

# Tema 4: Casos de Uso y Diagramas de Casos de Uso

Análisis y Diseño del *Software*  
Grado en Ingeniería Informática

Departamento de Estadística, Informática y Matemáticas  
Universidad Pública de Navarra-*Nafarroako Unibertsitate Publikoa*

Curso académico 2022-2023



Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

## Objetivos del tema

- Identificación y modelado de requisitos del sistema a desarrollar.
- Descripción de casos de uso y diagramas de casos de uso.

- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio

- 1 **Requisitos**
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio

# Extracción y análisis de requisitos

- La especificación de los requisitos del sistema a desarrollar se desglosa en dos tareas.
- **Extracción de requisitos**
  - Es un proceso de indagación sobre las ideas que el cliente tiene del sistema a desarrollar.
  - Como resultado se obtiene una descripción del sistema en términos comprensibles tanto por el cliente como por el equipo de desarrollo.
- **Análisis de requisitos:** estudia los requisitos informales obtenidos del cliente para definir una especificación técnica del sistema a desarrollar (modelo para análisis).

# Extracción de requisitos

- La comprensión del cliente se limita al dominio del problema pero no incluye el dominio de la solución no se puede esperar que el cliente verbalice exactamente lo que quiere.
- Es necesaria una relación dialéctica entre cliente y equipo de desarrollo.
  - Ayuda al cliente a formular sus requisitos.
  - Ayuda al equipo a entender esos requisitos.

# Técnicas para extracción de requisitos

- **Cuestionarios:** pedir al usuario final que responda una lista de preguntas pre-formuladas.
- **Análisis de tareas:** observar a los usuarios finales en su entorno operativo.
- **Escenarios:** bosquejar el uso del sistema como una colección (incompleta) de interacciones concretas entre actores y sistema.
- **Casos de uso:** confeccionar abstracciones que agrupen diferentes escenarios en una misma clase.

- El énfasis del equipo de desarrollo durante la identificación de actores y la descripción de escenarios es comprender el dominio de la aplicación.
- Se busca una comprensión del ámbito del sistema y del soporte que hay que proporcionar a los procesos de trabajo de los usuarios.
- Una vez que se han descrito escenarios operativos e identificado los actores que intervienen, el equipo de desarrollo formaliza los escenarios en casos de uso.



- 1 Requisitos
- 2 Identificación de actores**
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio

# Concepto de actor

- Los actores son abstracciones de roles que las entidades que interactúan con el sistema adoptan en esa interacción.
- No necesariamente son humanos, pueden ser sistemas externos.
- Se conectan a los casos de uso mediante asociaciones.
- Una asociación entre un actor y un caso de uso indica una comunicación/interacción directa entre ambos

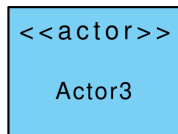
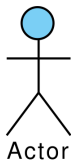
# Utilidad de los actores

- La identificación de actores tiene dos propósitos:
  - Definir las fronteras del sistema a desarrollar.
  - Encontrar las diferentes perspectivas desde las cuales el equipo de desarrollo necesita ver el sistema para poder construirlo.
- En las etapas iniciales de la extracción de requisitos resulta difícil distinguir entre actores y componentes del sistema (objetos de análisis, subsistemas).
  - **Ejemplo:** un subsistema de base de datos puede parecer a veces un actor, mientras que otras veces puede parecer parte del sistema a desarrollar.
- Los actores quedan completamente perfilados una vez se ha definido la frontera del sistema a desarrollar.
  - Los actores quedan fuera de la frontera.
  - Objetos y subsistemas quedan dentro de la frontera.

## Preguntas para identificar actores

- ¿A qué grupos de usuarios da soporte el sistema?
- ¿Qué grupos de usuarios emplean las principales funcionalidades del sistema?
- ¿Qué grupos de usuarios emplean funcionalidades secundarias como mantenimiento y administración?
- ¿Con qué *software* (interno o externo) debe interactuar el sistema?

# Actores en los diagramas



Actor4

- Tras la identificación de cada actor hay que determinar la funcionalidad que el sistema debe aportar a ese actor.
- Esa información se extrae definiendo escenarios y formalizando casos de uso.

- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios**
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio

## Concepto de escenario

- Un escenario es una descripción informal de una prestación simple del sistema desde el punto de vista de un actor concreto.
  - Los escenarios proporcionan un material común que es comprensible por desarrolladores y clientes.
- Los escenarios permiten a los desarrolladores ir dando forma a los requisitos.
  - En su descripción es posible identificar actores.
  - Su descripción se puede generalizar en casos de uso.
- Los escenarios se centran en situaciones específicas o eventos concretos entre el sistema a desarrollar y sus usuarios.
  - A diferencia de los casos de uso que proporcionan completas descripciones generalizadas de la interacción entre usuarios y sistema.



- **Ejemplo:** escenario de un sistema de gestión de alertas.

- Durante su patrulla en coche, los agentes *Agente A* y *Agente B* detectan humo saliendo de una nave de un polígono industrial.
- *Agente B* abre el mapa de su terminal y obtiene las coordenadas geográficas para ubicar la nave y las introduce en su aviso de alerta junto con una indicación de la posición de la columna de humo (esquina noroeste de la nave) y el nivel de emergencia. Añade al aviso una petición de unidades médicas móviles debido a que el área parece bastante ocupada. Confirma su entrada y espera confirmación del aviso.
- *Sargento Supervisor*, oficial de guardia en la central, recibe notificación de la alerta. Revisa la información recibida y decide atender la alerta. Seguidamente destina una unidad de bomberos y otra unidad médica al sitio del incidente, recibe estimación del tiempo de llegada y devuelve confirmación del aviso y tiempo estimado de llegada a *Agente B*.
- *Agente B* recibe la confirmación y el tiempo de llegada de las unidades médicas solicitadas.

- **Ejemplo:** observaciones sobre el escenario descrito.
  - Describe una situación concreta (instancia) de aviso de incidente con fuego.
  - No describe todas las posibles situaciones en las que se puede dar un aviso de incendio.
  - Presenta a ciertos participantes en el escenario (*Agente A*, *Agente B* y *Sargento Supervisor*) y describe sus roles en ese escenario.
  - Esos roles se pueden generalizar en los actores *FieldOfficer* y *Dispatcher*.

## Otros escenarios que surgen al interrogar al cliente

- Un accidente de automóvil sin heridos ocurre en la autopista. Agentes de tráfico realizan el atestado y dirigen el tráfico mientras retiran los coches accidentados.
- Un gato doméstico se ha encaramado en un árbol. Se destina un camión de bomberos a rescatar al gato. Dado que el incidente es de baja prioridad, la unidad destinada hace la ruta sin forzar el paso. Entretanto, el dueño del gato sube al árbol, cae y se rompe una pierna, necesitando una ambulancia para ser trasladado al hospital.
- Un terremoto sin precedentes daña edificios y carreteras, produciendo múltiples incidentes y activando un plan nacional de emergencias. Las carreteras dañadas impiden dar respuesta a los avisos.

# Diseño de escenarios

- No intenta abstraer ideas genéricas, sino que se centra en descripciones concretas de instancias particulares.
- Está orientada a describir comportamientos y funcionalidades; no está orientada a tecnología.
- No trata de ser completa y es bastante informal.
- Intenta dar forma al producto deseado, más allá de las ideas iniciales del cliente.

# Preguntas guía para preparar escenarios

- ¿Cuáles son las tareas que el actor quiere que el sistema realice?
- ¿A qué información accede el actor? ¿Quién genera esos datos? ¿Se pueden modificar o eliminar? ¿Por quién?
- ¿Sobre qué cambios externos necesita el actor informar al sistema? ¿Con qué frecuencia? ¿Por quién?
- ¿Sobre qué eventos necesita el sistema informar al actor? ¿Con qué latencia?

- La respuesta a esas preguntas se construye con diferentes fuentes:
  - Requisitos expuestos por el cliente
  - Necesidades de los usuarios del sistema
  - Observaciones de campo realizadas por el equipo de desarrollo

## Especificación orientada a escenarios

- El diseño de escenarios es un proceso iterativo.
- Cada escenario debe considerarse como un documento de trabajo susceptible de ser *iterado* (aumentado o reorganizado) cuando los requisitos, el criterio de aceptación del cliente o la situación de desarrollo cambie.
- La generalización de escenarios y la identificación de los casos de uso de más alto nivel que el sistema tiene que soportar habilitan a los desarrolladores a definir el alcance del sistema.

- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso**
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio



# Concepto de caso de uso

- Se emplean casos de uso para describir el **comportamiento externo** del sistema, es decir, el comportamiento desde el punto de vista de los actores.
- Un caso de uso especifica todos los posibles escenarios en los que los actores interactúan con una pieza de funcionalidad concreta del sistema.
  - En ese sentido, un escenario es una instancia de un caso de uso.
- Cada caso de uso es iniciado por un actor. Tras su arranque pueden intervenir otros actores e iniciar otros casos de uso.
- La descripción de un caso de uso consiste en una secuencia de eventos que muestran la interacción completa de los actores con el sistema cuando activan una funcionalidad concreta del mismo.
  - Esa interacción se denomina de forma genérica **comunicación**.

- **Ejemplo:** descripción del caso de uso *ReportEmergency*.

`https://miaulario.unavarra.es/access/content/group/  
2022_0_240501_1_G/teoria/ADS-4%3A%20casos%20de%20uso/  
reportemergency.pdf`

## Cómo nombrar un caso de uso

- El nombre debe ser una locución verbal que denote el objetivo del actor que inicia el caso de uso (qué es lo que actor iniciador trata de conseguir del sistema).
- Ese nombre debe reflejar la perspectiva del actor iniciador, no del sistema.
- **Ejemplo:** el nombre *ReportEmergency* indica que el actor iniciador *FieldOfficer* está intentando informar de una emergencia al sistema (y, por tanto, informar al actor *Dispatcher*).
  - Un nombre como *RecordEmergency* recogería la perspectiva del sistema.
  - Un nombre como *AttemptToReportEmergency* reflejaría la actividad desarrollada durante el caso de uso, no el objetivo.

- Para nombrar a los actores se deben usar locuciones nominales.
- **Ejemplo:** los actores participantes del escenario *ReportEmergency*
  - Los nombres *Agente A* y *Agente B* se generalizan como *FieldOfficer*; el rol *FieldOfficer* es el actor iniciador.
  - El nombre *Sargento Supervisor* se convierte en el actor *Dispatcher*.

# Partes de un caso de uso

- Partes obligatorias
  - Actores participantes
  - Descripción de la condición de entrada
  - Descripción del flujo de eventos
  - Descripción de la condición de salida
  - Descripción de excepciones (describe qué ocurre cuando las cosas van mal)
  - Descripción de requisitos no funcionales
- Se pueden incluir otras secciones para describir flujos excepcionales de eventos, reglas e invariantes que el caso de uso debe cumplir durante el flujo de eventos.

## Descripción del flujo de eventos

- Se usa lenguaje natural informal para describir la secuencia de pasos desde que se inicia el caso hasta que se completa.
  - **Ejemplo:** el caso de uso *ReportEmergency* debe describir todas las interacciones entre actores y sistema desde el inicio del aviso hasta la recepción de la confirmación.
- La longitud de la descripción no debiera exceder de dos o tres páginas.
- Si fuera mayor, se debe pensar en descomponer el caso en casos más simples relacionados mediante *include* y *extend*.

## Exigencias de la descripción del flujo de eventos

- Debe mostrar la relación causal entre pasos consecutivos.
- Debe mostrar claramente la frontera del sistema: se debe distinguir entre pasos completados por los actores y los completados por el sistema.
- Cada paso se describe en voz activa (no en voz pasiva) para explicitar el sujeto (actor o sistema) que completa el paso.
- Las excepciones deben ser descritas separadamente.
- No debe describir ni mencionar elementos de la interfaz de usuario del sistema (menú, botones, orden concreta, etc)

- **Ejemplo:** caso de su mal especificado.

`https://miaulario.unavarra.es/access/content/group/  
2022_0_240501_1_G/teoria/ADS-4%3A%20casos%20de%20uso/  
badreportemergency.pdf`

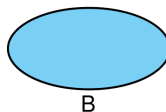
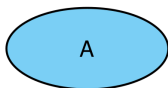


# Pautas para el modelado de requisitos de un sistema

- Establecer el contexto del sistema.
- Identificar los actores y organizarlos en jerarquías.
  - Grupos que requieren ayuda del sistema para completar sus tareas.
  - Grupos necesarios para ejecutar las funciones del sistema.
  - Grupos que interactúan con *hardware* externo o con otros sistemas de *software*.
  - Grupos que realizan funciones secundarias de administración mantenimiento.
- Considerar el comportamiento que cada actor espera del sistema o requiere que el sistema proporcione.
- Nombrar comportamientos comunes mediante casos de uso.
- Factorizar el comportamiento común y el comportamiento variante de cada caso.
- Modelar en un diagrama de casos de uso; incluir notas para requisitos no funcionales.

# Casos de uso en los diagramas

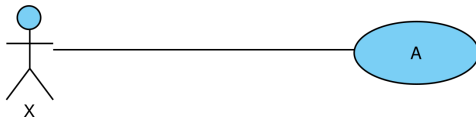
- Los casos de uso siempre están dentro de un recuadro que identifica y delimita el sistema
  - Los actores están fuera del sistema



- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama**
  - Actores y casos de uso
  - Entre actores
  - Entre casos de uso
  - Ejemplo relaciones
- 6 Errores comunes
- 7 Ejercicio

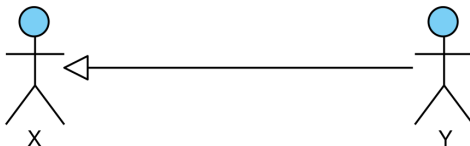
# Asociación

- Línea continua simple entre actores y casos de uso
- Los actores participan en el caso de uso
- Todos los actores deben estar relacionados con al menos un caso de uso
- Es posible mostrar multiplicidad
  - Mas de un actor con el mismo rol participa



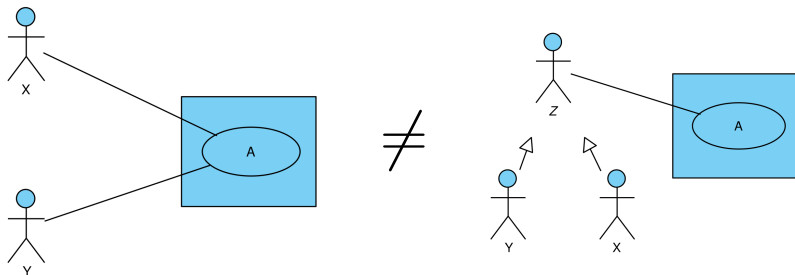
# Generalización/Herencia

- Se usa para generalizar y definir comportamientos comunes
- Y hereda de X
  - Y también participa en todas las asociaciones de X
- Si no hay instancias de un actor se le puede etiquetar como {abstract}



Entre actores

# Ejemplo generalización actores



# Relaciones entre casos de uso

- Ayudan a reducir tanto la complejidad como la redundancia en la descripción de los requisitos del sistema.
- Tres tipos de relaciones:
  - **Inclusión** para descomponer casos de uso largos en otros más cortos.
  - **Extensión** para tratar flujos excepcionales debidos a eventos alternativos.
  - **Generalización** para refinar casos de uso abstractos.

# Relación <<include>>

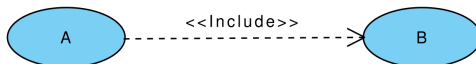
- Un caso de uso base incorpora expresamente el comportamiento de otro caso de uso.
  - El flujo de eventos debe indicar el evento concreto que arranca el caso incluido.
  - En el diagrama es corriente incluir alguna indicación del punto de inclusión.
- Dos posibilidades de empleo:
  - Descomposición funcional de comportamientos complejos (reduce complejidad de descripción)
  - Reutilización de funcionalidades ya definidas (elimina redundancias)



Entre casos de uso

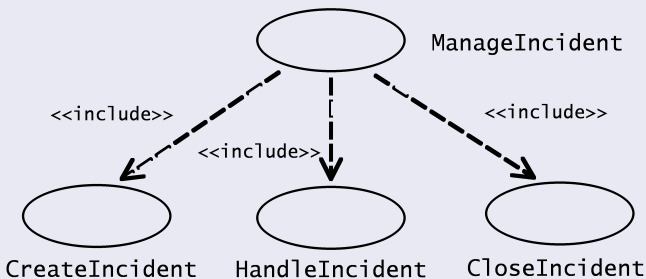
# Significado <<include>>

- A incluye a B
- El comportamiento de B esta integrado en A
- Cuando A se ejecuta, B siempre se ejecutara



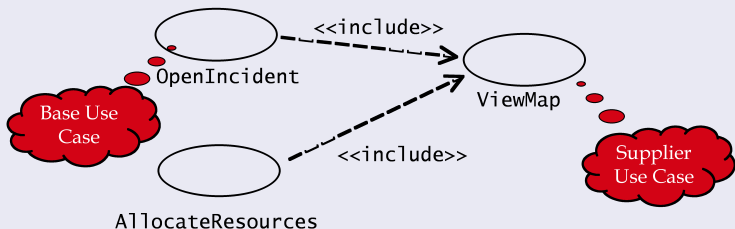
# Descomposición funcional

- **Problema:** el comportamiento de interacción de un caso de uso es demasiado largo o complejo.
- **Solución:** describir esa interacción como agregación de un conjunto de interacciones más simples definidas como casos de uso.
- **Ejemplo:** el caso de uso que describe la gestión de un incidente se descompone en partes.



# Reutilización de funcionalidad existente

- **Problema:** existen redundancias (partes idénticas) en el flujo de eventos de dos o más casos de uso. ¿Cómo reutilizar flujos de eventos en lugar de duplicarlos?
- **Solución:** la asociación de inclusión entre el caso de uso *A* y el caso de uso *B* indica que una instancia del caso *A* presenta todo el comportamiento del caso de uso *B*.
- **Ejemplo:** el caso de uso *ViewMap* describe una funcionalidad que se incluye en los casos *OpenIncident* y *AllocateResources*.

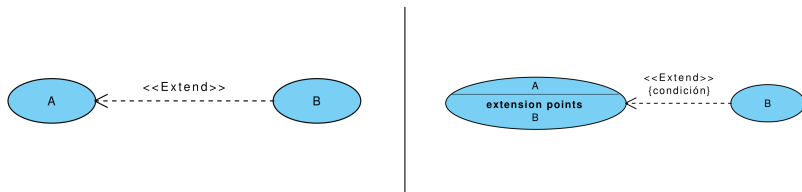


# Relación <<extend>>

- Un caso de uso *A* extiende otro caso de uso base *B* si, bajo ciertas condiciones, este último incluye el comportamiento del caso *A*.
- Permite separar flujos de eventos corrientes de flujos de eventos que ocurren bajo circunstancias específicas.
  - Como ejemplos de extensiones puede pensarse en la invocación de ayuda en línea, la cancelación de una transacción, o la forma de afrontar un fallo de conexión.
- En la descripción del flujo de eventos del caso extendido, el evento que arranca el caso extensión se indica como precondition del mismo.

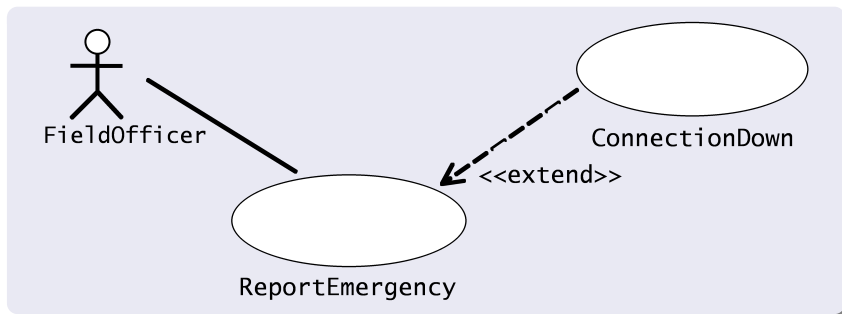
# Significado <<extend>>

- B extiende a A
- A puede usar el comportamiento de B
- Cuando A se ejecuta, B es opcional
- Se puede especificar una condición en el enlace
  - La condición también se puede mostrar con una nota
- En A se puede especificar un *extension point* que define cuando se introduce B



## Ejemplo <<extend>>

- **Ejemplo:** la comunicación entre patrulla y central se corta y el aviso de la patrulla no llega a la central; desde central deben notificar esa situación a la patrulla y las medidas para proseguir.



# Heurística para decidir entre inclusión y extensión

- Emplea <<include>> para comportamiento que se repite en dos o más casos de uso.
- Emplea <<extend>> para comportamiento excepcional, opcional o infrecuente.
- Aplica las reglas anteriores con criterio para no sobreestructurar el modelo.

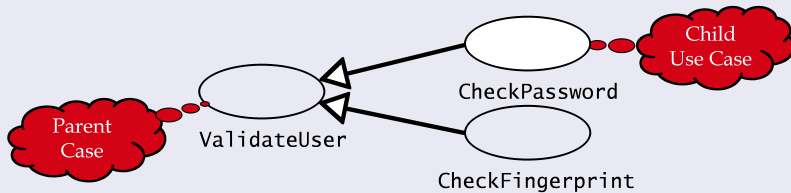
# Ejemplo

- **Ejemplo:** incorporación del caso de uso *ConnectionDown* al caso *ReportEmergency*.
  - Mediante inclusión:  
`https://miaulario.unavarra.es/access/content/group/2022_0_240501_1_G/teoria/ADS-4%3A%20casos%20de%20uso/includedown.pdf`
  - Mediante extensión:  
`https://miaulario.unavarra.es/access/content/group/2022_0_240501_1_G/teoria/ADS-4%3A%20casos%20de%20uso/extenddown.pdf`



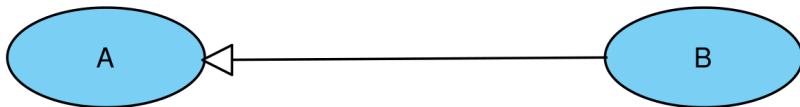
# Generalización/Herencia

- Un caso de uso incorpora el comportamiento de otro caso de uso, que concreta o modifica.
- **Ejemplo:** el caso de uso *ValidateUser* es responsable de autenticar (verificar identidad) al usuario. Como requisito del cliente, el sistema a desarrollar debería permitir autenticación mediante contraseña y/o mediante huella dactilar.

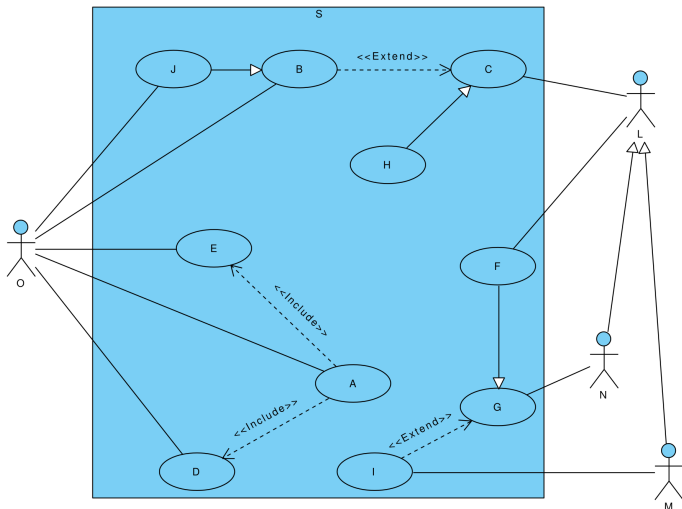


# Significado generalización

- A generaliza a B
- B hereda el comportamiento de A y sus relaciones
  - Asociaciones, <<include>> o <<extend>>
  - B decide que y como se ejecuta



# Ejemplo final

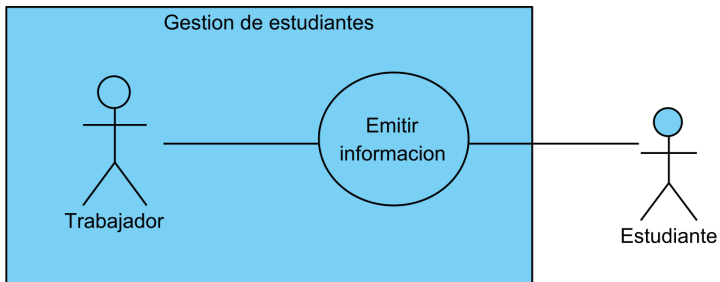


- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes**
- 7 Ejercicio

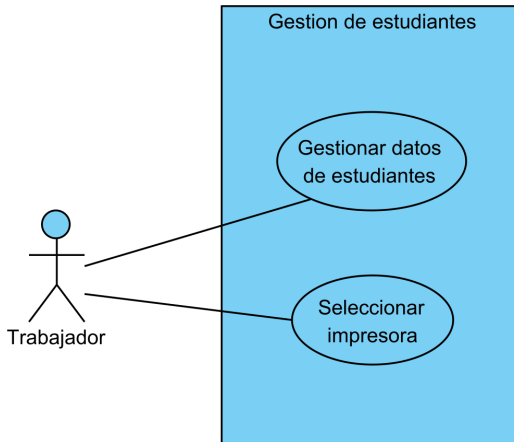
# Modelado de procesos



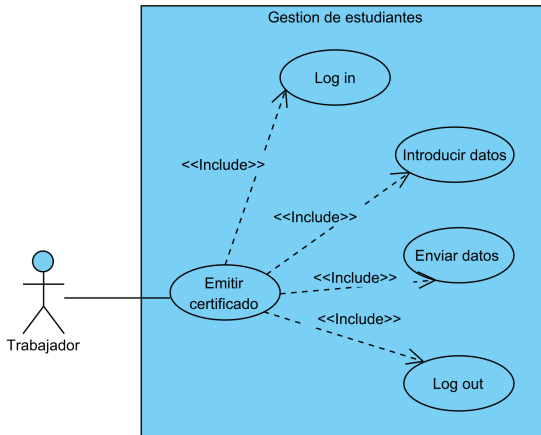
# Limites incorrectos del sistema



# Mezclar niveles de abstracción

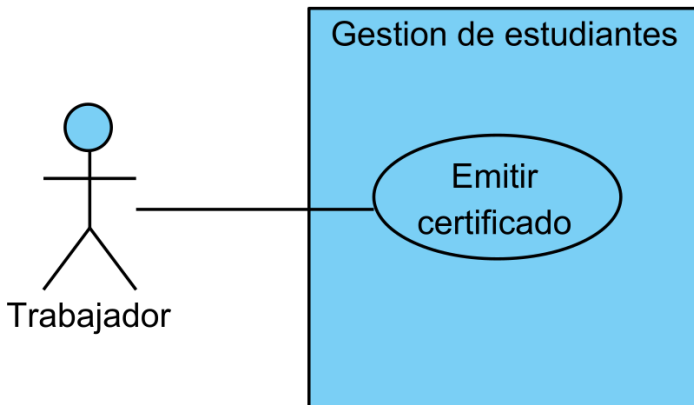


# Descomposición de las funcionalidades

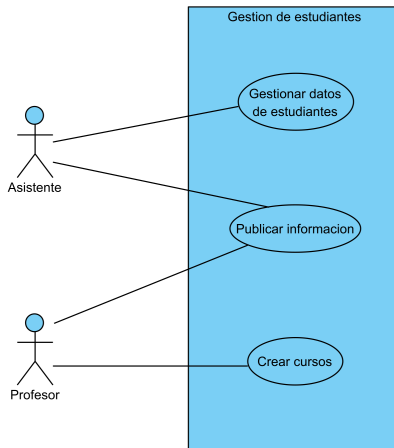




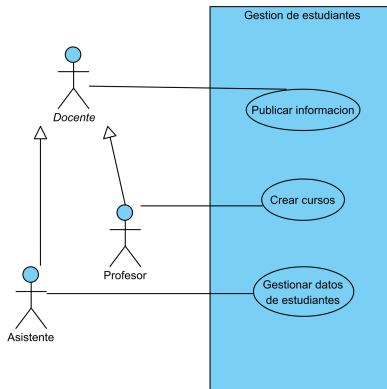
# Descomposición de las funcionalidades (correcto)



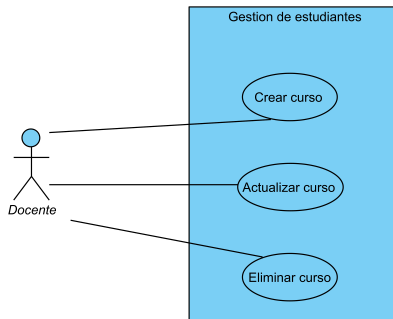
# Asociaciones incorrectas



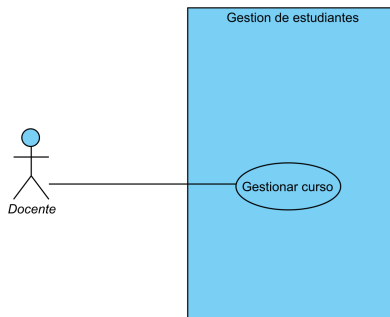
# Asociaciones incorrectas (correcto)



# Casos de uso redundantes



# Casos de uso redundantes (correcto)



- 1 Requisitos
- 2 Identificación de actores
- 3 Identificación de escenarios
- 4 Identificación de casos de uso
- 5 Relaciones entre elementos del diagrama
- 6 Errores comunes
- 7 Ejercicio**

# Gestión estudiantes

## Diseñar una oficina de apoyo a estudiantes de una universidad

- Los estudiantes se pueden matricular en cursos, inscribirse o abandonar. La inscripción implica matricularse.
- Los estudiantes reciben expedientes impresos por un empleado. Los profesores envían las notas a la oficina. Un sistema de notificación informa a los estudiantes automáticamente de la disponibilidad de sus expedientes.
- Hay dos tipos de empleados: (todos los empleados pueden dar información)
  - Se encargan exclusivamente de la gestión de tramites de estudiantes (matricula, inscripción, abandono)
  - Realizan el resto de tareas (administrativos)
- Los administrativos entregan expedientes cuando los estudiantes van a recogerlos. También pueden crear cursos, cuando realizan esta tarea tienen la opción de reservar aulas.

# Solución

