

DIAGRAMAS DE CASOS DE USO

- Se emplean para obtener y documentar los requisitos funcionales del sistema.
- Se modela un caso de uso por cada proceso/acción que pueda llevar a cabo el sistema.
- Además de realizar el diagrama, se documenta cada caso de uso, indicando: precondiciones, postcondiciones, acciones que lleva a cabo, el actor que inicia el caso de uso, etc.

Descripción de casos de uso	
Nombre:	
Actor principal:	
Actores secundarios:	
Precondiciones:	
Postcondiciones:	
Función:	
Descripción	

- **Los elementos que componen los diagramas de casos de uso son:**
 - EL sistema.
 - Los actores.
 - Los casos de uso.
 - Las relaciones:
 - Asociación.
 - Inclusión.
 - Extensión.
 - Generalización.
- Relación de asociación: Se utiliza para representar la interacción entre los actores y los casos de uso del sistema y se representan con una línea recta que une a ambos.
- Relación de inclusión: Se utiliza cuando el comportamiento de un caso de uso se incluye dentro del comportamiento de otro. Se representa con una flecha de trazo discontinuo desde el caso de uso que incluye hasta el caso de uso incluido, con la etiqueta <<include>>. Los casos de uso incluidos se pueden compartir, así evitamos repetirlos. También se pueden usar para estructurar el diagrama en varios niveles de detalle, aunque no conviene abusar de ellos.

- Relación de extensión: se utiliza cuando un caso además aporta un comportamiento adicional en determinadas circunstancias o cuando se cumple cierta condición. Se representa con una flecha de trazo discontinuo que apunta al caso que queremos extender, con la etiqueta <<extend>>.
- Relación de generalización: se utiliza para expresar que un caso de uso especializado es una forma particular de conseguir los objetivos de otro caso de uso más general. Se representa mediante una flecha de trazo continuo acabada en punta triangular hueca que apunta hacia el caso más general.

¿Cómo distinguir las relaciones entre casos?

A veces no está del todo claro cuando debemos de utilizar la extensión, la generalización o la inclusión. En caso de duda, podemos guiarnos por las siguientes reglas:

- La inclusión equivale a “copiar y pegar” un caso de uso dentro de otro. Si imaginamos los casos de uso como líneas de un programa, sería equivalente a incluir el código de un caso dentro del otro.
- La extensión se utiliza cuando un caso añade funcionalidad a otro dependiendo de alguna condición. Si imaginamos los casos de uso como líneas de un programa, equivaldría a incluir una condición if(), de forma que si se cumple la condición, se ejecuta el caso de uso B a mayores del A.
- La generalización sirve para indicar que varios casos de uso tienen el mismo comportamiento, pero lo llevan a cabo de distinto modo. Equivale a la herencia en la programación orientada a objetos (OOP), donde las clases hijas pueden reescribir parte del código de la clase padre.

Recomendaciones:

- Los casos de uso describen las iteraciones más importantes con el sistema, no su funcionamiento interno.
- Conviene mantener los casos de uso lo más simples posibles. Todos los implicados en el proyecto deben de ser capaces de entender el diagrama de casos de uso. Si se complica en exceso, pierde su utilidad.
- Hay que tener en cuenta que los diagramas de casos de uso se elaboran durante la fase de especificación de requisitos y que estos son cambiantes por naturaleza. Por ello conviene no ahondar demasiado de detalles, ni descomponer cada caso de uso en sub casos, ya que, si estos cambian, habremos malgastado el tiempo.

- En cualquier caso, es preferible acompañar al diagrama de una buena descripción del caso de uso.
- En caso de no estar implementando un sistema de casos de uso real, sino que se intente resolver un ejercicio, ha de intentar ceñirse al enunciado en la medida de lo posible.