

# 1. ORGANIZACIÓN EN MODELOS

Para iniciar la diagramación usando el MS Visio 2002, es necesario seguir la siguiente secuencia File\New\Software\UML Model Diagram. Se abre una página y en el lado izquierdo aparecen todas las plantillas de iconos organizadas según el diagrama o modelo UML que se puede construir.

Para poder organizar los contenidos de un proyecto adecuadamente, Visio nos proporciona un Explorador de Modelos, que si no se visualiza por defecto, se puede mostrar con la opción UML\View\ModelExplorer. Por defecto, presenta el aspecto de la Figura 1.1.

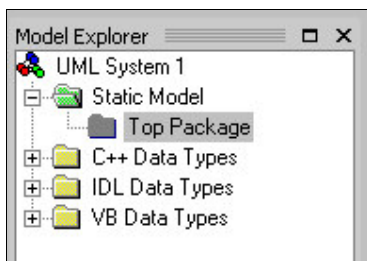


Figura 1.1 Explorador de Modelos por defecto.

Cada uno de los iconos anteriores admite su propio menú de opciones, usando el botón derecho del ratón. Una de las opciones más útiles es Rename, ya que por defecto Visio asigna nombres numerados a todos los elementos e interesa usar una nomenclatura acorde a lo que se intenta expresar. A continuación se sugiere una organización de modelos.

## 1.1 NIVEL PRINCIPAL

En la Figura 1.2 se ilustra los distintos modelos en los que se puede organizar un proyecto. Como puede verse, el nivel raíz se puede utilizar para dar un nombre al proyecto y los demás modelos para organizar contenidos adecuadamente.

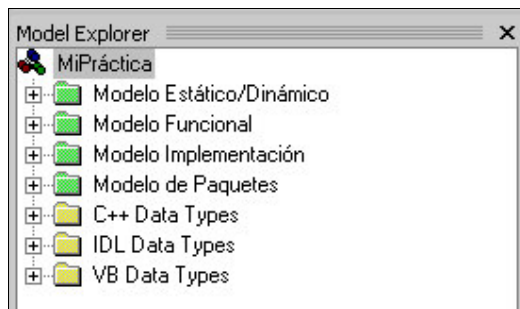


Figura 1.2 Modelos principales.

## 1.2 MODELO FUNCIONAL

En la Figura 1.3, se ilustra los contenidos del Modelo Funcional. Como puede verse, existe un paquete por defecto, llamado Top Package; pulsando el botón derecho sobre él, se muestra el menú

New, que permite añadir elementos básicos UML (clases, casos de uso,...) o diagramas completos a seleccionar, opción que abre una página para incorporar los elementos deseados.

En la Figura 1.3, se ha añadido un Diagrama de Casos de Uso en el que se ha diagramado un actor y dos casos de uso. Del mismo modo, pulsando el botón derecho del ratón sobre un icono de caso de uso, se accede a su menú New, que permite añadirle un Modelo de Actividad. Este automáticamente lleva vinculada una página a la que se puede cambiar el nombre (como a todos los elementos) y sirve para ubicar los iconos del Diagrama de Actividad, ligado a dicho caso de uso.

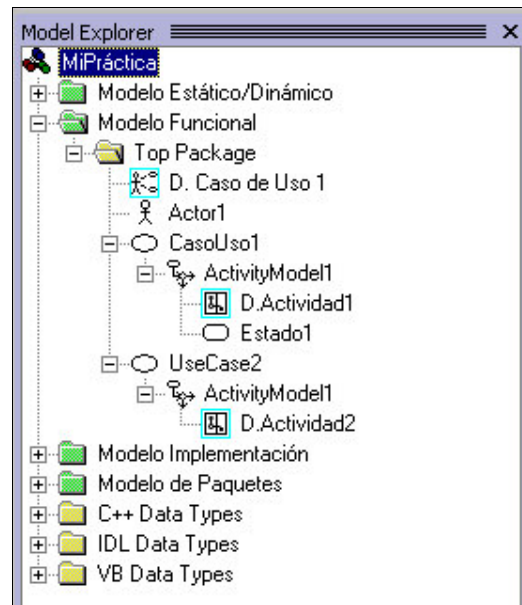


Figura 1.3 Modelo Funcional.

### 1.3 MODELO ESTATICO / DINAMICO

En la Figura 1.4, se muestra una organización posible para el Modelo Estático y Dinámico. Como puede verse, los diagramas más señalados cuelgan del Top Package y se pueden tener tantas páginas de diagramas de clases y de secuencia como sean necesarios. También se muestra cómo a una clase se le ha vinculado su propio diagrama de estados.

Se sugieren los siguientes diagramas:

- Diagrama de Clases del Dominio
- Diagrama de Clases de Interacción Hombre-Máquina y con otros Sistemas Externos
- Diagrama de Clases del Componente Gestor de Datos
- Tantas páginas de diagramas como sean necesarios, para los Diagramas de Secuencia.

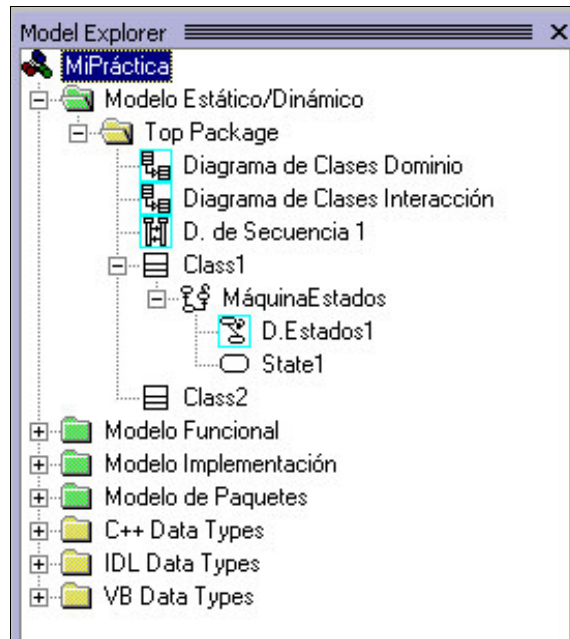


Figura 1.4 Modelado Estático y Dinámico.

## 1.4 MODELO DE PAQUETES

Para poder representar de una manera sencilla un Diagrama de Paquetes y sin usar realmente toda la potencia que Visio ofrece en este caso, merece la pena crear un modelo de paquetes como se ilustra en la Figura 1.5.

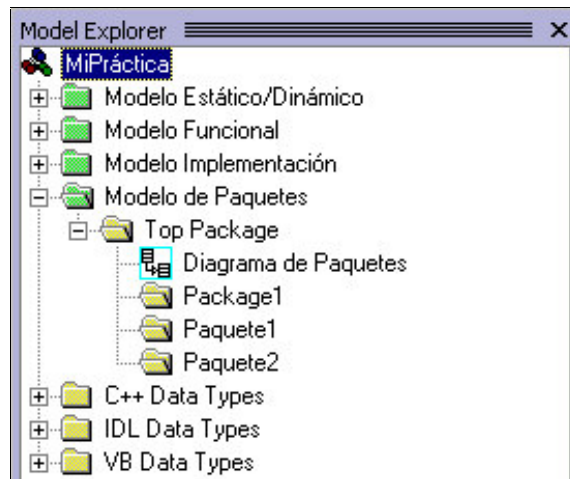


Figura 1.5 Modelo de Paquetes

Realmente, este paquete sólo sirve como soporte de una página en la que crear un diagrama de paquetes pero sin ningún vínculo con las clase que hipotéticamente deberían contener. Esto se ha hecho así para que los nombres las clases sean nombre simples y no nombres completos que hacen que los gráficos queden visualmente complicados. Nótese que cada vez que se pinte un paquete en el diagrama, Visio genera automáticamente la carpeta amarilla que lo representa, además de una

página para ubicar elementos gráficos de UML (esa parte no se visualiza en la Figura 1.5 ya que ha sido borrada con la opción Delete).

## 1.5 MODELO DE IMPLEMENTACION

Como puede verse en la Figura 1.6, el último modelo incluye el Diagrama de Componentes y el de Despliegue, diagramas que a este nivel de detalle no tienen mayor dificultad.

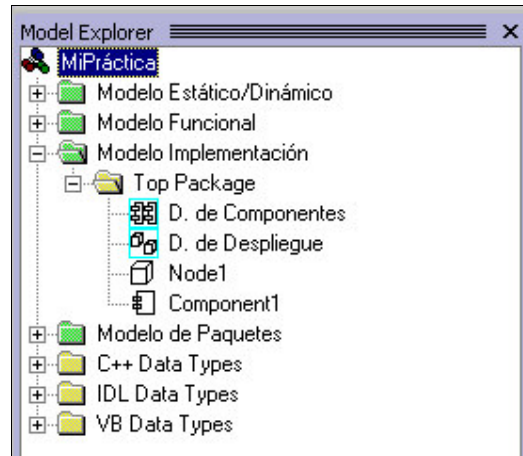


Figura 1.6 Modelo de Implementación.

## 2. DIAGRAMAS DE CASOS DE USO (UML USE CASE)

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento.

Un caso de uso describe un conjunto de secuencias, donde cada secuencia representa la interacción de los elementos externos al sistema (sus actores) con el propio sistema (y con sus abstracciones claves).

### 2.1 ENLACE “ACTOR – CASO DE USO”

Los actores sólo se pueden conectar a los casos de uso a través de asociaciones. Una **asociación** entre un actor y un caso de uso indica que el actor y el caso de uso se comunican entre sí, y cada uno puede enviar y recibir mensajes.

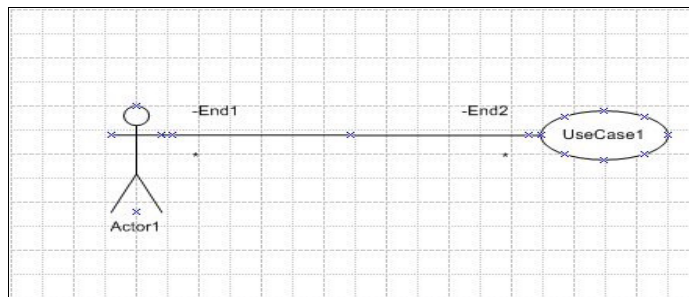


Figura 2.1 Asociación entre un actor y un caso de uso (apariencia inicial).

Por defecto, este es el aspecto que toma una asociación realizada bajo Microsoft Visio. A continuación se describen los pasos necesarios a ejecutar sobre el diagrama anterior, para adaptarlo a nuestras necesidades:

- *Asignar un nombre significativo al actor y al caso de uso: hacer “doble clic” sobre cada uno de ellos y rellenar el campo “Name”.*
- *Eliminar la multiplicidad y los roles de los extremos de la asociación: hacer “doble clic” sobre la asociación, y eliminar con la tecla de retroceso los campos “End Name” y “Multiplicity” dejándolos en blanco.*

End Name	Aggregation	Visibility	Multiplicity	IsNavigable
none		private		<input type="checkbox"/>
none		private		<input type="checkbox"/>

Figura 2.2 Eliminación de roles y multiplicidades.

- Una forma más adecuada de hacerlo y que sirve para mostrar / ocultar en general todos los adornos que UML incorpora es pulsando con el botón derecho el icono deseado y seleccionando '**Show Display Options**'. Aparece un conjunto de opciones donde seleccionar qué se desea mostrar y qué no. Además, en la parte inferior permite que las opciones elegidas se apliquen a todos los iconos ya existentes en la página o a los siguientes que se vayan a pintar.

## 2.2 ENLACE “ACTOR – ACTOR”

En este caso, pueden definirse categorías generales de actores (como “Dependiente”) y especializarlos (como “Jefe”) a través de relaciones de **generalización**.

Esta representación se consigue arrastrando un objeto de tipo “Extends” sobre la cuadrícula. Obsérvese no obstante que por defecto, su representación oculta el nombre que haya sido asociado al actor padre. Para solucionar este percance, es necesario acudir a la barra de herramientas y seleccionar la opción “Connection Point Tool”.

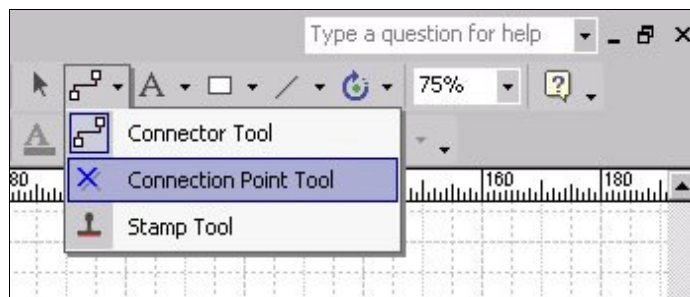


Figura 2.3 Selección de la herramienta “Connection Point Tool”.

Esta herramienta permite que aquel nuevo punto de conexión seleccionado, el cual adopta un tono rosáceo, pueda ser desplazado dentro de la cuadrícula y con ello, solventar el problema antes mencionado.

## 2.3 ENLACE “CASO DE USO – CASO DE USO”

Los casos de uso también pueden organizarse especificando relaciones de generalización, inclusión y extensión entre ellos. Estas relaciones se utilizan para factorizar el comportamiento común (extrayendo ese comportamiento de los casos de uso en los que se incluye) y para factorizar variantes (poniendo ese comportamiento en otros casos de uso que lo extienden).

Si bien todas ellas se representan arrastrando un objeto de tipo “Extends” sobre la cuadrícula, resulta imprescindible modificar este objeto según sea necesario, con motivo de diferenciar unas relaciones de otras.

- **Relación de generalización:** hacer “doble clic” sobre la relación y dejar en blanco el campo “Stereotype”.

Figura 2. 4 Omisión del campo “Stereotype” para la relación de generalización.

- **Relación de *extensión*:** hacer “doble clic” sobre la relación y elegir en el campo “Stereotype” la opción “extends”. Una vez cambiado el estereotipo de la relación debe cambiarse el aspecto de esta, para ello pulsar botón derecho sobre la misma y seleccionar la opción: *Format* → *Line*. En el nuevo cuadro de dialogo que aparece como resultado, deben modificarse los campos “Pattern” (cambia la trama de la línea), seleccionando el segundo tipo, y “Begin” (cambia el extremo de la línea), seleccionando la opción “Navigable”.

Figura 2.5 Modificación de la apariencia grafica de la relación de extensión.

- **Relación de *inclusión*:** por defecto, *Microsoft Visio* **no** incluye dentro de los estereotipos predefinidos el asociado a este tipo de relación. Incluye un icono para una versión anterior en la que el estereotipo se denominaba *USES*. Para adaptarnos adecuadamente a las últimas versiones de *UML*, lo primero que debe hacerse es declarar el estereotipo *INCLUDE* dentro de la lista de estereotipos que maneja la herramienta. Para ello, acudir al menú principal y seleccionar la opción: *UML* → *Stereotypes*.
- En el nuevo cuadro de dialogo que aparece, debe pulsarse la opción “New” para que sea introducido un nuevo estereotipo a la lista (con el nombre de “Stereotype1” por defecto). A continuación debe modificarse su descripción, escribiendo bajo el campo “Stereotype” su nombre, es decir “include”, y seleccionando bajo el campo “Base Class” la opción “Generalization”.

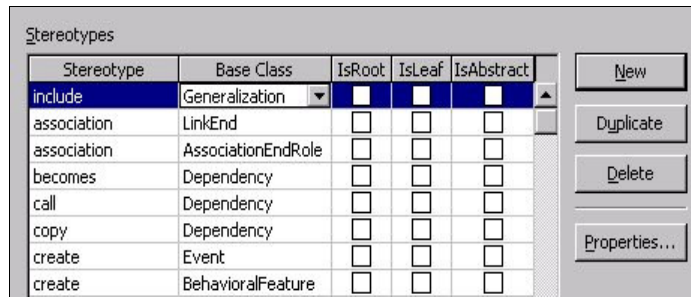


Figura 2.6 Adición del estereotipo “include” a la lista de predefinidos.

Incluido el nuevo estereotipo, hacer “doble clic” sobre la relación y comprobar que ahora dentro del campo “Stereotype” figura el nuevo estereotipo creado. Seleccionar entonces el estereotipo “include” y a continuación proceder a modificar la apariencia gráfica de la relación, siguiendo los pasos indicados para la relación de extensión.

El aspecto grafico final que han de tomar las tres relaciones explicadas anteriormente: generalización, extensión e inclusión, es el reflejado por el siguiente diagrama:

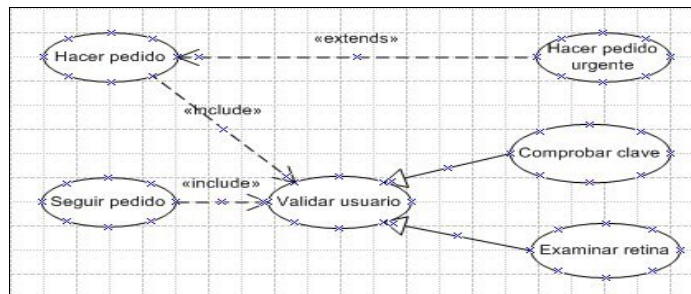


Figura 2.7 Aspecto optimo de las relaciones de generalización, extensión e inclusión.

## 2.4 MODELADO DEL CONTEXTO DEL SISTEMA

Modelar el contexto de un sistema implica dibujar una línea alrededor de todo el sistema y asegurar qué actores quedan fuera del sistema e interactúan con él.

Esta representación se consigue arrastrando un objeto de tipo “System Boundary” sobre la cuadrícula, el cual debe ser dotado de un nombre significativo (haciendo “doble clic” sobre el), así como de un tamaño apropiado.

## 2.5 REPRESENTACIÓN DE REFERENCIA

El siguiente diagrama muestra la apariencia que han de poseer los diagramas de Casos de uso desarrollados.



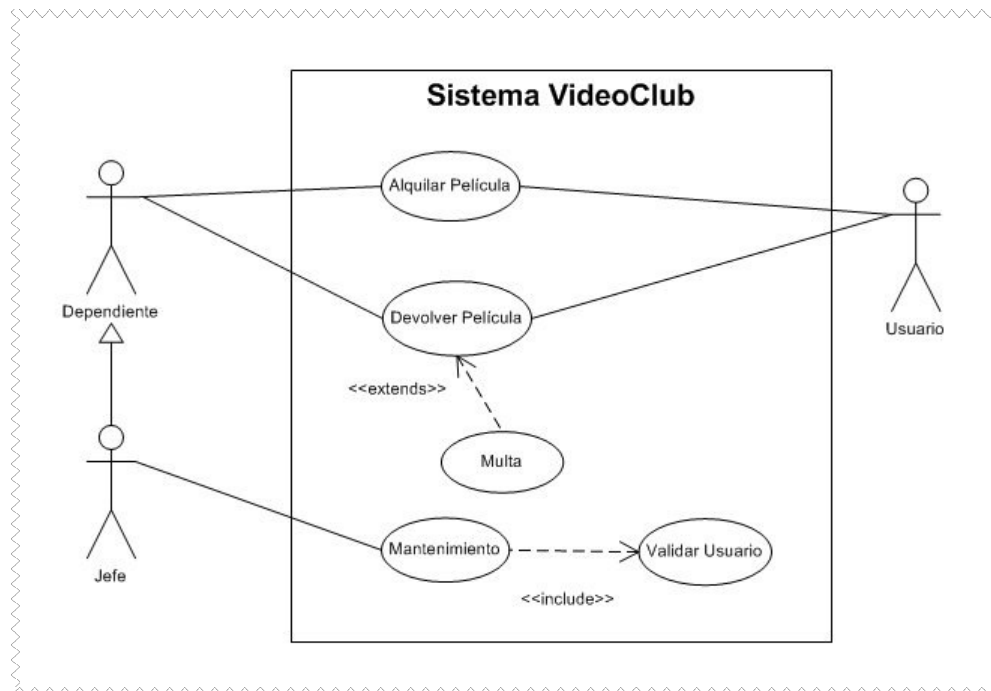


Figura 2.8 Representación de referencia – Diagrama de Casos de uso.



### 3. DIAGRAMAS DE ACTIVIDAD (UML ACTIVITY)

Un diagrama de actividades muestra el flujo de control entre actividades. Los diagramas de actividades contienen: Estados de actividad (estados compuestos) y estados de acción (estados simples), Transiciones (bifurcaciones, divisiones y uniones) y Objetos.

#### 3.1 TRANSICIONES

Cuando se completa la acción o la actividad de un estado, el flujo de control pasa inmediatamente al siguiente estado de acción o estado de actividad. Este flujo se especifica con transiciones que muestran el camino de un estado de actividad o estado de acción al siguiente.

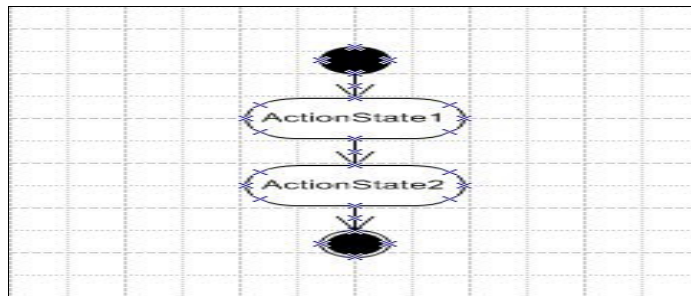


Figura 3.1 Transiciones secuenciales.

En realidad, un flujo de control tiene que empezar y parar en algún sitio (a menos, por supuesto, que sea un flujo infinito, en cuyo caso tendrá un principio pero no un final). Por lo tanto, como se aprecia en la figura, se debe especificar un estado inicial (un círculo relleno, objeto de tipo "Inicial State") y un estado final (un círculo relleno dentro de una circunferencia, objeto de tipo "Final State").

Las transiciones secuenciales son frecuentes, pero no son el único tipo de camino que se necesita para modelar un flujo de control. En los diagramas de actividad, se pueden incluir **bifurcaciones**, que especifican caminos alternativos, elegidos según el valor de alguna expresión booleana.

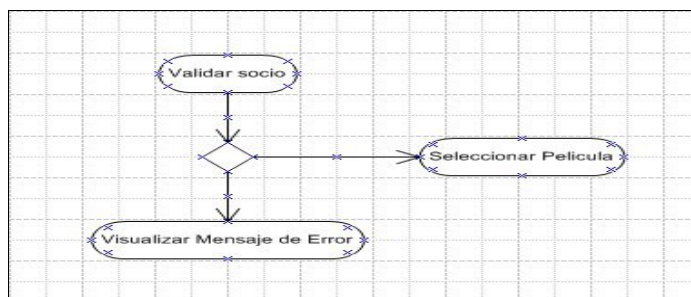


Figura 3.2 Bifurcación (apariencia inicial).

Como se muestra en la figura 3.2, una bifurcación se representa con un rombo (objeto de tipo "Decision"). Una bifurcación puede tener una transición de entrada y dos o más de salida. En cada transición de salida se coloca una expresión booleana, que se evalúa sólo una vez al entrar en la bifurcación. Para colocar dichas expresiones, es necesario hacer "doble clic" sobre cada una de las transiciones de salida, habilitar la opción "Guard", y rellenar el campo "Body" con las expresiones asociadas (utilizar la tecla de retroceso para colocar el cursor al comienzo del cuerpo).

### 3.2 REPRESENTACIÓN DE REFERENCIA

Este diagrama muestra la apariencia que han de poseer los diagramas de actividad desarrollados.

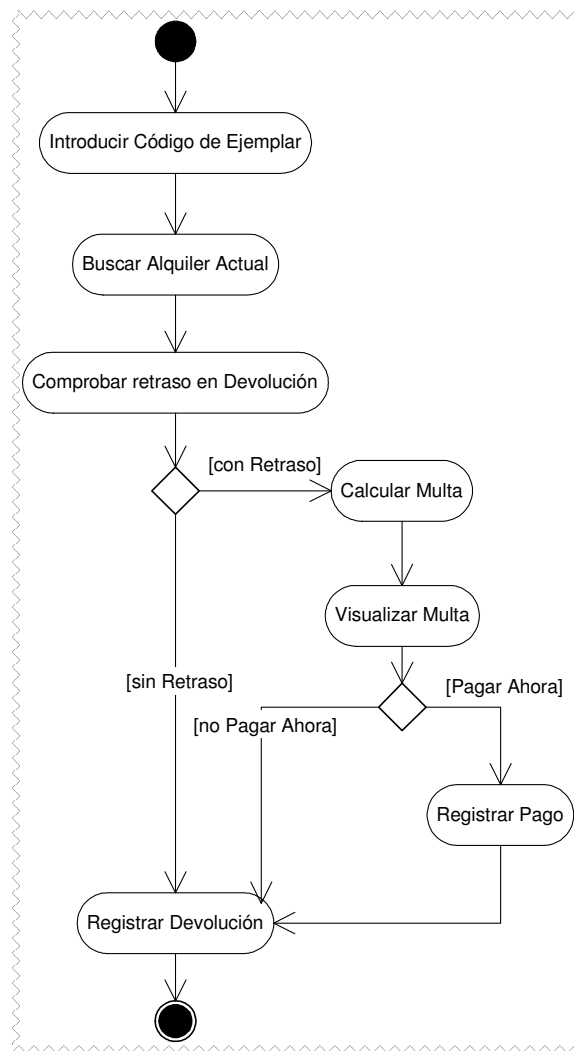


Figura 3.3 . Representación de referencia - Diagrama de Actividad.

## 4. DIAGRAMAS DE CLASES (UML STATIC STRUCTURE)

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos y se utilizan para modelar la vista de diseño estática de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas de base de datos.

Un diagrama de clases es un diagrama que muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.

### 4.1 CARACTERÍSTICAS AVANZADAS DE LAS CLASES

Las clases son los bloques de construcción más importantes de cualquier sistema orientado a objetos. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

A continuación, se expone explícitamente la secuencia de pasos que ha de ejecutarse para configurar y adecuar este bloque de construcción según nuestras necesidades. Esta configuración es gestionada desde el nuevo cuadro de dialogo que surge al hacer “doble clic” sobre un objeto de tipo “Class”, dentro del cual, debe prestarse especial atención a las pestañas “Class”, “Attributes” y “Operations”.

- **Pestaña “Class”:**
  - Asignar un nombre significativo a la clase: rellenar el campo “Name”.
  - *Especificar que la clase es abstracta en caso de no poder tener instancias directas:* marcar la casilla “IsAbstract”. La consecuencia visual de la activación de esta casilla consiste en mostrar el *nombre* de la clase en cursiva.

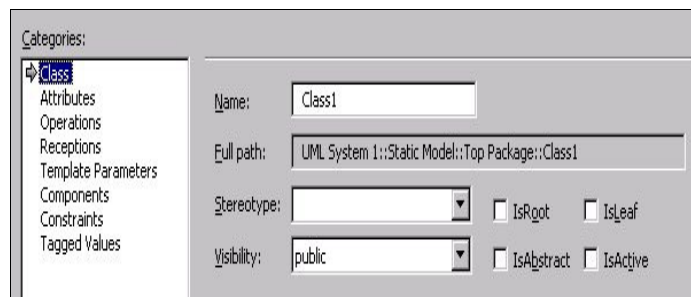


Figura 4.1 Configuración de las propiedades generales de una clase.

- Aplicar un estereotipo sobre la clase: *lo primero que debe hacerse es declarar el nuevo estereotipo dentro de la lista de estereotipos que maneja la herramienta. Para ello, acudir al menú principal y seleccionar la opción: UML → Stereotypes.*
- *En el nuevo cuadro de dialogo que aparece, debe pulsarse la opción “New” para que sea introducido un nuevo estereotipo a la lista (con el nombre de “Stereotype1” por defecto). A*

continuación debe modificarse su descripción, escribiendo bajo el campo “Stereotype” su nombre, y seleccionando bajo el campo “Base Class” la opción “Class”.

Incluido el nuevo estereotipo, comprobar ahora que en el cuadro de dialogo anterior “UML Class Properties”, y dentro del campo “Stereotype”, figura el nuevo estereotipo creado y permite su selección.

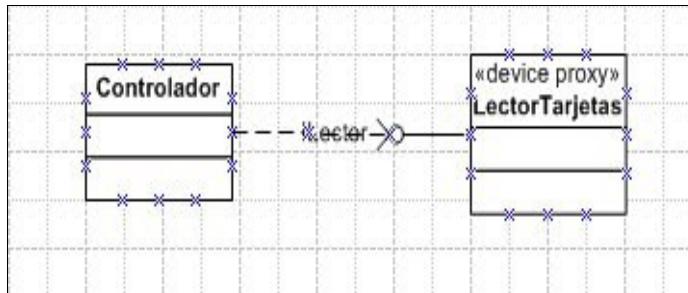


Figura 4.2 Aplicación de un estereotipo sobre una clase.

- **Pestaña “Attributes”:**

Esta pestaña permite configurar los atributos de la clase que se esté modelando, y que serán compartidos por todos los objetos de la misma.

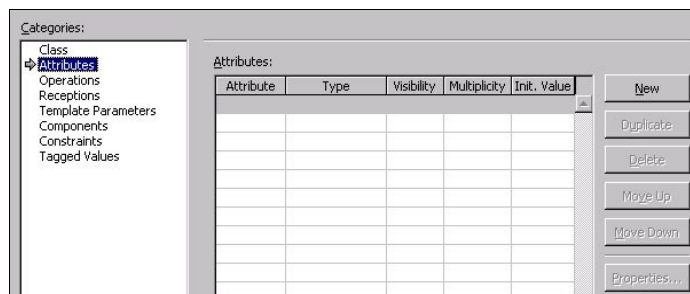


Figura 4.3. Configuración de los atributos de una clase.

Para crear un nuevo atributo dentro de una clase, debe pulsarse la opción “New” y a continuación proceder a especificar sus características:

- Asignarle un nombre significativo: rellenar el campo “Attribute”.
- Especificar su tipo: rellenar el campo “Type” en base a los siguientes criterios.
  - Tipo primitivo de datos → VB:: xxx
  - Tipo complejo de datos → Top Package:: xxx
  - Expresión de cardinalidad: seleccionar la opción “Properties...” y rellenar los campos “Prefix” y “Suffix” con la parte de la expresión que preceda y siga

respectivamente al tipo de datos que maneje esta, y que habrá de ser declarado previamente en base a los dos criterios anteriores.

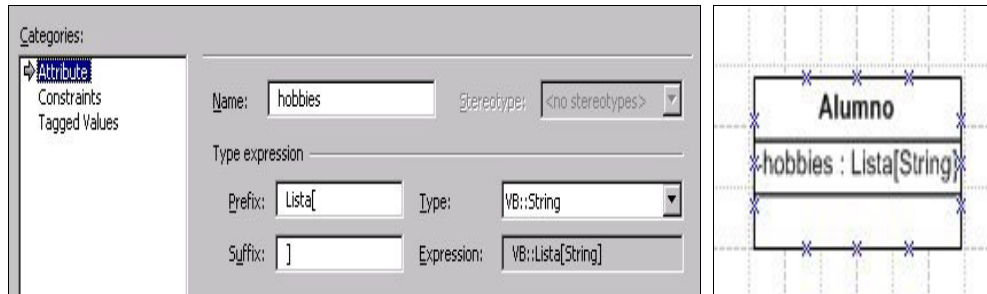
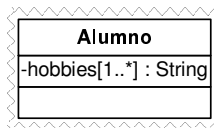


Figura 4.4 Especificación de una expresión como tipo de un atributo.

- Multiplicidad o Cardinalidad: rellenar el campo “Multiplicity” con uno de los valores posibles. Este mecanismo de expresión es más correcto que el previamente señalado y es el que realmente contempla el estándar de la OMG.



#### • Pestaña “Operations”:

Esta pestaña permite configurar las operaciones de la clase que se este modelando, y que serán compartidas por todos los objetos de la misma.

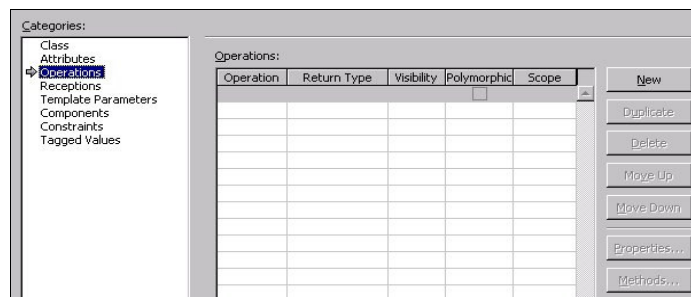


Figura 4.5 Configuración de las operaciones de una clase.

Para crear una nueva operación dentro de una clase, debe pulsarse la opción “New” y a continuación proceder a especificar sus características:

- Asignarle un nombre significativo: rellenar el campo “Operation”.
- *Especificar el tipo del valor de retorno:* rellenar el campo “Return Type” en base a los siguientes criterios.
  - Tipo primitivo de datos → VB:: xxx

- Tipo complejo de datos → Top Package:: xxx
- Estructura contenedora → una vez seleccionado el tipo primitivo o complejo, seleccionar la opción “Properties...” donde se permite la introducción de un ‘Prefix’ y un ‘Suffix’, que posibilitan la especificación de la expresión deseada (tal y como se ha explicado previamente en la sección relativa a los atributos).
  - *Especificar los parámetros de entrada:* seleccionar la opción “Properties...” y situarse sobre la pestaña “Parameters” dentro del nuevo cuadro de dialogo. A continuación, seleccionar la opción “New” para crear un nuevo parámetro y proceder a especificar sus características (“Parameter” o nombre del parámetro y “Type” o tipo del parámetro) en consonancia con el procedimiento seguido hasta ahora. Nótese que la primera línea de la rejilla de datos queda reservada para el Retorno, pudiéndose tratar el mismo como un parámetro más.

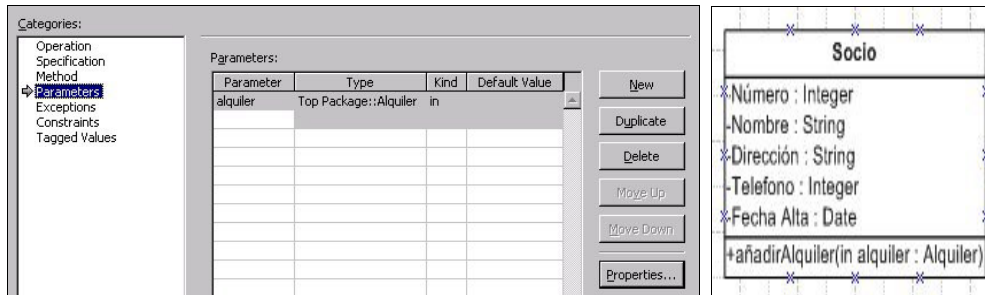


Figura 4.6 Especificación de parámetros de entrada para una operación.

Para concluir este punto advertir cómo el tratamiento de los interfaces (objetos de tipo “Interface”) resulta parejo al tratamiento especificado anteriormente para las clases, salvando las siguientes excepciones:

- Los interfaces son bloques de construcción estereotipados por defecto (bajo el estereotipo «interface») para ser diferenciados de los clasificadores anteriores.
- El cuadro de dialogo que gestiona las configuraciones de los interfaces presenta las siguientes pestañas a tener en especial consideración: Interface (equivalente a la pestaña “Class” en el contexto de los objetos de tipo “Class”) y “Operations” (con idéntica mecánica de configuración que la presentada para los objetos de tipo “Class”).

## 4.2 CARACTERÍSTICAS AVANZADAS DE LAS RELACIONES

Al modelar los elementos que constituyen el vocabulario de un sistema, también hay que modelar cómo se relacionan entre sí estos elementos.

Una relación es una conexión entre elementos. En el modelado orientado a objetos, los cuatro tipos más importantes de relaciones son las dependencias, las generalizaciones, las asociaciones y las



realizaciones. Gráficamente, una relación se dibuja como una línea, con diferentes tipos de líneas para distinguir las diferentes relaciones.

#### 4.2.1 Dependencia

Una dependencia es una relación de uso, la cual especifica que un cambio en la especificación de un elemento puede afectar a otro elemento que lo utiliza, pero no necesariamente a la inversa. Las dependencias se deben aplicar cuando se quiera representar que un elemento utiliza a otro.

Gráficamente, una dependencia se representa como una línea discontinua (objeto de tipo "Dependency"), dirigida hacia el elemento del que se depende.

#### 4.2.2 Generalización

Una generalización es una relación entre un elemento general (llamado superclase o padre) y un caso más específico de ese elemento (llamado subclase o hijo). Las generalizaciones se deben utilizar cuando se quiera mostrar relaciones padre/hijo.

Gráficamente, la generalización se representa como una línea dirigida continua (objeto de tipo "Generalization"), con una gran punta de flecha vacía, apuntado al padre.

#### 4.2.3 Asociación

Una asociación es una relación estructural que especifica que los objetos de un elemento están conectado con lo elementos de otro. Dada una asociación entre dos clases, se puede navegar desde un objeto de una clase hasta un objeto de la otra clase, y viceversa. Es legal que ambos extremos de una asociación estén conectados a la misma clase. Las asociaciones se deben utilizar cuando se quiera representar relaciones estructurales.

Gráficamente, una asociación se representa como una línea continua (objeto de tipo "Binary Association") que conecta la misma o diferentes clases.

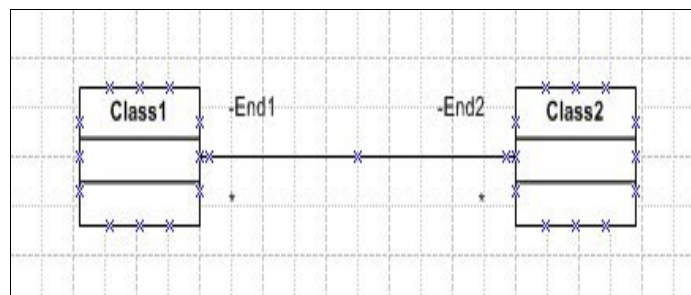


Figura 4.7 Relación de asociación entre dos clases (apariciencia inicial).

A parte de esta forma básica, existen cuatro adornos que se aplican a las asociaciones y que deben ser especificados, a los que se accede realizando "doble clic" sobre la relación de asociación que se desee modelar.

- **Nombre:** *una relación de asociación puede tener un nombre, que se utiliza para describir la naturaleza de la relación.*
- Para especificar el nombre de una asociación, resulta necesario rellenar el campo ("Name") asociado a las propiedades de la relación. Sin embargo, en Microsoft Visio el nombre recientemente asignado no será visualizado por defecto. Para que este sea mostrado, debe seleccionarse la relación de asociación, acudir al menú principal de la herramienta y escoger la opción: Shape → Actions → Shape Display Options (también accesible con el botón derecho del ratón). En el nuevo cuadro de diálogo, ha de activarse la opción "Name" para dicha relación de asociación, posibilitando *entonces la visualización de su nombre asociado.*
- **Rol:** *cuando una clase participa en una asociación, tiene un rol específico que juega en la asociación; un rol es simplemente la cara que la clase de un extremo de la asociación presenta a la clase del otro extremo.*
- Para especificar los roles de cada una de las clases que aparecen en una relación de asociación, es necesario rellenar los campos "End Name" asociados a cada uno de los extremos de la misma.

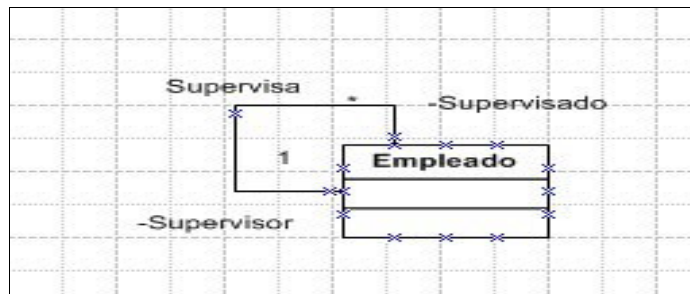


Figura 4.8 Especificación de los roles de las clases que participan en una asociación.

En caso de no desear aplicar este adorno, resulta necesario suprimir el nombre dado por defecto a los extremos de la asociación. Para ello, seleccionar la relación de asociación, acudir al menú principal de la herramienta y escoger la opción: Shape → Actions → Shape Display Options. En el nuevo cuadro de diálogo, desactivar las opciones "First end name" y "Second end name".

- Además, resulta de especial relevancia, destacar la posibilidad de poder aplicar esta nueva configuración de la relación de asociación, sobre el resto de diagramas dibujados bajo la cuadrícula actual. Para ello, activar la opción "Apply to subsequently dropped UML shapes of the same type in the current drawing window page".

Figura 4.9 Ocultación de los roles de las clases que participan en una asociación.

- **Multiplicidad:** cuando se está modelando un diagrama de clases, es importante señalar cuantos objetos pueden conectarse a través de una instancia de una asociación. Este “cuantos” se denomina multiplicidad del rol de la asociación, y se escribe como una expresión que se evalúa a un rango de valores o a un valor explícito.

Para especificar la multiplicidad de cada uno de los extremos de una asociación, es necesario rellenar el campo “Multiplicity” asociado a cada uno de estos con una de las expresiones establecidas.

- **Agregación:** una asociación normal entre dos clases representa una relación estructural entre iguales. A veces, se desea modelar una relación “todo/parte”, en la cual una clase representa al elemento total (el “todo”), que consta de elementos más pequeños (las “partes”). Este tipo de relación se denomina *agregación*, la cual representa una relación del tipo “tiene-un”, o sea, un objeto del todo tiene objetos de la parte.

En realidad, la agregación es sólo un tipo especial de asociación y se especifica añadiendo a una asociación normal un rombo vacío en la parte del todo, que se consigue al seleccionar en el campo “Aggregation” de este extremo la opción de “shared”.

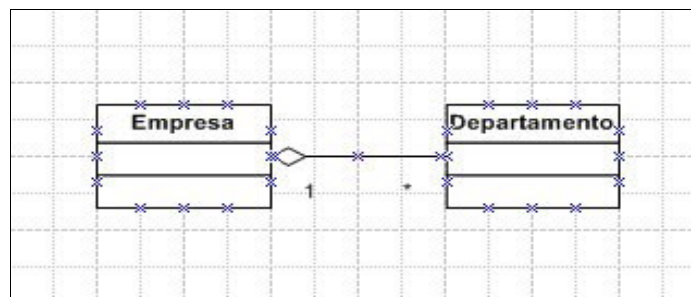


Figura 4.10 Agregación.

Hasta el momento, han sido definidos cuatro adornos básicos que se aplican a las asociaciones: nombre, rol en cada extremo de la asociación, multiplicidad en cada extremo y agregación. Sin embargo, para usos avanzados hay otras propiedades que permiten modelar detalles sutiles, como la calificación, algunas variantes de la agregación, las clases asociación y las restricciones.

- **Calificación:** un calificador es un atributo de una asociación cuyos valores particionan el conjunto de objetos relacionados con un objeto a través de una asociación. El objeto origen,

junto con los valores de los atributos del calificador, devuelven un objeto destino (si la multiplicidad del destino es como máximo uno) o un conjunto de objetos (si la multiplicidad del destino es muchos).

Para especificar un calificador sobre una asociación, es necesario hacer “doble clic” sobre ella para acceder a las propiedades de la relación, seleccionar el extremo sobre el que se quiere asociar el calificador, pulsar sobre la opción “Properties...” y situarse sobre la pestaña “Qualifier Attributes” dentro del nuevo cuadro de diálogo. A continuación, seleccionar la opción “New” para crear un nuevo atributo del calificador y proceder a especificar sus características (“Attribute” o nombre del atributo y “Type” o tipo del atributo) en consonancia con el procedimiento seguido hasta ahora.

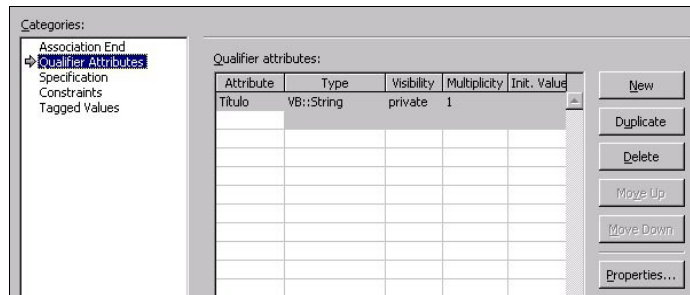


Figura 4.11 Especificación de los atributos de un calificador.

- Sin embargo, en Microsoft Visio el calificador recientemente especificado no será visualizado por defecto. Para que este sea mostrado, debe seleccionarse la relación de asociación, acudir al menú principal de la herramienta y escoger la opción: *Shape → Actions → Shape Display Options*. En el nuevo cuadro de diálogo, ha de activarse la opción “Properties” para dicha relación de asociación, posibilitando entonces la visualización de su calificador asociado.

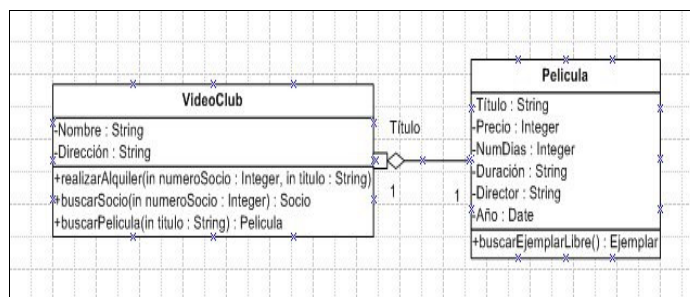


Figura 4.12 Calificación.

- **Composición:** la composición es una forma de agregación, con una fuerte relación de pertenencia y vidas coincidentes de la parte con el todo. Esto significa que, en una composición, un objeto puede formar parte sólo y exclusivamente de un todo.

La composición es realmente un tipo especial de asociación, y se especifica adornando una asociación simple con un rombo relleno en el extremo del todo. Para conseguir esta

representación, hacer “doble clic” sobre la asociación para acceder a las propiedades de la relación, y seleccionar en el campo “Aggregation” de este extremo la opción de “composite”.

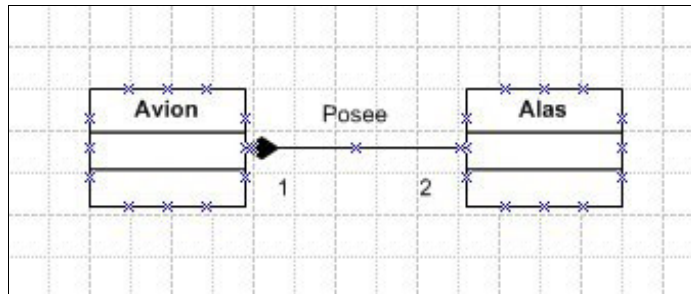


Figura 4.13 Composición.

- **Clases asociación:** una clase asociación es un elemento de modelado con propiedades tanto de asociación como de clase. Una clase asociación puede verse como una asociación que también tiene propiedades de clase, o una clase que también tiene propiedades de asociación.

Para representar una clase asociación, es necesario utilizar un objeto de tipo “Association Class”, el cual en primera instancia ha de ser conectado con las dos clases que vayan a conformar la relación de asociación. A partir de aquí, la clase asociación debe ser configurada dotándola de un nombre, atributos... procediendo de igual manera que la detallada para el caso de las clases comunes (objetos de tipo “Class”).

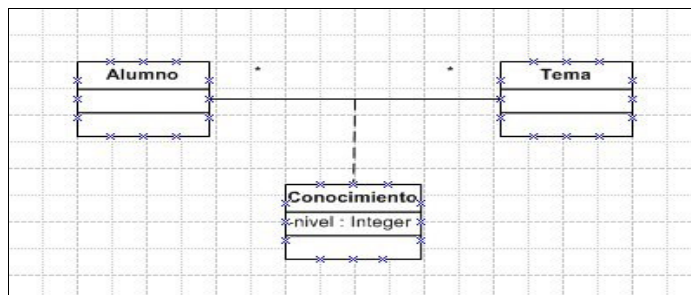


Figura 4.14 Clases asociación.

- **Restricciones:** una restricción especifica condiciones que deben cumplirse para que el modelo esté bien formado.

Una restricción se representa como una cadena de caracteres entre llaves junto al elemento asociado. Para conseguir esta ilustración, ha de arrastrarse un objeto de tipo “Constraint” sobre la cuadrícula y a continuación proceder a configurar el mismo. En primer lugar, debe definirse la restricción en sí, para ello hacer “doble clic” sobre la misma y rellenar el campo “Body” con su especificación (utilizar la tecla de retroceso para colocar el cursor al comienzo del cuerpo), indicando que esta será realizada utilizando como lenguaje “OCL”.

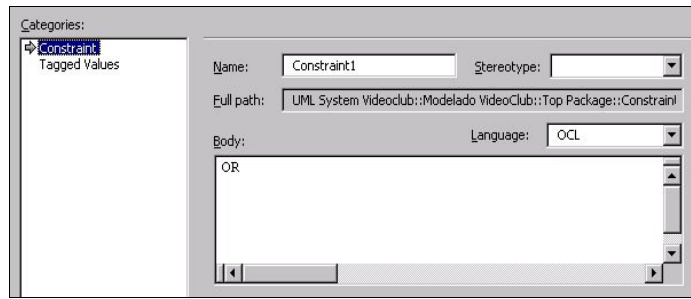


Figura 4.15 Especificación de una restricción.

- A continuación, resulta necesario cambiar el aspecto visual de la restricción dado que, Microsoft Visio representa por defecto las mismas utilizando el mismo marco que el reservado para las notas. Para que este no sea mostrado, debe seleccionarse la restricción, acudir al menú principal de la herramienta y escoger la opción: *Shape → Actions → Shape Display Options*. En el nuevo cuadro de dialogo ha de desactivarse la opción “Constraint Shape”, impidiendo entonces la visualización del marco asociado por defecto.
- Destacar de nuevo en este punto, la posibilidad de poder aplicar esta nueva configuración de restricción, sobre el resto de diagramas dibujados bajo la cuadrícula actual. Para ello, activar la opción “Apply to subsequently dropped UML shapes of the same type in the current drawing window page”.

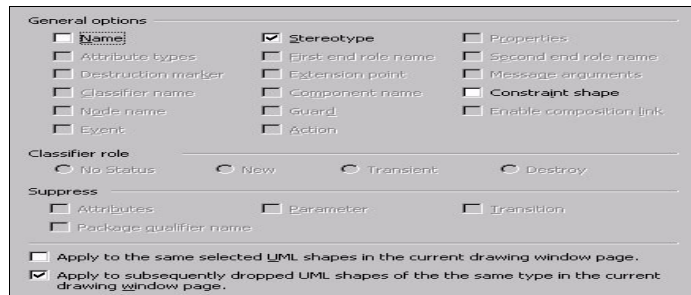


Figura 4.16 Ocultación del marco establecido por defecto para las restricciones.

Por último y en caso de especificar una restricción respecto a relaciones, advertir cómo esta debe ser conectada directamente con las asociaciones para las que debe considerarse, seleccionando para ello ‘2-element Constraint’, que deja el texto de la restricción libre o el ‘OR-Constraint’ que lleva como restricción predefinida una OR.

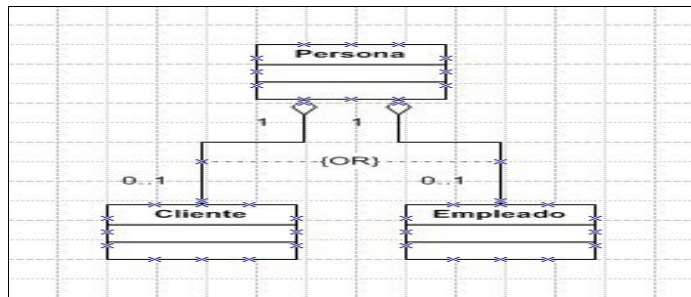


Figura 4.17 Restricciones.

Para concluir y dentro del ámbito de las restricciones debe tenerse en cuenta que, cuando se crea un modelo con UML, se trabaja dentro de las reglas que impone UML. Sin embargo, si se necesita expresar nueva semántica sobre la cual UML no dice nada o si se necesitan modificar las reglas de UML, entonces hay que escribir una restricción.

Por ejemplo, la siguiente figura muestra entre otras cosas que, cada “Empleado” puede trabajar en cero o mas “Compañías” y que cada “Compañía” puede tener cero o mas “Empleados”. El diagrama además indica que cada “Compañía” debe tener exactamente un “Empleado” como gerente y cada “Empleado” puede ser el gerente de cero o mas “Compañías”.

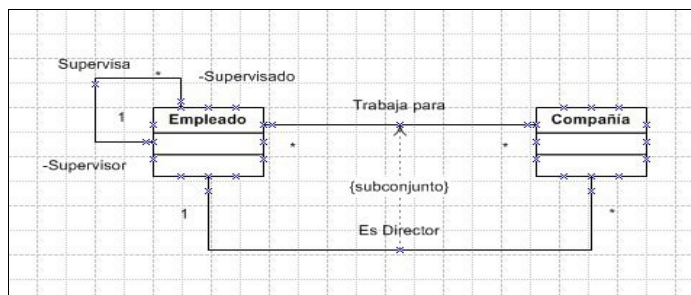


Figura 4.18 Modelado de nueva semántica.

Toda esta semántica puede expresarse con UML básico. Sin embargo, afirmar que un gerente también debe trabajar para una compañía es algo que afecta a varias asociaciones y no puede expresarse con UML básico. Para especificar este invariante, hay que escribir una restricción que muestre al gerente como un subconjunto de los trabajadores de la compañía, conectando las dos asociaciones mediante una restricción, tal y como puede apreciarse en al Figura 4.18.

## Realización

Una realización es una relación semántica entre clasificadores, en la cual un clasificador especifica un contrato que otro clasificador garantiza que cumplirá.

Las realizaciones se deben aplicar cuando se quiera especificar la relación entre una interfaz y la clase o el componente que proporciona una operación o servicio para ella.

Gráficamente, una realización se puede representar de dos maneras: de la forma canónica (con el estereotipo «interface» y la línea dirigida discontinua con la punta de flecha vacía) y de la forma abreviada (con la notación en forma de piruleta, figura 4.2).

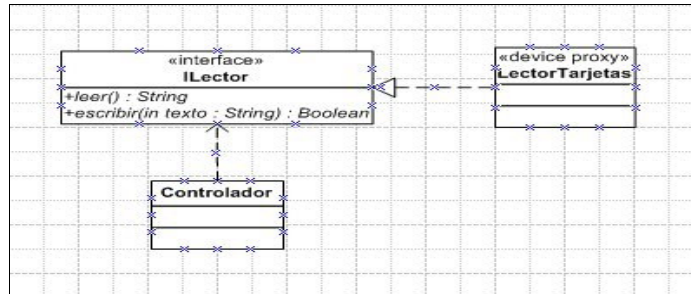


Figura 4.19 Realización de una interfaz (forma canónica).

### 4.3 REPRESENTACIÓN DE REFERENCIA

Este diagrama muestra la apariencia que han de poseer los diagramas de clases desarrollados.

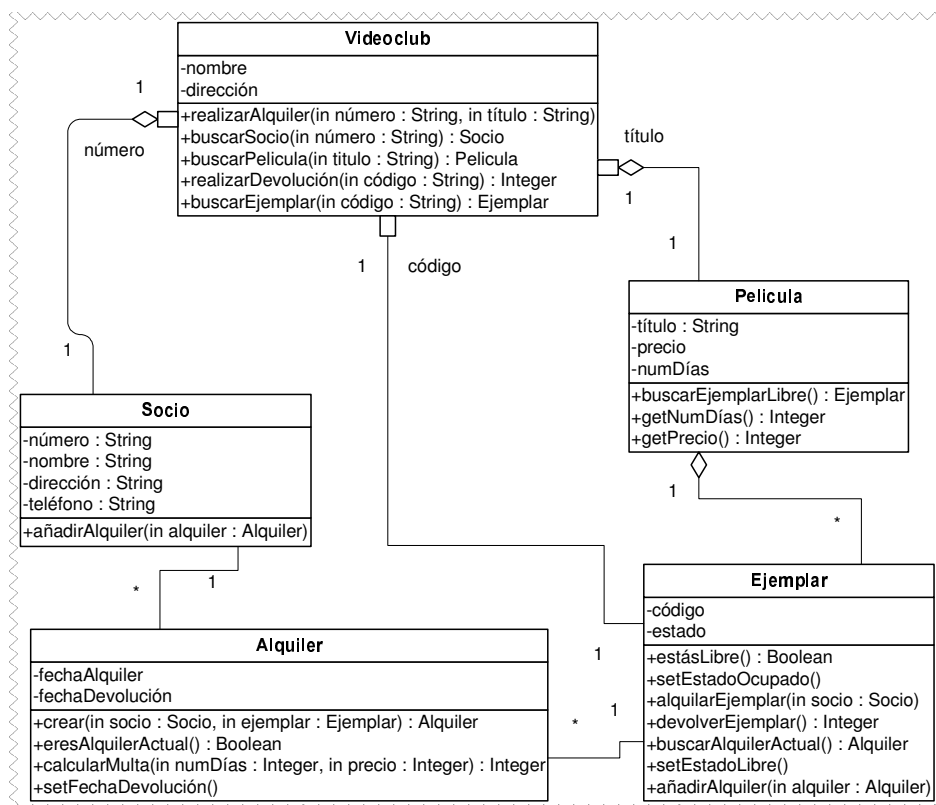


Figura 4.20 Representación de referencia - Diagrama de Clases.