

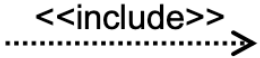



Diagramas de casos de uso

Interacciones del sistema y su entorno. Los diagramas de casos de uso muestran la funcionalidad del sistema desde el punto de vista de un observador externo. *¿Qué hace el sistema?*. Capturan los requisitos funcionales del sistema y define los límites entre el sistema y los elementos externos. Un diagrama de casos de uso es un grafo que se compone de los siguientes elementos:

- Actor: Representa a cualquier elemento que intercambia información con el sistema. Los actores se encuentran fuera del sistema. Se clasifican en:
- Caso de uso: Describen una interacción entre el actor y el sistema
- Asociación: Representa una relación entre los elementos del diagrama
- Escenario: Describe una secuencia concreta de pasos que describe una interacción entre el usuario y el sistema. Proporciona un ejemplo de lo que ocurre cuando alguien interactúa con el sistema. Es un caso concreto de la interacción

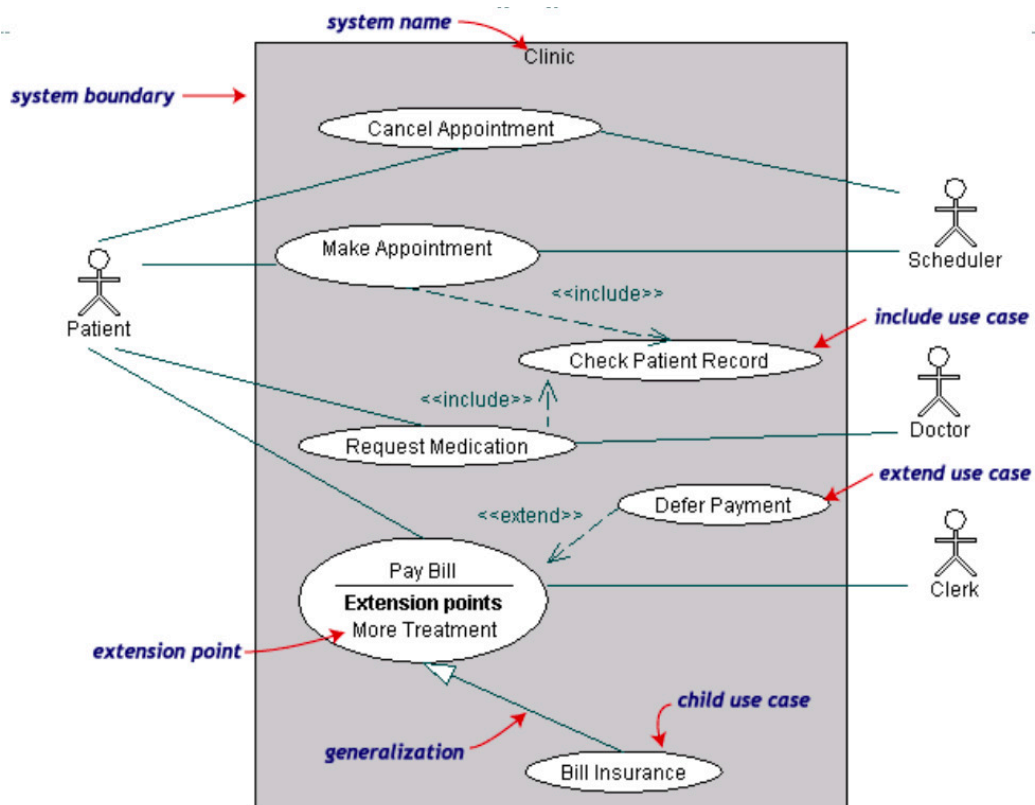
Relación	Función	Notación
Asociación	Interacción entre un actor y un caso de uso	
Generalización	Relación entre un caso general y otro más específico que hereda sus características (especializado -> general)	
Include	Inserta un fragmento de comportamiento. Se usa para factorizar los casos (dividirlo en componentes más sencillos)	
Extends	Inserta un comportamiento adicional bajo determinadas condiciones (caso -> except)	





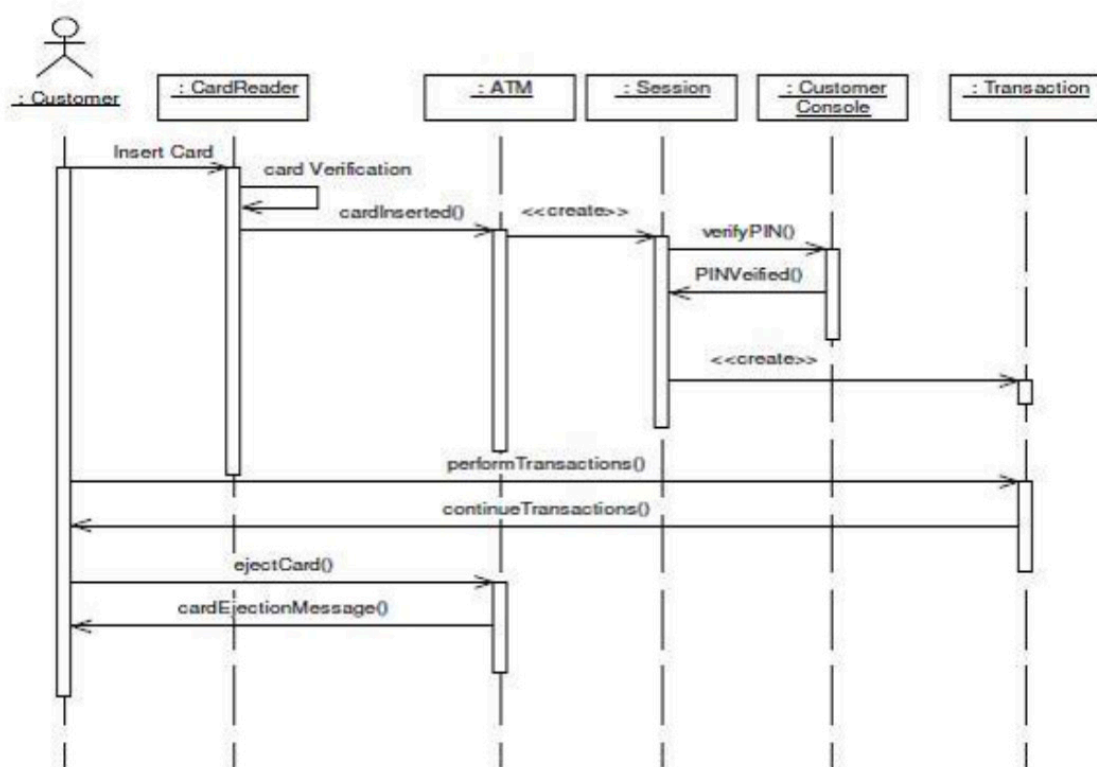






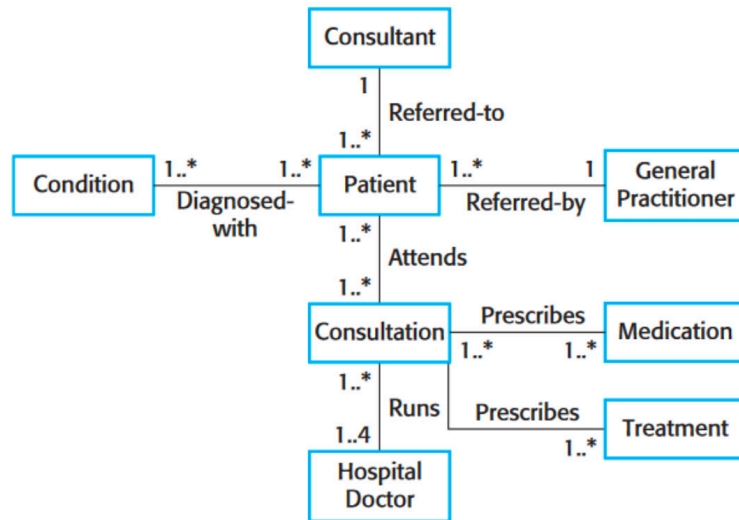
Diagramas de secuencia

Los diagramas de secuencia se utilizan para modelar las interacciones entre los actores y los objetos dentro de un sistema. Un diagrama de secuencia muestra la secuencia de interacciones que tienen lugar durante un caso de uso particular (escenario). Los objetos y los actores involucrados se muestran en la parte superior del diagrama, con una línea de puntos trazada verticalmente a partir de estos. Las interacciones entre los objetos se indica por las flechas con etiquetas



Diagramas de clases

Se utilizan para mostrar las clases del sistema bajo desarrollo y las asociaciones entre estas clases. Una clase se puede considerar como una definición de un tipo de objeto de la misma. Una asociación es un enlace entre clases que indica que existe alguna relación entre estas clases. Durante las primeras etapas del proceso de ingeniería del software los objetos representan algo en el mundo real



Tipos de diagramas de clase

- Conceptual: Modela el problema y los recursos del sistema. Contiene clases del negocio
- De implementación: Modela la solución y los requisitos del código. Se pueden usar para generar código

Clases y objetos

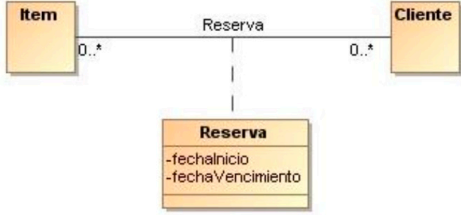

- Objeto: Es un concepto o entidad única inequívocamente identificable. Un objeto es una instancia de una clase. Especificación: `Objeto : Clase`
- Clase: Es una plantilla que describe las propiedades y comportamiento de los objetos. Se define su estructura mediante atributos y operaciones
 - Atributos: Representan la información almacenada en una clase. Especificación: `visibilidad nombre : tipo = valor {propiedad}`
 - Operación: Define una propiedad de comportamiento de una clase. Especificación `visibilidad nombre : tipo {propiedad}`





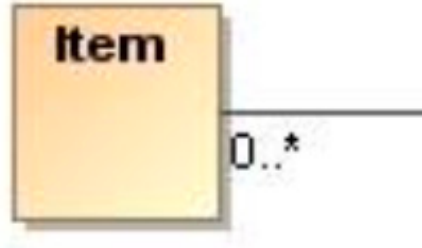
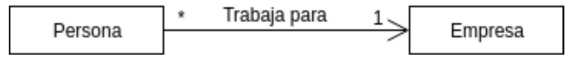
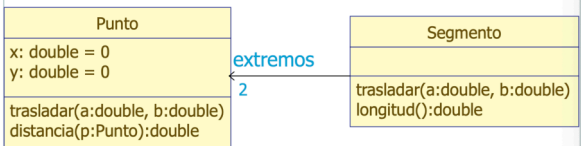


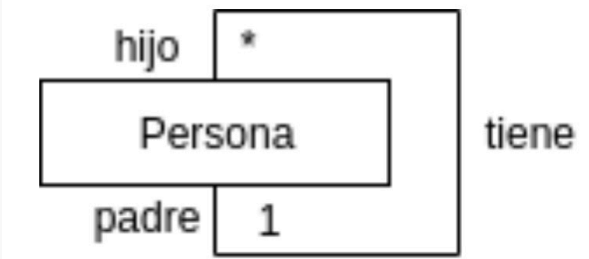
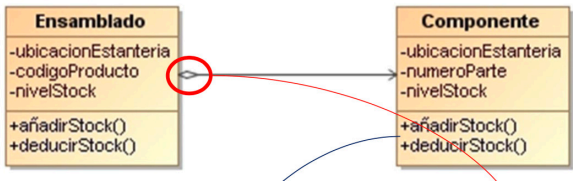
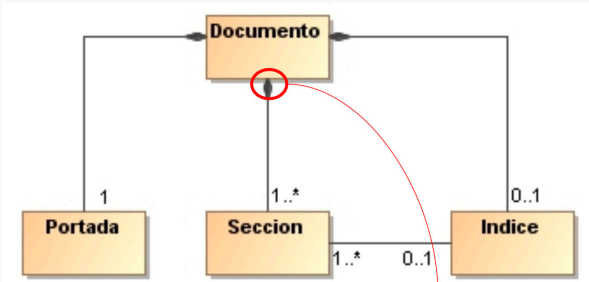
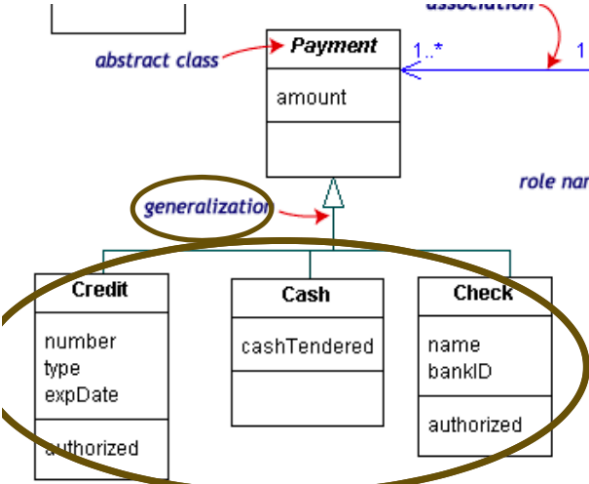


Clases especiales	Descripción	Ejemplo
Clase asociación	Se conecta a una asociación para definir propiedades de la asociación (similar a la tabla intermedia muchos-muchos en BBDD)	
Clase abstracta	Es una clase no instanciable. Contiene operaciones abstractas que las clases derivadas definen. Se indican con el nombre en cursiva	

Asociaciones

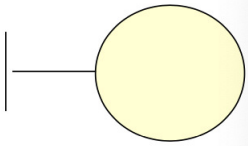
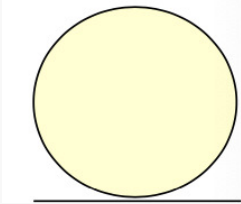
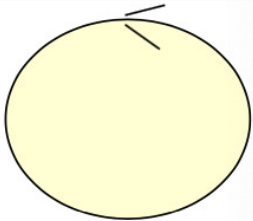
Relación estructural entre dos clases. Especifica que objetos de un tipo tienen una referencia a objetos de otro tipo. Esto permite a una clase acceder a los datos y comportamiento de otra

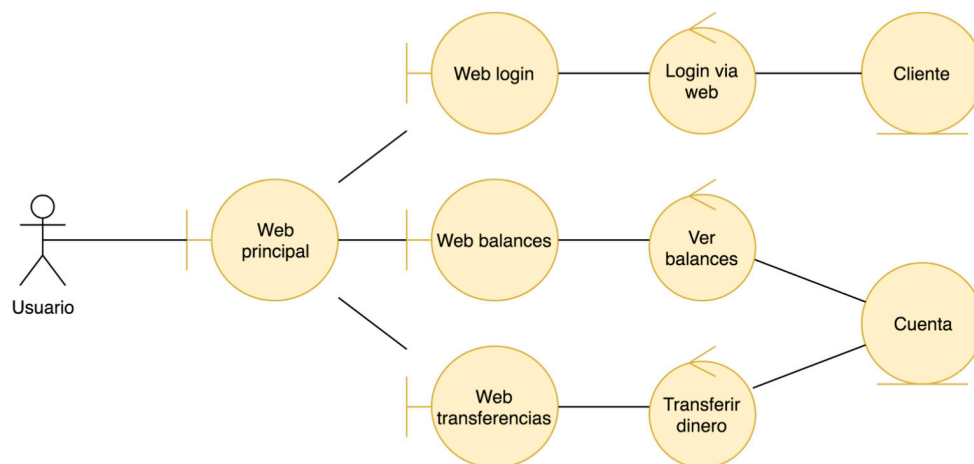
Nombre	Descripción	Ejemplo
Multiplicidad	Indica cuantos objetos de un extremo de la asociación están relacionados con un objeto del otro extremo	
Navegabilidad	Una flecha indica la dirección en la que una asociación puede ser preguntada. Nos permite conocer quien posee la implementación de la asociación. Si no tiene flecha, es bidireccional	
Rol	Indica el papel que la clase de un extremo de la asociación presenta a la clase del otro extremo	

Asociación reflexiva	Una clase puede ser principio y final	
Agregación	Cuando una clase es parte de otra. Se indica con un rombo vacío sobre el objeto del que es parte	
Composición	Indica una relación de dependencia, es decir, un objeto no puede existir el resto de objetos a los que está conectado dejan de existir. Se representa con un rombo relleno sobre el objeto del que dependen	
Generalización	Se utiliza para indicar que una o varias clases son especializaciones de otra	

Clases de análisis

Representan abstracciones de lo que serán una o varias clases en el diseño. Se centra en los requisitos funcionales y no contiene operaciones ni signaturas

Nombre	Descripción	Símbolo
Interfaz	Modela la interacción entre el sistema y los actores. Son abstractas y solo definen la información y las peticiones que intercambian entre el sistema y sus actores	
Entidad	Modelan la información persistente. Derivan de una clase del modelo del negocio o del dominio	
Control	Modelan los aspectos dinámicos internos del sistema. representan el control de un caso de uso concreto. Representan cálculos o lógica del sistema	



Diagramas de actividad

Modela la realización de una actividad (workflow). Están formados por un punto inicial y un punto final y contienen elementos condicionales que modulan como actuar