

UD 06.

DIAGRAMAS DE CASOS DE USO

Entornos de Desarrollo
CFGs DAW

Autor: Cristina Álvarez Villanueva

Revisado por:

Fco. Javier Valero - franciscojavier.valero@ceedcv.es

M^a Carmen Safont - mariacarmen.safont@ceedcv.es

2019/2020

Versión:191129.1005

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

NOTA: Este trabajo es derivado del trabajo original realizado por Cristina Álvarez Villanueva.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE DE CONTENIDO

1. UML.....	4
2. EL DIAGRAMA DE CASOS DE USO.....	5
3. ELEMENTOS DEL DIAGRAMA DE CASOS DE USO.....	6
4. IDENTIFICAR ACTORES.....	7
5. IDENTIFICAR LOS CASOS DE USO.....	8
6. IDENTIFICAR RELACIONES.....	14
6.1 <i>Relación de asociación</i>	14
6.2 <i>Relación de inclusión</i>	14
6.3. <i>Relación de extensión</i>	15
6.4. <i>Relación de generalización</i>	16
7. HERRAMIENTAS DE DISEÑO.....	16
8. Bibliografía.....	17

UD06. DIAGRAMAS DE CASOS DE USO

1. UML

Para realizar labores de diseño de software, es vital realizar **modelos o diagramas** representando gráficamente la estructura de la aplicación y su funcionalidad, de una manera fácil de entender y rápida de crear. Antiguamente, los diagramas de cada diseñador eran únicos, con lo que era complicado que varios desarrolladores pudieran entenderlos.

Para solucionarlo, se creó el estándar UML. UML o el Lenguaje de Modelo Unificado (**Unified Modeling Language**) es un **lenguaje gráfico** para visualizar, especificar y documentar las partes del desarrollo de software.

UML ha pasado por muchas versiones, siendo la actual **UML 2.x**, soportada y respaldada por OMG (*Object Management Group*). Esta versión tiene 13 tipos de diagramas, divididos en 3 categorías: 6 para la estructura estática de la aplicación o del sistema, 3 para tipos generales de comportamiento y 4 para otros aspectos de las interacciones:

- **Diagramas de estructura:** son la parte estática del modelo. Se centra en los elementos que deben existir en el sistema modelado.
Incluyen el diagrama de clases, de objetos, de componentes, de estructura compuesta, de paquetes y de implementación o despliegue.
- **Diagramas de comportamiento:** son la parte dinámica del modelo. Se centran en lo que debe suceder en el sistema.
Incluyen el diagrama de casos de uso, el diagrama de actividad y el de estado.
- **Diagramas de interacción:** son los derivados del diagrama de comportamiento más general. Se centran en el flujo de control y de datos entre los elementos del sistema modelado.
Incluyen el diagrama de secuencia, de comunicación, de tiempos y de vista de interacción.

El objetivo de **UML** es guiar a los desarrolladores en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes.

La eficiencia se mide en términos de **coste, calidad y tiempo** de desarrollo. El paso desde la determinación de las necesidades del cliente hasta la implementación no es sencillo. Los desarrolladores comienzan capturando los requisitos del cliente en forma de casos de uso en el modelo de casos de uso.

Después analiza y diseñan el sistema para cumplir los casos de uso, creando en primer lugar un modelo de análisis, diseño e implementación.

Por último los desarrolladores preparan un modelo de prueba que le permiten verificar que el sistema proporciona la funcionalidad descrita en los casos de uso.

En el presente tema vamos a ver el **diagrama de casos de uso**, posteriormente veremos lo demás.

2. EL DIAGRAMA DE CASOS DE USO

El **diagrama de casos de uso** (en adelante DCU) se emplea para entender el uso del sistema. Muestra un conjunto de actores, las acciones (*casos de uso*) que se realizan en el sistema, y las relaciones entre ellos.

«El diagrama de casos de uso describe el comportamiento del sistema desde el punto de vista de un usuario/sistema que interactúa con él.»

(Ramos, 2014:262)

Entonces, los **casos de uso** (en adelante CU) modelan el sistema desde el punto de vista del usuario. Así, son una herramienta muy buena para obtener los requisitos de software durante la fase de análisis de un proyecto. Deben cumplir los siguientes objetivos:

- Definir los requisitos funcionales y operativos del sistema, diseñando un conjunto de escenarios que faciliten una descripción de cómo se usará el sistema
- Dar una descripción clara de cómo el usuario interactúa con el sistema y al revés
- Dar una base para la validación de las pruebas

El diagrama es la representación visual de los casos de uso, a través de notación UML.

3. ELEMENTOS DEL DIAGRAMA DE CASOS DE USO

El **diagrama de casos de uso** está formado por:

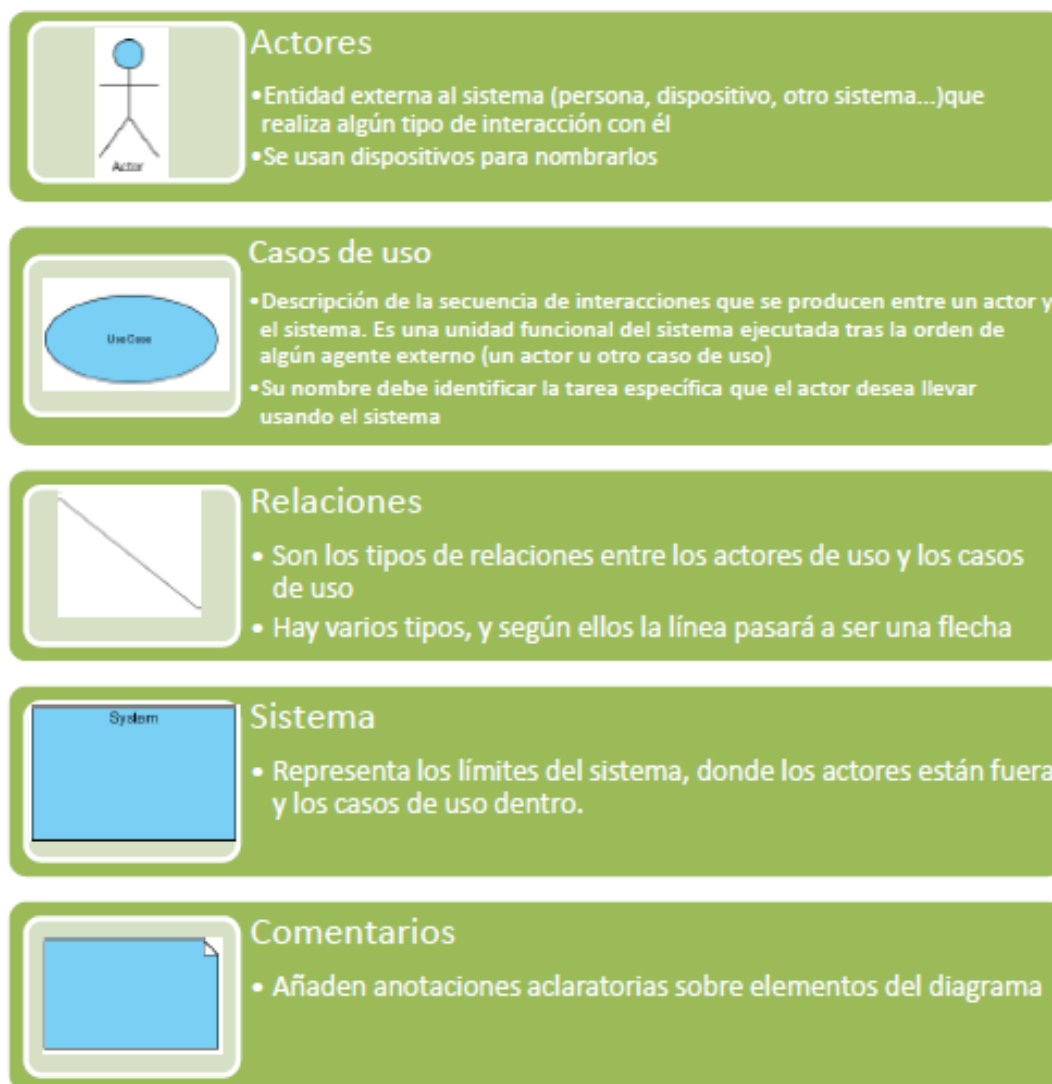


Figura 1: Elementos del **diagrama de casos de uso**

Los CU siempre los inician los actores para pedir o modificar información sobre el sistema, o bien para informar al sistema de que ha ocurrido algo en el entorno. El nombre del caso de uso debe coincidir con el objetivo del actor principal, que es normalmente el que comienza el caso de uso.

Ejemplo 1: una máquina expendedora donde un cliente compra una chocolatina y el reponedor abastece la máquina de ellas.

Los casos de uso serán: *Comprar Chocolatina* y *Reabastecer*:

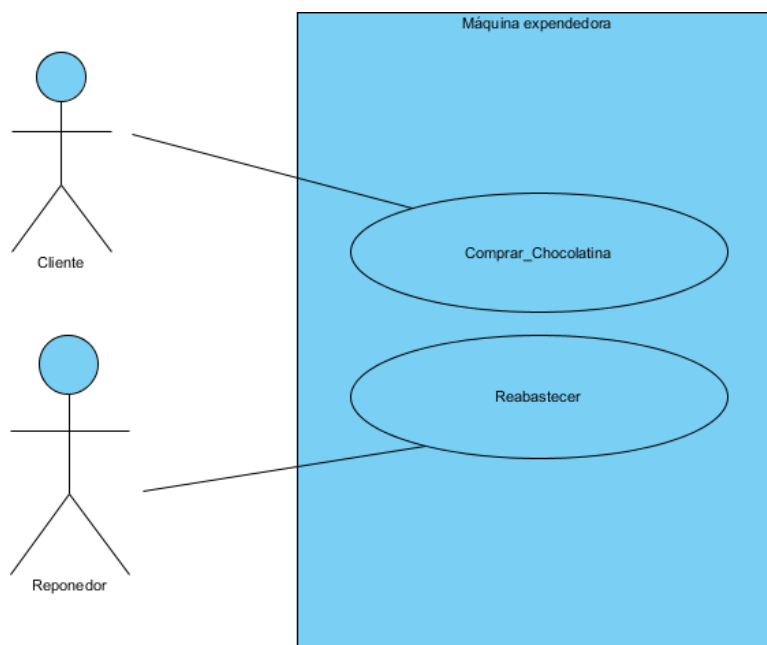


Figura 2: Ejemplo 1

4. IDENTIFICAR ACTORES

Para identificar a los actores debemos saber **qué** y **quié**n interactúa con el sistema y **qué roles** desempeñan cuando interactúan con él. Para ello, debemos considerar que:

- Los actores son siempre externos al sistema
- Interactúan directamente con él
- Representan roles que personas y elementos tienen con el sistema
- Su nombre describe su rol
- Una misma entidad puede interpretar varios roles como actores distintos

Los actores implicados en un caso de uso pueden ser de dos tipos:

- **Principales:** los que activan el caso de uso.
- **Secundarios:** los que interactúan con el caso de uso una vez activado.

Suelen ser otros sistemas, dispositivos o componentes externos.

Ejemplo 2. Siguiendo con el Ejemplo 1 un cliente puede ser un *cliente puntual* o un *cliente registrado*. Y el reponedor puede ser un *administrativo* o el propio *gerente*.

5. IDENTIFICAR LOS CASOS DE USO

Para identificar los casos de uso necesitamos entender *qué hace el sistema* y *qué hace el actor al interactuar con él*. Para ello, nos podemos preguntar cosas como:

- ¿Qué tareas hacen los actores relacionados con el sistema?
- ¿Qué información crea/lee/guarda/modifica/borra el actor?
- ¿Notifica el actor de los cambios externos ocurridos al sistema?
- ¿Notifica el sistema de los cambios internos a un actor?
- ¿Interactúa el sistema con algún sistema externo?

Los casos de uso se especifican en modo texto, y se puede **crear una plantilla** con elementos como estos:

- **Nombre:** el nombre del CU
- **ID:** el identificador del CU
- **Breve descripción:** lo que se espera que haga el CU
- **Actores implicados:** principales o secundarios
- **Precondiciones:** las condiciones a cumplir antes de que empiece el CU
- **Curso normal:** pasos cronológicos que llevan a finalizar con éxito el CU
- **Postcondiciones:** condiciones que han de cumplirse al acabar el CU
- **Alternativas:** desviaciones del curso normal del CU (errores, excepciones, etc). El sistema deberá informar de ellas al usuario.

Ejemplo 3. Supongamos un *caso de venta*.

En este sistema, un actor cliente puede tener dos casos de uso:

Comprar_Productos y *Buscar_Productos*.

Y un actor administrativo puede tener *Insertar_Productos*.

En ese caso, veamos cómo sería el diagrama:

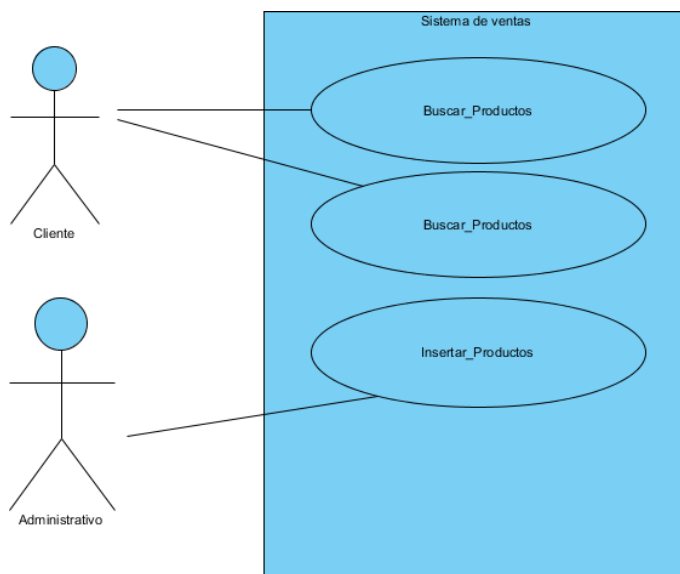


Figura 3: Diagrama de caso de uso del ejemplo 3

Y la plantilla del caso de uso BuscarProductos sería así:

Nombre: BuscarProductos
ID: CU-1
Descripción: El cliente pide consultar productos que cumplan una serie de criterios de búsqueda. El sistema muestra los datos de los productos solicitados
Actores: Cliente
Precondiciones: Se requiere que el cliente esté registrado en el sistema
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El cliente selecciona buscar productos 2. El sistema solicita los criterios de búsqueda 3. El cliente introduce los criterios solicitados 4. El sistema busca productos que cumplan los criterios introducidos por el cliente 5. El sistema encuentra productos y se los muestra al cliente
Postcondiciones: El sistema muestra los productos encontrados
Alternativa 1: <ol style="list-style-type: none"> 5. El sistema no encuentra productos. 6. El sistema informa al cliente de que no hay productos que cumplan el criterio de búsqueda 7. Fin del caso de uso

Figura 4: Ejemplo 3 → Buscar_Productos (basado en Ramos, 2014:266)

Vemos que en el punto 5 se ha establecido una alternativa: si el sistema no encuentra los productos según el criterio de búsqueda.

Vamos a mejorar este ejemplo añadiendo un **paso condicional y un paso repetitivo**:

Nombre: Buscar_Productos ID: CU-1
Descripción: El cliente pide consultar productos que cumplan una serie de criterios de búsqueda. El sistema muestra los datos de los productos solicitados
Actores: Cliente
Precondiciones: Se requiere que el cliente esté registrado en el sistema
Curso normal del caso de uso: <ol style="list-style-type: none">1. El cliente selecciona buscar productos2. El sistema solicita los criterios de búsqueda3. El cliente introduce los criterios solicitados4. El sistema busca productos que cumplan los criterios introducidos por el cliente5. SI el sistema encuentra algún producto: PARA cada producto encontrado El sistema muestra los datos SINO El sistema informa al cliente que no hay productos que cumplan con el criterio de búsqueda6. El cliente repite los pasos 2 a 6 hasta finalizar las búsquedas
Postcondiciones: Ninguna
Alternativa 1: Ninguna

Figura 5: Ejemplo 3 → Buscar_Productos (basado en Ramos, 2014:267)

Y ahora veamos cómo sería el **CU de Insertar_productos**:

Nombre: Insertar_Productos ID: CU-2
Descripción: El administrativo solicita insertar un producto en el sistema. El sistema inserta los datos del producto introducido por el administrativo.
Actores: Administrativo
Precondiciones: Se requiere que el administrativo esté registrado en el sistema.
Curso normal del caso de uso: <ol style="list-style-type: none">1. El administrativo solicita insertar un producto2. El sistema solicita los datos del producto3. El administrativo introduce los datos4. El sistema valida los datos introducidos por el administrativo5. Si los datos son correctos: El sistema inserta el producto SINO El sistema comunica al cliente que ha introducido los datos erróneos.6. El administrativo repite los pasos 2 a 6 hasta finalizar la inserción del producto
Postcondiciones: Ninguna
Alternativa 1: Ninguna

Figura 6: Ejemplo 3 → Insertar_Productos (basado en Ramos, 2014:268)

Y queda por ver **el caso de uso de *Comprar_Productos***:

Nombre: Comprar_Productos
ID: CU-3
Descripción: El cliente selecciona los productos a comprar y escribe la cantidad deseada. El sistema comprueba si hay existencias y muestra el importe por cada producto seleccionado. Cuando el cliente no desea comprar más productos, el sistema muestra el importe total de la compra y solicita al cliente el pago de la misma.
Actores: Cliente
Precondiciones: Se requiere que el cliente esté registrado en el sistema
Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El cliente selecciona comprar productos 2. El cliente selecciona un producto y escribe la cantidad. 3. El sistema valida los datos introducidos. 4. El sistema muestra el importe y el total acumulado. 5. El cliente repite los pasos 2 a 4 hasta indicar fin de compra. 6. El sistema calcula el importe total y lo muestra al cliente. 7. El sistema finaliza la compra y solicita la forma de pago al cliente. 8. El cliente selecciona la forma de pago. 9. El sistema solicita los datos al cliente según la forma de pago elegida. 10. El cliente introduce los datos. 11. El cliente realiza el pago. 12. El sistema muestra la confirmación del pago y registra la compra del cliente.
Postcondiciones: El cliente realiza la compra de productos.
Alternativa 1: <ol style="list-style-type: none"> 3.1. El producto seleccionado no tiene existencias. 3.2. El sistema informa al cliente que no hay existencias. 3.3. Continuar con el paso 5 del curso normal.
Alternativa 2: <ol style="list-style-type: none"> 6.1. Si el importe total es 0: <ol style="list-style-type: none"> 6.1.1. Fin del caso de uso.
Alternativa 3: <ol style="list-style-type: none"> 10.1. Los datos introducidos por el cliente no son correctos. 10.2. El sistema informa al cliente. 10.3. Fin del caso de uso.

Figura 7: Ejemplo 3 → Comprar_Productos (basado en Ramos, 2014:269)

Como se observa, se ha optado por alternativas en vez de por pasos condicionales en el curso normal.

Para usar esta documentación hemos de tener en cuenta que:

- Se han de usar frases sencillas
- No se deben hacer referencias a elementos de la interfaz de usuario (ej. página principal, pantalla de ingreso, botones, etc.)
- Se deben evitar detalles arquitectónicos (ej. decir “el sistema almacena el pedido” en vez de “el sistema almacena el pedido en la base de datos MySQL”).

Los casos de uso son muy limitados ya que se describen con texto informal. Por lo tanto, deberemos usar una nueva notación para especificar este comportamiento interno. En UML se propone usar una notación llamada “**Diagrama de Actividad**”. Lo veremos en próximas unidades.

EJERCICIO. Prueba a hacer **un caso de uso** describiendo los pasos que sigue un usuario para identificarse en un sistema.

El usuario debe estar registrado.

El sistema le pedirá su nombre y su contraseña.

Si los datos son correctos le dará acceso, y si no se le bloqueará tras tres intentos fallidos.

6. IDENTIFICAR RELACIONES

Existen varios tipos de relaciones:




Relación	Notación
Asociación	
Extensión	
Inclusión	
Generalización	

Figura 8:Tipos de relaciones en DCU

Veamos cada una de ella en detalle y con ejemplos.

6.1 Relación de asociación

Es la línea de comunicación entre un actor y un caso de uso en el que participa. Permite expresar que un actor realiza determinada tarea.

Se representa mediante una línea continua. Algunos autores lo hacen con una línea continua que acaba en forma de flecha abierta.

En el caso del ejemplo 3 sería:

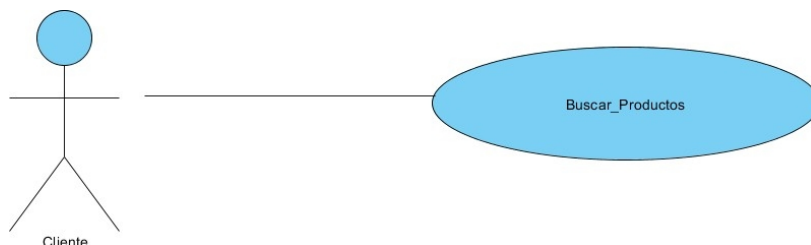


Figura 9: Relación de asociación

6.2 Relación de inclusión

Permite que un caso de uso base incluya el comportamiento de otro caso de uso.

Se realizan con aquellas relaciones **entre un caso de uso general, y casos de uso que siempre debe realizarse**.

Esta relación de inclusión es **indispensable** que ocurra para que el CU base sea terminado con éxito.

Representa acciones que siempre se realizan en varios casos de uso. Son como subrutinas.

Se representa mediante una flecha discontinua con la palabra «**include**».

La relación apunta al caso de uso a ser incluido, es decir, **del caso general al caso más concreto**. Ver el ejemplo.

Ejemplo 4. Imaginemos un sistema de viajes aéreos. Tiene dos casos de uso (realizar una reserva y modificarla). Ambos comparten pasos, en concreto, comparten cómo reservar asientos. Podríamos dejarlo así:

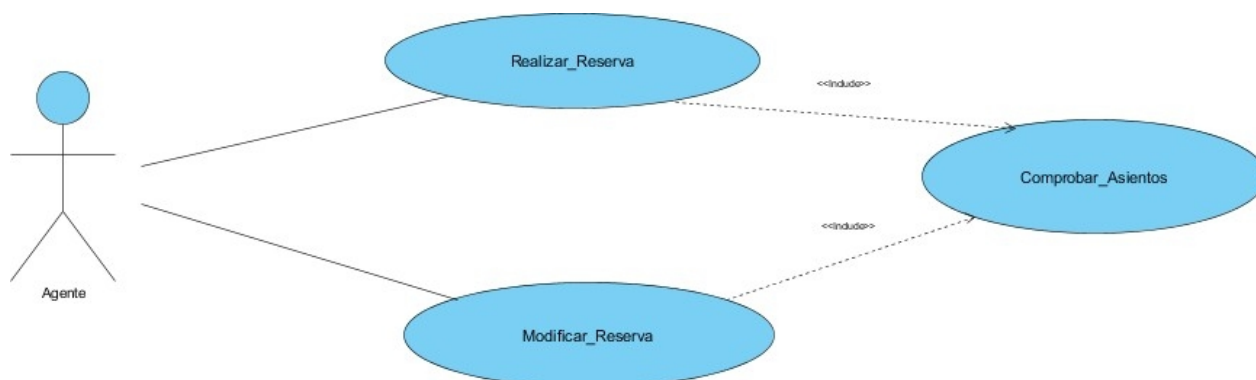


Figura 10: Uso de caso incluido. (Ramos, 2014: 271)

6.3. Relación de extensión

Permite que un caso de uso **extienda su comportamiento** con uno o más fragmentos de comportamiento de otro. Se utiliza para especificar que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.

Se representa mediante una flecha discontinua con la palabra «**exclude**». La relación apunta al caso de uso que extenderá.

A diferencia de en la inclusión, **el CU extendido no sabe nada del CU que lo extiende**.

Por tanto, esta relación de extensión **no es indispensable** que ocurra, pero si lo hace le da un valor extra.

El CU base sí debe tener una descripción textual que indique qué puntos podrían ser ampliados.

Ejemplo 5. Vamos a ampliar parte del ejemplo 5. Vamos a incluir en Realizar_Reserva la situación de que se puedan reservar asientos de primera clase. En este caso tenemos:

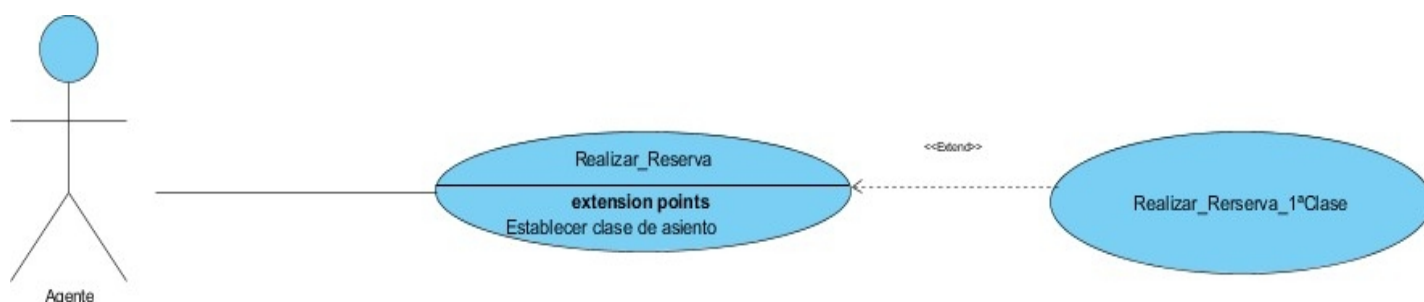


Figura 11: Uso de caso extendido (Ramos, 2014: 271)

Prueba a hacer la representación gráfica de un sistema personal donde un director puede hacer dos operaciones: consultar y modificar datos de los empleados. Para ello, necesita localizar los detalles del empleado a modificar o a consultar.

6.4. Relación de generalización

Es como la generalización entre clases. El caso de uso hijo **hereda** el comportamiento y el significado del caso de uso padre.

Es decir, indica que un elemento es el caso general de otro. Se usa cuando se tiene **uno o más CU que son especificaciones de un CU más general**.

La relación de generalización se puede hacer entre actores o entre casos de uso.

Ejemplo 6. Si tenemos un sistema de ventas donde se pueden buscar libros o CDs de música, podemos representarlo como un CU padre de Buscar_Productos y dos especializaciones (libros y CDs):

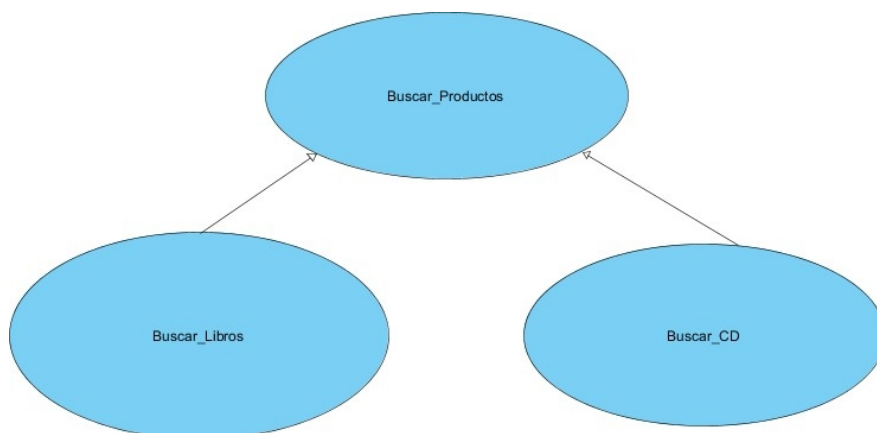


Figura 12: Uso de generalización (Ramos, 2014:275)

7. HERRAMIENTAS DE DISEÑO

Existen múltiples herramientas software para el **desarrollo de sistemas UML**. Algunas herramientas CASE de modelado UML son:

- **Argo UML**: de código abierto, diseñada en Java y con licencia pública Eclipse (EPL 1.0). Multiplataforma y en 10 idiomas.
- **LibreOffice Draw**: existen galerías online con los elementos UML que se pueden incorporar a LibreOffice para realizar los diagramas.
- **Visual Paradigm**: además de modelar en UML ofrece generación de informes y capacidades de ingeniería de código, ya que permite navegar entre el código y su visualización. De código propietario pero con licencia gratuita para la versión estándar. Integrable en Eclipse y en Netbeans.

Y además, IDEs como **Eclipse o Netbeans** permiten incluir módulos o plugins sobre UML.

8. BIBLIOGRAFÍA

[1] Argo UML

<http://argouml.tigris.org>

[2] Umbrella

<http://umbrello.kde.org>

[3] Visual Paradigm

www.visual-paradigm.com

[4] Ramos, A.; Ramos, MJ (2014): *Entornos de desarrollo*, Garceta, Madrid

[5] Casado, C. (2012): *Entornos de desarrollo*, RA-MA, Madrid