Tipos de relaciones en diagramas de Casos de Uso

Hace unas semanas hablamos en este blog de una duda que frecuentemente me plantean los alumnos a la hora de modelar diagramas de clases con UML. En esta ocasión vamos a intentar resolver otra consulta que me suelen plantear a la hora de modelar diagramas de casos de uso, se trata de los diferentes tipos de **relaciones** que podemos encontrar entre **casos de uso**.

Antes de nada diremos que **los casos de uso** fueron ideados por Jacobson a principios de los noventa y están inspirados en el concepto de escenario, el cual ya había sido utilizado para describir procesos. Los casos de uso especifican un comportamiento deseado del sistema, representan **requisitos funcionales** del mismo. Es importante resaltar que describen qué hace el sistema, no cómo lo hace. UML nos dice que: “Un caso de uso especifica un conjunto de secuencias de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un particular actor”. Los casos de uso nos sirven de base para elaborar los aspectos funcionales del pliego de condiciones y nos dan soporte en las etapas de modelado, desarrollo y validación de software.

# Elementos de un caso de uso

**Conjunto de secuencias de acciones:** cada secuencia representa un posible comportamiento del sistema.

**Actores:** se tratan de los roles que pueden jugar los agentes que interactúan con el sistema. Los roles son jugados por personas, dispositivos, u otros sistemas. Podríamos distinguir entre actores primarios, para los cuales el objetivo del caso de uso es esencial y actores secundarios, que interactúan con el caso de uso, pero cuyo objetivo no es esencial.

**Variantes:** son versiones especializadas, un caso de uso que extiende a otro o un caso de uso que incluye a otro.

Como veremos a continuación, en los diagramas de casos de uso se muestran:

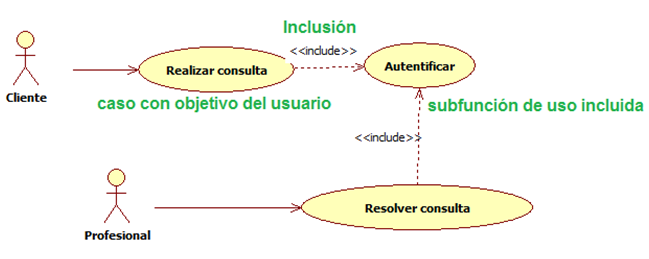
* casos de uso (representados en forma de elipses),
* actores (en forma de personajes)
* y relaciones (en forma de líneas y/o flechas).

UML define cuatro tipos de **relaciones en los diagramas de casos de uso**:

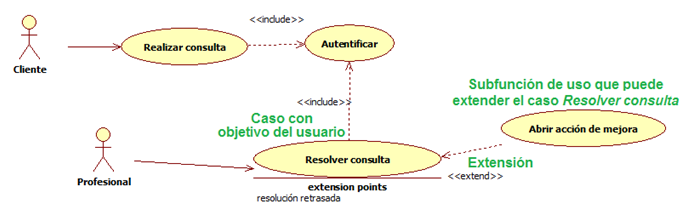
* **Comunicación:** Relación (asociación) entre un actor y un caso de uso. El estereotipo de la relación de comunicación es: <<communicate>> aunque generalmente no se estipula ningún nombre, como podemos apreciar en el siguiente *ejemplo de comunicación*:



* **Inclusión:** Un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia. La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso, también puede utilizarse para estructurar un caso de uso describiendo sus sub-funciones. El caso de uso incluido existe únicamente con ese propósito, ya que no responde a un objetivo de un actor. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<include>>. Algunos casos de uso típicos de inclusión son: comprobar, verificar, buscar, validar, autentificar o *login*. En principio, no deberíamos abusar de este tipo de relación, para no hacer una descomposición funcional del sistema. A partir de UML 1.3 la relación <<include>> reemplazó al denominado <<uses>>. Veamos un *ejemplo de inclusión* entre casos de uso:

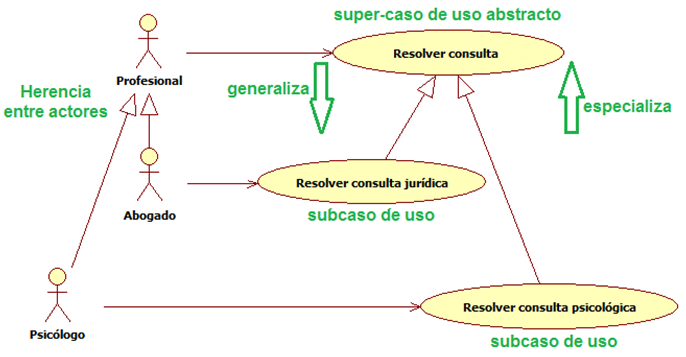


* **Extensión:** Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso. En el caso de uso base, la extensión se hace en una serie de puntos concretos y previstos en el momento del diseño, llamados **puntos de extensión**, los cuáles no son parte del flujo principal. La relación de extensión sirve para modelar: la parte opcional del sistema, un sub-flujo que sólo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado. Este tipo de relación produce confusión y no debería utilizarse en exceso. Conviene su uso sólo para insertar un nuevo comportamiento no previsto en un caso de uso existente. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<extend>>. Veamos un *ejemplo de extensión:*



En este ejemplo usamos la relación de extensión entre los casos de uso *Abrir acción de mejora* y *Resolver consulta.* En este caso tendremos el punto de extensión “resolución retrasada” (en el caso de uso *Resolver consulta*) debido a que cuando haya pasado un tiempo estipulado por la organización (por ejemplo 3 días laborales) se abrirá una acción de mejora para dejar constancia del retraso y realizar posteriormente las acciones pertinentes, de ahí que digamos que el caso de uso *Abrir acción de mejora* es una sub-función de uso que puede extender al caso de uso *Resolver consulta.*

* **Especialización y generalización de los casos de uso**: Un caso de uso (sub-caso) hereda el comportamiento y significado de otro, es decir las relaciones de comunicación, inclusión y extensión del súper-caso de uso. En muchas ocasiones este súper-caso de uso es abstracto y corresponde a un comportamiento parcial completado en el sub-caso de uso. O dicho de otra manera, Los casos de uso “hijo” son una especialización del caso de uso “padre”. En la medida de lo posible debería evitarse puesto que produce cierta confusión en algunas ocasiones. Veamos un *ejemplo de generalización*:



Como podemos ver en este último ejemplo también pueden existir vínculos de generalización o **herencia entre actores**. Los actores especializados (Abogado y Psicólogo) heredan los casos de uso del actor general (Profesional). La punta de flecha apunta al actor más general. Hay que reseñar que los actores especializados pueden tener otros casos de uso propios que no estarán disponibles para los demás actores. Este tipo de herencia entre actores sí que se usa frecuentemente puesto que nos simplifica considerablemente el diagrama, nos ahorra un número importante de relaciones de comunicación entre actores y casos de uso y nos sirve para esclarecer visualmente la jerarquía entre actores del sistema.