

Diagrama de casos de uso

El diagrama de casos de uso es uno de los diagramas incluidos en UML 2.5, estando este clasificado dentro del grupo de **diagramas de comportamiento**. Es, con total seguridad, el diagrama más conocido y es utilizado para representar los actores externos que interactúan con el sistema de información y a través de que funcionalidades (casos de uso o requisitos funcionales) se relacionan. Dicho de otra manera, muestra de manera visual las distintas funciones que puede realizar un usuario (más bien un tipo de usuario) de un Sistema de Información.

En este documento se incluye información sobre cómo construir este diagrama.

Lo primero es saber cuál es su finalidad. El diagrama de casos de uso, dependiendo de la profundidad que le demos, puede ser utilizado para muchos fines, entre ellos podemos encontrar los siguientes:

- **Representar los requisitos funcionales.**
- **Representar los actores** que se comunican con el sistema. Normalmente los actores del sistema son los usuarios y otros sistemas externos que se relacionan con el sistema. En el caso de los usuarios hay que entender el actor como un “perfil”, pudiendo existir varios usuarios que actúan como el mismo actor.
- **Representar las relaciones** entre requisitos funcionales y actores.
- **Guiar el desarrollo** del sistema. Crear un punto de partida sobre el que empezar a desarrollar el sistema.
- **Comunicarse de forma precisa entre cliente y desarrollador.** Simplifica la forma en que todos los partícipes del desarrollo, incluyendo el cliente, perciben como el sistema funcionará y ofrecerá una visión general común del mismo.

Contenido

- [1 Elementos de un diagrama de casos de uso](#)
 - [1.1 Actores](#)
 - [1.2 Casos de uso](#)
 - [1.3 Relaciones](#)
- [2 Descripción de requisitos funcionales y no funcionales](#)
- [3 Cómo dibujar un diagrama de casos de uso](#)
- [4 Ejemplos de un diagrama de casos de uso](#)

1.1 Elementos de un diagrama de casos de uso

Un diagrama de casos de uso está compuesto, principalmente, de 3 elementos: **Actores, Casos de uso y Relaciones**.

1.1.1 Actores

Como ya hemos comentado en la presentación, un actor es algo o alguien externo al sistema que interactúa de forma directa con el sistema. Cuando

decimos que interactúa nos referimos a que aporta información, recibe información, inicia una acción...

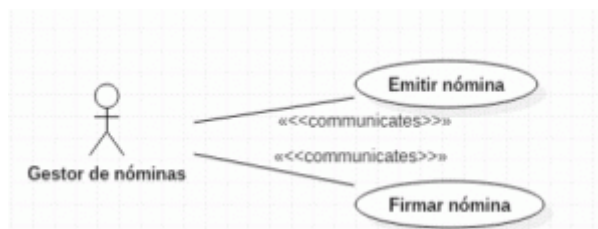
Se representan con una imagen de un “muñeco de palo” con el nombre del actor debajo.



Representación de un actor

Existen dos tipos de actores: Los usuarios y los sistemas.

No hay que entender los usuarios como personas singulares, sino como “perfiles o roles” que identifican a un tipo de usuario, pero no al usuario en sí. Por ejemplo, en una aplicación de gestión de nóminas, un actor de este tipo podría ser “gestor de nóminas” que se encarga de emitir y firmar nóminas. Este rol podría ser tomado, por ejemplo, por cualquier individuo del personal de recursos humanos y, además, por el jefe de la empresa. Es un ejemplo muy sencillo, pero como puedes ver, un actor no representa a una única persona o a un único usuario.



Ejemplo de actor

Por otro lado, los actores pueden ser otros sistemas que también interactúan con nuestro propio sistema. Un ejemplo podría ser, en nuestra aplicación de nóminas, un sistema que almacene las nóminas firmadas a modo de archivo. En este caso cuando se firma la nómina se recibe la misma por el sistema de archivo, por tanto el caso de uso se relaciona con el actor.

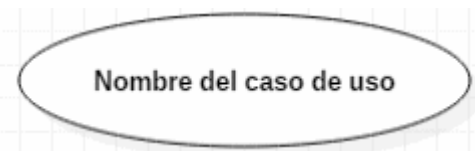
En ocasiones este tipo de actores no se representa con un “hombre de palo” porque puede dar la sensación de que es un usuario y queda poco intuitivo.

1.1.2 Casos de uso

Un caso de uso se utiliza para representar una de las funcionalidades que realiza el sistema. Es una secuencia de acciones que hace el sistema y que producen un resultado que puede percibir un usuario.

Formalmente hablando, un caso de uso es una clasificación de comportamiento que especifica una unidad de funcionalidad completa y que está realizada por uno o más sujetos que se relacionan con el caso de uso colaborando para ello con uno o más actores y que produce un resultado que tiene alguna utilidad para cualquier de esos actores.

Se representan con una elipse que incluye en su interior el nombre del caso de uso.



Representación de un caso de uso

Existen muchos ejemplos de casos de uso. Algunos podrían ser: Crear pedido, Listar productos, Enviar correo. Cualquier acción que realice la aplicación.

Las especificaciones anteriores a UML 2.5 requerían que un caso de uso sea invocado por un actor. En UML 2.5 esto se eliminó, lo que significa que podría haber algunas situaciones en las que la funcionalidad del sistema la inicie el propio sistema y, al mismo tiempo, brinde resultados útiles a un actor. Por ejemplo, el sistema podría notificar a un cliente que se envió la orden, programar la limpieza y el archivo de la información del usuario, solicitar información de otro sistema, etc.

1.1.3 Relaciones

Las relaciones **conectan los casos de uso** con los actores o los casos de uso entre sí.

Cuando conectan un actor con un caso de uso representa que ese actor **interactúa** de alguna manera con ese caso de uso y se representa con una línea continua con la identificación **<<communicates>>**.

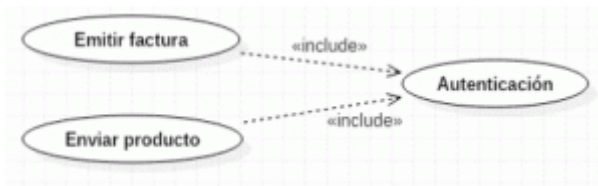
Cuando conectan casos de uso entre sí se pueden diferenciar dos tipos de relaciones: **<<include>>** y **<<extends>>**. En español a veces se usa la nomenclatura **<<usa>>** y **<<extiende>>**:

- **<<include>>**: Se utiliza para representar que un caso de uso **utiliza siempre** a otro caso de uso. Es decir, un caso de uso se ejecutará obligatoriamente (lo incluye, lo usa). Se representa con una flecha discontinua que va desde el caso de uso de origen al caso de uso que se incluye.



Relación include entre dos casos de uso

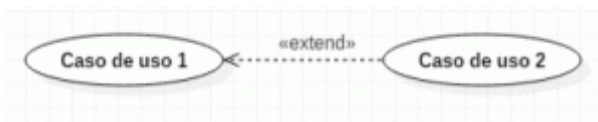
Un uso típico de este tipo de relaciones se produce cuando dos casos de uso **comparten una funcionalidad**. Esa funcionalidad es extraída de los dos y se crea un caso de uso nuevo que se relaciona con los anteriores con un include.



Ejemplo de uso de include

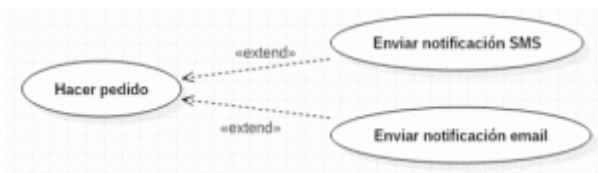
En este ejemplo, los casos de uso emitir factura y enviar producto ejecutarán ambos el caso de uso autenticación.

- **<<extend>>**: Este tipo de relaciones se utilizan cuando un caso de uso tiene un comportamiento **opcional**, reflejado en otro caso de uso. Es decir, un caso de uso puede ejecutar, normalmente dependiendo de alguna condición o flujo del programa, otro caso de uso. Se representa con una flecha discontinua que va desde el caso de uso opcional al original.



Relación extend entre dos casos de uso

Un ejemplo de esta relación podría ser la siguiente:



Ejemplo de relaciones extend

En este supuesto el caso de uso Hacer pedido puede dar lugar (o no) a otros dos casos de uso: Enviar notificación SMS y Enviar notificación email. Se supone que, cuando un usuario hace un pedido, el sistema le permite elegir si quiere que se envíe una notificación de ese pedido por SMS o por email.

Existe, además, otra relación denominada **generalización** que consiste en hacer que un elemento herede el comportamiento de otro. Aunque se puede utilizar entre casos de uso, es más común utilizarlo entre actores, haciendo que uno de los actores tenga acceso a las funcionalidades de otro. Se representa con una flecha con la punta hueca que va desde el elemento que hereda al elemento heredado:



Generalización entre dos actores

1.2 Descripción de requisitos funcionales y no funcionales

Es común en este tipo de diagramas describir cada caso de uso junto con la secuencia de pasos necesaria para completarlo y las posibles excepciones hasta definir todas las situaciones posibles. Esta descripción servirá de guía para el desarrollo, la profundidad de las situaciones que se traten dependerá de cada fase del proyecto o de cada situación en particular.

Existen dos tipos de requisitos:

- Requisitos funcionales
- Requisitos no funcionales

Los requisitos suelen ser plasmados junto a la siguiente información:

- **Identificador y nombre descriptivo:** Se utiliza una identificación única para diferenciarlo de los demás y un nombre descriptivo que suele coincidir con el objetivo que los actores esperan alcanzar al realizar el caso de uso.
- **Versión**
- **Autores**
- **Objetivos asociados**
- **Requisitos asociados**
- **Descripción:** Este campo debe completarse de forma distinta en función de si el caso de uso es abstracto o concreto.
- **Precondición:** se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el **caso de uso**.
- **Secuencia normal:** **secuencia** normal de interacciones del caso de uso. En cada paso, un actor o el sistema **realiza una o más acciones, o se realiza otro caso de uso**.
- **Postcondición:** se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.
- **Excepciones:** **especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado, lo que modifica el flujo "normal" del caso de uso.**
- **Importancia**
- **Urgencia**
- **Comentarios**

Ejemplos de representación de un requisito en forma de tabla:

| RF-01 | Acceso Aplicación | |
|-----------------------------|--|---|
| Versión | Versión 1.0 | |
| Autores | Alonso Quijano | |
| Objetivos Asociados | OBJ-01: Acceso Controlado a la Aplicación Software. | |
| Requisitos asociados | RI-01: Información de los Usuarios. | |
| Descripción | El sistema deberá comportarse como se describe en el siguiente caso de uso cuando un usuario decida acceder a la aplicación. | |
| Precondición | El usuario tiene que disponer de un nombre de usuario y una contraseña para poder acceder y deberá tener el acceso habilitado. | |
| Secuencia normal | Paso | Acción |
| | 1 | El usuario solicita al sistema entrar en la aplicación. |
| | 2 | El sistema solicita al usuario que introduzca el nombre de usuario y su contraseña. |
| | 3 | El usuario introduce su nombre y su contraseña. |
| | 4 | El sistema comprueba los datos introducidos. |
| | 5 | Si los datos son correctos el sistema muestra la página de inicio de la aplicación. |
| Excepciones | Paso | Acción |
| | 5 | Si el nombre de usuario no es correcto. El sistema muestra un mensaje. Ir al paso 2. |
| | 5 | Si la contraseña no es correcta. El sistema muestra un mensaje. Ir al paso 2. |
| | 5 | Si el usuario no tiene el acceso habilitado a la aplicación. El sistema muestra un mensaje. Ir al paso 2. |
| Postcondición | Si el nombre de usuario y la contraseña son correctos accede a la pantalla de inicio de la aplicación. | |
| Importancia | Vital. | |
| Urgencia | Inmediatamente. | |
| Comentarios | Ninguno. | |

Modelado de un requisito funcional

| RNF-01 | Entorno de Explotación |
|-----------------------------|--|
| Versión | Versión 1.0 |
| Autores | Alonso Quijano |
| Objetivos asociados | OBJ-05: Funcionamiento óptimo por usuario estándar |
| Requisitos asociados | |
| Descripción | El sistema deberá funcionar sin ningún tipo de limitación en equipos con: Pentium IV a 2,4 GHz, con 1 GBytes de RAM y al menos 6 GBytes de disco duro. |
| Importancia | Vital. |
| Urgencia | Inmediatamente. |
| Estabilidad | Alta. |
| Comentarios | Ninguno. |

Modelado de un requisito no funcional

1.3 Cómo dibujar un diagrama de casos de uso

A la hora de dibujar un diagrama de casos de uso te recomendamos que compruebes que has realizado previamente todas estas tareas, respondiendo a las preguntas que te escribimos a continuación:

- **Recopilar fuentes de información:** ¿cómo se supone que debo saber eso?
- **Identificar actores potenciales:** ¿qué usuarios utilizan los bienes y servicios del sistema empresarial?
- **Identificar posibles casos de uso:** ¿a qué bienes y servicios pueden recurrir los actores?
- **Conectar** los casos de uso: ¿quién puede hacer uso de los bienes y servicios del sistema empresarial?
- **Describir actores:** ¿a quién o qué representan los actores?
- **Buscar más casos de uso:** ¿Qué más debe hacer el sistema?
- **Documentar** casos de uso: ¿qué sucede exactamente en cada caso de uso?
- **Relacionar modelos** entre casos de uso empresarial: ¿qué actividades se realizan repetidamente?
- Verificar la vista, ¿todo es correcto?

Los pasos se han escrito en este orden a propósito, ya que es la forma lógica de seguirlos. Sin embargo, este orden no es obligatorio, ya que en la práctica, los pasos individuales a menudo se superponen unos con otros.

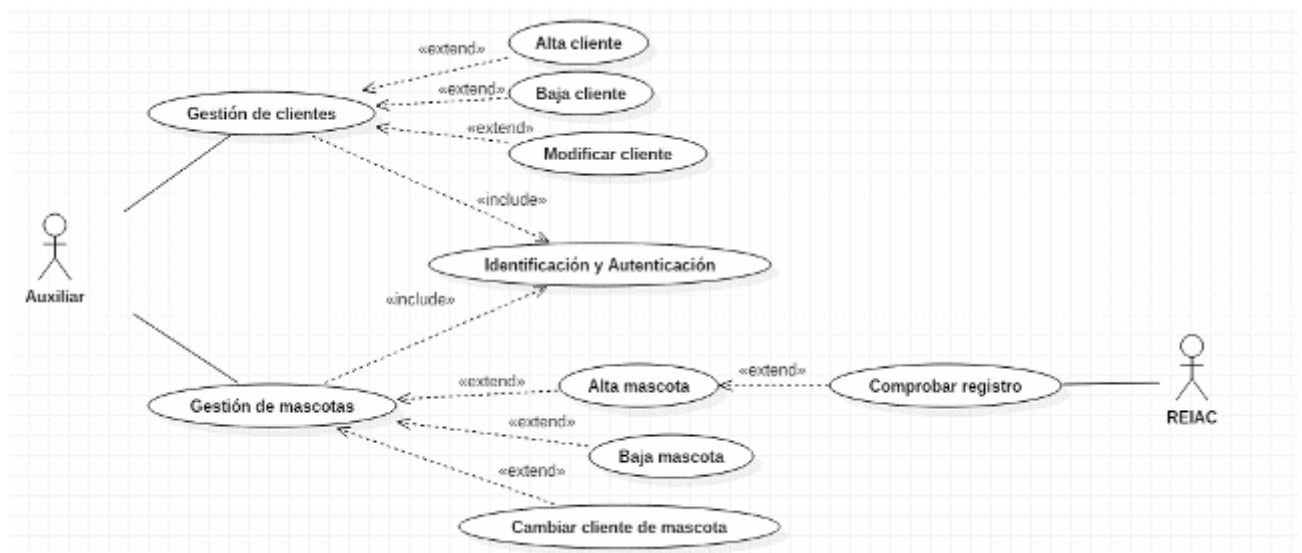
Para poder seguir los pasos de una forma óptima, es importante comprender el negocio/sistema para conseguir seguir cada paso individual. En algunos casos también es necesario consultar a los expertos o consultores del negocio. No tiene sentido aferrarse a la visión personal del analista, si este no tiene mucho conocimiento del área de negocio de la aplicación.

1.4 Ejemplos de un diagrama de casos de uso

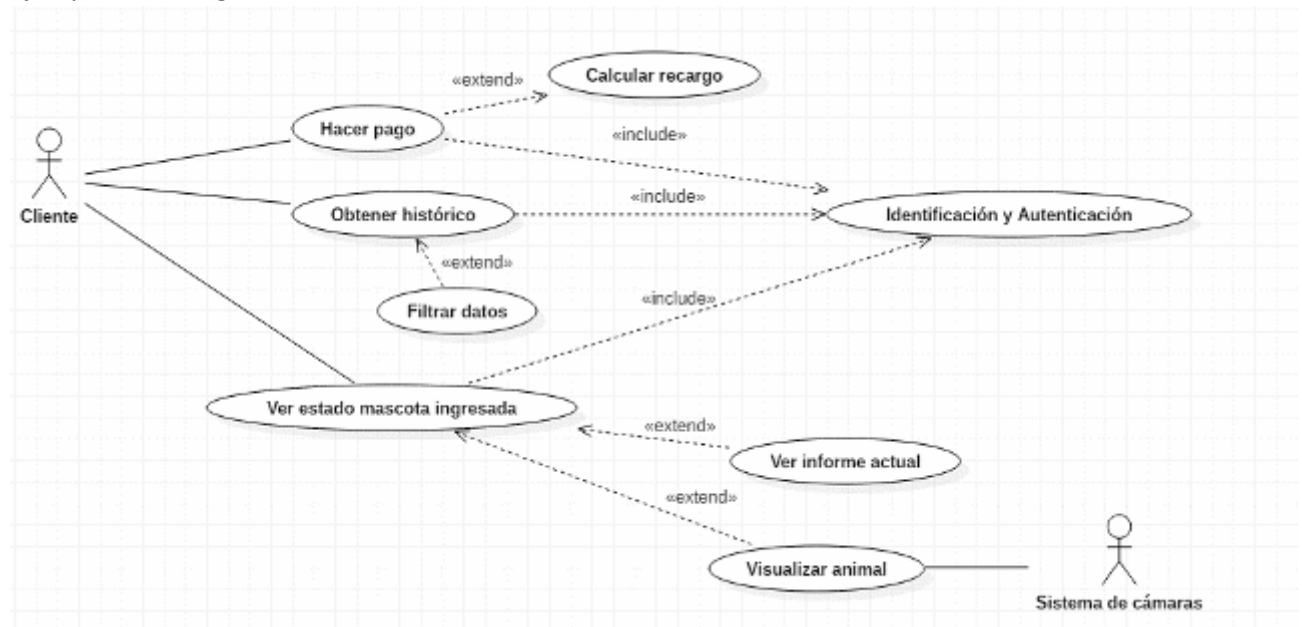
A modo de ejemplo se propone un ejercicio de un diagrama de casos de uso que consiste en el diseño de una aplicación que gestione los trámites a realizar en una clínica veterinaria en base a las siguientes premisas:

- La clínica veterinaria almacena datos de contacto de todos sus clientes como pueden ser: Nombre, Apellidos, DNI, Fecha de nacimiento, Teléfono o Email. Estos datos son introducidos y gestionados por los auxiliares, que ejercen las funciones administrativas.
- Además se almacena información de cada uno de las mascotas de las que es dueño cada cliente. Obviamente, cada cliente puede tener más de una mascota, pero cada mascota solo puede pertenecer a un único cliente. Se permite, además, cambiar el dueño de una mascota por otro.
- Al dar de alta un nuevo animal, se comprobará en el registro del REIAC (Red Española de Identificación de Animales de Compañía) si el animal está correctamente dado de alta. Este proceso únicamente se hará en animales que tengan la obligación de estar identificados.
- Cada vez que un veterinario realiza una consulta sobre un animal, esta queda almacenada incluyendo datos básicos como: Tiempo de consulta, Identificación de la persona que lo ha tratado, Animal tratado, Importe total, Resolución, Recetas... Para calcular el tiempo de la consulta el veterinario tendrá un botón en la aplicación donde pueda pulsar cuando comienza la consulta para calcular el tiempo a modo de cronómetro y otro botón para finalizar.
- En caso de que el animal se quede ingresado en la clínica, el cliente debe ser capaz de acceder al estado en tiempo real del animal. Además podrá comunicarse con una cámara que tendrá el animal colocado, donde podrá ver su situación actual. La gestión de estas cámaras no corresponde al sistema, sino que se utilizará una aplicación ya presente en el veterinario.
- Las recetas y otros documentos relacionados con el servicio se incluirán en un gestor de contenidos que ya está en funcionamiento en la clínica veterinaria.

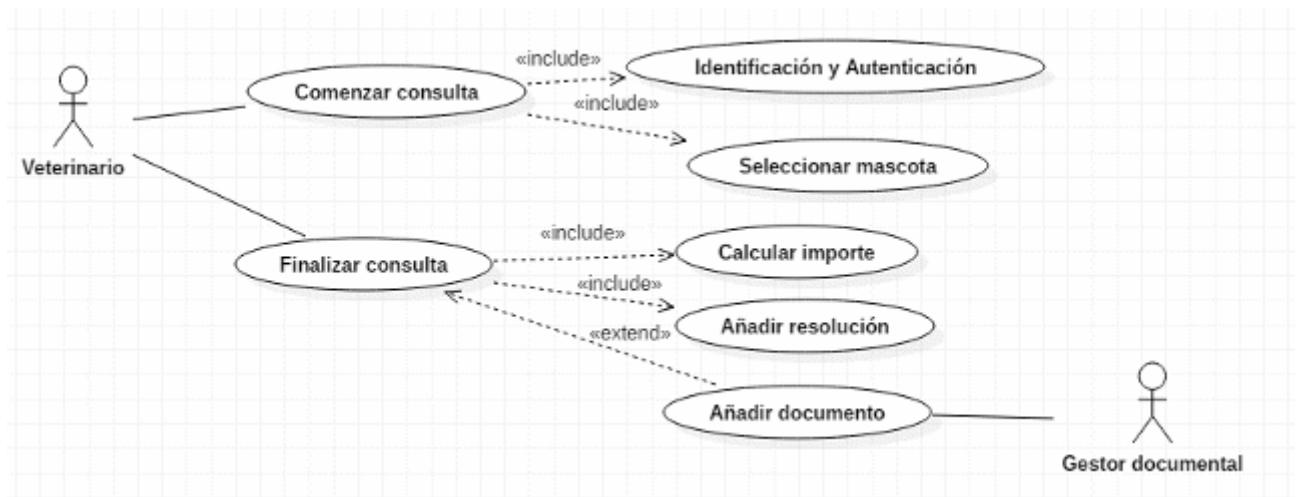
- Una vez terminado el servicio, el cliente no tiene por qué realizar inmediatamente el pago, sino que puede identificarse posteriormente en la aplicación vía web y realizar el pago. Si el cliente tarda más de una semana se efectuará un recargo sobre el precio inicial.
- Además, el cliente debe ser capaz de obtener un histórico de todas las consultas que ha recibido cualquiera de sus mascotas.



Ejemplo Diagrama de casos de uso del actor "auxiliar"



Ejemplo Diagrama de casos de uso del actor "Cliente"



Ejemplo Diagrama de casos de uso del actor “veterinario”

No obstante, dependiendo del nivel de profundidad, el diagrama puede variar significativamente descomponiendo, añadiendo, omitiendo o fusionando alguno de los casos de uso que se han expuesto.