**Diagramas de casos de uso**

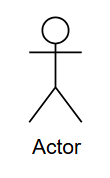
Un diagrama de casos de uso es una forma de diagrama de comportamiento (un diagrama de comportamiento es un tipo de diagrama de modelado de sistemas para representar como se comporta un sistema o algunas de sus partes a ciertos eventos o estímulos) mejorado. Este tipo de diagrama forma parte del modelado orientado a objetos y es muy común en el análisis y diseño de software.

El uso de este tipo de diagramas nos mostrara quien y como interactúa con el sistema, define la funcionabilidad del sistema y nos proporciona una base para posteriores diseños entre otras ventajas-

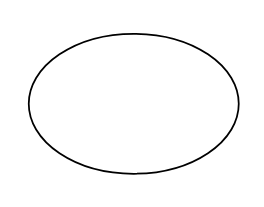
**ELEMENTOS DE LOS DIAGRAMAS DE CASOS DE USO**

Los elementos básicos de un diagrama de casos de uso son los siguientes:

* Actores: Representan a las personas, otros sistemas, o entidades externas que interactúan con el sistema. Se dibujan como muñecos (stick figures).



* Casos de uso: Son acciones o funciones que el sistema pueden realizar. Se representan como óvalos.



* Relaciones: Conectan actores con casos de usos o casos de usos entre si.

**RELACIONES EN LOS DIAGRAMAS DE CASO DE USO**

RELACIONES ENTRE ACTOR Y CASO DE USO

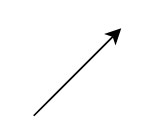
La asociación entre un actor y un caso de uso se denomina relación de comunicación. Un actor puede ser:

* Activo: Cuando el actor inicia es uso de un caso de uso. Si eso usa un flecha como relación esta debe apuntar al caso de uso, si no hay flecha el actor es activo de por si.
* Pasivo: Cuando el caso de uso es iniciado por el software y no por el actor. La flecha debe apuntar al actor.

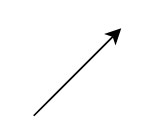
RELACIONES ENTRE CASO DE USO Y CASO DE USO

Existen tres tipos de relaciones:

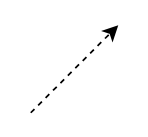
* Asociación: Esta relación se representa mediante un línea y se usa para representar la comunicación entre un actor y un caso de uso, entre casos de uso o entre actores.
* Generalización o especialización: Se usa cuando existe un comportamiento común a varios casos de uso, pero que presentan una diferencia. De esta forma los casos de uso específicos “heredan” el comportamiento del caso de uso generalizado. La relación se representa con una línea con flecha. Esta relación solo se puede realizar entre dos casos de uso o entre dos actores, nunca entre un actor y un caso de uso.
  + Por ejemplo un tienda que maneja varios sistemas de realizar pedidos: realizar pedido en tienda, online o por teléfono. Todos tiene pasos diferentes para el poder realizar el pedido pero se puede definir un caso de uso generalizado que sea “Realizar pedido”.



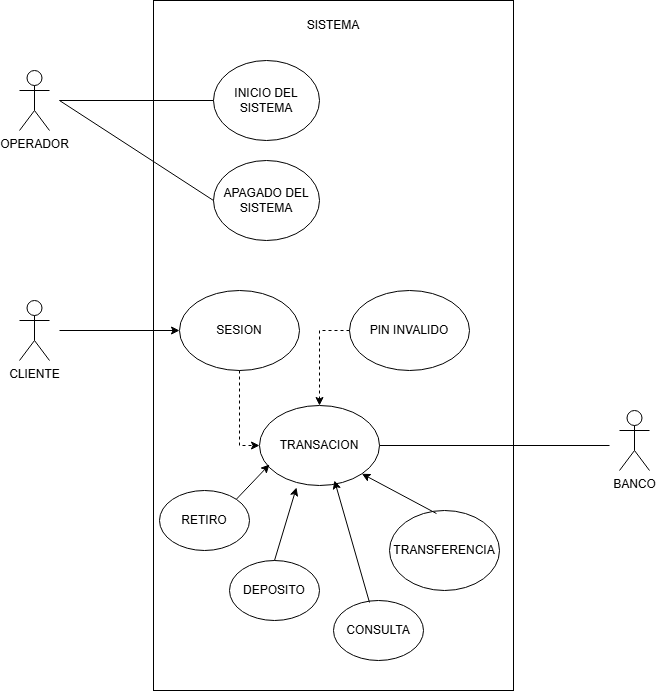
* Include: Utilizada cuando existen una serie de pasos comunes y de observación obligatoria entre varios casos de uso. Se representa mediante una flecha punteada que parte del caso de uso base hacia el caso de uso incluido. Esta relación solo se puede establecer entre dos casos de uso.
  + Por ejemplo un sistema de ventas con tres casos de uso: Ingresar pedido, facturar y registrar pago. Para estos tres casos de uso necesitamos primero buscar a que cliente asociaremos el pedido, creando asi un caso de uso llamado “Seleccionar cliente” con una relación de inclusión con los casos de uso anteriores.



* Extend: Se utiliza para describir comportamientos adicionales que añaden actividades a uno o varios casos de uso. Un caso de uso puede funcionar como añadido a otro o varios casos de uso. Esta relación se representa igual que una relación de include solo que esta vez la flecha parte del caso de uso extendido al caso de uso base. Esta relación solo se puede establecer entre casos de uso.
  + Por ejemplo, un sistema de banca donde tenemos un procedimiento de autentificación. Si el usuario se encuentra en otro país el banco pedirá otro sistema para autentificarse. Este otro sistema puede ser un caso de uso extendido con el nombre de “Sistema de autentificación alternativo” ya que no siempre será utilizado. Por último este caso de uso extendido podrá también ser utilizado por otros casos de uso como por ejemplo uno que permita hacer una transacción.



**EJEMPLO**



En este ejemplo podemos observar el funcionamiento de un cajero automático. Vemos tres actores los cuales interactúan con los diferentes casos de uso del sistema, que son las diferentes actividades que puede ejecutar el cajero automático.

**Diagramas de clases**

Un diagrama de clases es un tipo de diagrama estructural (un diagrama estructural es un tipo de diagrama que representa la estructura estática de un sistema, es decir, una estructura que no va a cambiar con el tiempo) el cual representa una clase de un sistema y sus relaciones.

En el diseño de software orientado a objetos, las clases son las encargadas de crear y manipular objetos. Por lo tanto, las clases son elementos esenciales dentro de este desarrollo.

Estos diagramas de clases no existen por si solos ya que dependen de los diagramas de casos de uso y están vinculados con los diagramas de objetos y de comunicación. Gracias a la combinación de estos diagramas podemos tener una visión mas clara del software.

Estos diagramas tienen dos propósitos principales:

* Visualizar las clases de un sistema y sus propiedades.
* Mostrar y analizar las relaciones entre clases.

Además este tipo de diagramas son la base para los diagramas de componentes y despliegue que muestran los aspectos de hardware y software de un sistema.

Estos diagramas nos muestran el funcionamiento, el comportamiento y las relaciones de un sistema. Nos ayudara con la creación de clases, sus atributos, sus métodos y sus relaciones. También podremos ver como afectará a las clases existentes el añadir una nueva clase.

**ELEMENTOS DE LOS DIAGRAMAS DE CLASE**

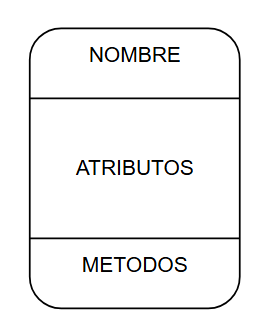
Estos diagramas contienen clases y sus interacciones. Estas se representan mediante un rectángulo que contiene el nombre de la clase, sus atributos y sus métodos. Según el nivel de detalle que necesitaremos determinaremos si mostrar los atributos y los métodos.

Las interacciones de estos diagramas muestran las relaciones entre las clases utilizando líneas y flechas. Las flechas tienen significados específicos para diferenciar las relaciones, como por ejemplo las de herencia, las bidireccionales o las unidireccionales. Estas clases también pueden agruparse en paquetes.

NOTACIONES DEL DIAGRAMA DE CLASES

Estas notaciones incluyen las etiquetas de lo miembros de la clase, su visibilidad y los paquetes.

* Clases: El nombre de una clase refleja el nombre de un objeto. Este nombre va en el centro en la sección superior arriba, en negrita y en mayúsculas. El nombre de una clase es obligatorio.
* Atributos: La sección del centro incluye los atributos de una clase. Estos describen las características de una clase, las cuales se aplicaran a los objetos.
  + Por ejemplo, en una clase que sea “**TARJETA DE DEBITO**”, los atributos serian el número de tarjeta y el propietario. Para cada objeto creado de la clase, los atributos tendrán diferentes valores.
* Métodos: La última parte del rectángulo es para los métodos de la clase. Los métodos son las acciones que la clase realiza.
  + Por ejemplo, en la clase “**TARJETA DE DEBITO**”, podemos tener un método que sea el poder cambiar un número de tarjeta y su propietario.
* Visibilidad: Los signos de mas y menos representan si los atributos de una clase o métodos son visibles o no. La almohadilla se puede usar para proporcionar una visibilidad protegida. La visibilidad nos permite decidir si una clase y sus métodos y atributos se pueden utilizar fuera de la propia clase.
* Paquete: Los paquetes sirven para agrupar clases y se visualizan mediante una caja con una etiqueta con el nombre en mayúsculas del paquete. El uso de paquetes nos ayudar a determinar una mejor visibilidad.



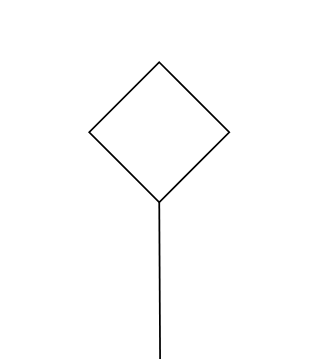
NOTACIONES DE LAS RELACIONES DE LOS DIAGRAMAS DE CLASES

Los diagramas de clases se conectan mediante líneas, notaciones textuales o números de multiplicidad para poder observar sus relaciones.

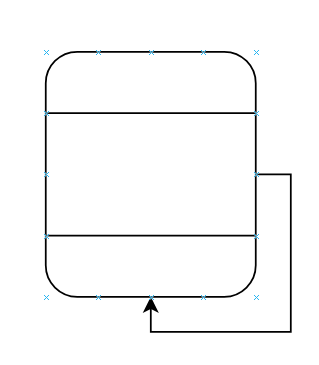
* Asociación: Una asociación conecta dos clases en cualquier conexión lógica. Es la relación mas utilizada y se ve como una línea recta. Las asociaciones pueden ser bidireccionales, es decir, que ambas clases saben que la otra existe o unidireccionales, las cuales se representan mediante una punta de flecha plana.



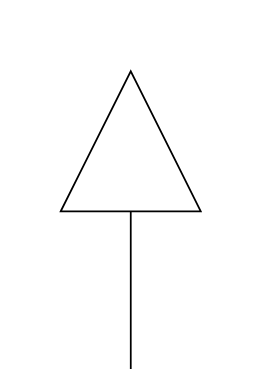
* Agregación y composición: La agregación y la composición son subconjuntos de la relación de asociación.
  + En la agregación las clases secundarias siguen existiendo independientemente de la principal y se representan con un diamante vacío junto a la clase principal.
  + La composición es lo opuesto y se representa mediante un diamante relleno junto a la clase principal.



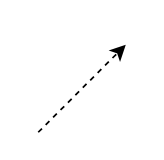
* Asociación reflexiva: La asociación reflexiva es la única relación unilateral y está representada mediante una línea que hace un bucle junto al modelo de la clase. Su propósito es representar una relación entre una instancia de una clase y otra instancia de la misma clase.



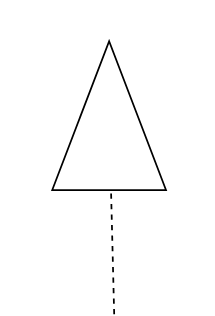
* Herencia y generalización: Este tipo de relaciones representan como clases heredan propiedades de otras clases. Esta relación se representa mediante una línea recta y una forma de triángulo hueco apuntando a la clase independiente.



* Dependencia: Esta relación se representa mediante una línea escalonada con una flecha que parte desde la clase dependiente. Esta relación representa que si hay un cambio en la superclase lo habrá en la subclase.

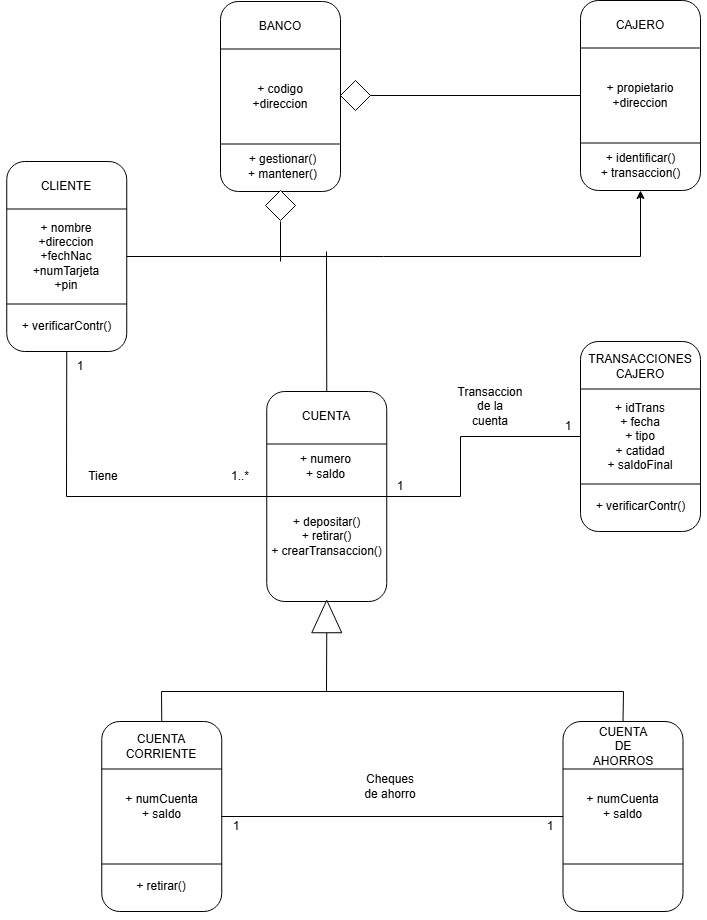


* Realización: Las clases conectadas por este tipo de relación indica que una clase utiliza un comportamiento definido por otra clase. Hay dos elementos en esta relación, la superclase y la subclase. Esta relación se representa mediante una línea escalonada y una forma de triángulo hueco.



* Multiplicidad: Esta relación es un conjunto de números colocados en las intersecciones de líneas. La multiplicidad define cuantas instancias participan en una relación. Los dígitos muestran el número de instancias de una clase que están vinculadas a una instancia de otra clase.

**EJEMPLO**



En este ejemplo podemos observar la representación de las diferentes clases de un sistema de un banco. Las clases se ven representadas mediante un rectángulo con las esquinas redondeadas divididas en secciones para los atributos, el nombre y los métodos. Estas clases están conectadas mediante relaciones diferentes como de herencia o de asociación. También podremos observar una descripción de la relación junto a su multiplicidad.

**Diagramas de actividades**

Un diagrama de actividades es una representación visual de acciones, restricciones, requisitos y otros factores que intervienen en la realización de tareas.

Una actividad en UML es un método para crear representaciones visuales que muestran las relaciones entre diferentes elementos. Estos diagramas representan el flujo de lo que sucede en el sistema, visualizando el comportamiento dinámico de un sistema mostrando el flujo de una actividad a otra. Cada componente de un diagrama de actividades se llama elemento y la actividad es el elemento de mas alto nivel dentro del diagrama.

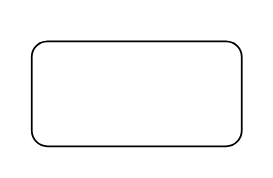
Este tipo de diagramas se pueden usar para rastrear la lógica de un algoritmo, representar los pasos en un caso de uso, modelar el proceso de negocio o flujo de trabajo o describir el flujo de acciones para cualquier actividad. También son muy útiles en las etapas de planificación para aclarar un flujo de trabajo o para las operaciones de un proyecto que ya está completo para poder entenderlo o mejorarlo.

Mapear actividades de esta manera puede revelar nueva información, ayudar a identificar ineficiencias y proporcionar otros beneficios importantes.

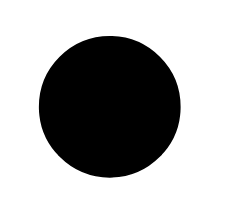
**ELEMENTOS DE LOS DIAGRAMAS DE ACTIVIDADES**

Estos son los elementos que componen un diagrama de actividades:

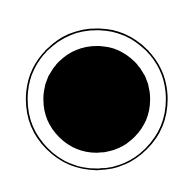
* Acción: Este elemento representa un paso dentro de una actividad. Las acciones suelen representarse con rectángulos con esquinas redondeadas.



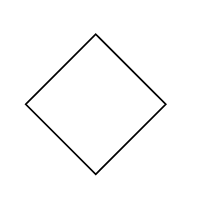
* Nodo de inicio: Esto es el punto de partida o el evento desencadenante de una actividad. Este se representa por un círculo negro sólido.



* Nodo final: Este es el paso final de una actividad, cuando el flujo dinámico terminado es retomado por otra actividad. Se representa mediante un círculo negro solido dentro de un círculo blanco mas grande.



* Nodo de control: Este elemento controla el flujo entre otros nodos. Se representa mediante un diamante con un flujo de entrada y dos o mas flujos de salida.



* Flujos de control: Estos también se conocen como bordes de control. Representa el flujo de control de un elemento a otro, con una línea sólida.



* Flujos de objetos: También se conocen como bordes de objeto. Estos representan el flujo dirigido de objetos de un elemento a otro y se representan mediante una línea punteada.

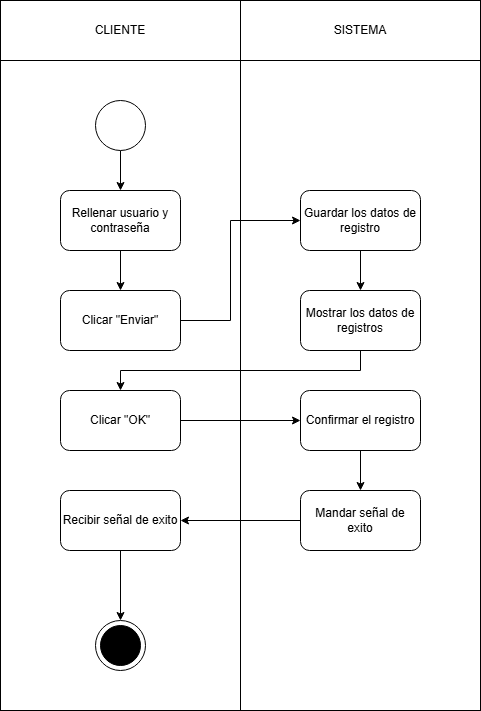


* Partición de actividad: Es una columna o fila que se utiliza para mostrar áreas de responsabilidad para diferentes actores. Estos también se conocen como diagramas de carriles.

**COMPARACIÓN DE LOS DIAGRAMAS DE ACTIVIDADES CON OTROS DIAGRAMAS**

Un diagrama de actividades describe el flujo de actividad de sistema a sistema, pero dependiendo del diseño y de las necesidades puede que no aborde a los usuarios ni a los actores. Sin embargo, en los diagrama de caso de uso si que se preocupa por los usuarios y los actores.

**EJEMPLO**



En este ejemplo podemos ver como es el proceso de registro de un cliente dividido en dos columnas. En la primera podemos ver la acciones del cliente y en la segunda las acciones del sistema.