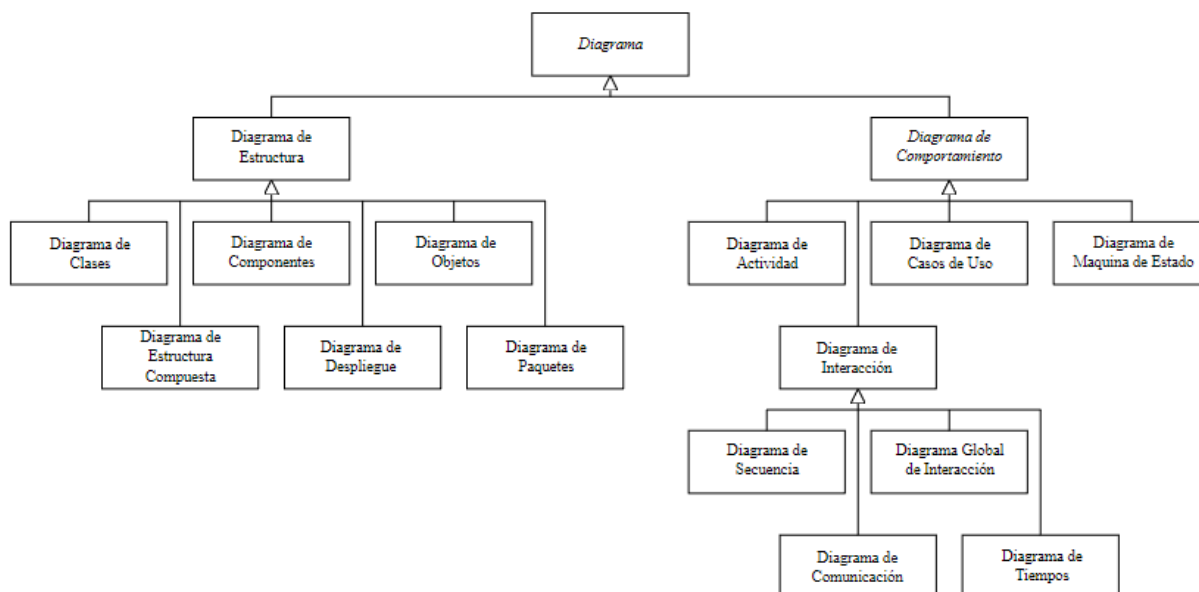


# IRAT

## UML

Lenguaje unificado de modelado por sus siglas en ingles. Es un lenguaje gráfico para visualizar, especificar construir y documentar un sistema.

### Tipos de diagramas en UML



### Diagramas estructurales:

Muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción. Existen un total de 7 tipos de diagramas de estructuras:

- **Diagrama de clases:** es el más utilizado, Es el bloque principal en cualquier solución orientada a objetos.
- **Diagrama de componentes:** Muestra la relación estructural de los componentes de un sistema de software. Se utiliza cuando se trabaja con sistemas complejos

que tienen muchos componentes. Los componentes se comunican entre sí mediante interfaces.

- **Diagrama de despliegue:** Muestra el hw de su sistema y el Sw de ese hw.
- **Diagramas de objetos:** Diagramas de instancia son muy similares a los diagramas de clase. Muestran la relación entre los objetos pero usan ejemplos del mundo real. Se utilizan para mostrar cómo se verá un sistema en un momento dado. Debido a que hay datos disponibles en los objetos a menudo se utilizan para explicar relaciones complejas entre objetos.
- **Diagrama de paquetes:** muestra como un sistema está dividido en agrupaciones lógicas y dependencias entre esas agrupaciones.
- **Diagrama de perfiles:** es un nuevo tipo introducido en UML 2.0 Se usa muy raramente.
- **Diagrama de estructura compuesta:** Se utiliza para mostrar la estructura interna de una clase.

## Diagramas de Comportamiento

Muestran el comportamiento dinámico de los objetos en el sistema.

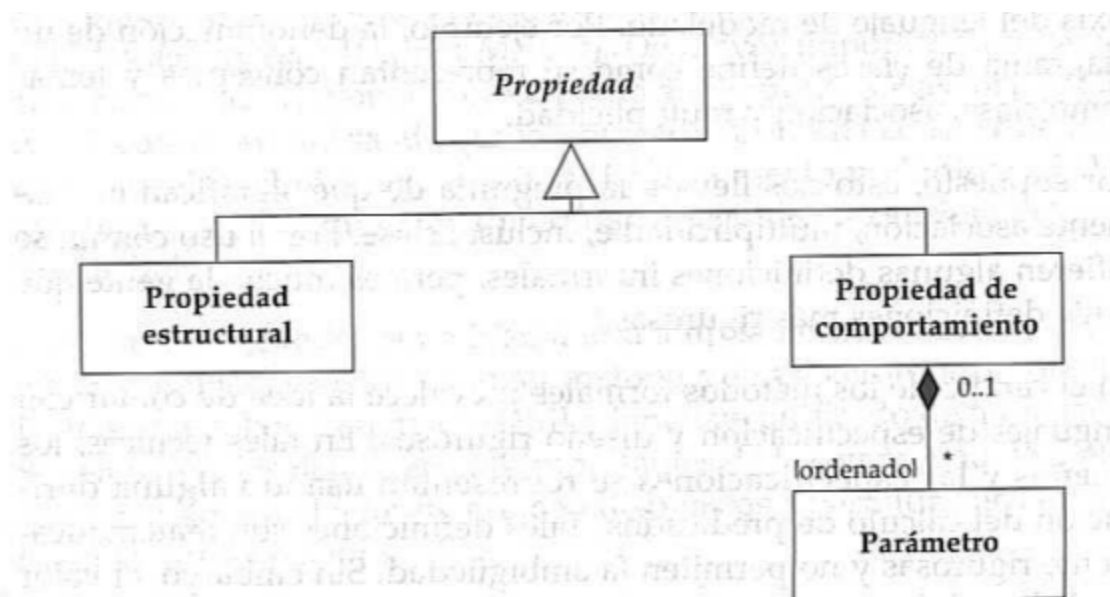
- **Diagrama de actividades:** representan flujos de trabajo de forma gráfica.
- **Diagrama de casos de uso:** es el más conocido, ofrece una visión general de los actores involucrados en un sistema. Las diferentes funciones que necesitan esos actores y cómo interactúan estas diferentes funciones. Es un gran punto de partida en cualquier proyecto.
- **Diagrama de máquina de estados:** son similares a los diagramas de actividad aunque con las anotaciones y el uso cambian un poco. Son útiles para describir el comportamiento de los objetos que actúan de manera diferente de acuerdo con el estado que se encuentran en el momento.
- **Diagrama de interacción:** los diagramas de interacción incluyen distintos tipos de diagramas:

- Diagramas de secuencia: muestra cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones. Se muestran las interacciones para un escenario en particular. Los procesos se representan verticalmente y las interacciones se muestran como flechas.
- Diagramas de comunicación: acá el foco está en los mensajes pasados entre objetos.
- Diagramas de tiempos: Los diagramas de sincronización son similares, representan el comportamiento de los objetos en un marco de tiempo dado.
- Diagramas global de interacciones: Son similares a los diagramas de actividad. En estos muestran una secuencia de diagramas de interacción y en el orden en que sucede,

Maneras de usar UML:

Sketch, blueprint y lenguaje de programación

El más común de los tres es el UML sketch, en donde los desarrolladores utilizan UML para ayudar a comunicar algunos aspectos de un sistema.



**Figura 1-1: Extracto del metamodelo del UML 1.1**

## Diagrama de casos de uso:

Cómo vimos anteriormente este tipo de diagramas muestran de manera visual las distintas funciones que puede realizar un usuario de un sistema de información.

Finalidades:

- Representar los requisitos funcionales
- Representar los actores: que se comunican con el sistema.
- Representar las relaciones entre requisitos funcionales y actores.
- Guiar el desarrollo
- Comunicarse de forma precisa entre cliente y desarrollador.

## Elementos de un diagrama de casos de uso:

### Actores:

Es algo o alguien externo al sistema que interactúa de forma directa con el sistema (aporta información, recibe información, inicia una acción.)

Se representa con un muñeco de palitos con el nombre del actor abajo.



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.



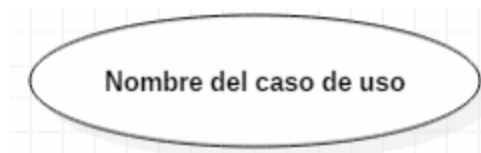
**Ejemplo de actor**

Esta tarea puede ser realizada por cualquier usuario, no solo por el gestor de nóminas, en caso de que él mismo no se encuentre puede realizarlo incluso el jefe de la empresa.

Casos de uso:

Un caso de estudio 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.

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



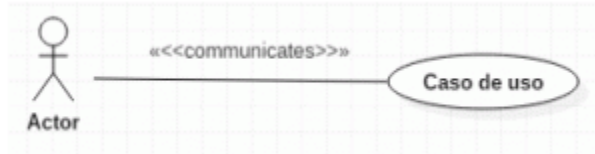
**Representación de un caso de uso**

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

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.

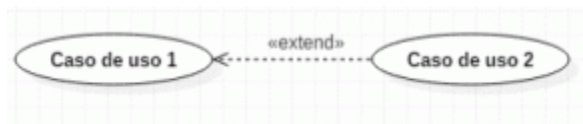


**Ejemplo de uso de include**

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:



## 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?. Para más información te recomiendo la [guía para identificar actores y casos de uso](#).
- **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.

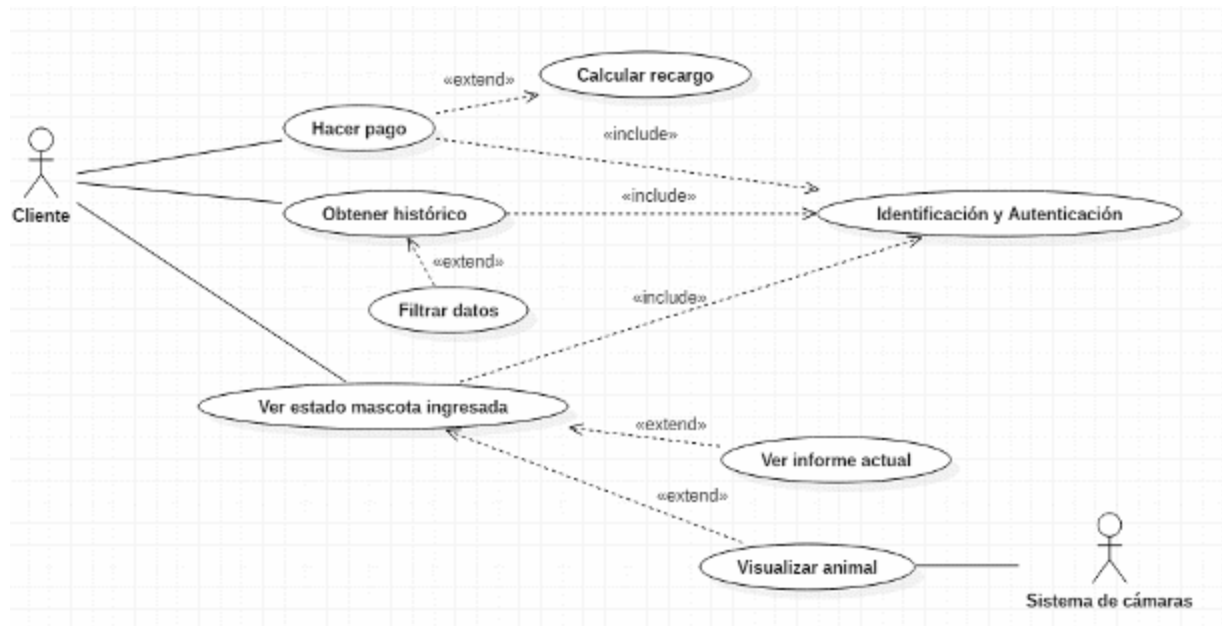
## Ejemplos de un diagrama de casos de uso

Ejemplo clínica veterinaria:

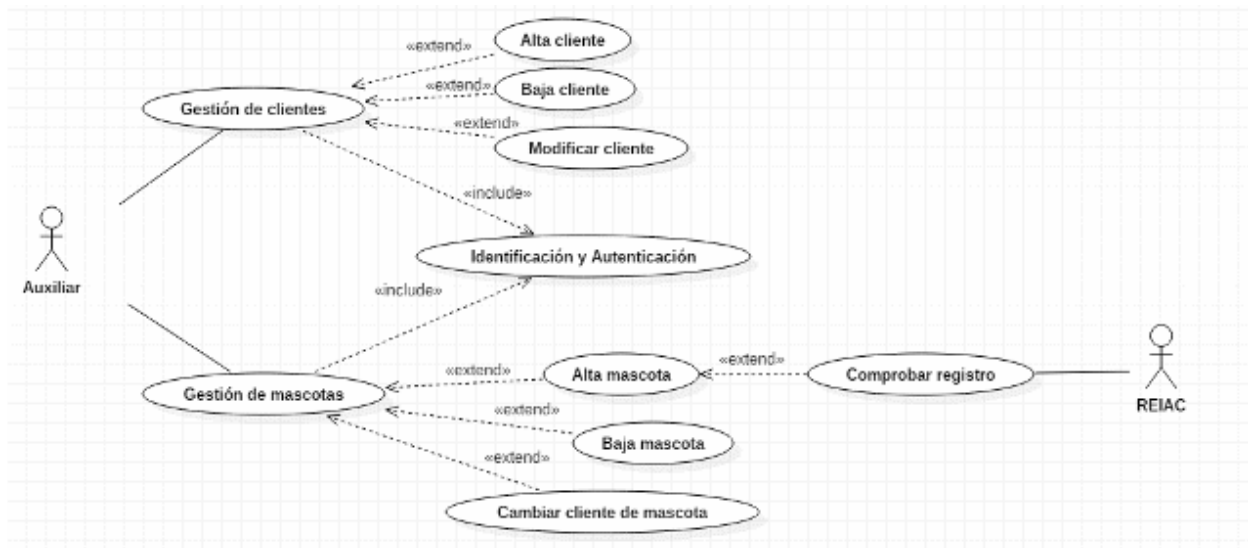
A modo de ejemplo se propone un ejercicio de un diagrama de casos de uso que consiste en el diseño de una aplicación que gestione los trámites a realizar en una clínica veterinaria en base a las siguientes premisas:

- 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.

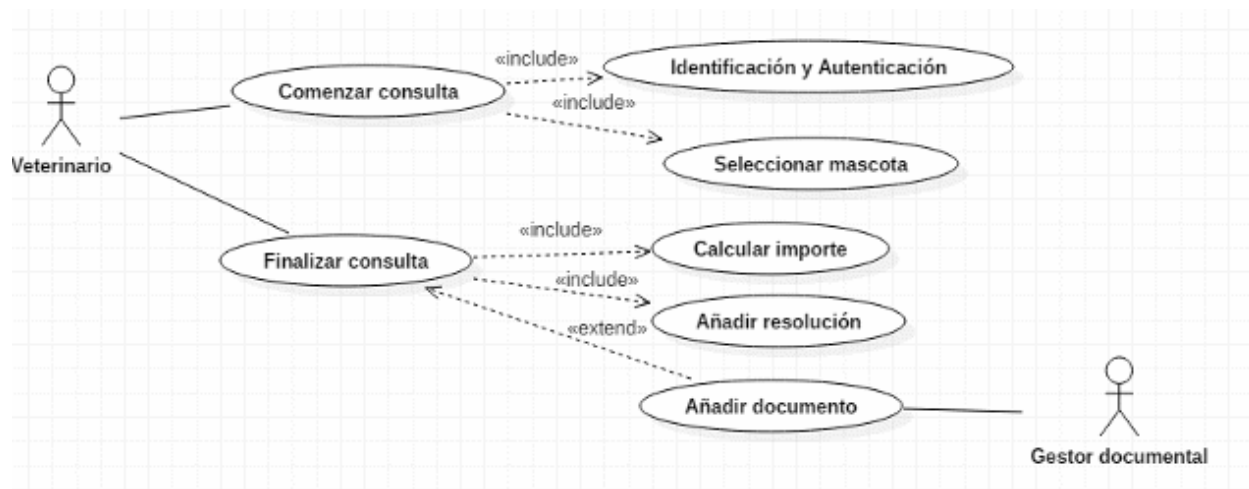
- En caso de que el animal se quede ingresado en la clínica, el cliente debe ser capaz de acceder al estado en tiempo real del animal. Además podrá comunicarse con una cámara que tendrá el animal colocada, donde podrá ver su situación actual. La gestión de estas cámaras no corresponde al sistema, sino 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 porque 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 «Cliente»**



**Ejemplo Diagrama de casos de uso del actor «auxiliar»**



**Ejemplo Diagrama de casos de uso del actor «veterinario»**