

TEMA 4. CASOS DE USO

ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN



Universidad
Francisco de Vitoria
UFV Madrid

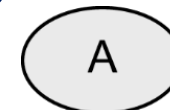
ÍNDICE

Tema 4: Diagramas de casos de uso

1. Introducción
2. Casos de uso
3. Actores
4. Relaciones entre casos de uso y actores
5. Relaciones entre casos de uso
6. Relaciones entre actores
7. Descripción de casos de uso
8. Buenas prácticas
9. Errores típicos
10. Elementos de notación

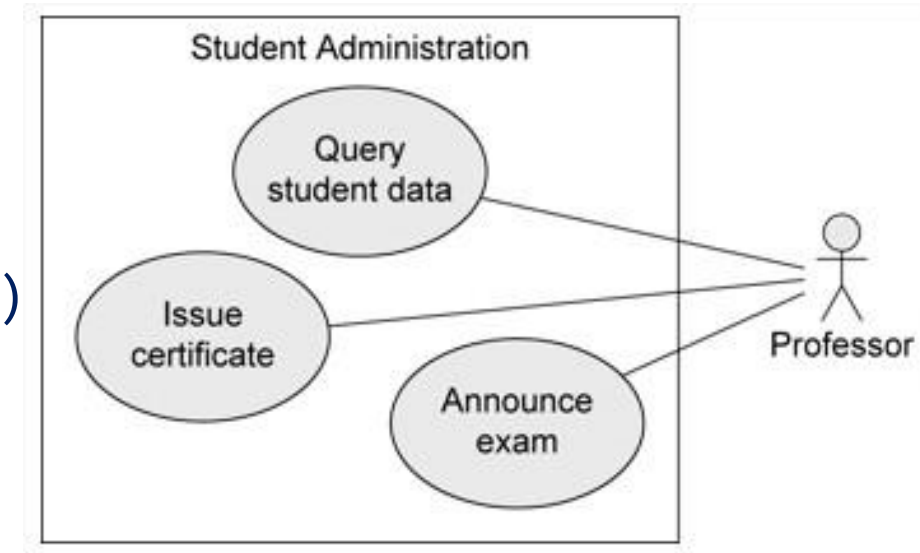
1.INTRODUCCIÓN

- El caso de uso es un concepto fundamental de muchos métodos de desarrollo orientados a objetos.
- Los diagramas de casos de uso expresan las **expectativas de los clientes**.
 - Resulta esencial en un diseño detallado.
- El diagrama de casos de uso se utilizan **durante todo el proceso de análisis y diseño**.
- Podemos usar un diagrama de casos de uso para responder las siguientes preguntas:
 - **¿Qué se está describiendo?** (el sistema).
 - **¿Quién interactúa con el sistema?** (los actores).
 - **¿Qué pueden hacer los actores?** (los casos de uso).



2.CASOS DE USO

- **El sistema** (¿qué describe?)
 - Sistema de gestion de alumnos.
- **Los actores** (¿quién interactúa con el sistema?)
 - Profesor.
- **Los casos de uso** (¿qué pueden hacer los actores?)
 - Consultar datos de alumnos.
 - Emitir un certificado.
 - Anunciar un examen.



Los actores y los casos de uso representan los conjuntos de todos los actores de un tipo y de todos los escenarios (no la instancia particular).
Ejemplo: escribir Pepe en vez de profesor.

2.CASOS DE USO

A

- Describe **la funcionalidad** que se espera que realice el **sistema**.
- Ofrece un resultado concreto **para uno o más actores** que **interactúan** con este caso de uso.
- "Entre un actor y el sistema, los casos de uso describen las **acciones e interacciones vinculadas con un objetivo funcional del actor**".
- El caso de uso va a convertirse en **un requerimiento funcional del cliente**.
- El **conjunto de todos los casos de uso** describe **la funcionalidad** que debe proporcionar un sistema.
 - Documenta la funcionalidad que ofrece un sistema. Todos ellos compondrán el **documento de especificación de requisitos**.
- Notaciones posibles para un caso de uso:

Más utilizada

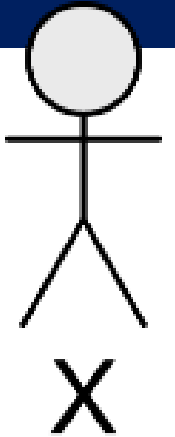
Query
student data

Query
student data

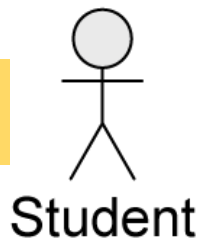
Query
student data

3. ACTORES

- Los actores interactúan con el **sistema** ...
 - Por la **ejecución de casos de uso**. Los actores inician la ejecución de casos de uso.
 - **Al ser utilizados por casos de uso**. Los actores aportan funcionalidad para la ejecución de casos de uso.
- Los **actores representan roles** que pueden adoptar los usuarios.
 - Se pueden representar múltiples roles de un mismo usuario.
- Los **actores no forman parte del sistema**, están fuera de los límites del sistema (fuera del cuadro que representa el sistema).
- Notaciones posibles para un actor:



Más utilizada



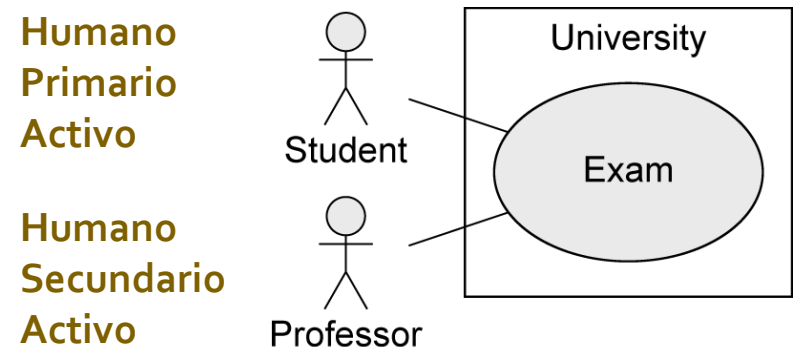
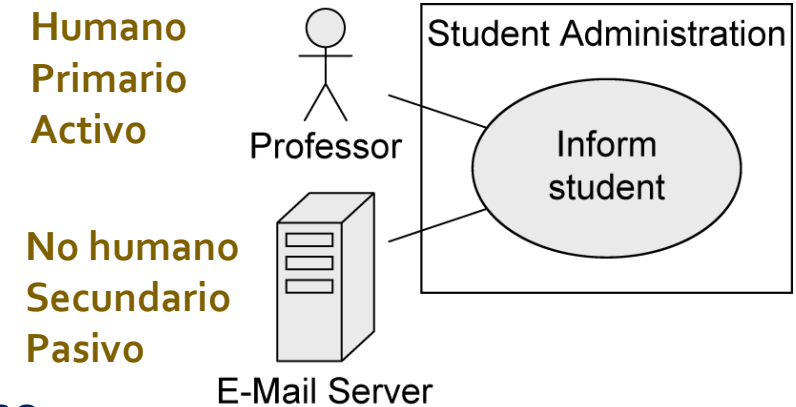
«actor»
Professor



E-Mail Server

3. ACTORES. CATEGORÍAS

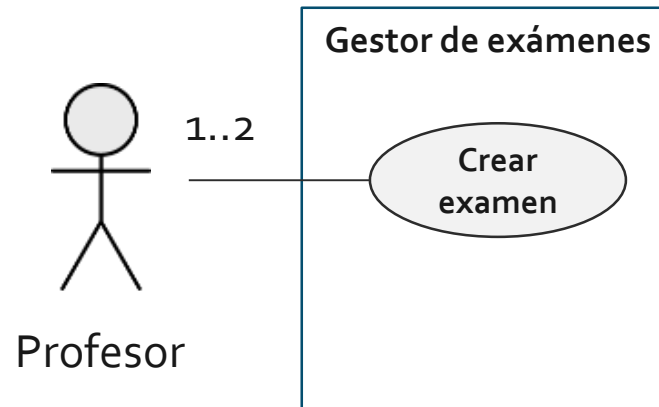
- **Humano**
 - Por ejemplo: estudiante, profesor.
- **No humano**
 - Por ejemplo: servidor de correo electrónico.
- **Primario:** es el encargado principal de la ejecución del caso de uso.
 - Para ellos el objetivo del caso de uso es **esencial** y constituye un **objetivo** del actor.
- **Secundario:** no recibe ningún resultado directo.
 - Para ellos el objetivo del caso de uso **no es esencial**, pero han de interactuar con él.
- **Activo:** inicia la ejecución del caso de uso.
- **Pasivo:** proporciona funcionalidad para la ejecución del caso de uso.



4.RELACIONES ENTRE CASOS DE USO Y ACTORES = ASOCIACIONES

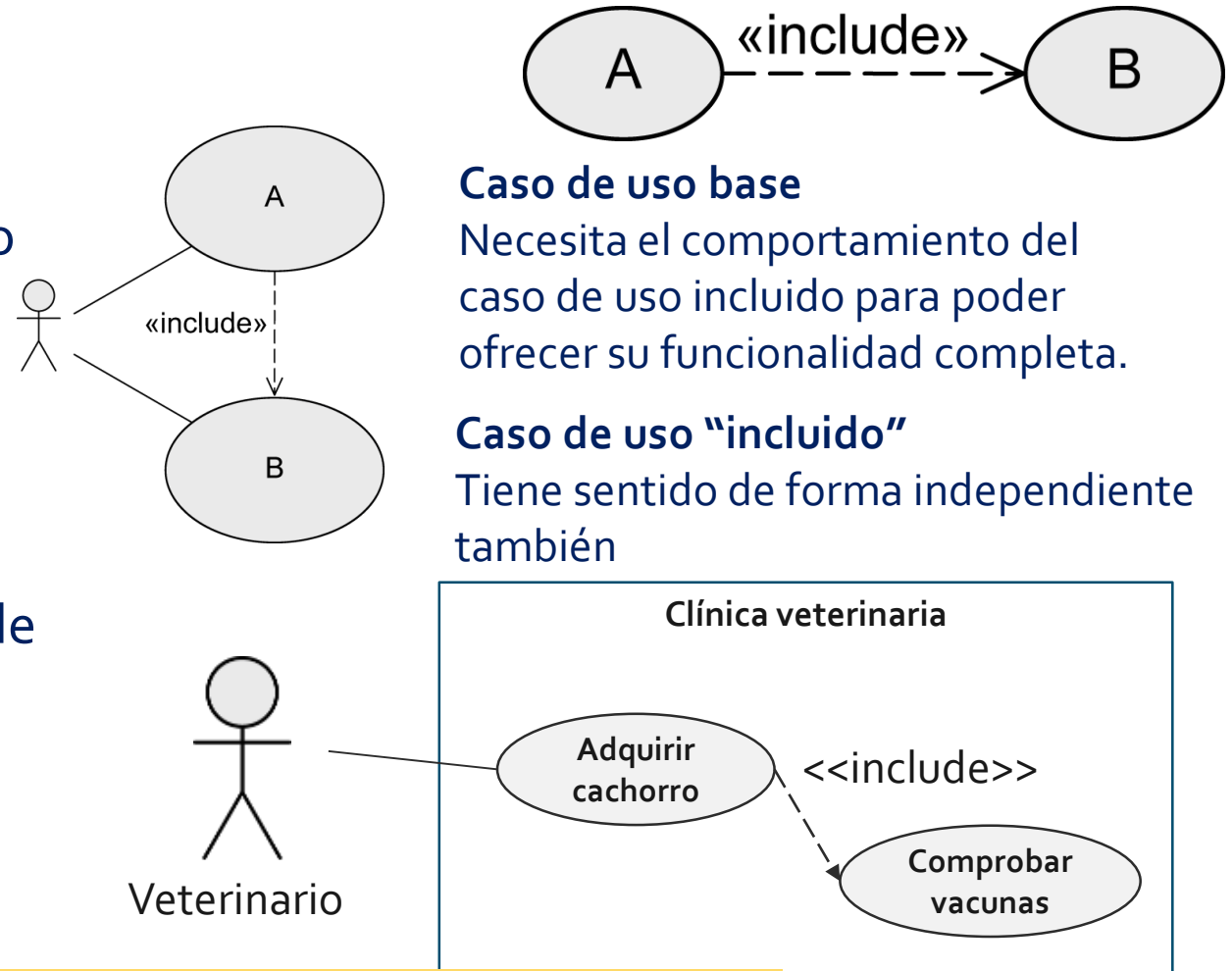
- Los actores están conectados con casos de uso a través de **líneas continuas (asociaciones)**.
- Cada actor debe comunicarse **con un caso de uso al menos**.
- Una **asociación suele ser binaria** (relación entre un actor y un caso de uso)
- Si no se especifica multiplicidad en la asociación se asume que es binaria (1 a 1).
- Es lo más común en la definición de casos de uso.

Sólo uno o dos profesores pueden crear un examen.
Sería una restricción en la normativa universitaria



4. RELACIONES ENTRE CASOS DE USO. RELACIÓN <<INCLUDE>> (ANTES <<USES>>)

- Un caso de uso base incorpora explícitamente el comportamiento de otro caso de uso en algún lugar de su secuencia.
- Por tanto, el comportamiento de un caso de uso (caso de uso “incluido”) se **integra en el comportamiento de otro caso de uso** (caso de uso base).
- **Include** → **enriquece** un caso de uso con otro y ayuda a **compartir una funcionalidad** que sea común entre varios casos de uso. También puede servir para **estructurar** un caso de uso concretando sus subfunciones.
- El caso de uso “incluido” existe únicamente con ese propósito porque no es el objetivo principal del actor.



Caso de uso base

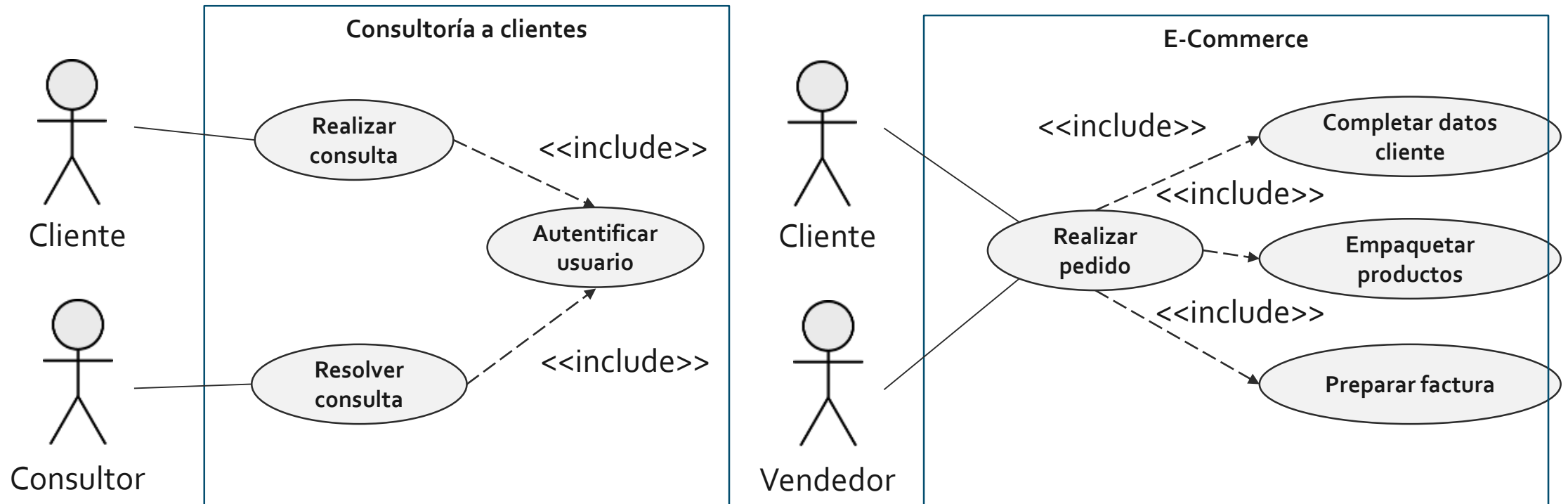
Necesita el comportamiento del caso de uso incluido para poder ofrecer su funcionalidad completa.

Caso de uso “incluido”

Tiene sentido de forma independiente también

A la hora de adquirir un cachorro para venderlo después se verifica que tenga las vacunas correctas. El caso de uso Adquirir Cachorro incluye esa comprobación de vacunas.

4. RELACIONES ENTRE CASOS DE USO. RELACIÓN <<INCLUDE>> (ANTES <<USES>>)

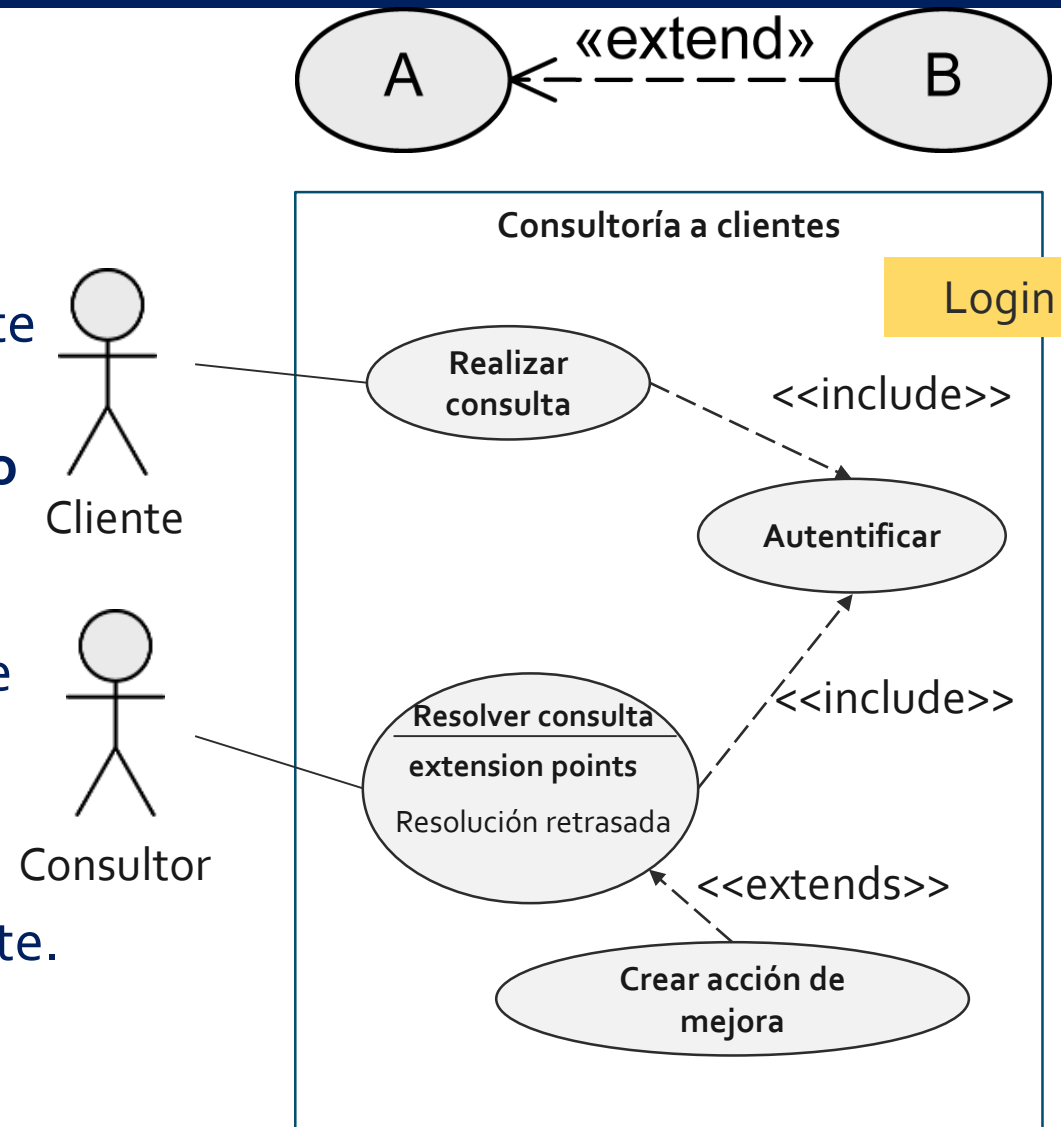


Include. No debe utilizarse en exceso para no llegar a nivel de diseño detallado en la fase de análisis.

4.RELACIONES ENTRE CASOS DE USO.

RELACIÓN <<EXTEND>>

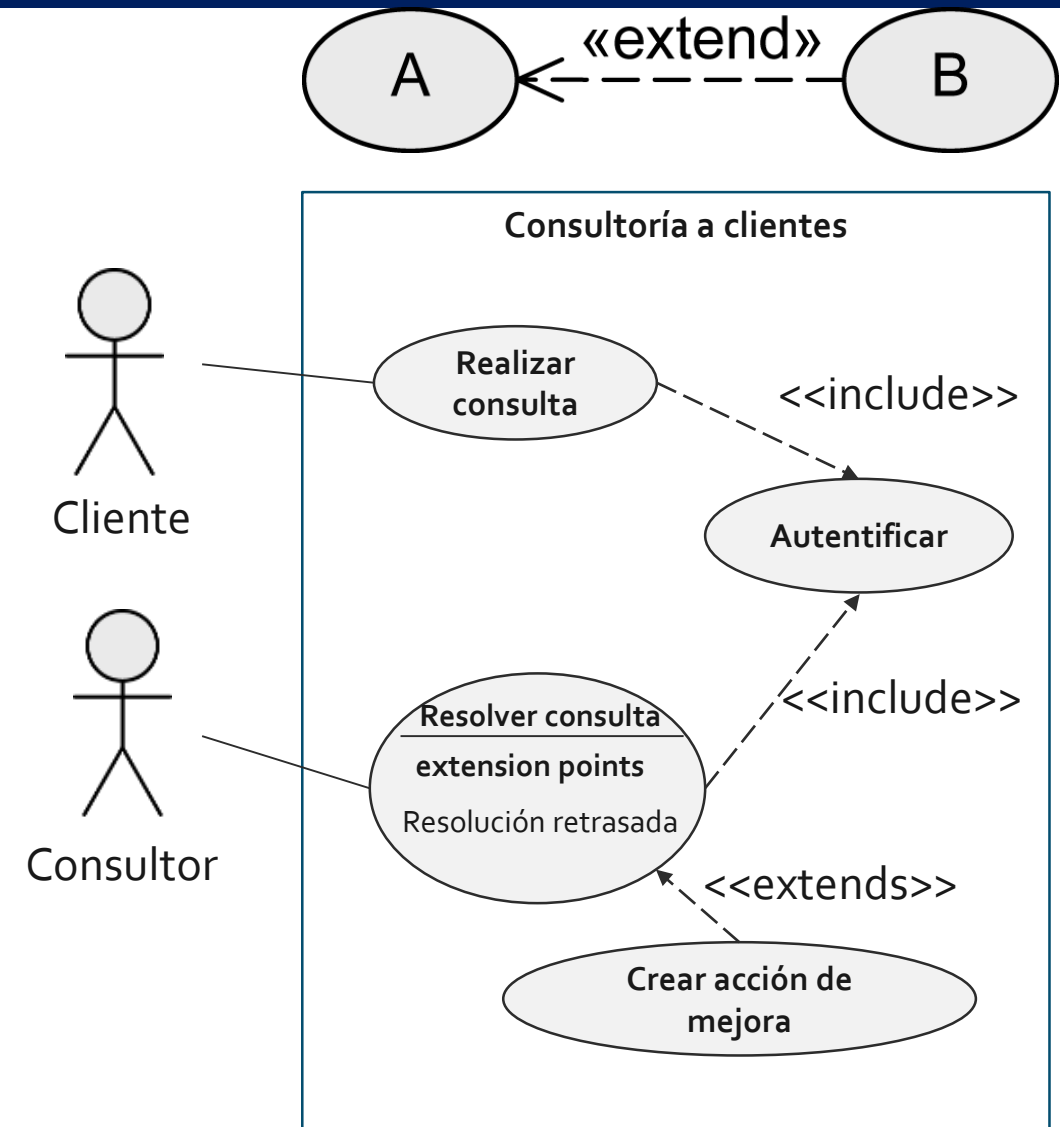
- Un caso de uso base **añade** implícitamente el **comportamiento de otro caso de uso**.
- En el **caso de uso base**, la extensión se hace sobre unos puntos concretos y previstos en el momento del diseño, llamados **puntos de extensión**. Se escriben directamente dentro del caso de uso.
- El caso de uso extendido **puede incluir comportamiento del caso de uso que se extiende, aunque no tiene que incluir todo** el comportamiento.
- La relación de extensión sirve para diseñar una parte que sea opcional en el sistema, o un subflujo que sólo se da bajo unas condiciones concretas.
- Se recomienda utilizarlo sólo para **insertar un nuevo comportamiento no previsto** en un caso de uso existente.



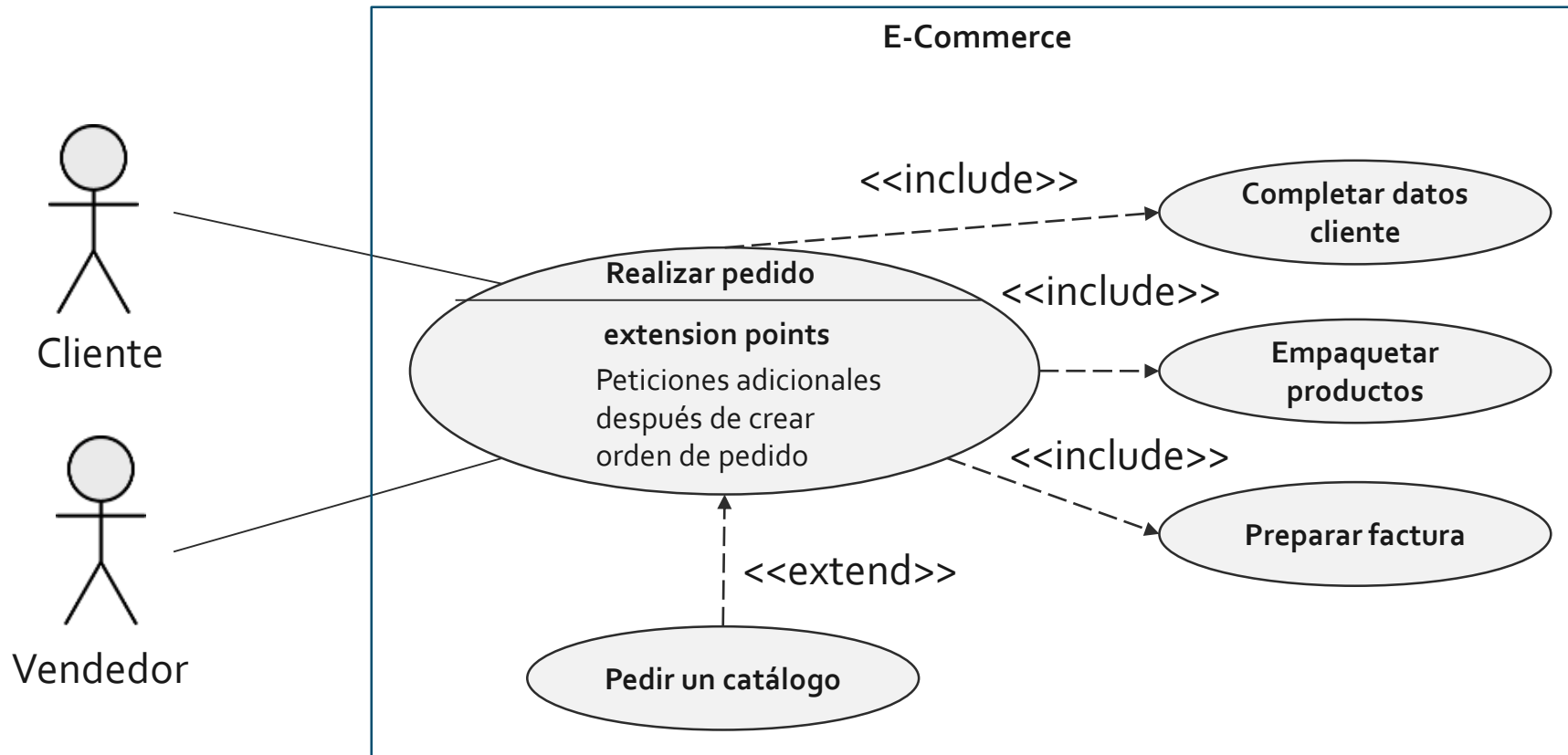
4.RELACIONES ENTRE CASOS DE USO.

RELACIÓN <<EXTEND>>

- Autenticar: validar el perfil del usuario.
- Extends:
 - Por ejemplo: cuando pase un tiempo definido por la empresa sin haber recibido la resolución a su consulta, se crea una acción de mejora para dejar constancia del retraso y llevar a cabo ciertas acciones.
 - Se puede decir que “Crear acción de mejora” es una subfunción de uso “Resolver consulta”.



4.RELACIONES ENTRE CASOS DE USO. RELACIÓN <<EXTEND>>

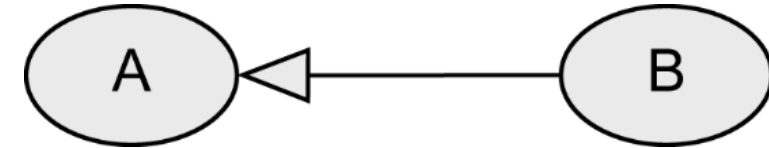


Extend. No debería utilizarse en exceso porque a veces produce confusión y se suele utilizar para excepciones

4.RELACIONES ENTRE CASOS DE USO.

RELACIÓN GENERALIZACIÓN

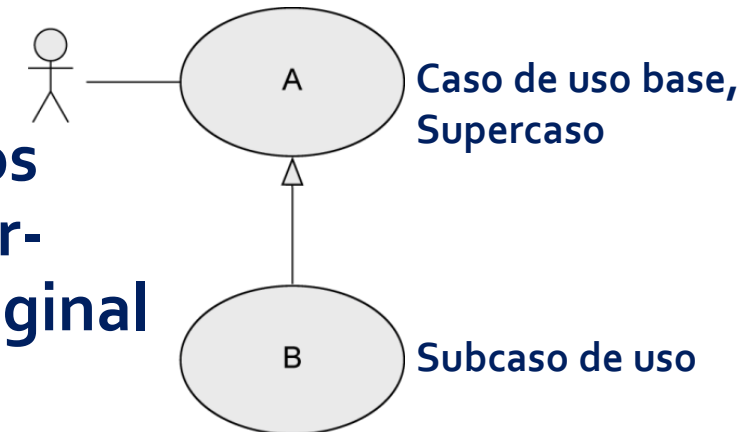
- B hereda de A.



- Un caso de uso B (subcaso) **hereda el comportamiento y significado** de otro caso de uso A. Y por tanto, las relaciones de comunicación, las **asociaciones**.

- Los casos de uso “hijo” son una **especialización** del caso de uso “padre”.

- En el **caso de uso “hijo”** se especifican los **comportamientos extra** que es necesario **añadir al caso de uso “padre”** (supercaso), para representar una **funcionalidad diferente a la original**

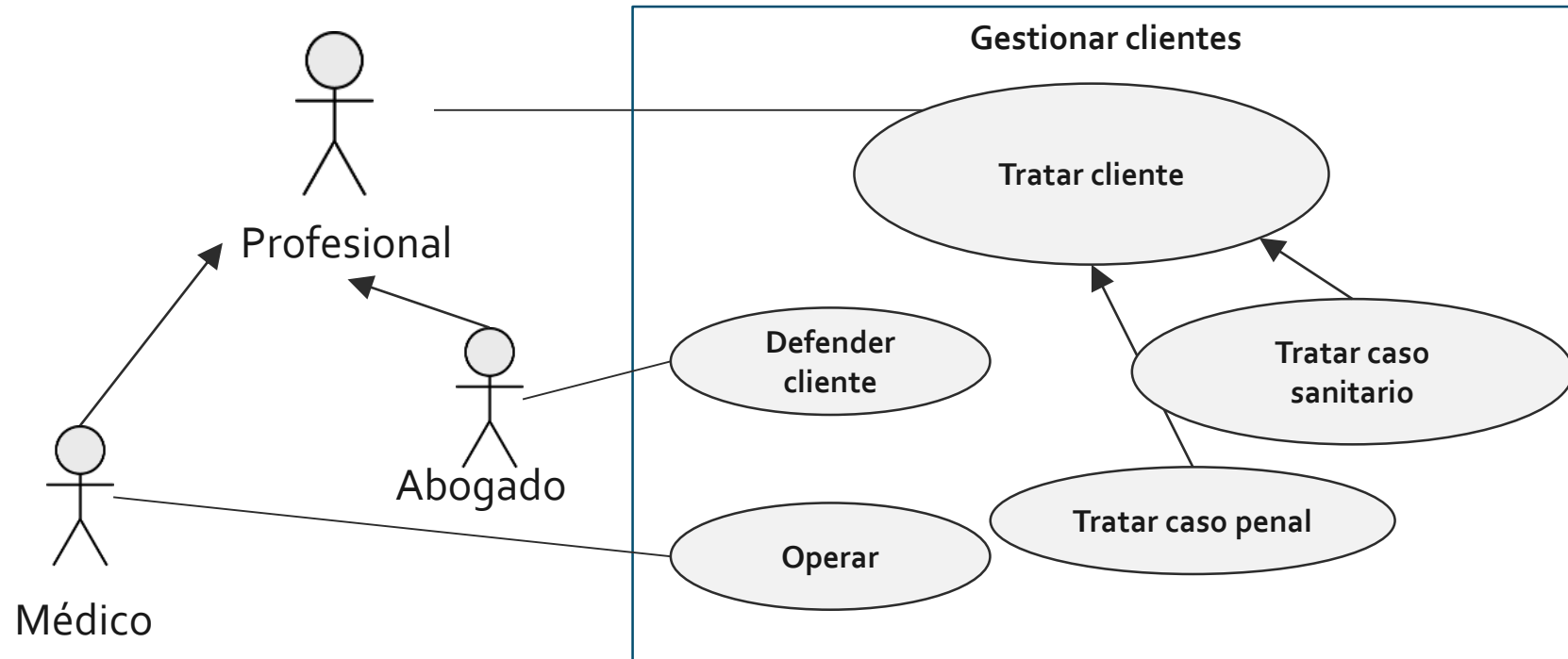


4.RELACIONES ENTRE CASOS DE USO. RELACIÓN GENERALIZACIÓN

- **B** hereda el comportamiento de **A** y puede extenderlo o sobrescribirlo. **B** también hereda todas las relaciones de **A** .
- **B** adopta la funcionalidad básica de **A** pero **decide por sí mismo qué parte de A se ejecuta o cambia**.
- **A** puede ser etiquetado como {abstract}
- **A** es una **generalización**, luego no tiene sentido como tal. Los casos de uso **B** lo **particularizan** y la acción cobra sentido.

4.RELACIONES ENTRE CASOS DE USO. RELACIÓN GENERALIZACIÓN

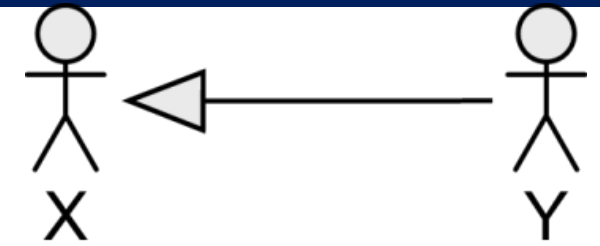
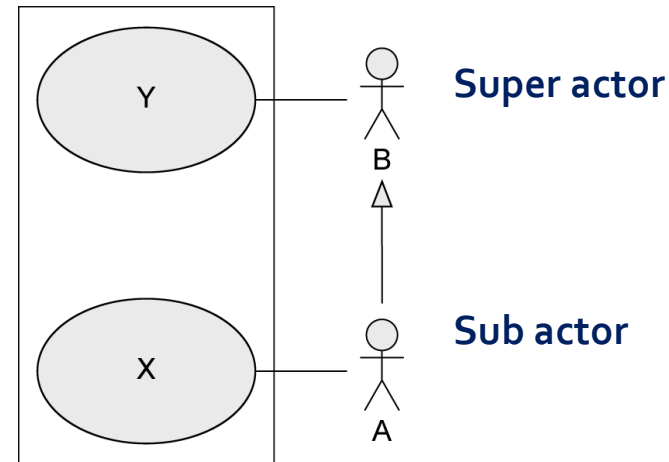
Cada médico y cada abogado es un profesional.
Cada profesional puede ejecutar el caso de uso "Tratar cliente"
Sólo los abogados pueden defender al cliente.
Sólo los médicos pueden operar



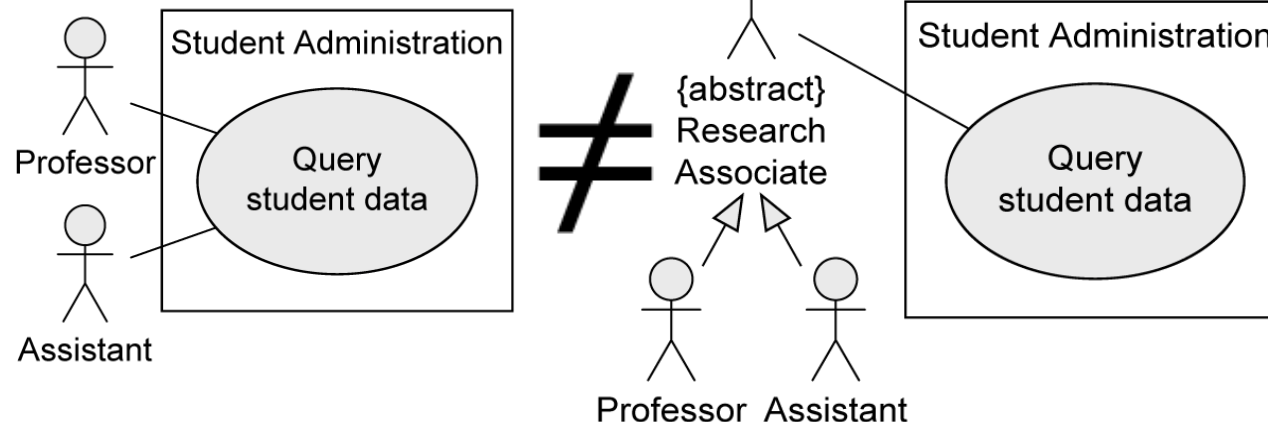
En la medida de lo posible debería evitarse, puede llevarnos a un detalle no apropiado para esta fase o puede confundir

4.RELACIONES ENTRE ACTORES. GENERALIZACIÓN

- Actor **A** hereda de actor **B** .
- **A** puede comunicarse con **X** y **Y** .
- **B** sólo puede comunicarse con **Y** .
- Se permite la **herencia múltiple** .



Para ejecutar la query necesitamos a Profesor y al Asistente.
Un profesor y un asistente participan en el caso de uso Query.



Los dos actores tienen un actor “padre” común que es el que ejecuta el caso de uso.

Un profesor o un asistente pueden participar en el caso de uso.

Si no hay instancia de un actor, puede utilizarse la etiqueta {abstract}

7.DESCRIPCIÓN DE CASOS DE USO

- **Enfoque estructurado (se puede especificar la información en un cuadro a modo textual)**
 - Nombre.
 - Breve descripción.
 - Requisito previo: requisito previo para una ejecución exitosa
 - Condición posterior: estado del sistema después de una ejecución exitosa
 - Situaciones de error: errores relevantes para el dominio del problema
 - Estado del sistema ante la ocurrencia de un error
 - Actores que se comunican con el caso de uso
 - Disparador: eventos que inician / inician el caso de uso
 - Proceso estándar: pasos individuales a seguir
 - Procesos alternativos: desviaciones del proceso estándar

[A. Cockburn: Writing Effective Use Cases, Addison Wesley, 2000]

7.DESCRIPCIÓN DE CASOS DE USO

- **Nombre:** Reserva de sala
- **Breve descripción:** un empleado reserva una sala de conferencias en la universidad para un evento.
- **Requisito:** El empleado está autorizado a reservar salas de conferencias.
- **Condición posterior:** Se reserva una sala de conferencias.
- **Situaciones de error:** No hay sala de conferencias libre.
- **Estado del sistema en caso de error:** El empleado no ha reservado una sala de conferencias.
- **Actores:** Empleado
- **Desencadenante:** el empleado requiere una sala de conferencias.



Descripción caso de uso → Requisito funcional

Name:	Reserve lecture hall
Short description:	An employee reserves a lecture hall at the university for an event.
Precondition:	The employee is authorized to reserve lecture halls. Employee is logged in to the system.
Postcondition:	A lecture hall is reserved.
Error situations:	There is no free lecture hall.
System state in the event of an error:	The employee has not reserved a lecture hall.
Actors:	Employee
Trigger:	Employee requires a lecture hall.
Standard process:	(1) Employee selects the lecture hall. (2) Employee selects the date. (3) System confirms that the lecture hall is free. (4) Employee confirms the reservation.
Alternative processes:	(3') Lecture hall is not free. (4') System proposes an alternative lecture hall. (5') Employee selects the alternative lecture hall and confirms the reservation.

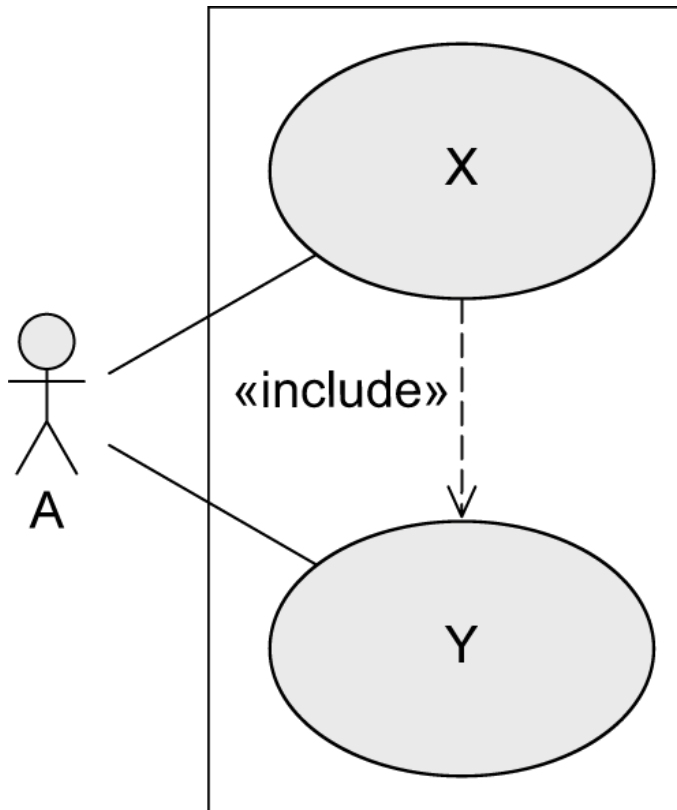
7.DESCRIPCIÓN DE CASOS DE USO

- **Proceso estándar** (pasos a seguir por la acción que representa el caso de uso):
 - (1) El empleado inicia sesión en el sistema.
 - (2) El empleado selecciona la sala de conferencias.
 - (3) El empleado selecciona la fecha.
 - (4) El sistema confirma que la sala de conferencias está libre.
 - (5) El empleado confirma la reserva.
- **Procesos alternativos:**
 - (4 ') La sala de conferencias no está libre.
 - (5 ') El sistema propone una sala de conferencias alternativa.
 - (6 ') El empleado selecciona la sala de conferencias alternativa y confirma la reserva.

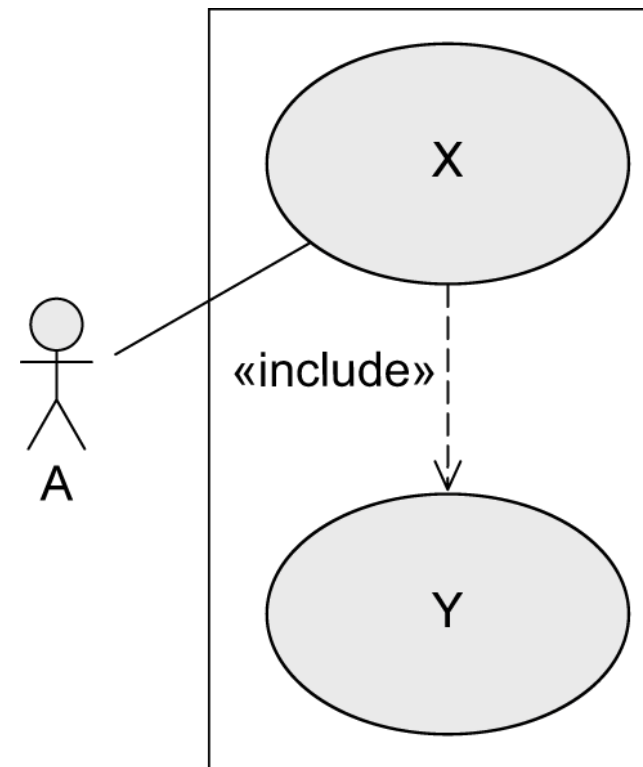
8. BUENAS PRÁCTICAS

«include»

UML standard



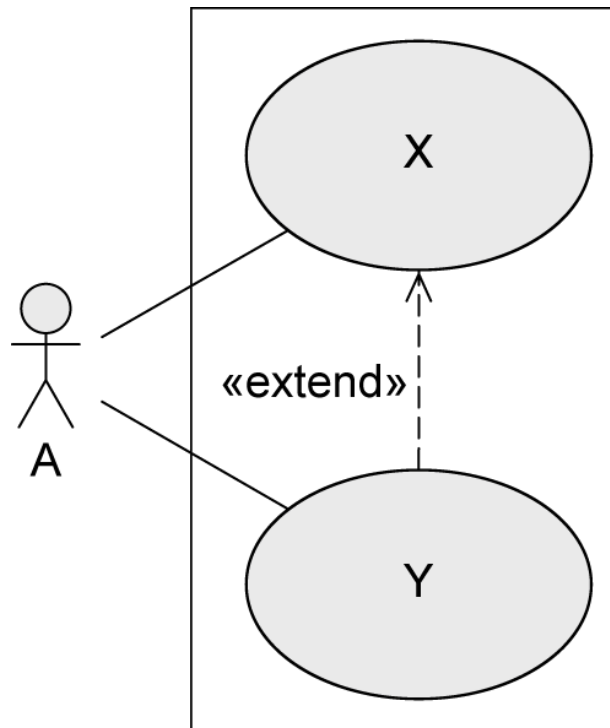
Buena práctica



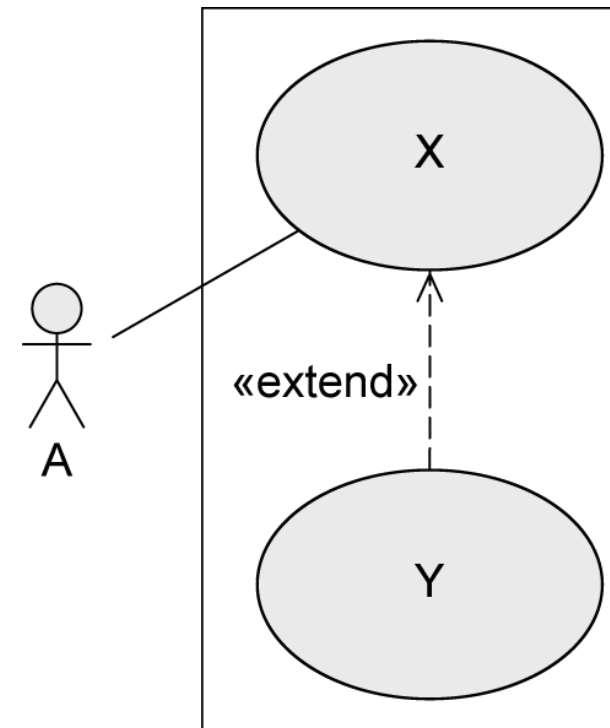
8. BUENAS PRÁCTICAS

«extend»

UML standard



Buena práctica



8. BUENAS PRÁCTICAS

IDENTIFICACIÓN DE ACTORES. Para identificarlos...

- ¿Quién participa en los principales casos de uso?
- ¿Quién necesita apoyo para su trabajo diario?
- ¿Quién es responsable de la administración del sistema?
- ¿Cuáles son los dispositivos / sistemas (software) externos con los que debe comunicarse el sistema?
- ¿Quién está interesado en los resultados del sistema?

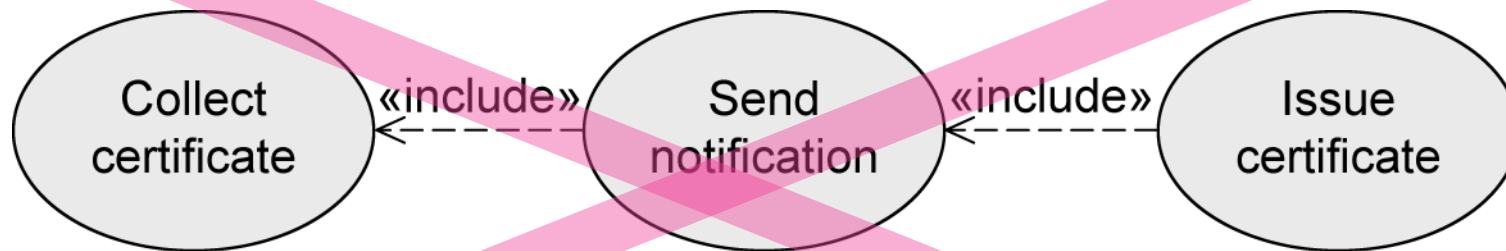
8. BUENAS PRÁCTICAS

IDENTIFICACIÓN DE CASOS DE USO

- ¿Cuáles son las principales tareas que debe realizar un actor?
- ¿Un actor quiere consultar o incluso modificar información contenida en el sistema?
- ¿Un actor desea informar al sistema sobre cambios en otros sistemas?
- ¿Se debe informar a un actor sobre eventos inesperados dentro del sistema?

9. ERRORES TÍPICOS

- Diseñar flujogramas.
- Los diagramas de casos de uso no diseñan procesos o flujos de trabajo.



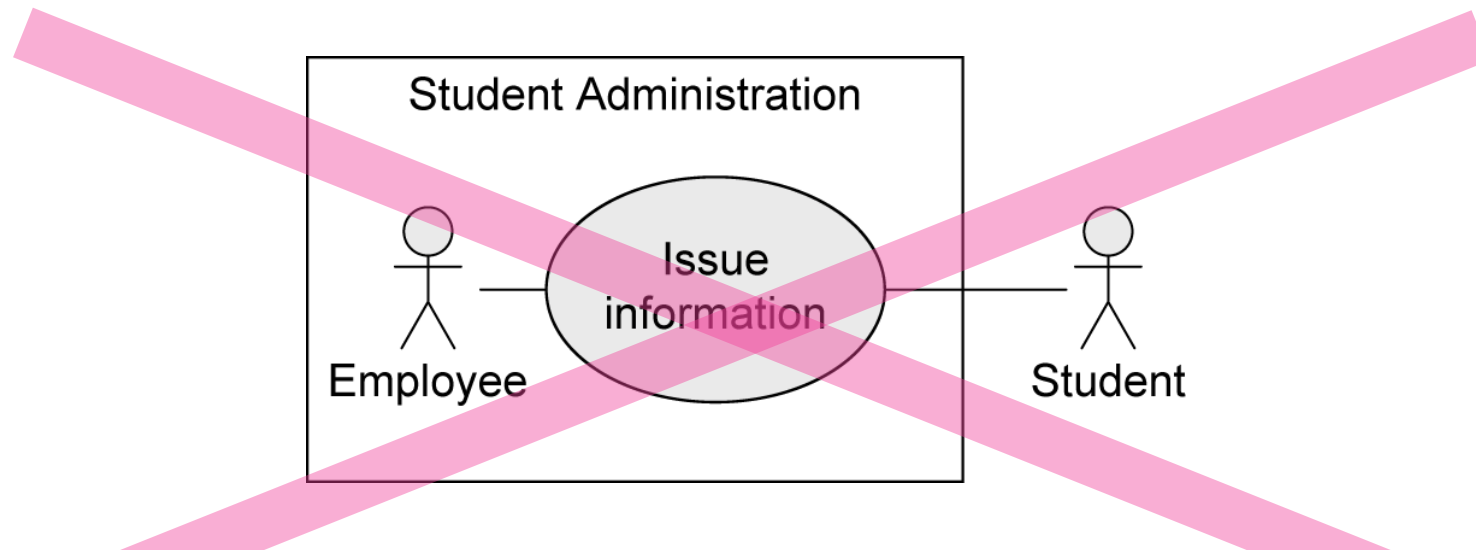
La funcionalidad que ofrece uno de estos casos de uso no es parte de la funcionalidad que ofrece otro caso de uso. Por eso, los casos de uso deben usarse independientemente unos de otros.

El ponente debe haber enviado el certificado de asistencia al curso a la Universidad. Es decir, debe emitir el certificado.

Elaborar certificado, Enviar notificación y Emitir certificado pueden conectarse cronológicamente siguiendo una línea de tiempo pero esto no se representa en los casos de uso.

9. ERRORES TÍPICOS

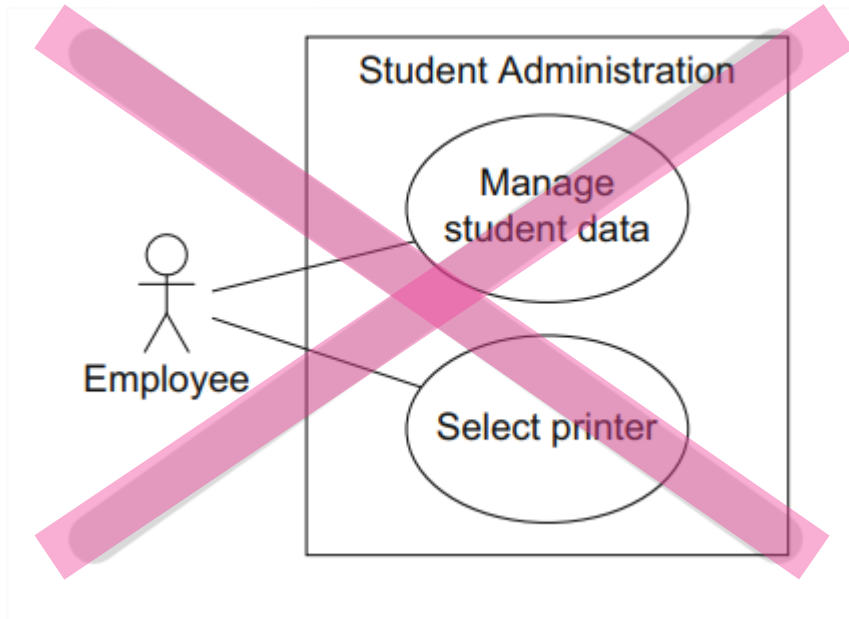
- **Establecer límites del sistema incorrectamente:**
- los actores no son parte del sistema y por esa razón ¡están ubicados fuera de sus límites!



Los actores siempre están fuera de los límites del sistema: si se van a ubicar dentro del sistema, son parte del mismo y por tanto no deben diseñarse como actores. Si no son parte del sistema y son necesarios para la ejecución de los casos de uso, deben ser representados como actores, fuera de los límites del sistema.

9. ERRORES TÍPICOS

- **Mezclar niveles de abstracción:**
- Al identificar casos de uso, siempre hay que asegurarse de que estén ubicados en el mismo nivel de abstracción.
- Hay que evitar representar un nivel más abstracto en el mismo diagrama.

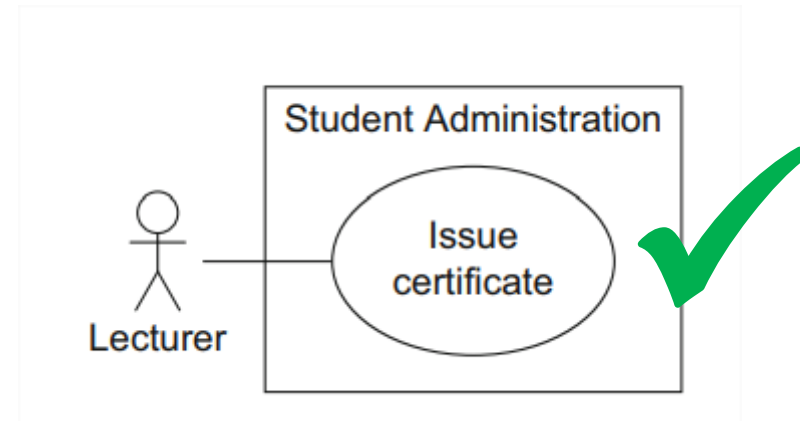
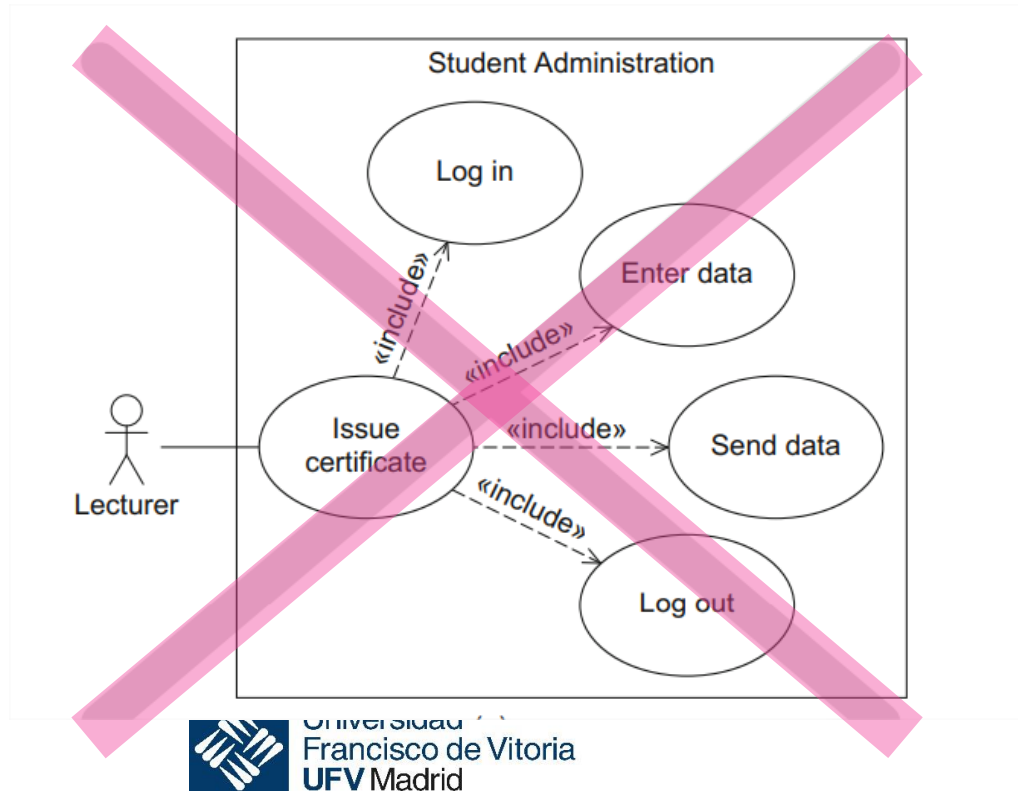


Primero creamos un diagrama de casos de uso con casos de uso que se basan en funcionalidades de gestión (manage student data) . Después, vamos explotando o detallando los casos de uso hasta llegar a los requisitos técnicos (seleccionando una impresora)

9. ERRORES TÍPICOS

Descomponer funcionalmente

- Los casos de uso, incluso los casos de uso incluidos o extendidos, siempre se pueden ejecutar de forma independiente. Si solo pueden ejecutarse dentro del alcance de otro caso de uso y no de forma independiente, no son casos de uso y deben no ser representado como tal.
- Su funcionalidad debe cubrirse en el descripción del caso de uso que los utiliza.



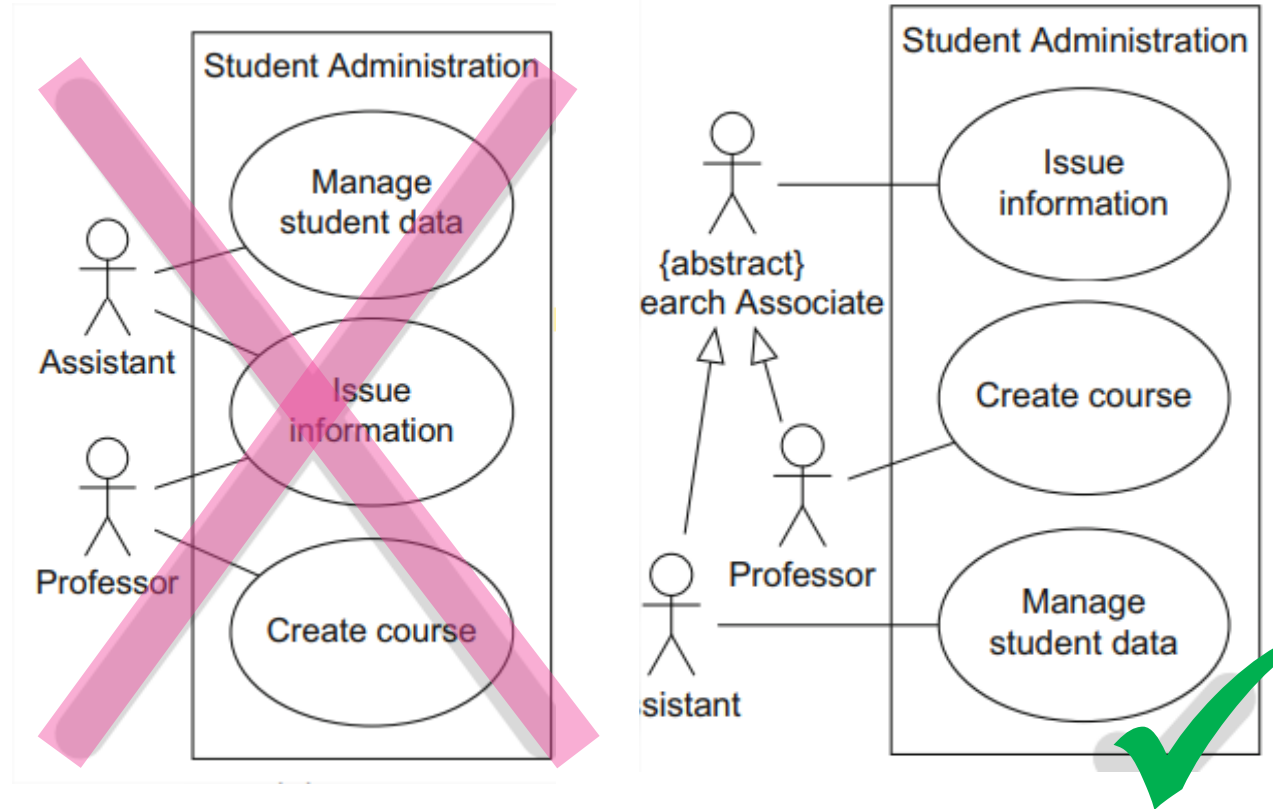
9. ERRORES TÍPICOS

Definir asociaciones incorrectas

- Si un caso de uso está asociado con dos actores, significa que ambos son necesarios para su ejecución (no uno u otro).

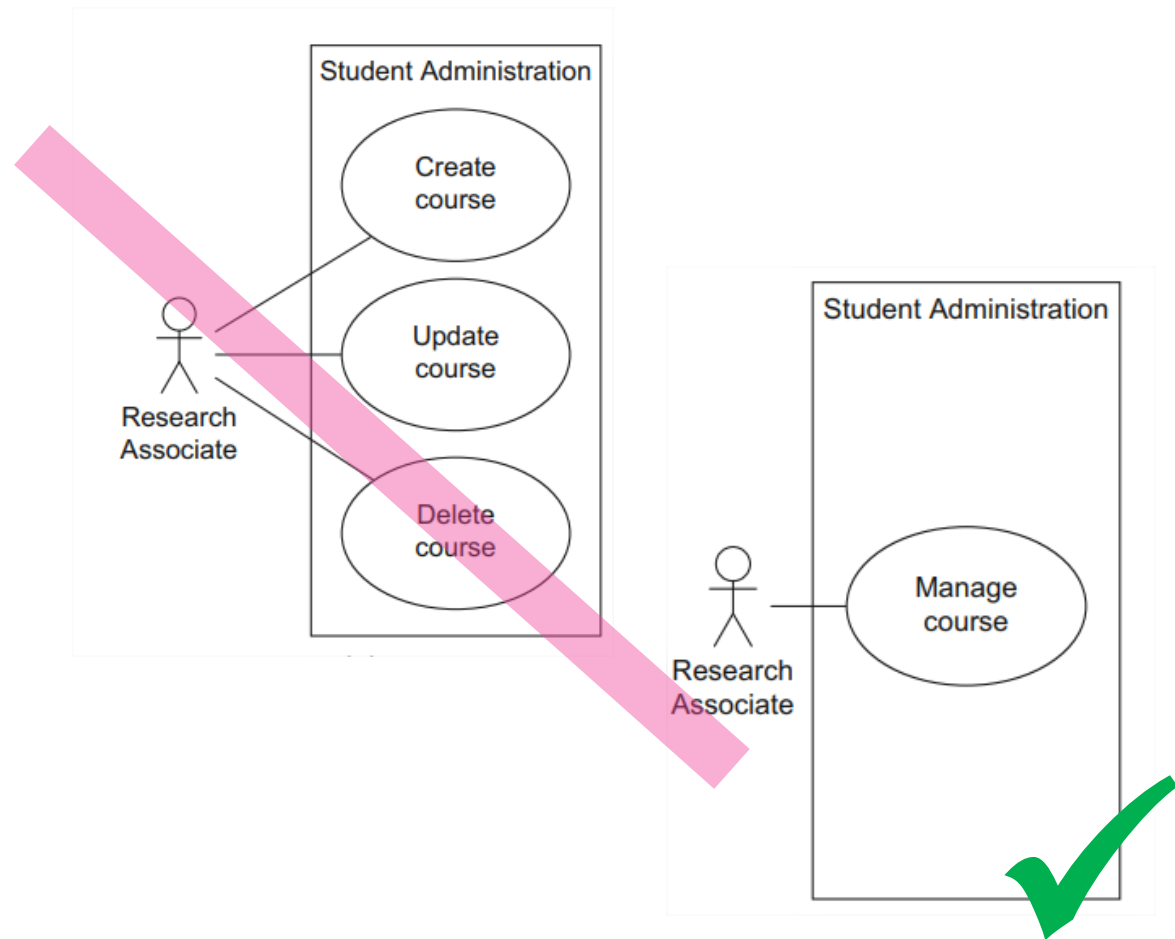
Los actores Asistente y Profesor están involucrados en la ejecución del caso de uso Generar información. Esto no tiene sentido.

Para resolver el error, representamos un nuevo actor abstracto "Investigador Asociado" del que heredan los dos actores Asistente y Profesor. Cada actor realizará la tarea que le corresponde y sólo él será necesario para ejecutar el caso.

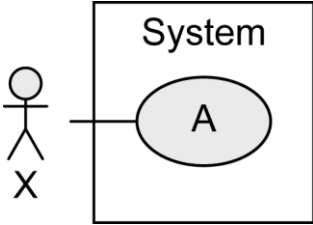
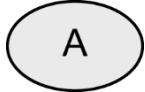
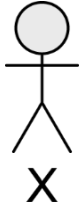


9. ERRORES TÍPICOS

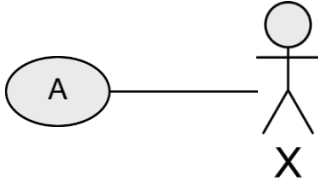
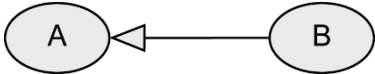
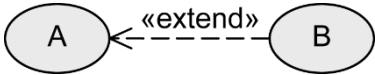
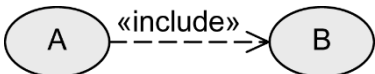
- **Diseñar casos de uso redundantes.**
- Al definir los casos de uso, es muy tentador crear un caso de uso separado para cada situación que necesitemos representar. Podemos definir casos de uso separados para crear, actualizar y eliminar cursos.
- Para resolverlo, tiene sentido agrupar casos de uso que tienen el mismo objetivo, es decir, la gestión de un curso. Los pasos individuales se especificarán en la descripción del proceso estándar.



10. ELEMENTOS DE NOTACIÓN UML

Nombre	UML	Descripción
Sistema		Expone el límite entre el sistema y los usuarios del mismo
Caso de uso		Parte o funcionalidad del sistema
Actor		Representa los usuarios del sistema

10. ELEMENTOS DE NOTACIÓN UML

Nombre	UML	Descripción
Asociación		Relación entre casos de uso y actores
Generalización		Relación de herencia entre actores o casos de uso
Relación Extend		<u>B extiende A</u> : uso opcional del caso de uso B sobre el caso de uso A
Relación Include		<u>A incluye B</u> : el caso de uso A hace un uso específico del caso de uso B