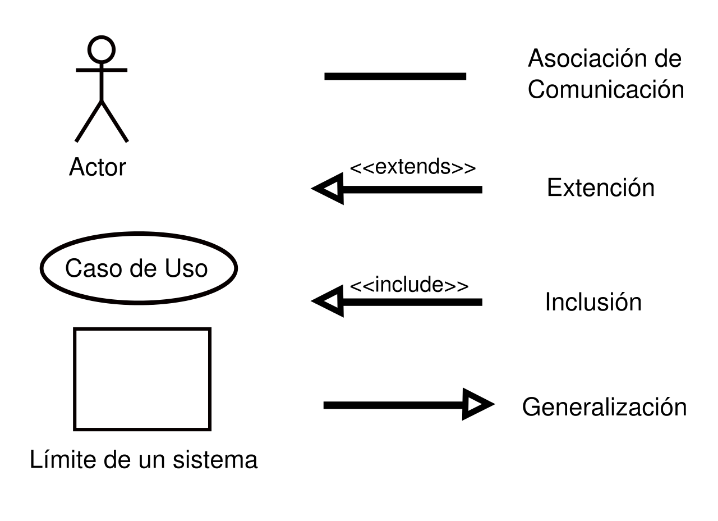
Caso de Uso

Un **caso de uso** es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. Los personajes o entidades que participarán en un diagrama de caso de uso se denominan actores. En el contexto de [ingeniería del software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_del_software), un diagrama de caso de uso representa a un sistema o subsistema como un conjunto de interacciones que se desarrollarán entre casos de uso y entre estos y sus actores en respuesta a un evento que inicia un actor principal. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

Su uso es común para la captura de [requisitos funcionales](https://es.wikipedia.org/wiki/Requisito_funcional), especialmente con el paradigma de la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), donde se originaron, si bien puede utilizarse con resultados igualmente satisfactorios con otros [paradigmas de programación](https://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n).



# Un poco de historia en la programación

En 1986, [Ivar Jacobson](https://es.wikipedia.org/wiki/Ivar_Jacobson), importante contribuyente al desarrollo de los modelos de [UML](https://es.wikipedia.org/wiki/UML) y [proceso unificado](https://es.wikipedia.org/wiki/Proceso_unificado), creó el concepto de caso de uso. ​ Se han realizado muchas mejoras al concepto que se estableció entonces, pero probablemente la más influyente y significativa, en términos de definición del término caso de uso, fue la de [Alistair Cockburn](https://es.wikipedia.org/w/index.php?title=Alistair_Cockburn&action=edit&redlink=1" \o "Alistair Cockburn (aún no redactado)) en el libro [*Escribir casos de uso efectivos*](http://books.google.com/books/about/Writing_effective_use_cases.html?id=VKJQAAAAMAAJ) publicado en el año 2000.

Durante los [años 1990](https://es.wikipedia.org/wiki/A%C3%B1os_1990) los casos de uso se convirtieron en una de las prácticas más comunes para la captura de [requisitos funcionales](https://es.wikipedia.org/wiki/Requisito_funcional), especialmente con el desarrollo del paradigma de la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), donde se originaron, si bien puede utilizarse con resultados igualmente satisfactorios con otros paradigmas de programación.

# Definiciones básicas

## Actores

Se le llama actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo.

En el caso de los seres humanos se pueden ver a los actores como definiciones de rol por lo que un mismo individuo puede corresponder a uno o más Actores. Suele suceder sin embargo, que es el sistema quien va a tener interés en el tiempo. Es frecuente encontrar que nuestros sistemas deben efectuar operaciones automáticas en determinados momentos; y siendo esto un requisito funcional obvio, resulta de interés desarrollar alguna forma de capturar dicho requisito en el modelo de caso de uso final.

# Tipos de relaciones

* Comunica (*<<communicates>>*): Relación (asociación) entre un actor y un caso de uso que denota la participación del actor en dicho caso de uso.
* Usa (*<<uses>>*) (o *<<include>>* en la nueva versión de UML): Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro.
* Generalización (sin estereotipo) (En UML 1.3) Indica que un caso de uso es una variante de otro. El caso de uso especializado puede variar cualquier aspecto del caso de uso base.
* Extiende (*<<extend>>, <<extiende>>*)(En UML 1.3): Es un estereotipo de dependencia. Ofrece una forma de extensión más controlada que la relación de generalización. El caso de uso base declara un conjunto de puntos de extensión, El caso de uso especializado solo puede alterar el comportamiento de los puntos de extensión marcados.

Se utiliza una relación de tipo <<extends>> entre casos de uso cuando nos encontramos con un caso de uso similar a otro pero que hace algo más que este (variante). En cambio, utilizaremos una relación tipo <<uses>> cuando nos encontramos con una parte de comportamiento similar en dos casos de uso y no queremos repetir la descripción de dicho comportamiento común.

En una relación <<extends>>, un actor que lleve a cabo el caso de uso base puede realizar o no sus extensiones. Mientras, en una relación <<include>> el actor que realiza el caso de uso base también realiza el caso de uso incluido.

En general utilizaremos <<extends>> cuando se presenta una variación del comportamiento normal, e <<include>> cuando se repite un comportamiento en dos casos de uso y queremos evitar dicha repetición.

Por último en un diagrama de casos de uso, además de las relaciones entre casos de uso y actor (asociaciones) y las dependencias entre casos de uso (<<include>> y <<extends>>), pueden existir relaciones de herencia ya sea entre casos de uso o entre actores.

Llamamos modelo de casos de uso a la combinación de casos de uso y sus correspondientes diagramas. Los modelos de casos de uso se suelen acompañar por un glosario que describe la terminología utilizada. El glosario y el modelo de casos de uso son importantes puntos de partida para el desarrollo de los diagramas de clases.

Por último se debe tener en cuenta, que aunque cada caso de uso puede llevar a diferentes realizaciones, es importante reflejar en cada representación el motivo que nos ha llevado a descartarla, si es el caso.

**Pasos para la Definición de un Caso de Uso:**

* ID
* NOMBRE
* REFERENCIAS CRUZADAS
* CREADO POR
* ÚLTIMA ACTUALIZACIÓN POR
* FECHA DE CREACIÓN
* FECHA DE ÚLTIMA ACTUALIZACIÓN
* ACTORES
* DESCRIPCIÓN
* TRIGGER
* PRE-CONDICIÓN
* POST-CONDICIÓN
* FLUJO NORMAL
* FLUJOS ALTERNATIVOS
* INCLUDES
* FRECUENCIA DE USO
* REGLAS DE NEGOCIO
* REQUISITOS ESPECIALES
* NOTAS Y ASUNTO

# Normas de aplicación

Los casos de uso evitan típicamente el lenguaje técnico, prefiriendo la lengua del usuario final o del experto del campo del saber al que se va a aplicar. Los casos del uso son a menudo elaborados en colaboración por los analistas de requisitos y los clientes.

Cada caso de uso se centra en describir cómo alcanzar una única meta o tarea. Desde una perspectiva tradicional de la ingeniería de software, un caso de uso describe una característica del sistema. Para la mayoría de proyectos de software, esto significa que quizás a veces es necesario especificar decenas o centenares de casos de uso para definir completamente el nuevo sistema. El grado de la formalidad de un proyecto particular del software y de la etapa del proyecto influenciará el nivel del detalle requerido en cada caso de uso.

Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran lo que el actor hace o debe hacer para realizar una operación.

Un caso de uso debe:

* Describir una tarea del negocio que sirva a una meta de negocio.
* Tener un nivel apropiado del detalle.
* Ser bastante sencillo como para que un desarrollador lo elabore en un único lanzamiento.

Situaciones que pueden darse:

* Un actor se comunica con un caso de uso (si se trata de un actor primario la comunicación la iniciará el actor, en cambio si es secundario, el sistema será el que inicie la comunicación).
* Un caso de uso extiende otro caso de uso.
* Un caso de uso utiliza otro caso de uso.

# Facilidades

La técnica de caso de uso tiene éxito en sistemas interactivos, ya que expresa la intención que tiene el actor (su usuario) al hacer uso del sistema.

Como técnica de extracción de requisito permite que el analista se centre en las necesidades del usuario, qué espera este lograr al utilizar el sistema, evitando que la gente especializada en informática dirija la funcionalidad del nuevo sistema basándose solamente en criterios tecnológicos.

A su vez, durante la extracción (*extraction* en inglés), el analista se concentra en las tareas centrales del usuario describiendo por lo tanto los casos de uso que mayor valor aportan al negocio. Esto facilita luego la priorización del requisito.

Aunque comúnmente se asocian a la fase de Test de una aplicación, esta idea es errónea, y su uso se extiende mayormente a las primeras fases de un desarrollo.

# Limitaciones

Los casos de uso pueden ser útiles para establecer requisitos de comportamiento, pero no establecen completamente los [requisitos funcionales](https://es.wikipedia.org/wiki/Requerimiento_funcional) ni permiten determinar los requisitos no funcionales. Los casos de uso deben complementarse con información adicional como reglas de negocio, [requisitos no funcionales](https://es.wikipedia.org/wiki/Requerimiento_no_funcional), diccionario de datos que complementen los requisitos del sistema. Sin embargo la ingeniería del funcionamiento especifica que cada caso crítico del uso debe tener un requisito no funcional centrado en el funcionamiento asociado.