**Analisis**

**1 Casos de Uso**

Si bien no es una herramienta creada por UML, la solución que éste propone son los **casos de uso**. Los casos de uso son una forma de descomponer la funcionalidad del sistema en partes más pequeñas, cada una centrada en un uso único del sistema. Existen dos alternativas para especificar los casos de uso de un sistema, una gráfica y otra narrativa, las cuales se complementan entre sí.

**Notación: Caso de Uso – Forma Gráfica**

Los casos de uso se pueden representar en forma gráfica usando la notación descrita a continuación. Se representa mediante una elipse, y denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos representa la totalidad de operaciones desarrolladas por el sistema. Un caso de uso va acompañado de un nombre significativo. Ejemplos:


Figura 3.9: Ejemplo de Casos de Uso

**Notación: Actor**

Se representa mediante un símbolo que personifica una persona, y va acompañado de un nombre significativo.


Figura 3.10: Ejemplo de Actores

**Notación: Relaciones en un Diagrama de Casos de Uso**

- **Comunica** (communicates): relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado.

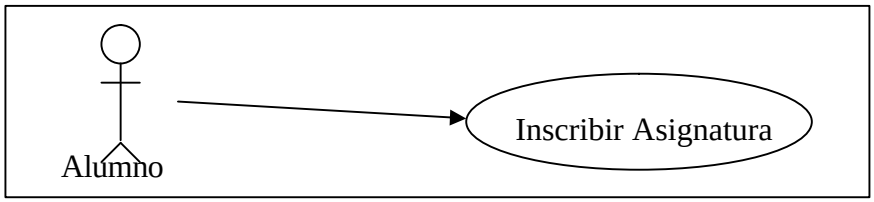



Figura 3.11: Ejemplo de relación *communicates.*

- **Usa** (include): relación entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro.

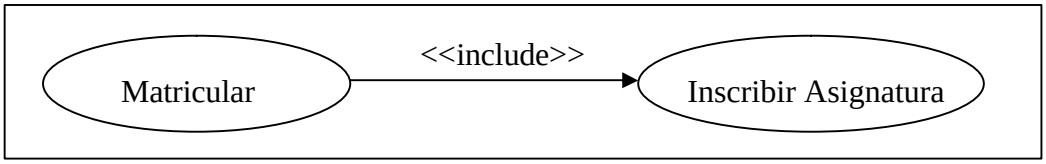



Figura 3.12: Ejemplo de relación *include*.

- **Extiende** (extend): relación entre dos casos de uso que se aplica cuando un caso de uso es una especialización de otro. Por ejemplo, podría tenerse un caso de uso que extienda la forma de pago a un cajero, la que podría ser al contado o con cheque.

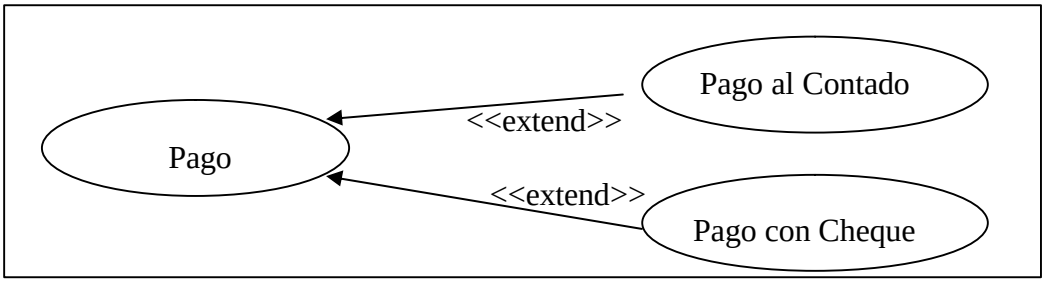



Figura 3.13: Ejemplo de relación *extend*.

**Notación: Caso de Uso – Forma Narrativa**

El caso de uso se puede ver como un documento narrativo que describe la secuencia de eventos de un actor que utiliza un sistema para completar un proceso. Los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican e incluyen los requerimientos en las historias que narran. El siguiente ejemplo corresponde a un caso de uso de **alto nivel**, que describe clara y concisamente el proceso de comprar artículos en una tienda cuando se emplea una caja una caja registradora en el punto de venta.

| **Caso de Uso** | **Comprar Productos** |
|---|---|
| **Actores** | Cliente (iniciador), Cajero |
| **Tipo** | Principal |
| **Descripción** | Un Cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el Cliente se marcha con los productos. |

La explicación del formato es:

- Caso de uso: nombre del caso de uso.
- Actores: lista de actores en la que se indica quién inicia el caso de uso.
- Tipo: que puede ser Primario, Secundario, Opcional, entre otros.
- Descripción: repetición del caso de uso de alto nivel o alguna síntesis similar.

UML no especifica un formato rígido; puede modificarse para atender las necesidades y ajustarse al espíritu de la documentación: ante todo, una combinación clara.

Conviene comenzar con los casos de uso de alto nivel para lograr rápidamente entender los principales procesos globales. Posteriormente, se pasa al llamado caso de uso **expandido** para mostrar más detalles, para lograr así un conocimiento más profundo de los procesos y de los requerimientos. El caso de uso expandido del caso de uso de alto nivel Comprar Productos se entrega en la página que sigue.

La estructura del formato expandido agrega a la del alto nivel lo siguiente:

- Propósito: intención del caso de uso.
- Referencias Cruzadas: casos de uso y funciones relacionadas con el sistema.
- Curso Normal de los Eventos: es la parte medular del formato expandido; describe los detalles de la conversión interactiva entre los actores y el sistema. Un aspecto esencial de la sección es explicar la secuencia más común de eventos: la historia normal de las actividades y el término exitoso de un proceso. No incluye situaciones alternativas.
- Cursos Alternativos: describe importantes opciones o excepciones que pueden presentarse en relación al curso normal. Si son éstas son complejas, se pueden expandir y convertir en nuevos casos de uso.

| **Caso de Uso** | **Comprar Productos** |
|---|---|
| **Actores** | Cliente (iniciador), Cajero |
| **Propósito** | Capturar una venta y su pago |
| **Tipo** | Principal y esencial |
| **Descripción** | Un Cliente llega a una caja con productos que desea comprar. El Cajero registra los productos y obtiene el pago. Al terminar la transacción, el Cliente se marcha con los productos. |
| **Referencias Cruzadas** | Casos de Uso: el Cajero debe haber terminado el caso de uso llamado *Registrar* |

**Curso Normal de los Eventos**

| **Acción de los Actores** | **Respuesta del Sistema** |
|---|---|
| 1.- Este caso de uso comienza cuando un Cliente llega a la caja con productos que desea comprar. | |
| 2.- El Cajero registra los productos. Si hay más de un producto, también puede introducir la cantidad. | 3.- Determina el precio del producto y agrega la información sobre él a la actual transacción de venta. |
| 4.- Al terminar el registro de los productos, el Cajero indica al sistema que terminó dicho proceso. | 5.- Calcula y presenta el total de la venta. |
| 6.- El Cajero le indica el total al Cliente. | |
| 7.- El Cliente escoge la forma de pago:<br>a) Si paga en efectivo, ver la sección *Pago en Efectivo*<br>b) Si paga con tarjeta con crédito, ver la sección *Pago con Tarjeta de Crédito*<br>c) Si paga con cheque, ver la sección *Pago con Cheque* | 8.- Registra la venta terminada. |
| | 9.- Actualiza los niveles de inventario. |
| | 10.- Genera un recibo. |
| 11.- El Cajero entrega el recibo al cliente. | |
| 12.- El Cliente se marcha con los productos comprados. | |

**Sección: Pago en Efectivo**

**Curso Normal de los Eventos**

| **Acción de los Actores** | **Respuesta del Sistema** |
|---|---|
| 1.- El Cliente da un pago en efectivo, posiblemente mayor que el total de la venta. | |
| 2.- El Cajero registra el efectivo recibido. | 3.- Entrega la diferencia al Cliente. |
| 4.- El cajero guarda el efectivo recibido y saca la diferencia. Luego, le entrega el vuelto al Cliente. | |

**Cursos Alternativos – Sección Pago en Efectivo**

- Línea 1: el Cliente no tiene suficiente efectivo. Puede cancelar o iniciar otro método de pago.
- Línea 4: la caja no tiene suficiente efectivo para pagar la diferencia. El Cajero pide más efectivo al supervisor o le pide al Cliente otro billete de menor valor u otra forma de pago.

**Diagramas de Casos de Uso – Ejemplo en Desarrollo**

En la figura 3.14 se muestra el diagrama de caso de uso de alto nivel del sistema; el rectángulo grande el límite del sistema; el nombre del sistema aparece sobre el rectángulo. El ícono de persona representa un actor que es un usuario humano. El ícono rectangular del sistema de respaldo representa un actor que es otro sistema. El nombre del actor Sistema de Respaldo está precedido por un string <<*actor*>>, el cual es llamado un *estereotipo* e indica que el Sistema de Respaldo es un actor. Este estereotipo y el ícono de persona son equivalentes; sin embargo, comúnmente, el primero se usa para representar sistemas y el segundo, usuarios humanos. Las elipses simbolizan casos de uso. Las líneas que conectan actores y casos de uso indican que los actores usan o participan en la funcionalidad provista por los casos de uso. Así, esta línea es llamada relación *comunica*.

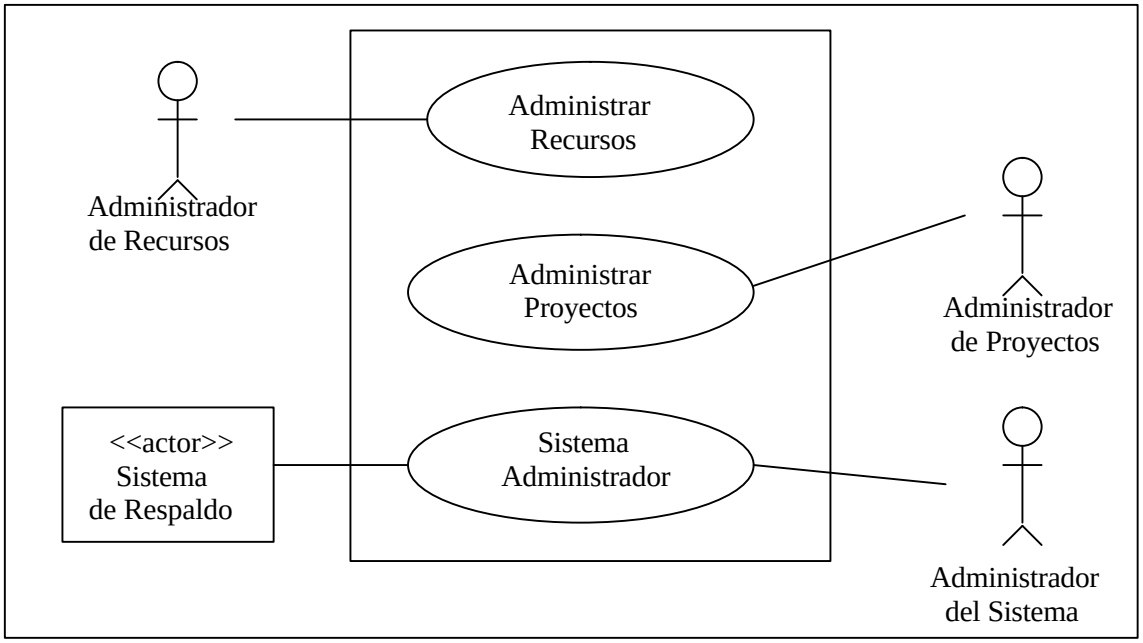



Figura 3.14: Diagrama de Caso de Uso a Alto Nivel

Cada caso de uso de se puede detallar en nuevos diagramas del mismo tipo, para mostrar mayor profundidad sobre el tema. A continuación se muestra los diagramas de casos de uso de dos de los tres subsistemas de la figura 3.14; en el primer caso (figura 3.15), el diagrama de caso de uso Administrar Recursos muestra las diversas funcionalidades que éste provee; así, los administradores de recursos pueden agregar, eliminar o actualizar información de habilidades. Dado que una habilidad debe ser encontrada en la base de datos de sistema antes de que pueda ser eliminada o actualizada, un caso de uso Actualiza Habilidad es usado. Las flechas desde los casos de uso Eliminar Habilidad y Actualizar Habilidad al caso de uso Encontrar son rotuladas con el estereotipo *include* para indicar que este último es llamado o incluido en los dos primeros casos de uso anteriores. Esta flecha es llamada una relación *include*. También se puede ver que los administradores de recursos pueden agregar, eliminar y actualizar información de recursos, y en particular, en el último caso pueden asignar o desasignar una habilidad a un recurso, lo que se puede hacer usando las operaciones que aparecen al final de la figura siguiente. Notar que las flechas

llevan el rótulo *extend*, para indicar que los casos de uso son opcionales en su utilización (en este caso, se usa uno u otro, pero no ambas a la vez, de ahí la opcionalidad).

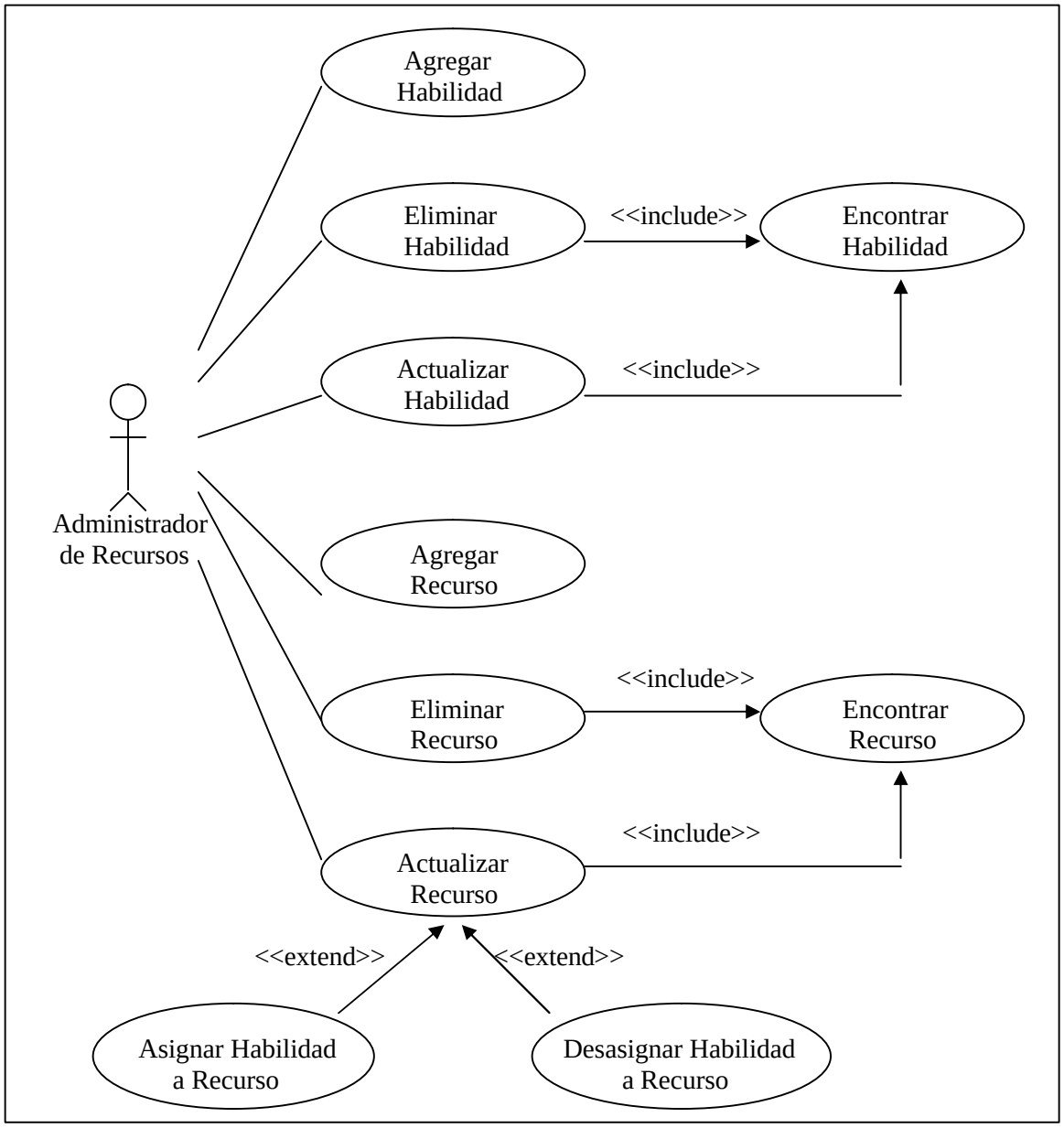



Figura 3.15: Diagrama de Caso de Uso Administrar Recursos

La figura 3.16 contiene el diagrama de caso de uso del Sistema de Administración.


Figura 3.16: Diagrama de caso de uso del Sistema de Administración.

## 2 Modelo Conceptual

Los actores y casos de uso son definidos para capturar los requerimientos de un sistema. La idea del análisis es traducir dichos requerimientos en un diagrama de clases inicial, el cual describe los principales tipos de objetos que existen en el sistema.

Un modelo de clases es un modelo del dominio del problema, que captura todos los requerimientos del sistema, este conocimiento es contenido en las clases, atributos y operaciones de las mismas, y en las asociaciones entre las mismas clases. En otras palabras, describen la estructura estática de un sistema, o cómo está estructurado más que cómo se comporta, conteniendo:

- Clases: representan entidades con características comunes. Estas características incluyen atributos, operaciones y asociaciones.
- Asociaciones: que simbolizan las relaciones entre dos o más clases, donde las asociaciones tienen características comunes. Estas características incluyen atributos y operaciones.

En el caso particular de un modelo de clases para la etapa de análisis, el diagrama correspondiente excluye las operaciones y otros elementos, hasta la etapa de diseño.

No siempre resulta fácil decidir cuáles clases son requeridas en el modelo. Así, es recomendable utilizar cierto número de heurísticas que pueden guiar el trabajo de definir las clases necesarias y eliminar las innecesarias.

Cuando se define una clase, se debe identificar sus operaciones (comportamiento) y los atributos (datos). Algunos atributos son más simples que otros; por ejemplo, el saldo de una cuenta bancaria es un valor simple, mientras que la lista de transacción de la misma cuenta es tan compleja que seguramente deberá ser representada por una segunda clase.

Los casos de uso son un punto de partida ideal para la identificación de las clases. La descripción del caso de uso, la del curso normal y la de los alternativos se pueden utilizar para identificar cuáles son las “cosas” y frases relevantes al quehacer. Así, a modo de ejemplo, si el caso de uso para la compra de productos tiene la siguiente definición, probables clases pueden ser las “cosas” que aparecen subrayadas en el texto:

*Un cliente llega a la caja con un canasto de productos. El cajero chequea cada producto. El precio de cada ítem es determinado por el sistema, y el precio de cada transacción es presentado al cliente y registrado. El total es señalado al cliente, quien realiza el pago, en efectivo o cheque.*

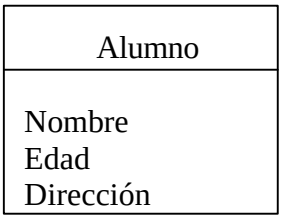
Algunos de los nombres subrayados están duplicados y son sinónimos, por lo que el paso siguiente consiste en revisar la lista de nombres y descartar los superfluos. Para el ejemplo anterior, *producto* e *ítem* son identifican el mismo concepto, y dado que *producto* es el mejor término, *ítem* es dejado de lado. Por otra parte, hay términos que están fuera del ámbito del sistema o no son relevantes, por lo que también deben de quedar fuera. Ejemplo: *canasto* no es de importancia, por lo que se debe dejar de lado.

Tras determinar los conceptos fundamentales a considerar, se debe proceder a construir el modelo conceptual del sistema. Para esto, UML provee un gran número de símbolos destinados a apoyar la labor a realizar, parte de la cual se presenta a continuación, señalando eso si que se explica

sólo lo utilizable en un modelo conceptual, postergando elementos adicionales para cuando se llegue a la etapa de diseño.

**Notación: Clase**

Las clases se representan por rectángulos con dos divisiones internas. Cada clase describe un conjunto de objetos con características y comportamiento similar. En las tres partes anteriores se ubican el nombre de la clase y sus atributos, respectivamente. Un ejemplo de clase es:




**Notación: Atributo**

Generalmente son de tipos simples, ya que los atributos de tipos compuestos se representan mediante asociaciones de composición con otras clases. La sintaxis de un atributo es:

*nombre = valor_inicial {propiedad}*

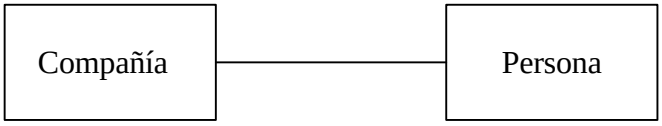
donde:

- nombre: si comienza con minúscula se considera un atributo de objeto.

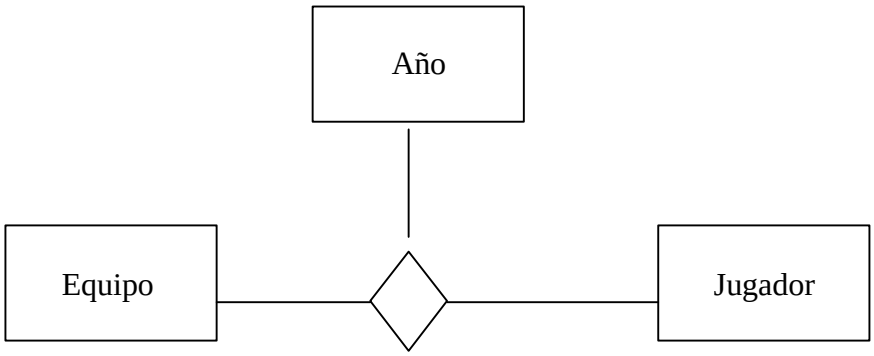
**Notación: Asociación (rol, multiplicidad, cualificador)**

Una asociación, en general, es una línea que une dos o más símbolos. Pueden tener varios tipos de adornos, que definen su semántica y características. Los tipos de asociaciones entre clases presentes en un diagrama de clases son:

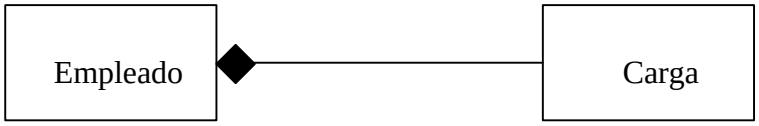
- **Asociación binaria**: se identifica como una línea sólida que une dos clases. Representa una relación de algún tipo entre las dos clases, que no exige dependencia existencial ni encapsulamiento. Ejemplo:



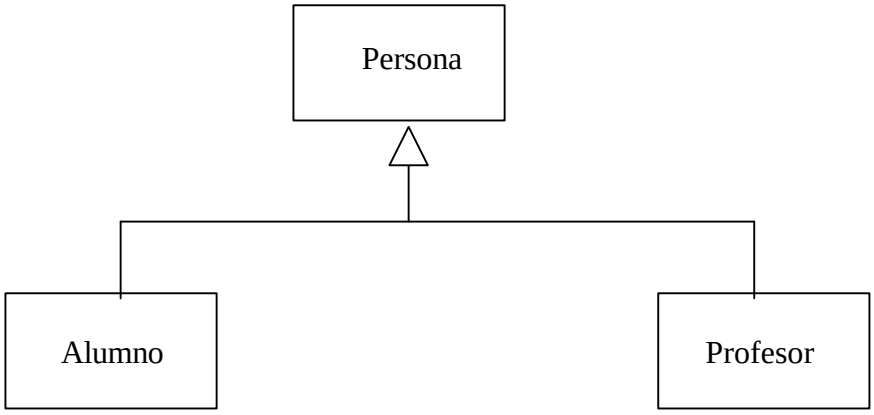
- **Asociación n-aria**: es una forma de expresar una relación entre tres o más clases. Se representa como un diamante del cual salen líneas de asociación a las clases. Ejemplo:




- **Agregación**: al comienzo de la línea que simboliza una asociación se puede ubicar un rombo que simbolice una asociación de composición o agregación compuesta (rombo ennegrecido), lo que ocurre cuando la entidad determina la existencia de la otra, o bien el concepto de agregación compartida (rombo blanco), si los objetos pueden existir más allá de la asociación. Ejemplo:

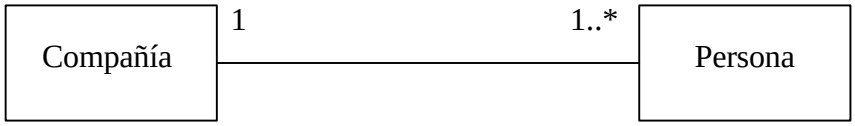



- **Generalización**: la relación de generalización denota una relación de herencia entre clases. Se representa dibujando un triángulo sin rellenar en el lado de la superclase. La subclase hereda todos los atributos y mensajes descritos en la superclase. Ejemplo:

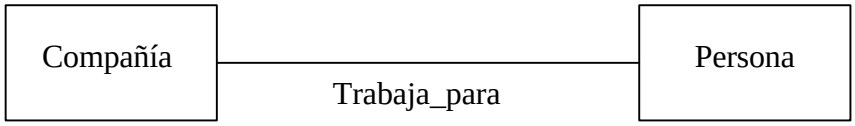



Cada asociación puede presentar algunos elementos adicionales que dan detalle a la relación, como son:

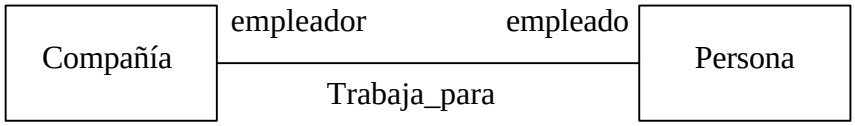
- **Multiplicidad**: describe la cardinalidad de la asociación. Cada asociación tiene, en ambos sentidos, una multiplicidad: **1** indica una ocurrencia; ***** indica 0 o más ocurrencias; **1..*** señala una o más; **1..40** indica de 1 a 40 ocurrencias; 3,5,8 indica que hay 3 ó 5 u 8 ocurrencias . Ejemplo:



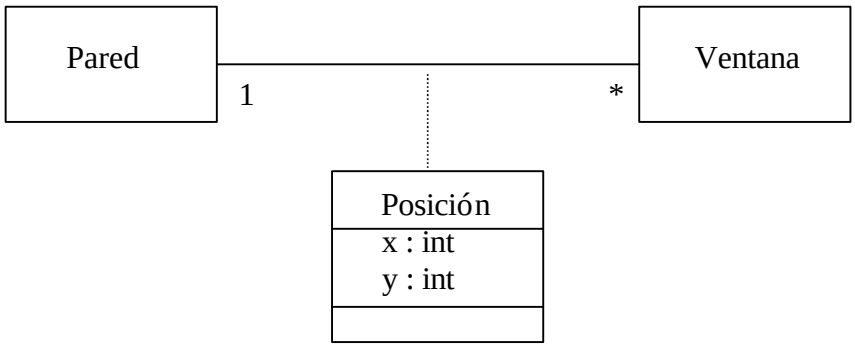

- **Nombre**: describe el significado de la relación; se agrega al nombre una punta de flecha que indica en qué sentido se debe leer la frase para interpretarla adecuadamente. Ejemplo:



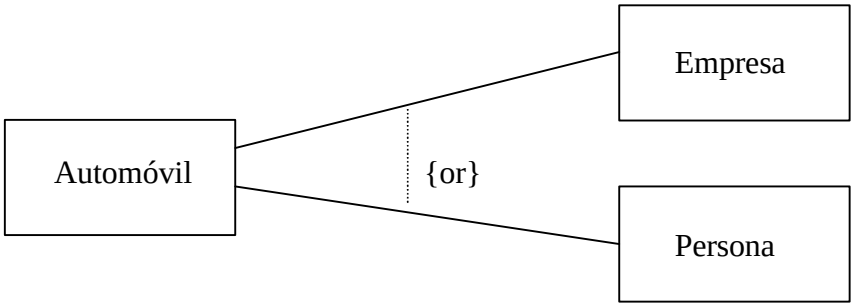

- **Rol**: identificado como un nombre al final de la línea, describe la semántica de la relación en el sentido indicado. Ejemplo:




- **Atributos**: como consecuencia de una relación puede necesitarse almacenar cierta información de detalle. Ésta se denota como una clase relacionada por una línea punteada a la relación. Ejemplo: considerar una relación entre *Muro* y *Ventana*, la cual tiene como detalle un objeto de la clase *Posición*; cabe notar que este objeto no podría tomarse como atributo de ninguna de las clases anteriores, ya que el contexto de su existencia está dado precisamente por la relación entre las dos clases.



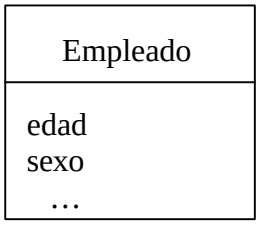
En forma adicional, en algunas ocasiones es necesario describir que una clase está relacionada con un objeto de una clase o de otra, pero no de ambas. Esto se denota por medio de una relación *or* exclusiva. Su representación es una línea punteada que une dos asociaciones, junto con la aclaración (por medio de una propiedad) del tipo de asociación. Ejemplo: si un automóvil tiene un único dueño, que puede ser una persona o una empresa, entonces…




**Notación: Restricciones**

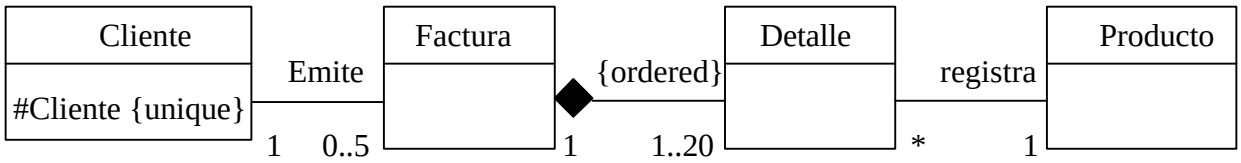
La asociación exclusiva del último ejemplo, expresada por el elemento *{or}*, es una de las tantas restricciones que se pueden incorporar en el modelo. El número y tipo de restricciones no es rígido o limitado, y puede tener tantas posibilidades como lo requiera el sistema en estudio.

Ciertos autores extienden UML a través del llamado Lenguaje de Restricciones sobre Objetos (OCL, Object Constraint Language), usado para expresar condiciones atachados a elementos de modelos. Cuando son incorporados a una clase, estas expresiones especifican reglas a los cuales los objetos de la clase deben adherirse. En algunos casos, cuando la expresión es muy compleja, se puede usar en su lugar lenguaje natural.

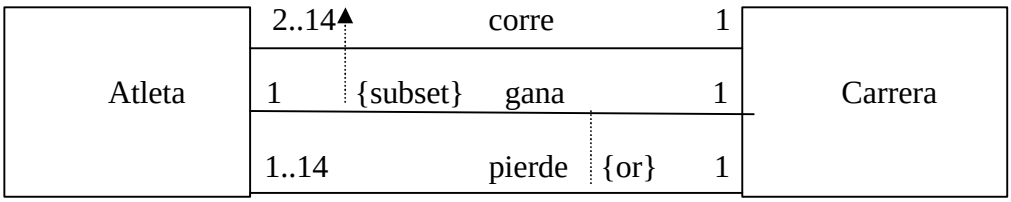



{sexo = “hombre” ⇒ edad in [18,65]}
{sexo = “mujer” ⇒ edad in [18,60]}

En general, se puede plantear situaciones como:




para señalar que el número de cliente es único y que los detalles de una factura deben respetar un orden entre ellos, y




para señalar que un atleta gana o pierde una carrera pero no las dos posibilidades a la vez, y que quien gana debe ser alguien que corre la carrera.

**Notación: Observación (comentario, nota)**

Es un comentario dentro de un diagrama, es decir aclaraciones a éste. Puede estar relacionado con uno o más elementos en el diagrama mediante líneas punteadas. Se representa mediante un rectángulo con su borde superior derecho doblado.

## Modelo Conceptual – Ejemplo en Desarrollo

Al construir el modelo conceptual para el sistema de Administración de Proyectos, se tiene:

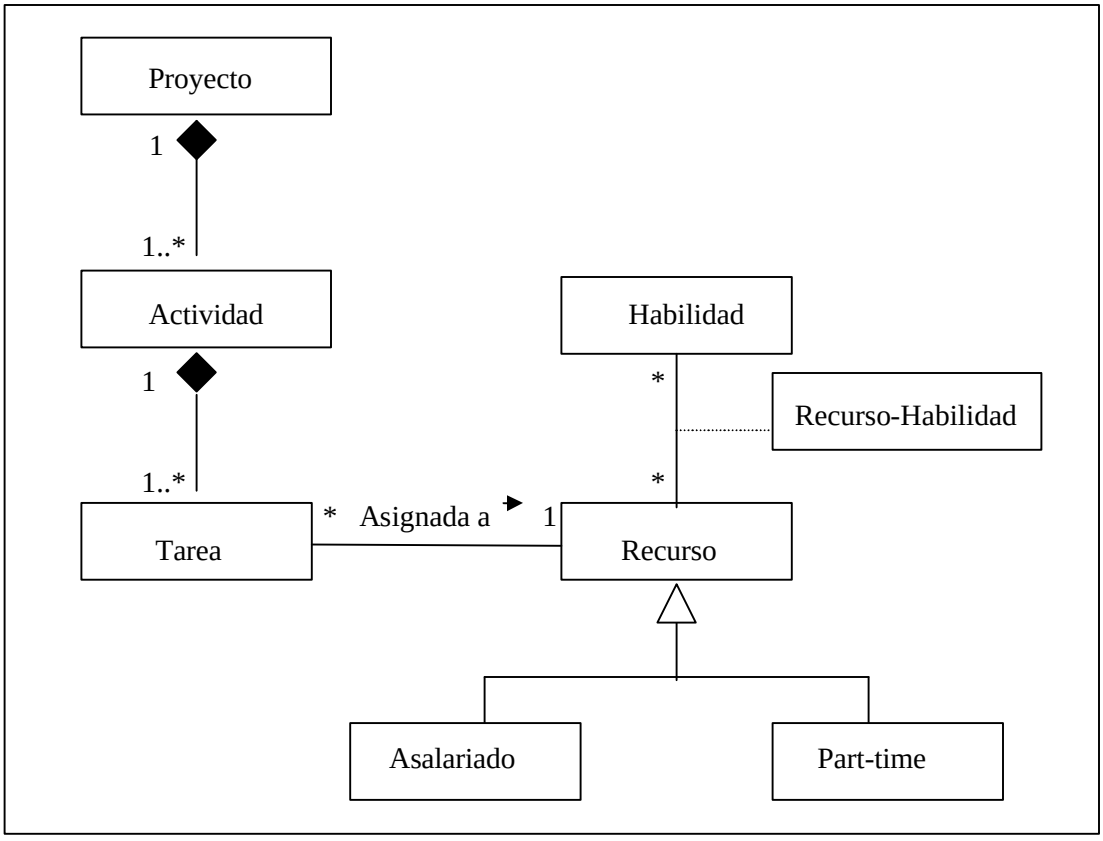



Figura 3.17: Ejemplo de diagrama conceptual de clases.

En este último caso, la flecha que relaciona las clases *asalariado* y *part-time* con *recurso*, representa una asociación de generalización (o bien, relación “es un”). Sólo por simplicidad, se han omitido los atributos de cada clase. Sin embargo, normalmente éstos se consideran, como por ejemplo:

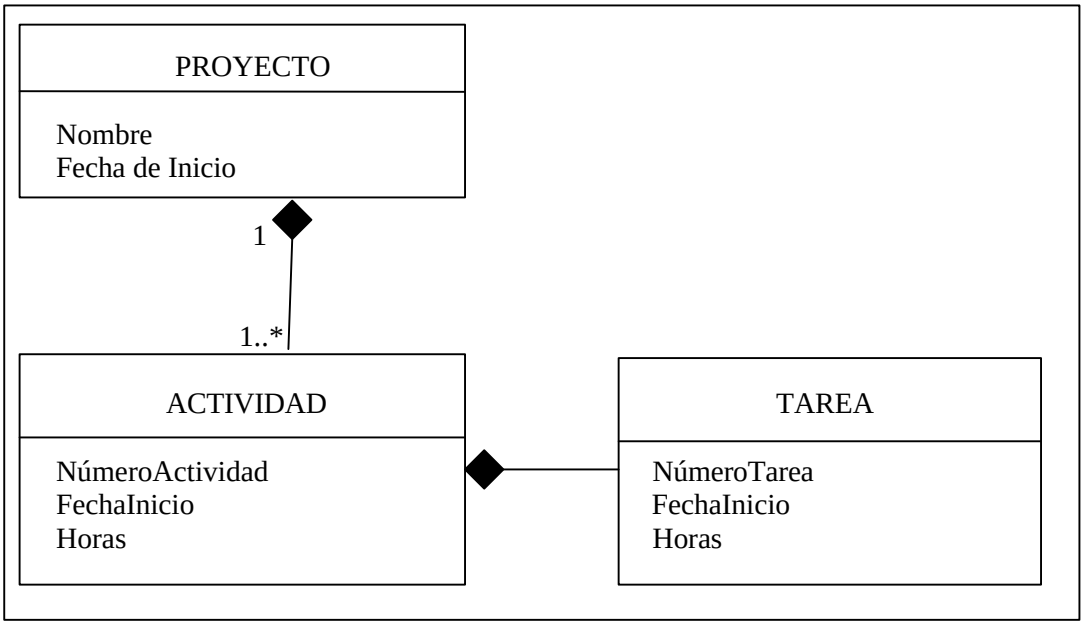



Figura 3.18: Diagrama de Clases de Alto Nivel para el SubSistema de Administración de Recursos

## 3 Escenarios e Interacciones entre los Objetos

Antes de iniciar el diseño lógico de cómo funcionará una aplicación de software, es necesario investigar y definir cómo su comportamiento como una “caja negra”. La identificación del comportamiento de un sistema, a través de los llamados escenarios y las interacciones entre los objetos, es una descripción de lo que el sistema hace, sin explicar la manera en que lo hace. Una parte de la descripción es son los diagramas de secuencia del sistema.

Los casos de uso indican cómo los actores interactúan con el sistema, indicando que es lo que en realidad se desea desarrollar. Durante la interacción, un actor genera eventos dirigidos a un sistema, solicitando alguna operación a cambio. Por ejemplo, cuando un cajero ingresa el código de barras de un producto, está diciendo al sistema que registre la compra. Con el evento se inicia una operación del sistema.

Los casos de uso dan descripciones de alto nivel; el modelo conceptual provee información preliminar de los tipos estáticos del sistema. Sin embargo, se requiere desarrollar un entendimiento más detallado del sistema, instanciando casos de uso usando las clases ya identificadas, e identificando asociaciones, operaciones y nuevas clases. Esto es resuelto por los escenarios, cada uno de los cuales provee una secuencia específica de interacciones entre los actores y las entidades en el sistema, basándose en la secuencia genérica de eventos que define el caso de uso correspondiente.

Así, un escenario es un camino a través de un caso de uso, una especie de ejemplo en acción de un caso de uso; es uno de los muchos caminos posibles a través del caso de uso correspondiente

(un caso de uso tiene muchos escenarios). Un escenario es una elaboración textual de un caso de uso, que puede ser representada gráficamente mediante los diagramas que provee UML, los que dan una descripción gráfica de las interacciones del actor y de las operaciones que dan origen.

Estos diagramas describen interacciones entre clases. Estas interacciones son modeladas como intercambios de mensajes. Estos diagramas se centran sobre clases y los mensajes que ellas se intercambian para cumplir algún comportamiento deseado. Los diagramas de secuencia son un tipo de diagrama de interacción.

Los diagramas de interacción contienen los siguientes elementos:

- Roles de Clases: representan roles que los objetos pueden jugar en la interacción.
- Líneas de vida: simbolizan la existencia de un objeto sobre un periodo de tiempo.
- Activaciones (o regiones de control): representan el tiempo durante el cual el objeto están realizando una operación.
- Mensajes: simbolizan la comunicación entre objetos.

En general, las acciones se consideran secuenciales o concurrentes, dependiendo si están a diferente o a un mismo nivel respecto de la línea de tiempo que se maneje. También está la posibilidad de realizar iterativamente una misma acción también se puede agregar a un diagrama de secuencia. Esto se puede hacer de dos formas (alternativas entre sí):

- Encerrar en un rectángulo aquel mensaje, con lo cual tanto este mismo como las secuencias en su interior se repiten hasta que se cumpla cierta condición. Una condición se especifica al final del rectángulo, pero siempre en su interior. Ejemplo de condición de término es [No más tareas].
- Establecer una línea de vida adicional que comienza en el mensaje que da inicio al ciclo, y que termina con el último mensaje que se encuentra en dicho ciclo. También se incluye una condición de término, tal como [Hasta que no hayan más Tareas].

## Diagramas de Secuencia – Ejemplo en Desarrollo

La figura 3.19 elabora el caso de uso Asignar Habilidad a Recurso (de la figura 3.15). Muestra cómo un administrador de recursos usa el sistema para asignar una habilidad a un recurso, y cómo las clases del sistema trabajan juntos para proveer esta funcionalidad. Los objetos al tope de la figura representan roles de clases; ellas se denominan así porque representan los objetos que participan en la interacción. Las líneas punteadas que se extienden desde cada objeto representan líneas de vida, y los rectángulos ubicados sobre las líneas de vida representan activaciones. Las flechas horizontales entre líneas de vida indican los mensajes intercambiados entre objetos, y son rotulados con el mensaje que es enviado entre los roles de clases. Un mensaje dispara una operación en el objeto receptor.

Un diagrama de secuencia muestra objetos, no clases. Difiere del modelo conceptual dado que éste contiene clases que significan la existencia de objetos. Si más de un objeto de una clase en particular tiene que aparecer en un diagrama de secuencia, múltiples roles de clases y líneas de vida son requeridas en el diagrama.

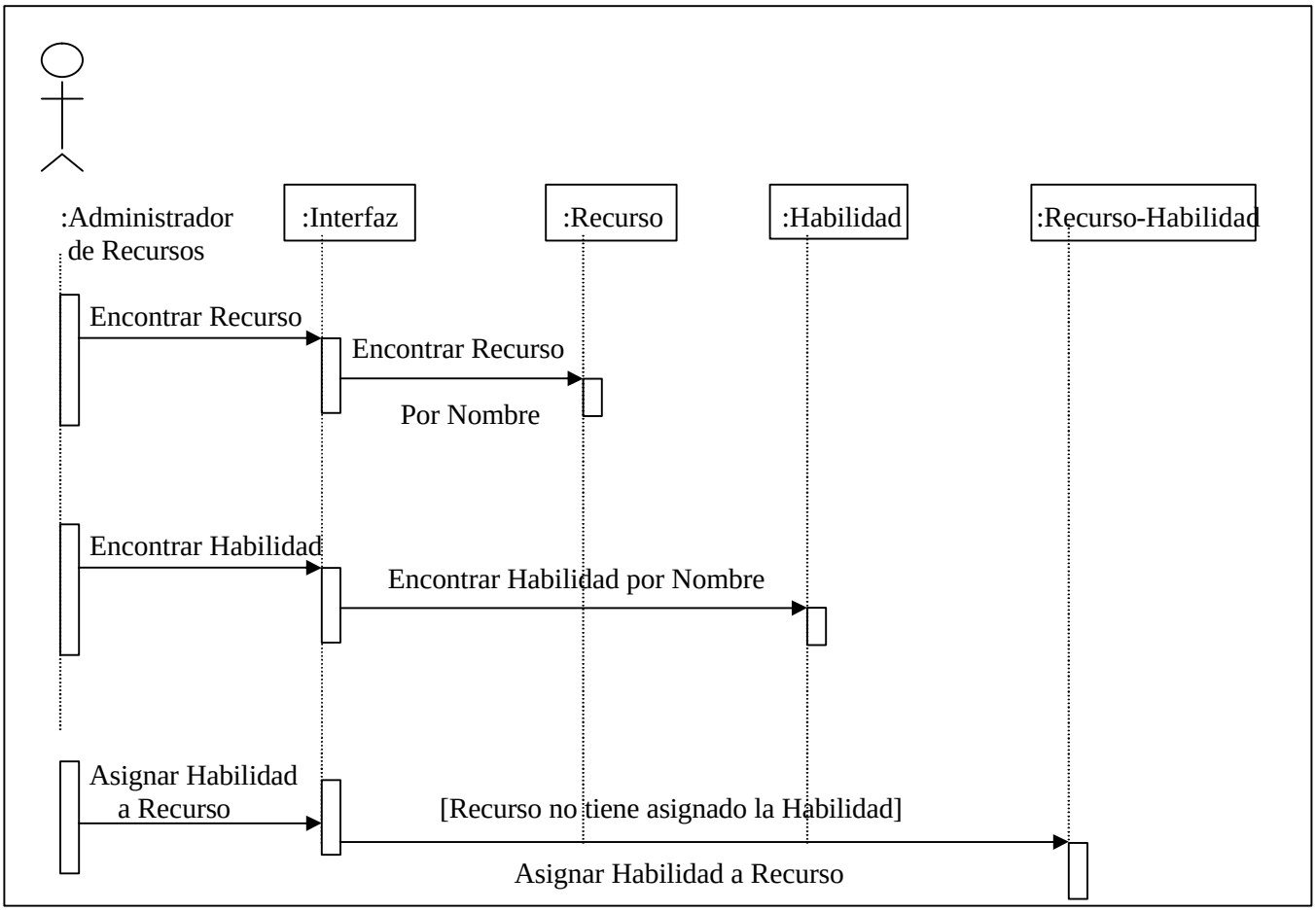



Figura 3.19: Diagrama de Secuencia para el Caso de Uso *Asignar Habilidad a Recurso*

Un administrador de recursos usará la ventana del administrador de recursos, la cual es una interfaz para encontrar un recurso, encontrar una habilidad y asignar la habilidad al recurso. Dicha ventana encontrará un recurso que usa un objeto de la clase Recurso, y una habilidad que usa un objeto de la clase Habilidad. La ventana asignará una habilidad a un recurso si la habilidad no está ya asignada al recurso. Esta condición es un *guardia* y se representa entre paréntesis cuadrados sobre el mensaje que origina desde la línea de vida de la ventana al rol de la clase Recurso-Habilidad.

El siguiente diagrama muestra cómo representa el caso de uso Eliminar Proyectos.

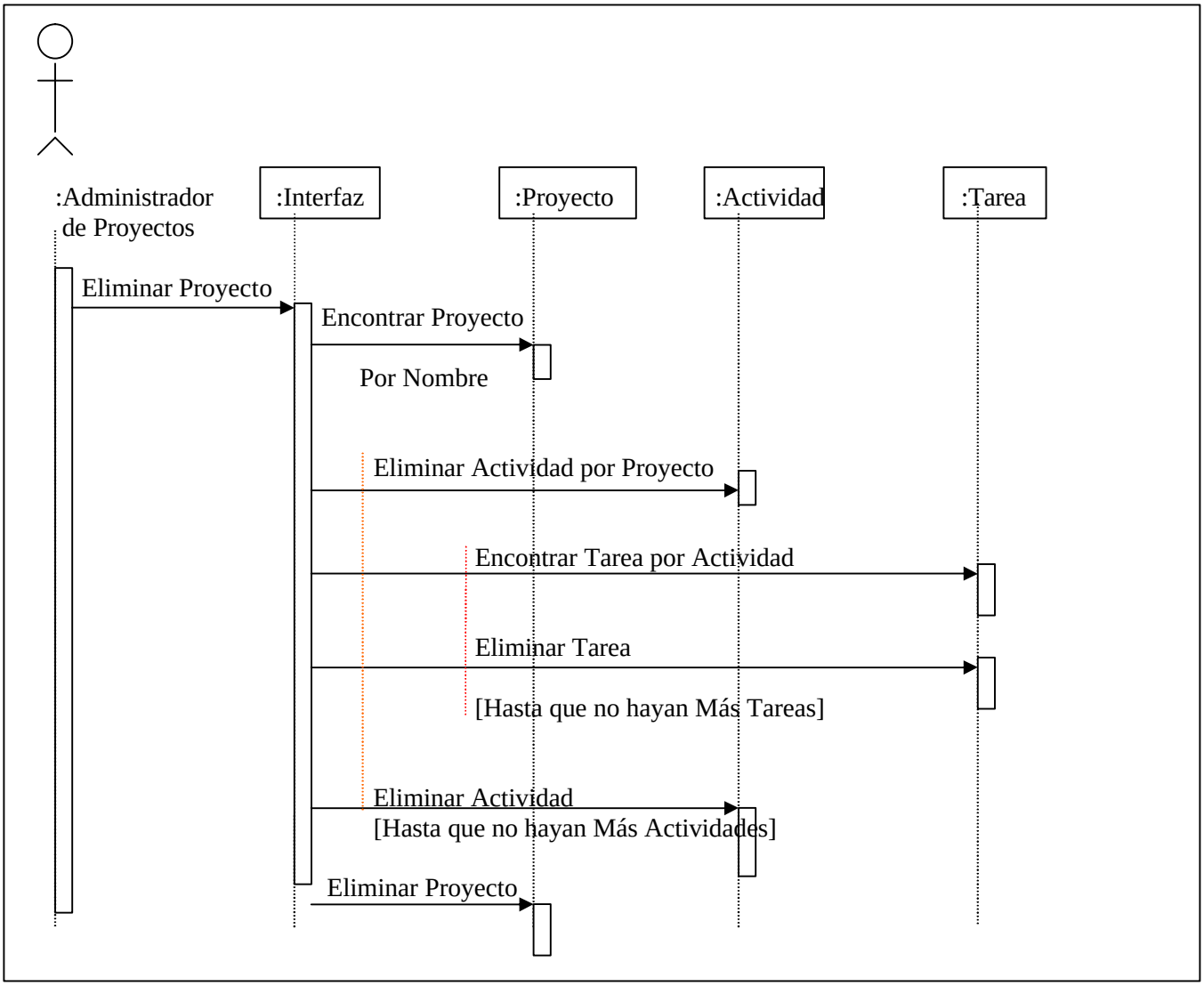



Figura 3.20: Diagrama de Secuencia para el Caso de Uso Eliminar Proyectos

## 4 Resumen

El siguiente diagrama resume las asociaciones entre los casos de uso, modelo conceptual, escenarios y diagramas de secuencia.

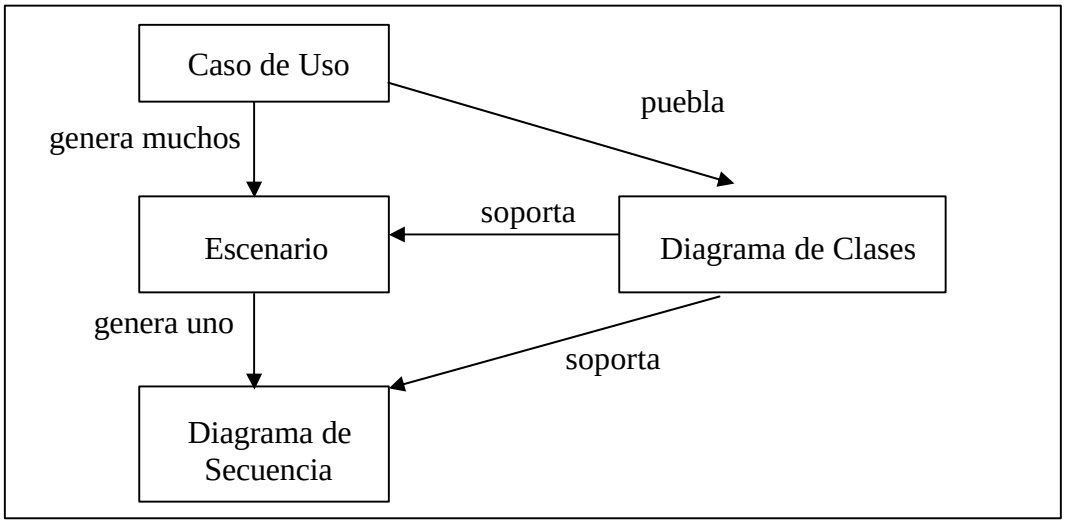



Figura 3.23: Relación entre algunos de los diagramas del análisis orientado al objeto.