

A close-up photograph of several folded teal-colored microfiber cloths stacked together. The cloths are folded in a way that creates multiple concentric circles, with the edges of the cloths visible where they overlap.

Modelado en Ingeniería de Software

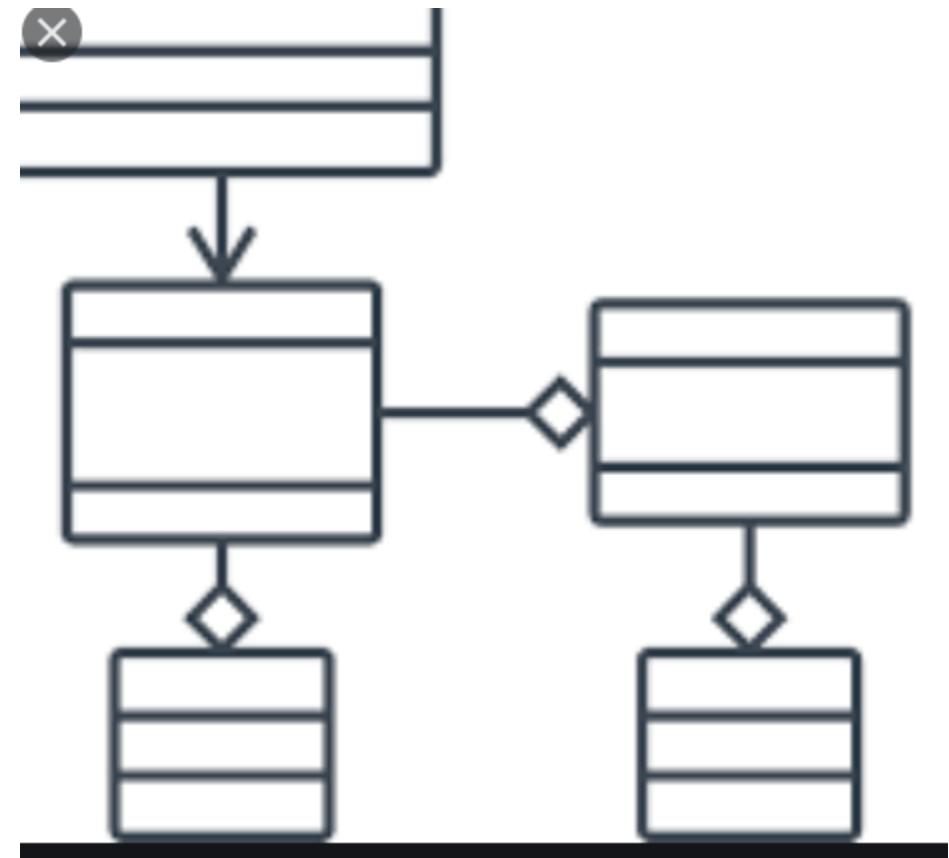
La importancia de modelar

- La concepción de un diseño de un sistema de software involucra imaginación, experiencia y conocimiento.
- Un modelo es una representación simplificada de la realidad que permite simular en diferentes situaciones el comportamiento de un sistema



Modelado de software OO

- Debe ser capaz de representar la información que el software transforma, la arquitectura y sus funciones.
- Representar las características que desean los usuarios y el comportamiento del sistema mientras la transformación tiene lugar.



Modelado



Los modelos deben cumplir estos objetivos en diferentes niveles de abstracción,

Desde el punto de vista del cliente

Representación en un nivel más técnico.



En el trabajo de ingeniería de software se crean dos clases de modelos:

Análisis y de diseño.

Modelado

Los Modelos de análisis (*modelos de requerimientos*)

- El de la información, el funcional y el de comportamiento.

Los *Modelos de diseño* representan características del software que ayudan a los profesionales a elaborarlo con eficacia:

- Arquitectura, interfaz de usuario y detalle en el nivel de componente.

Modelado de requerimientos



Tareas de modelado que conducen a la

especificación de los requerimientos y a la representación de un diseño del software que se va a elaborar.



El modelo de requerimientos es

un conjunto de modelos que muestra la primera representación técnica de un sistema.

Modelado de requerimientos

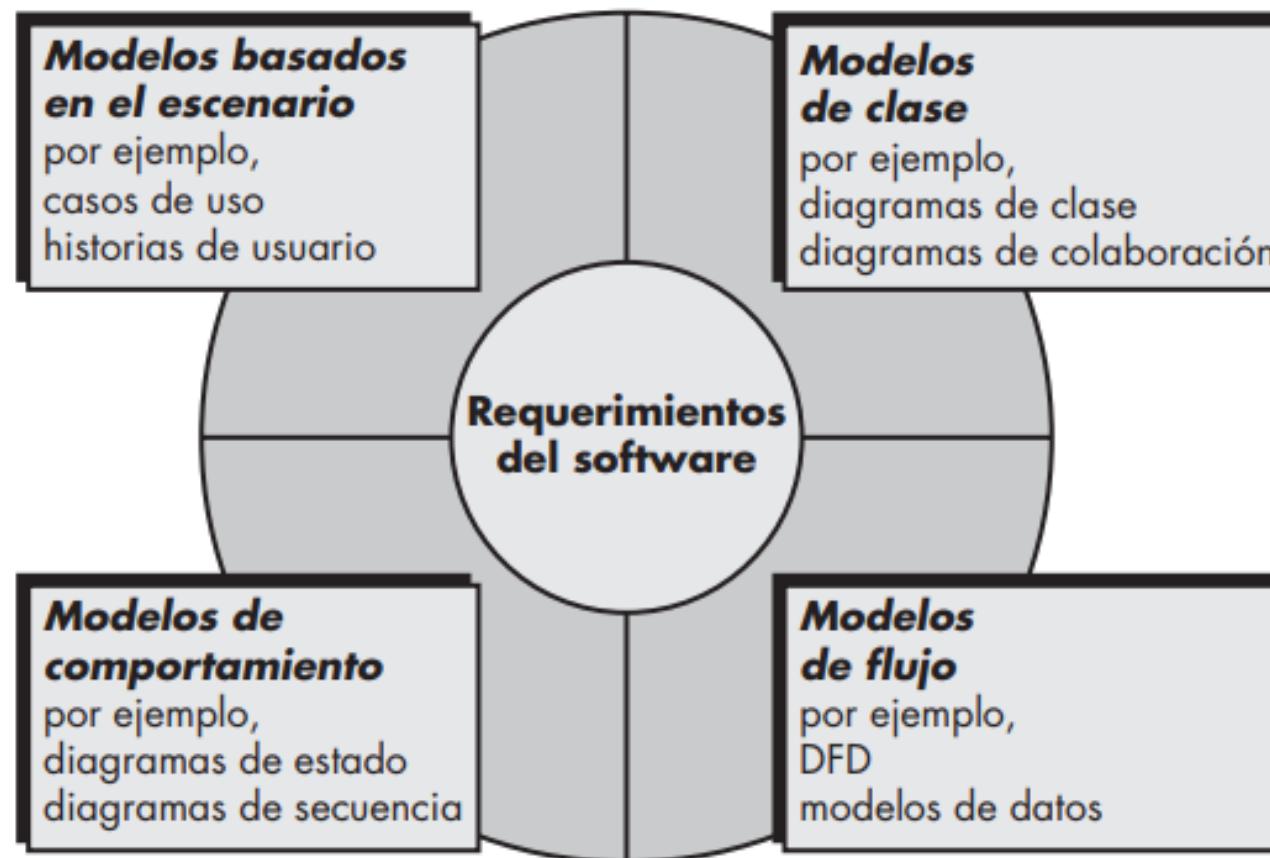
Para construir el modelo de requerimientos, se escoge una amplia variedad de

- representaciones basadas en texto y en diagramas.

Cada una de dichas representaciones

- da una perspectiva de uno o más de los elementos del modelo.

Elementos del modelo de análisis



UML

- UML (Unified Modeling Language) es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos de los sistemas de software.

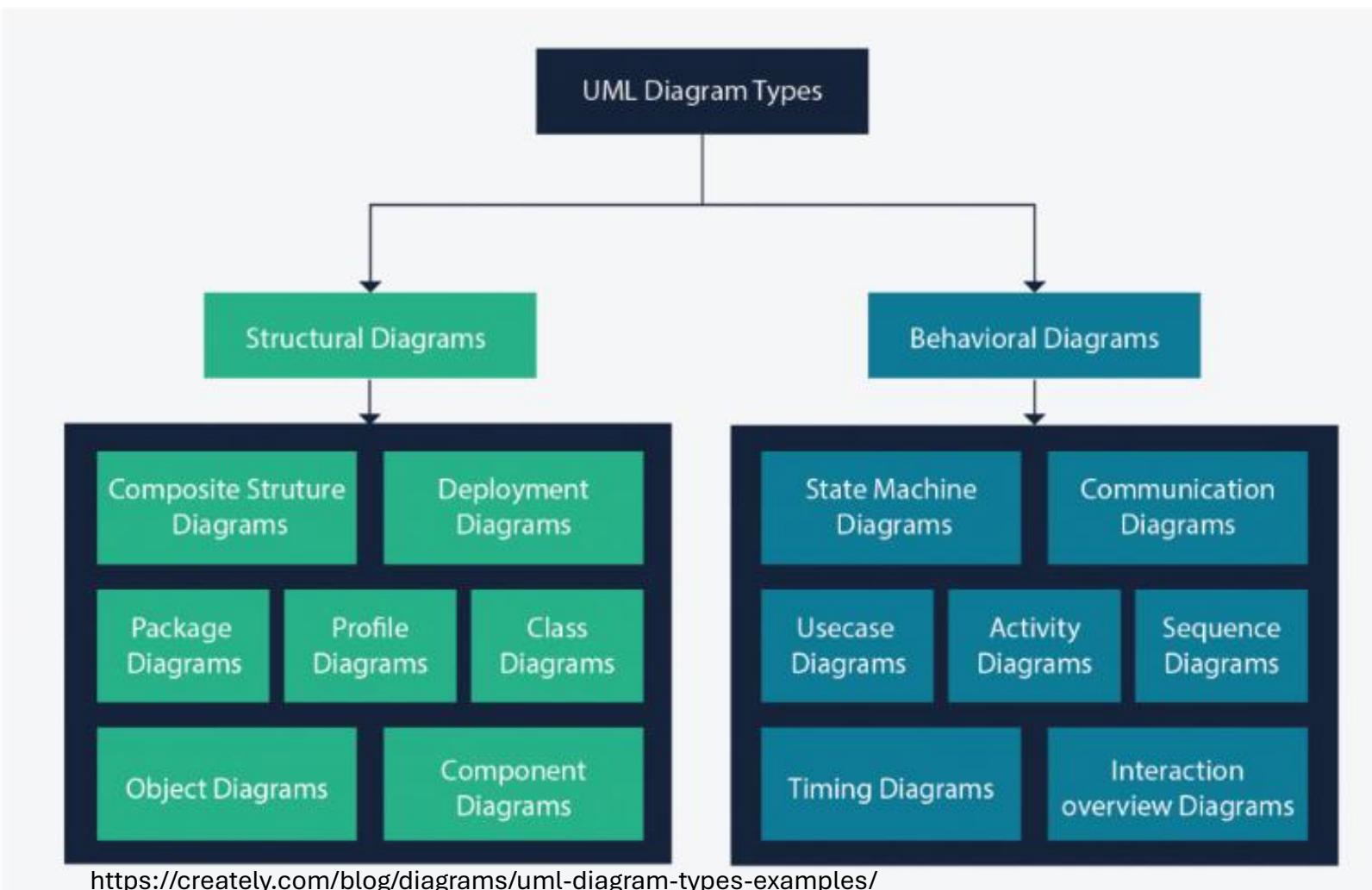


Antecedentes

- 1980 – La programación orientada a objetos
- Variedad de modelos para OO
- 1990 James Rumbaugh, Grady Booch e Ivar Jacobson crean UML.
- 1997 Uml estándar de la industria



UML



<https://creately.com/blog/diagrams/uml-diagram-types-examples/>

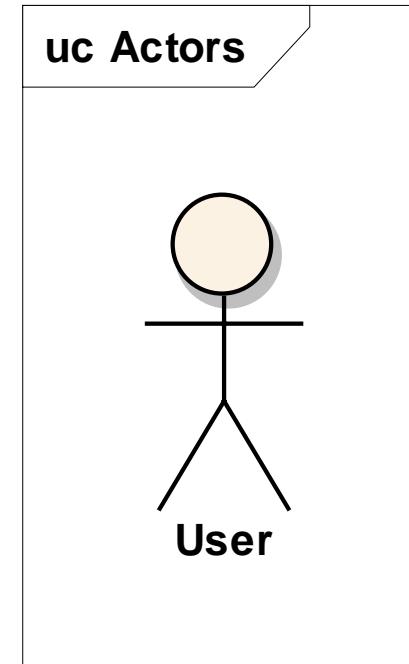
Diagrama de casos de uso

El Modelo de casos de uso es un catálogo de la funcionalidad del sistema descrita usando Casos de uso UML.

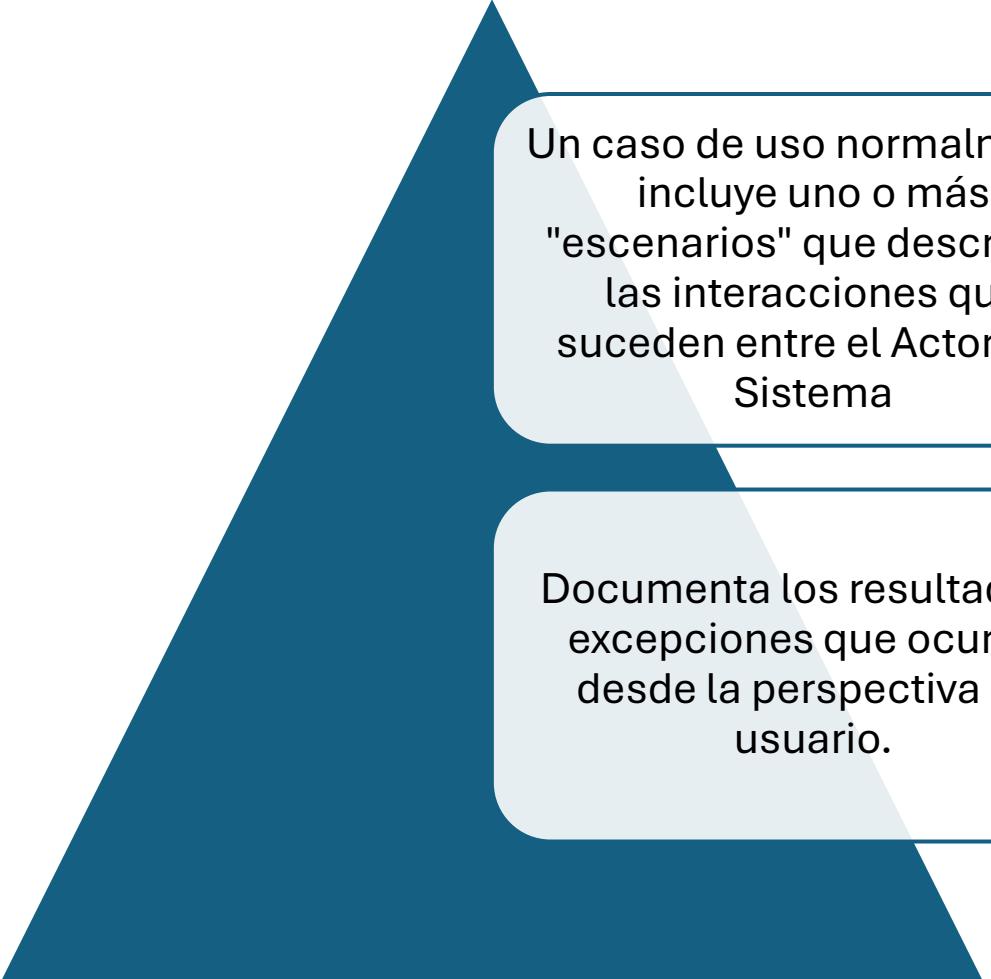
Cada Caso de uso representa una sola interacción repetida que un usuario o "actor" experimenta cuando usa el sistema.

Actores

- Los actores son los usuarios del sistema que se modelan.
- Cada actor tendrá un rol bien definido, y en el contexto de ese rol tiene interacciones útiles con el sistema.
- Una persona puede realizar el rol de más de un Actor.
- El rol de un actor se puede realizar por un sistema no humano, como otro programa de computadora.

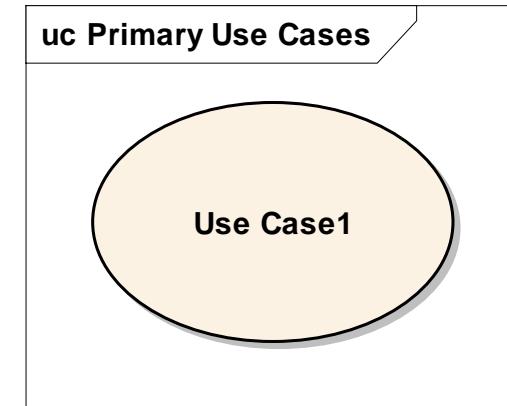


Casos de uso

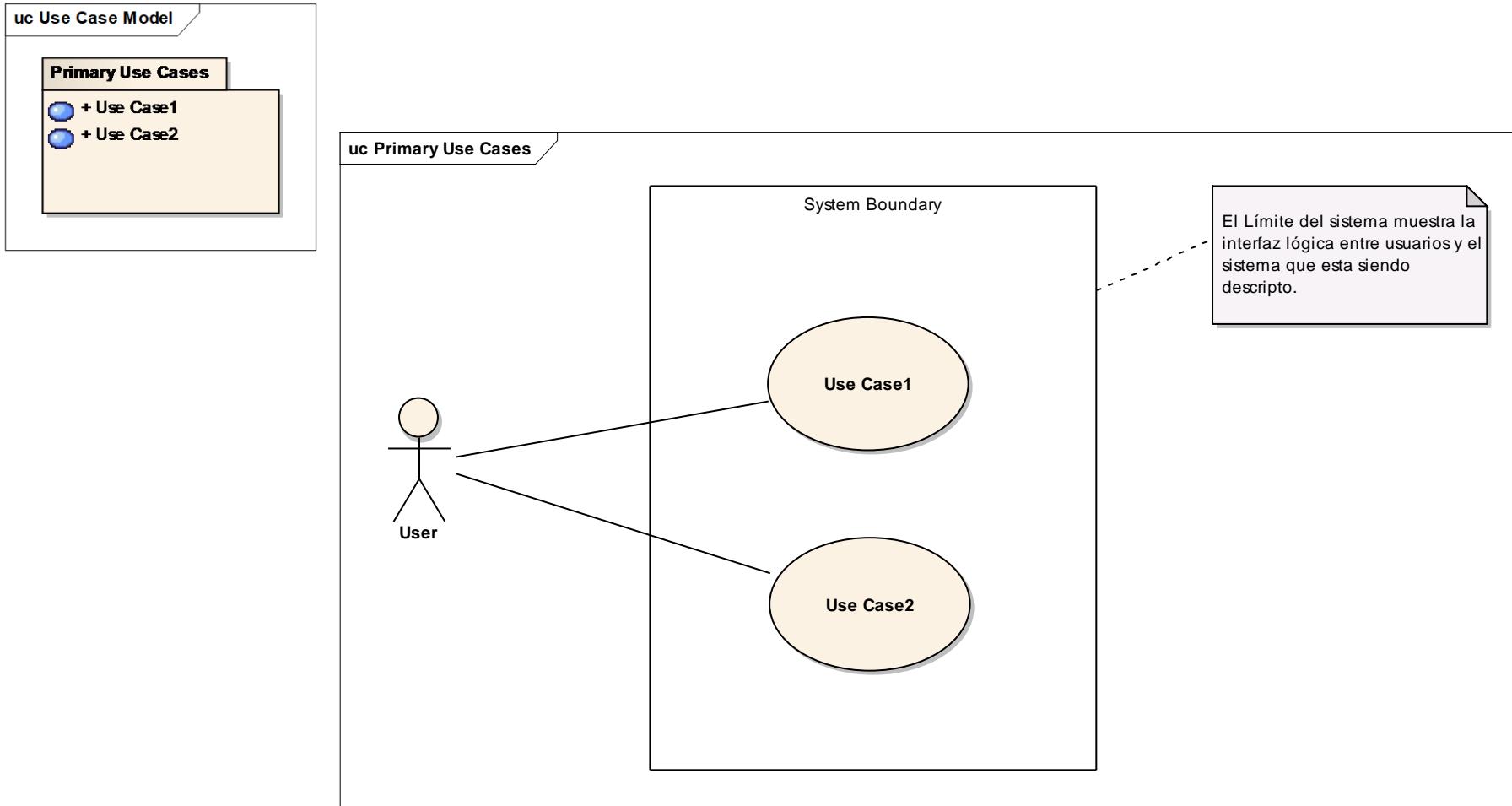


Un caso de uso normalmente incluye uno o más "escenarios" que describen las interacciones que suceden entre el Actor y el Sistema

Documenta los resultados y excepciones que ocurren desde la perspectiva del usuario.



Casos de uso



Relaciones entre casos de uso

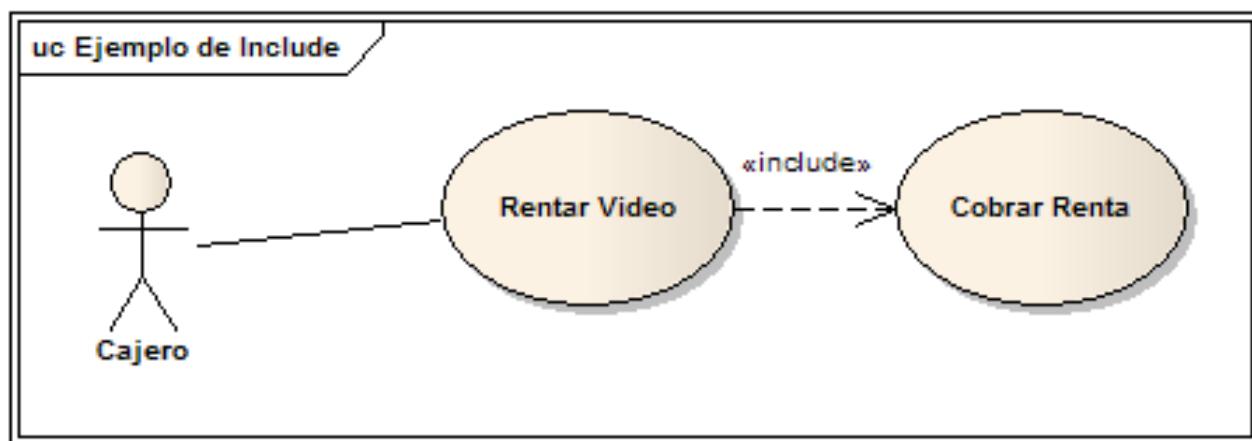
Son relaciones que usamos para ligar gráficamente dos casos de uso, cuyos flujos de eventos están unidos, normalmente en una sola sesión del usuario.

Include

Extends

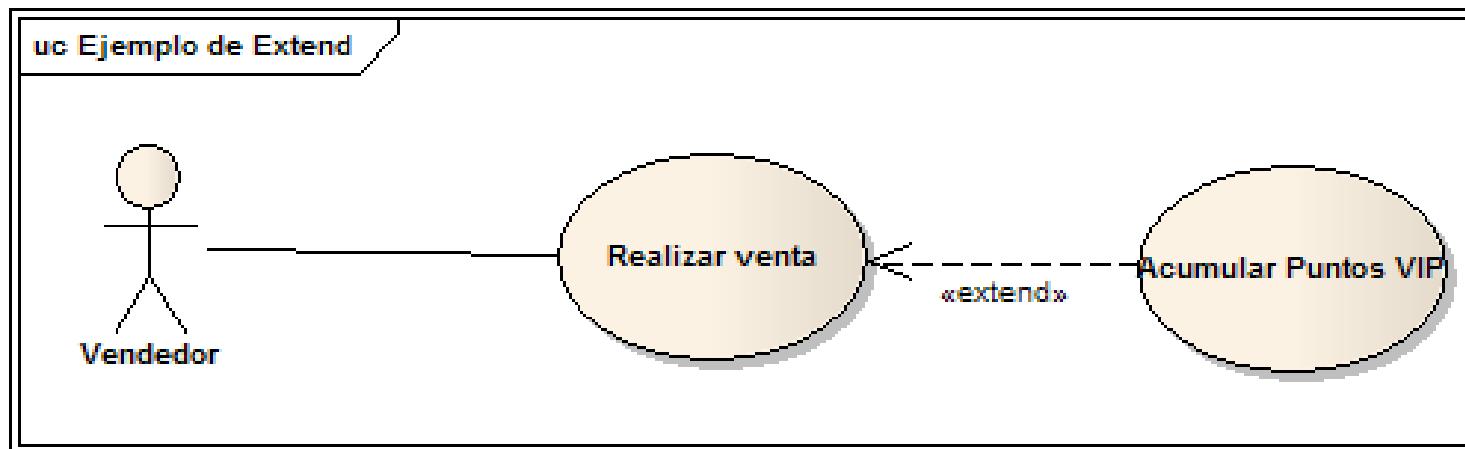
Include

- Cuando se relacionan dos casos de uso con un “include”, se dice que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido).
- Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien; pues no podría cumplir su objetivo

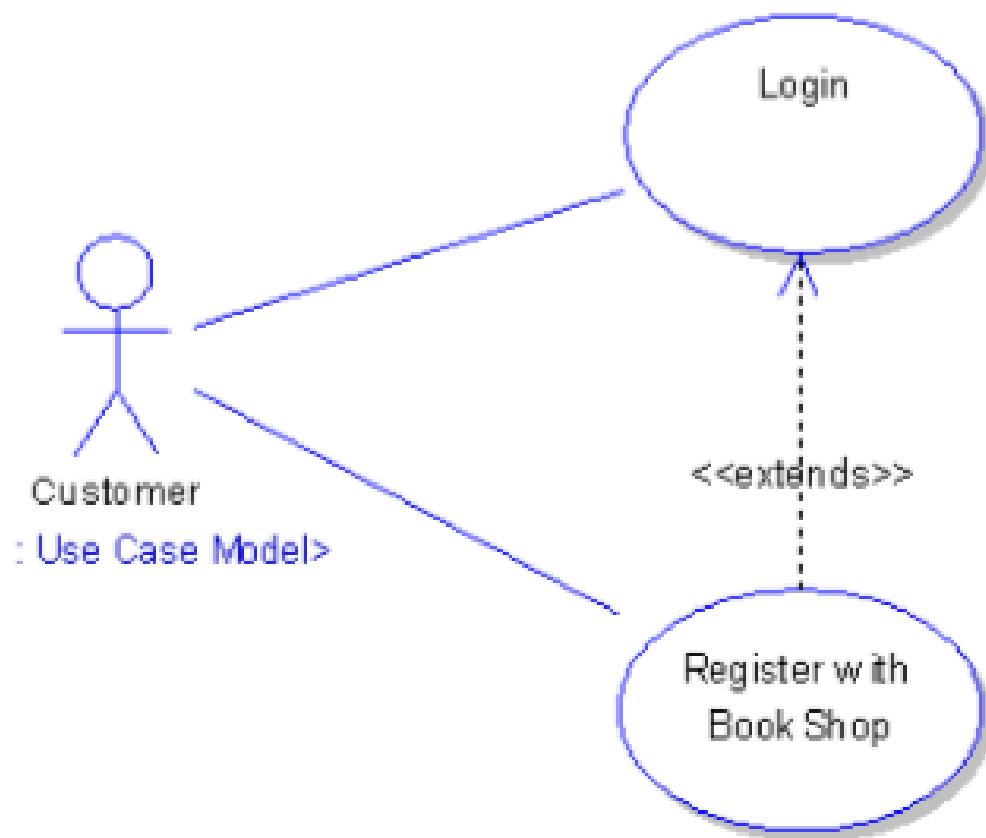


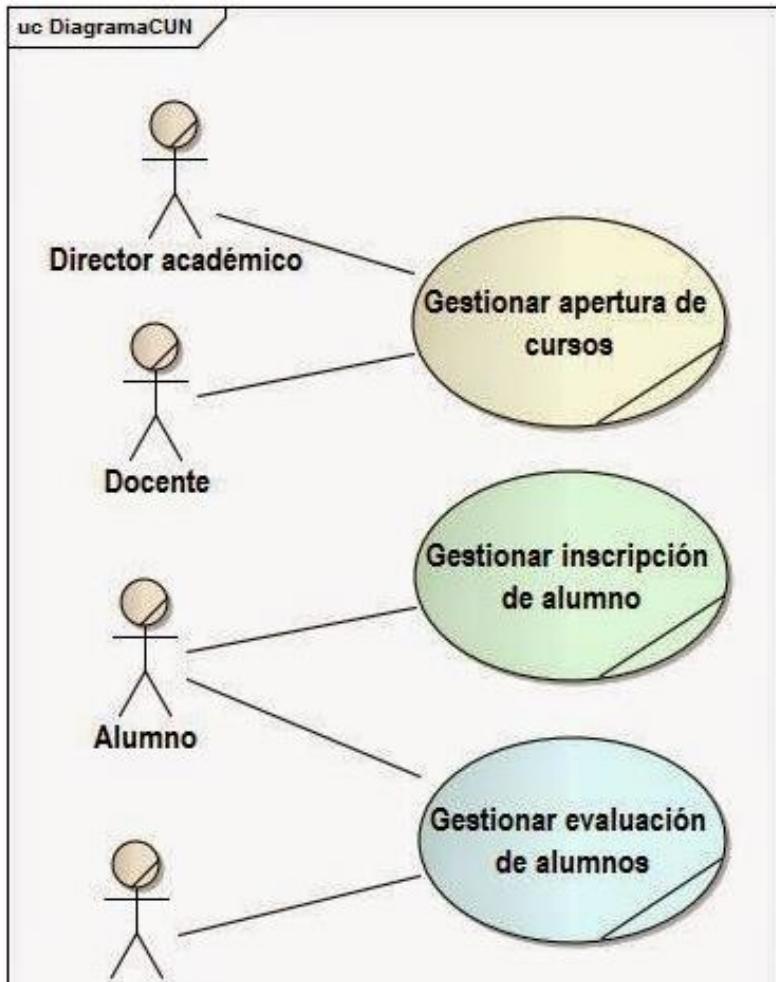
Extend

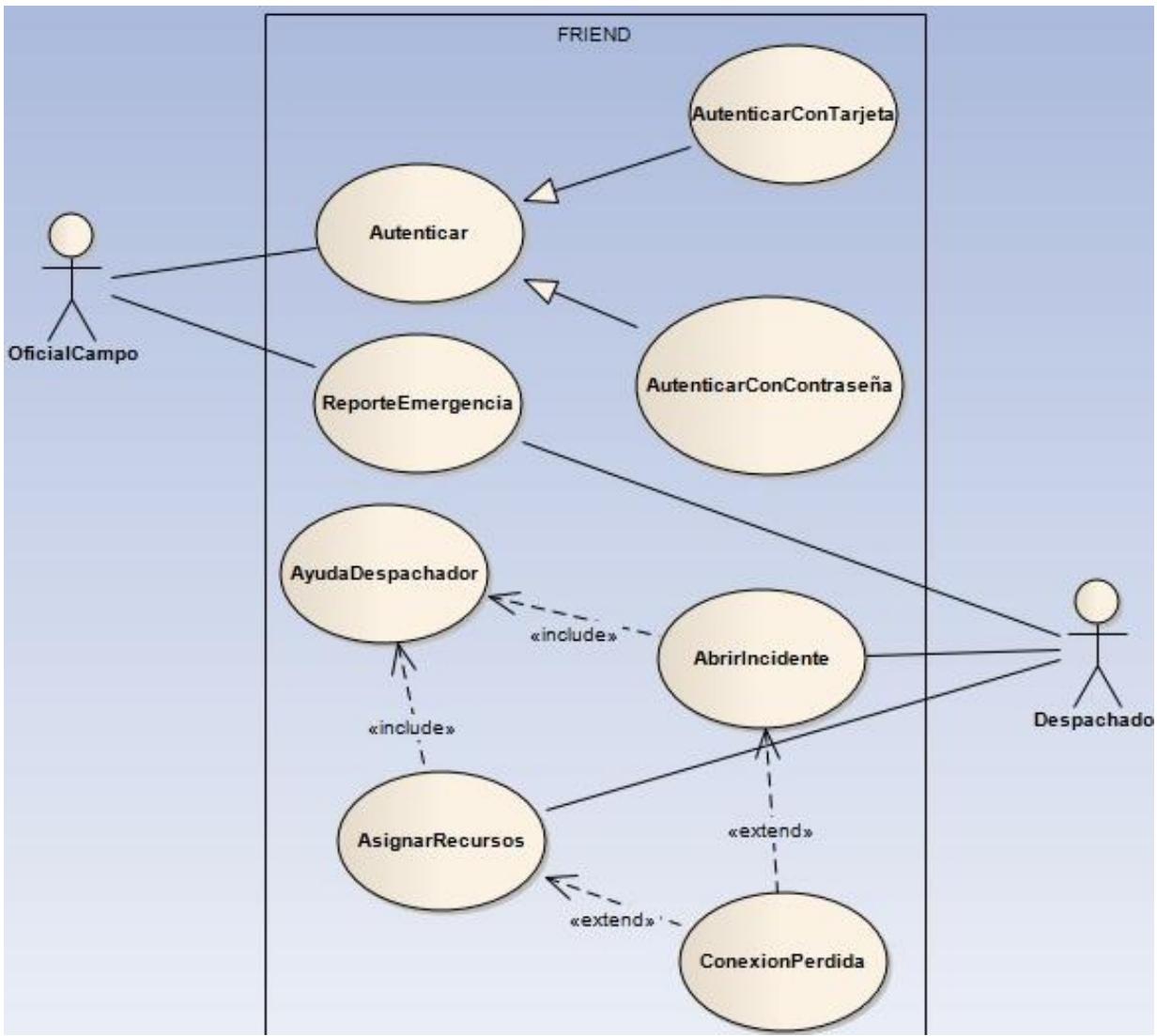
- Una de las diferencias básicas es que en el caso del “extend” hay situaciones en que el caso de uso de extensión no es indispensable que ocurra, y cuando lo hace ofrece un valor extra (extiende) al objetivo original del caso de uso base.



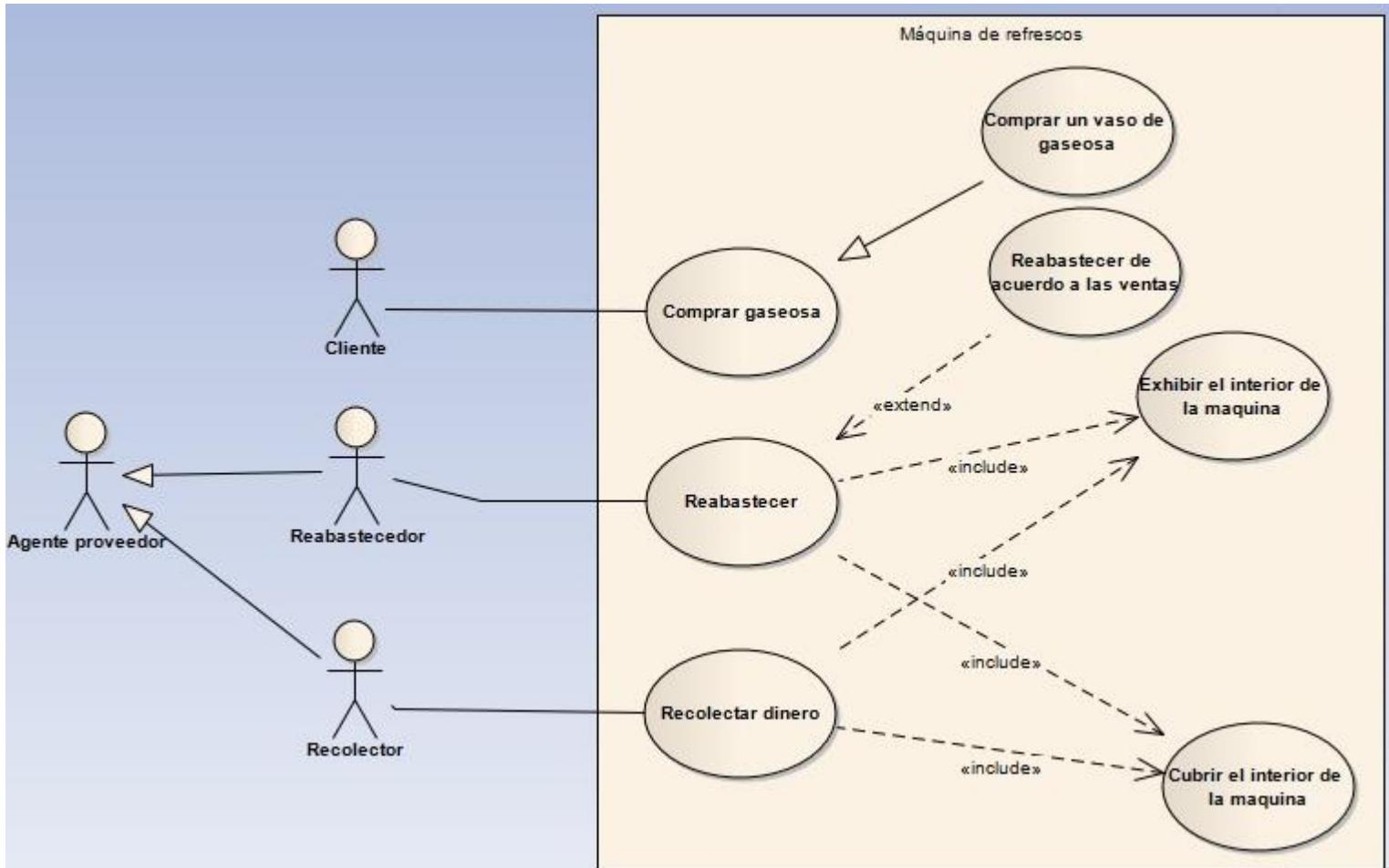
Ejemplos







Ejemplos



Herramientas CASE – Practica del problema de nomina – Casos de Uso

Problema Base

- Se desea hacer un sistema para gestión de monitorias académicas. Donde un alumno o usuario puede consultar, seleccionar pagar una monitoria. Los monitores, alumnos o egresados proponen, modifican y eliminan las monitorias en un tema y horario predeterminado. Se debe proveer por reportes y comentarios sobre el servicio.



Problema propuesto

- Se desea intercambiar, vender, regalar libros que ya no se usan. La idea es que una persona pueda ingresar un libro bajo distintos tipos de intercambio para que otro usuario lo use. Deben distinguirse personas que ingresan los libros y los que los adquieren. Se requiere crear un software que gestione el intercambio de libros, pueda controlar, sacar estadísticas, manejar pagos y opiniones acerca del servicio.

