuales, o atributos, candidatos [ver Figura 26].

Sin embargo, la imprecisión del lenguaje natural es un punto débil de este enfoque, por lo que se recomienda combinar esta técnica con la técnica anterior de la lista sugerida de categorías de clases conceptuales.

El objetivo final, independientemente de la técnica/s utilizada/s, sería el obtener una lista de clases conceptuales candidatas del dominio que deberá restringirse y acotarse en base al Escenario concreto que se contemple.

Escenario Principal de Éxito (Procesar Venta)
1. El <b>cliente</b> llega a un <b>terminal PDV</b> con <b>mercancías</b>
2. El <b>cajero</b> comienza una nueva <b>venta</b>
3. El <b>cajero</b> introduce el <b>identificador del artículo</b> .
4. El sistema registra la <b>línea de la venta</b> y presenta la <b>descripción del artículo, precio y suma</b> parcial. El cajero repite el paso 3 y 4 hasta que termine
5. El sistema presenta el total con los <b>impuestos</b> calculados
6. El Cajero le dice al cliente el total y solicita el <b>pago</b>
7. El cliente paga y el sistema gestiona el pago
8. El sistema registra la <b>venta</b> completa y envía la información de la venta y el pago al sistema de <b>contabilidad</b> externo y al sistema de <b>inventario</b> .
9. El sistema presenta el <b>recibo</b>
10. El cliente se va con el recibo y las mercancías (si es el caso)



Figura 26: Ejemplo del análisis lingüístico de los Casos de Uso para la identificación de las clases conceptuales del Modelo del Dominio.

#### 4.2.2.3 Visión del Cartógrafo

Para identificar y dar nombre a los conceptos a incluir en el Modelo del Dominio se puede usar la analogía con la labor que desarrolla un cartógrafo o *visión del cartógrafo*, resumida en los siguientes tres puntos:

- Utilizar los nombres existentes en el territorio: Así es importante utilizar el vocabulario del dominio para nombrar los conceptos (y sus características).
- Excluir características irrelevantes: Al igual que el cartógrafo elimina características no relevantes según la finalidad del mapa, por ejemplo, datos de población en un mapa de carreteras, un Modelo del Dominio debe intentar excluir conceptos del dominio que no sean pertinentes en base a los Requisitos o Casos de Uso considerados.
- No añadir elementos que no están: Si algo no pertenece al dominio del problema no debe añadirse de forma artificial al Modelo del Dominio.

#### 4.2.3 Clases Conceptuales de Especificación o de Descripción

En relación con las clases conceptuales a incluir en el Modelo del Dominio, debe tenerse en cuenta que en ocasiones será necesario utilizar lo que se denomina clases conceptuales de especificación o de descripción. Una clase conceptual de especificación o de descripción es una clase conceptual que recoge información sobre otra clase conceptual. Así, por ejemplo, siendo `Vaso` una clase conceptual, su clase conceptual de especificación o de descripción podría ser `DescripciondeVaso`. Estas clases conceptuales de especificación o de descripción se utilizarán para la tarea de especificar o describir las clases conceptuales y son distintas de las clases a las que describen. En general serán necesarias cuando se necesite la descripción de un artículo o servicio independientemente de la existencia actual de algún artículo o servicio, es decir, de alguna instancia de la clase conceptual. Su utilización hace que la eliminación de las instancias de la clase conceptual no tenga como resultado una pérdida de información que se necesita mantener, debido a la asociación incorrecta de dicha información con las instancias eliminadas. Además, su utilización reduce la información redundante o duplicada.

En la Figura 27 se presenta un ejemplo de la utilización de una clase conceptual de especificación o descripción `EspecificaciondeProducto` que sustituiría a una versión incorrecta en la que solo se utiliza la clase `Producto` que no permite separar las instancias u objetos de su descripción. La especificación de los productos implementada mediante la clase `EspecificaciondeProducto` tiene características comunes como el precio o el código de barras y esta descripción es independiente de las instancias u objetos en sí de la clase `Producto`, por lo que eliminar instancias u objetos concretos no implicaría necesariamente eliminar la descripción de ese tipo de instancias u objetos, a la par que se evita que exista redundancia en las instancias de la clase `Producto`.

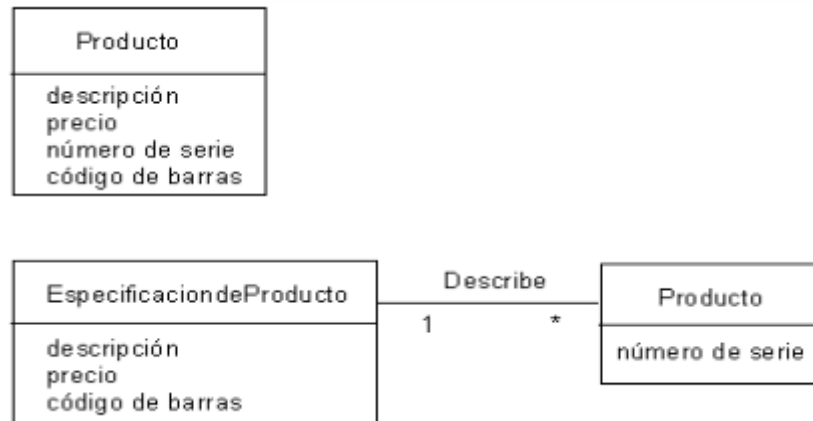


Figura 27: Ejemplo de uso de clases conceptuales de especificación o descripción.

#### 4.2.4 Atributos

Una vez identificadas las clases conceptuales es necesario identificar sus atributos para también incorporarlos al Modelo del Dominio de forma que sea posible satisfacer las necesidades de información de los Casos de Uso o Requisitos que se estén abordando en ese momento.

Los atributos deben tomar valores de tipos de datos simples – o primitivos<sup>18</sup> – (número, texto, etc.), puesto que los tipos complejos deberán ser en general modelados como conceptos. De hecho, un error típico es modelar un concepto complejo del dominio como un atributo. Incluso cuando un valor es de un tipo simple es más conveniente representarlo como concepto en diferentes situaciones como las que se enumeran a continuación:

- **Tiene operaciones asociadas.** Por ejemplo, un NIF podría tener asociada una operación de validación.
- **Tiene otros atributos asociados.** Por ejemplo, un precio de oferta podría tener fecha de fin.
- **Es una cantidad con una unidad.** Por ejemplo, un precio podría estar en pesetas o en euros.
- **Se compone de distintas partes o secciones.** Por ejemplo, un número de teléfono, el nombre de una persona, etc.

Es decir, en el dilema conceptos vs atributos podría decirse que en principio se crean atributos solo cuando su tipo de dato es simple y, además, en caso de duda sobre un elemento, éste se definirá como una clase conceptual en vez de como un atributo. De hecho, el número de atributos en un Modelo del Dominio debería ser escaso, únicamente para clases conceptuales en las que exista la necesidad de registrar información. Además, tal y como ya se ha comentado, es preferible que existan muchos conceptos (o clases conceptuales) a que existan pocos y se corra el riesgo de dejar fuera del análisis aspectos importantes del problema. Así, en el ejemplo de la Figura 28 se plantea el dilema sobre si *Destino* debiera ser un atributo de la clase conceptual *Vuelo* o bien una clase conceptual independiente de la clase conceptual *Vuelo*. En caso de duda, es preferible representar ambos

<sup>18</sup> Se denomina tipo primitivo o tipo elemental a los tipos de datos originales de un lenguaje de programación, esto es, aquellos que proporciona el lenguaje. Ejemplos típicos de tipos de datos primitivos son `char` (carácter), `int` (entero), `float` (real (coma flotante)) o `booleano` (lógico: Verdadero, Falso).

Vuelo y Destino (Aeropuerto) como conceptos separados y dejar así abierta la posibilidad de que alguna de dichas clases conceptuales, o ambas, pudiera inspirar la correspondiente clase software.

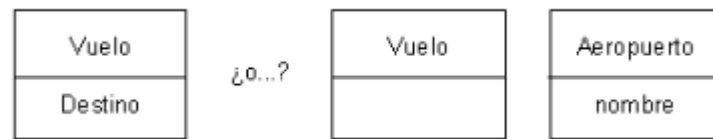


Figura 28: Dilema conceptos vs atributos.

Por otra parte, a la hora de añadir los atributos a las clases conceptuales del Modelo del Dominio debe tenerse en cuenta que, en general, no se especificarán los tipos de datos, salvo que aporte claridad. Además, tal y como ya se ha comentado, el tipo de dato de un atributo no debería ser un concepto del dominio como *Venta* o *Aeropuerto*, sino que siempre debería ser un tipo de dato simple.

Otra regla básica es que un atributo no podrá crear instancias diferentes de él, mientras que un concepto o clase conceptual sí podrá instanciarse. Así, por ejemplo, el concepto *Estudiante* tendría entidad para ser clase conceptual puesto que puede instanciarse, pero no parece suceder lo mismo con el concepto *número de matrícula*. Por tanto, parecería la opción más acertada el que *Estudiante* fuera una clase conceptual mientras que el *número de matrícula* fuera un atributo de dicha clase.

Finalmente, es importante no perder la perspectiva de que lo que se está creando en un Modelo del Dominio y no un Diagrama de Clases del Diseño. El no deslizarse al Diseño implicará que no se deberán añadir atributos que *referencien* objetos, de forma que las clases conceptuales se relacionen mediante asociaciones y no mediante atributos al estilo de las claves ajenas o foráneas utilizadas en el diseño de las bases de datos relacionales. Así, en la Figura 29 la segunda opción en la que la relación entre los conceptos *Cajero* y *TPDV* se establece mediante la asociación *usa*, se prefiere frente a la primera, en la que se estaría utilizando el atributo *númeroTPDV* para establecer el vínculo entre ambos conceptos o clases conceptuales.

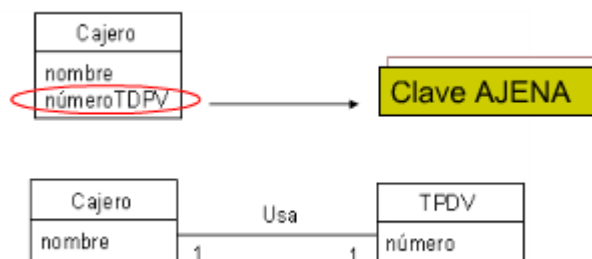


Figura 29: Ejemplo de relaciones mediante atributos.

#### 4.2.5 Asociaciones en el Modelo del Dominio

Las asociaciones en el Modelo del Dominio son relaciones entre conceptos que resultan de interés en el conjunto de Casos de Uso o Requisitos que se está tratando. Una asociación es una relación existente entre clases conceptuales, o entre instancias de dichas clases, que indica alguna conexión significativa e interesante. Las asociaciones en el Modelo del Dominio son abstracciones, es decir, que no representan conexiones entre entidades software por lo que se pueden definir asociaciones que luego no existan en la implementación. En el Modelo del Dominio es conveniente incluir las asociaciones para las que el conocimiento de la relación necesita mantenerse durante un cierto período de tiempo, es decir, asociaciones del tipo *necesita-conocer*. Lo prioritario es centrarse en este tipo de asociaciones, no obstante, estas se podrían completar con otras de menor relevancia, que podrían catalogarse como del tipo *solo-comprensión*, para enriquecer el conocimiento básico del dominio.

En la siguiente tabla se muestran algunas categorías de asociaciones típicas que podrían contemplarse en el Modelo del Dominio, junto con ejemplos pertenecientes al dominio o negocio de la gestión de supermercados y al dominio o negocio de la gestión de la reserva y compra de billetes de avión. [ver Tabla 3]

Categoría de Asociación	Ejemplos pertenecientes al dominio o negocio de la gestión de supermercados	Ejemplos pertenecientes al dominio o negocio de la gestión de la reserva y compra de billetes de avión
A es una parte física de B	Estantería – Supermercado	Ala – Avión
A es una parte lógica de B	Línea de Venta – Venta	Etapa de Vuelo – Ruta de Vuelo
A está físicamente contenido en B	Artículo – Estantería	Pasajero – Avión
A está lógicamente contenido en B	Descripción de Artículo – Catálogo	Descripción de Vuelo – Catálogo
A es una descripción de B	Descripción de Artículo – Artículo	Descripción de Vuelo – Vuelo
A es un elemento en una transacción o un informe B	Realización de una Reposición – Registro de Reposiciones Línea de Venta – Venta	Trabajo de Reparación – Registro de Reparaciones Realización de una Cancelación – Registro de Cancelaciones
A es registrado/archivado/capturado en B	Venta – Terminal de Caja	Reserva – Sistema de Reservas en línea
A es un miembro de B	Cajero – Supermercado	Piloto – Compañía Aérea
A es una subunidad organizativa	Sección – Supermercado	Mantenimiento – Compañía

de B		Aérea Ventas – Compañía Aérea
A usa o gestiona B	Cajero – Terminal de Caja	Piloto – Avión
A se comunica con B	Cliente – Cajero	Empleado de Agencia de Viajes – Pasajero
A está relacionado con una transacción B	Cliente – Pago	Pasajero – Reserva
A es una transacción relacionada con otra transacción B	Pago – Venta	Reserva – Cancelación
A está junto a B	Cajero – Terminal de Caja	Ciudad – Aeropuerto
A posee B	Supermercado – Terminal de Caja	Compañía Aérea – Avión
A es propiedad de B	Terminal de Caja – Supermercado	Avión – Compañía Aérea
A es un evento relacionado con B	Venta – Cliente	Despegue – Vuelo Aterrizaje – Vuelo
[...]	[...]	[...]

Tabla 3: Lista de asociaciones típicas a tener en cuenta entre conceptos en el Modelo del Dominio.

Debe tenerse en cuenta, no obstante, que es más importante identificar las clases conceptuales que identificar sus asociaciones. Ya que, demasiadas asociaciones podrían confundir un Modelo del Dominio en lugar de aclararlo<sup>19</sup> y su descubrimiento podría llevar además bastante tiempo, con beneficio marginal, por lo que se considera suficiente con identificar y representar las conexiones más significativas en el conjunto de los Casos de Uso y Requisitos abordados. Además, es preferible evitar el mostrar asociaciones redundantes.

Una vez identificadas las asociaciones existentes entre las clases conceptuales, estas se representan en el Modelo del Dominio con sus características: multiplicidad, etc., adecuadas.

Definidas las clases conceptuales, sus atributos y sus relaciones, es posible afirmar que ya se cuenta con un Modelo del Dominio, el cual no será un modelo definitivo, puesto que a lo largo del proceso se le pueden ir añadiendo conceptos que inicialmente no se hubieran considerado.

<sup>19</sup> Debe tenerse en cuenta que muchas líneas en un diagrama contribuirán a añadir ruido visual y harán que dicho diagrama resulte menos comprensible.

Nota de los profesores: Si detectas algún error, piensas que alguna información está presentada de manera confusa, o que sobra o falta algún contenido, por favor, envía un mensaje a [mperez@tel.uva.es](mailto:mperez@tel.uva.es). Gracias por anticipado. Vuestras sugerencias se utilizarán para mejorar esta documentación.