



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.

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.

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**.

• Actores

Como ya se ha 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.

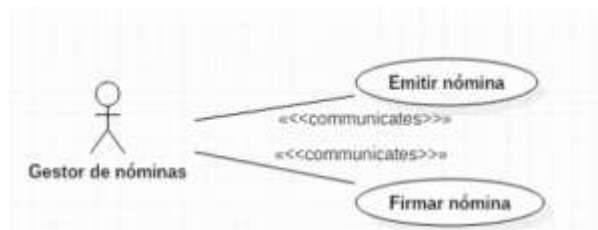
Se representan con una imagen de un “muñeco de palo” con el nombre del actor debajo.



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 podemos 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.

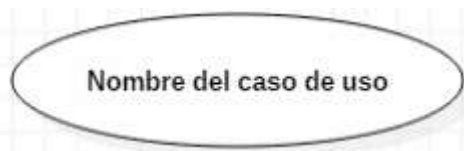
En ocasiones este tipo de actores no se representa con un “hombre de palo” porque puede dar la sensación de que es un usuario y queda poco intuitivo.

- **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

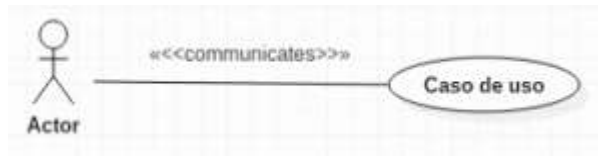
Existen muchos ejemplos de casos de uso. Algunos podrían ser: Crear pedido, Listar productos, Enviar correo. Cualquier acción que realice la aplicación.

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.

- **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>>.



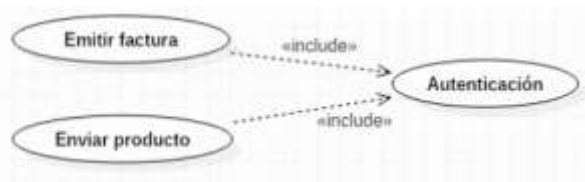
Cuando conectan casos de uso entre sí se pueden diferenciar dos tipos de relaciones: **<<include>>** y **<<extends>>**. En español a veces se usa la nomenclatura **<<usa>>** y **<<extiende>>**:

- ✓ **<<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

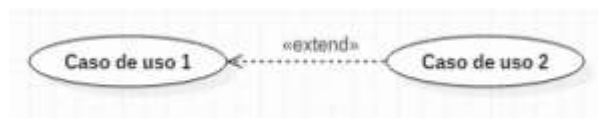
Un uso típico de este tipo de relaciones se produce cuando dos casos de uso **comparten una funcionalidad**. Esa funcionalidad es extraída de los dos y se crea un caso de uso nuevo que se relaciona con los anteriores con un include.



Ejemplo de uso de include

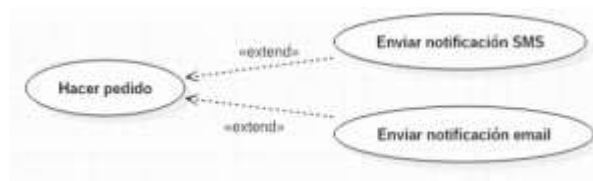
En este ejemplo, los casos de uso emitir factura y enviar producto ejecutarán ambos el caso de uso autenticación.

- ✓ **<<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

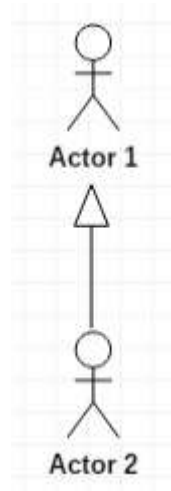
Un ejemplo de esta relación podría ser la siguiente:



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

Identificar actores y casos de uso

Para identificar actores y casos de uso se plantean las siguientes pautas y preguntas orientativas.

- **Identificar actores**

Encontrar actores es uno de los primeros pasos para definir los casos de uso del sistema. Cada tipo de fenómeno externo con el que el sistema debe interactuar está representado por un actor.

Como ya se ha comentado, un actor es cualquier cosa que intercambia datos con el sistema. Un actor puede ser un usuario, un hardware externo u otro sistema.

Las siguientes preguntas nos ayudarán a encontrar los actores que interactuarán con el sistema:

- ¿Qué grupos de usuarios requieren ayuda del sistema para realizar sus tareas?
- ¿Qué grupos de usuarios se necesitan para ejecutar las funciones principales más obvias del sistema?



- ¿Qué grupos de usuarios están obligados a realizar funciones secundarias, como el mantenimiento y la administración del sistema?
- ¿El sistema interactuará con cualquier hardware externo o sistema de software?

Cualquier individuo, grupo o fenómeno que se ajuste a una o más de estas categorías es un candidato para un actor.

Para determinar si tiene los actores adecuados (humanos), se puede intentar nombrar dos o tres personas que puedan actuar como actores, y luego ver si su conjunto de actores es suficiente para sus necesidades.

Puede ser difícil al principio encontrar los actores más adecuados, y no es probable que los encuentres a todos de inmediato porque no están aún identificados todos los casos de uso. Posteriormente, es posible que sea necesario revisar el modelo original, porque al principio hay una tendencia a modelar demasiados actores.

Hay que tener cuidado cuando se cambie actores; los cambios que se introduzcan también pueden afectar los casos de uso. Recordar que cualquier modificación a los actores constituye una alteración importante en las interfaces y el comportamiento del sistema.

El nombre del actor debe indicar claramente el papel del actor. Hay que asegurarse de que habrá poco riesgo en una etapa futura de confundir el nombre de un actor con otro.

Definir a cada actor escribiendo una breve descripción que incluya el área de responsabilidad del actor y para lo que el actor necesita el sistema. Debido a que los actores representan cosas fuera del sistema, no es necesario que se los describa en detalle.

• Identificar casos de uso

Cuando se completa el primer esquema de los actores, el siguiente paso es buscar los casos de uso del sistema. Los primeros casos de uso son muy preliminares, y sin duda tendrá que cambiarlos varias veces hasta que sean estables. Si la visión o los requisitos del sistema son deficientes, o si el análisis del sistema es impreciso, la funcionalidad del sistema no será clara. Por lo tanto, debe preguntarse constantemente si ha encontrado los casos de uso correctos. Además, debe estar preparado para agregar, eliminar, combinar y dividir los casos de uso antes de llegar a una versión final. Obtendrá una mejor comprensión de los casos de uso una vez que los haya descrito en detalle.

La mejor forma de encontrar casos de uso es **considerar lo que cada actor requiere del sistema**. Recordar que el sistema existe solo para sus usuarios y, por lo tanto, debe basarse en las necesidades de los usuarios. Reconocerá muchas de las necesidades de los actores a través de los requisitos funcionales establecidos en el sistema. Cada actor, sea humano o no, hay que hacer las siguientes preguntas:

- ¿Cuáles son las tareas principales que el actor quiere que realice el sistema?
- ¿El actor creará, almacenará, cambiará, eliminará o leerá datos en el sistema?
- ¿El actor necesitará informar al sistema sobre cambios repentinos y externos?
- ¿El actor necesita ser informado sobre ciertas ocurrencias en el sistema?
- ¿El actor realizará un arranque o apagado del sistema?

Las respuestas a estas preguntas representan los flujos de eventos que identifican casos de uso candidatos. No todos constituyen casos de uso separados; algunos pueden modelarse como variantes del mismo caso de uso. No siempre es fácil decir qué es una variante y qué es un caso de uso separado y distinto. Dependerá, en gran medida, del nivel de granularidad que queramos dar al esquema.



Además de los requisitos, el modelo empresarial de su organización es una valiosa fuente de información para determinar los casos de uso. El modelo de empresa describe cómo se puede incorporar el sistema de información en las operaciones existentes y, por lo tanto, le da una buena idea del entorno del sistema.

Un sistema puede tener varios modelos de casos de uso posibles. La mejor manera de encontrar el modelo “óptimo” es desarrollar dos o tres modelos, elegir el que prefiera y luego desarrollarlo más. Desarrollar varios modelos alternativos también lo ayuda a comprender mejor el sistema.

Cuando se haya delineado el primer modelo de caso de uso, se debe verificar que el modelo de caso de uso aborde todos los requisitos funcionales. Examinar cuidadosamente los requisitos para asegurarse de que todos los casos de uso cumplan con todos los requisitos.

Cada caso de uso debe tener un nombre que indique lo que se logra mediante sus interacciones con el (los) actor (es). El nombre debe tener varias palabras para ser entendido. No hay dos casos de uso que puedan tener el mismo nombre.

- **Describir cómo interactúan los actores y los casos de uso**

Debido a que es importante mostrar cómo se relacionan los actores con el caso de uso, al encontrar un caso de uso, se debe establecer qué actores interactuarán con él. Para hacer esto, se debe definir una asociación de comunicaciones que sea navegable en la misma dirección que la transmisión de señal entre el actor y el caso de uso.

Las transmisiones de señal generalmente van en ambas direcciones. Cuando este es el caso, debe permitir que las asociaciones de comunicaciones sean navegables en ambas direcciones. Definir, a lo sumo, una asociación de comunicación para cada par de actor y caso de uso.

También es recomendable describir brevemente cada asociación de comunicación que defina.

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

Cómo dibujar un diagrama de casos de uso

A la hora de dibujar un diagrama de casos de uso se recomienda comprobar haber realizado previamente todas estas tareas, respondiendo a las siguientes preguntas:

- **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.

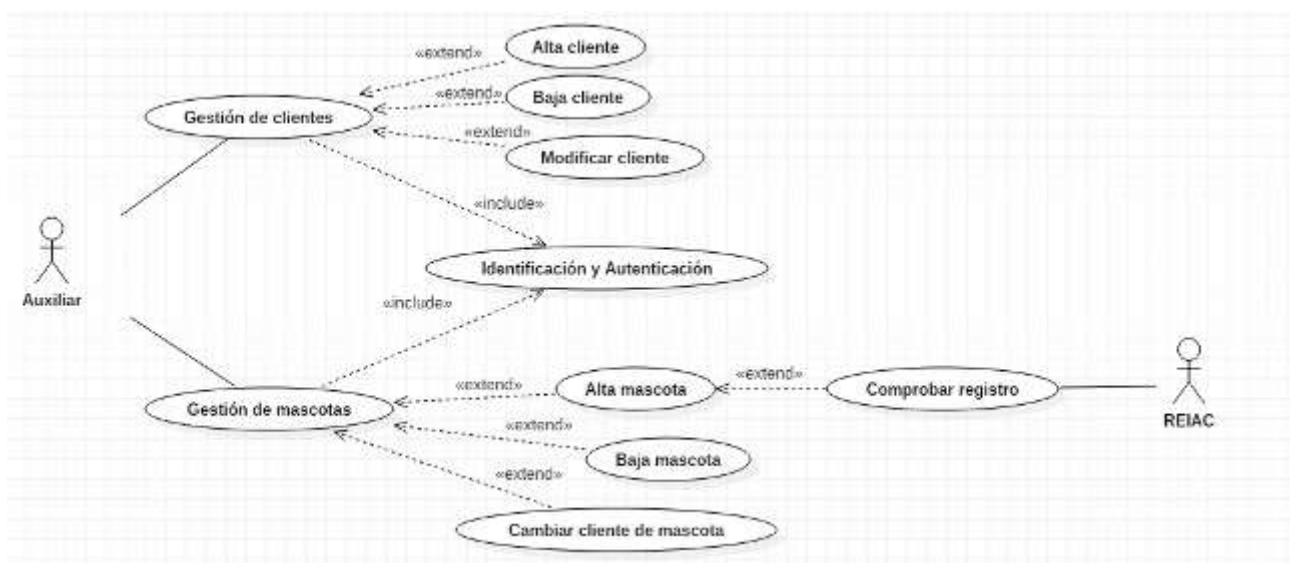
Ejemplo de un diagrama de casos de uso (Ejemplo extraído de la web)

Ejemplo clínica veterinaria:

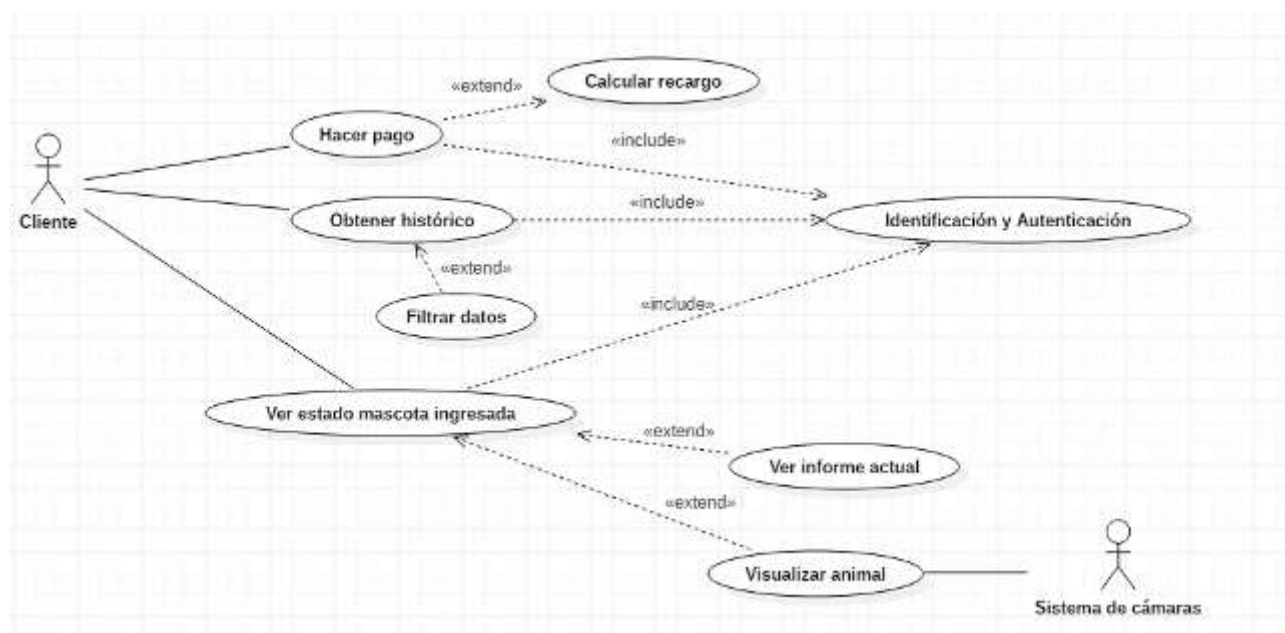
A modo de ejemplo se propone un ejercicio de un diagrama de casos de uso que consiste en el diseño de una aplicación que gestione los tramites a realizar en una clínica veterinaria en base a las siguientes premisas:



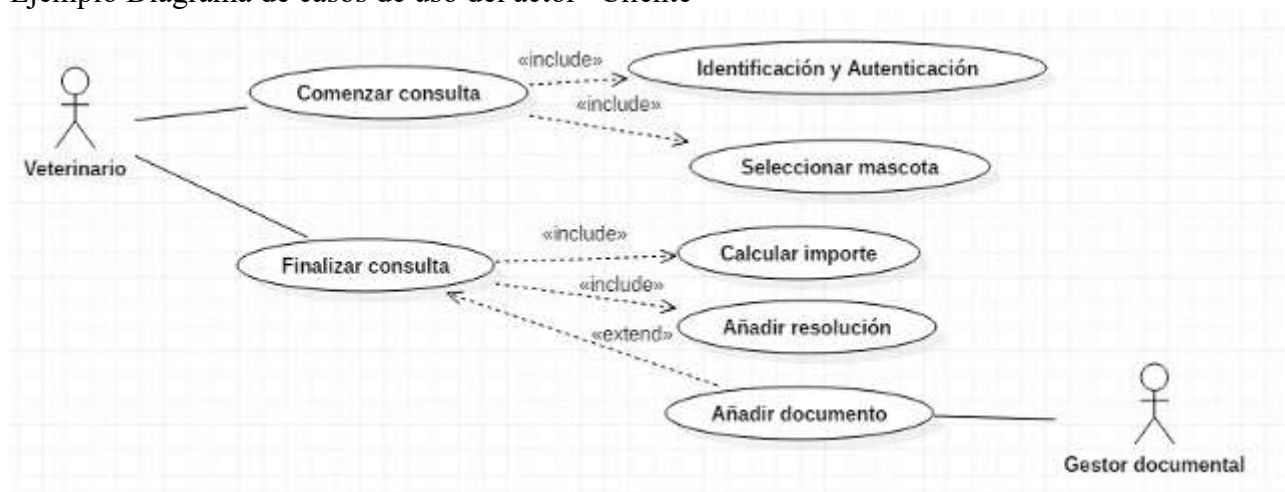
- La clínica veterinaria almacena datos de contacto de todos sus clientes como pueden ser: Nombre, Apellidos, DNI, Fecha de nacimiento, Teléfono o Email. Estos datos son introducidos y gestionados por los auxiliares, que ejercen las funciones administrativas.
- Además, se almacena información de cada 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 colocada, 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 porqué 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.