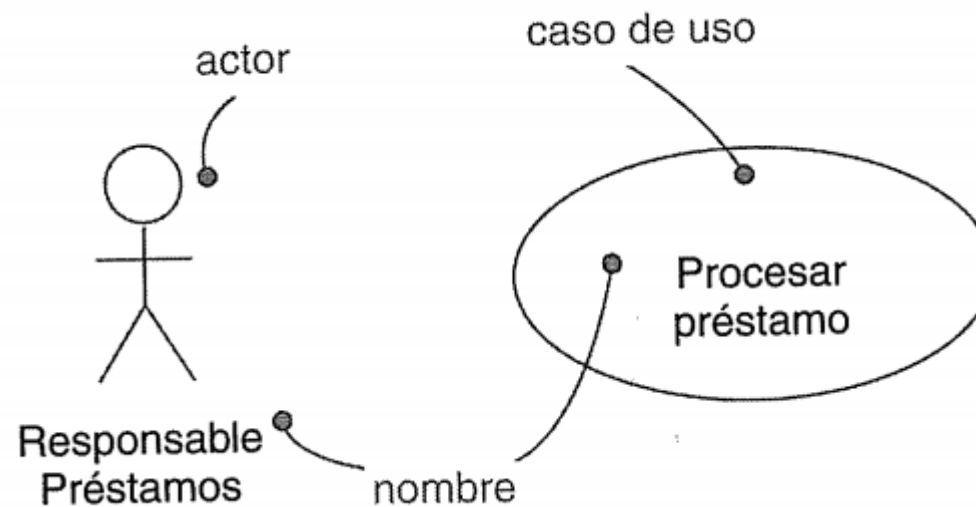


# Modelamiento

## Clase 02





# UML-Modelamiento del Sistema



# Requerimientos

Se describe las funcionalidades que debe ofrecer el sistema informático o software, estas funcionalidades deben incluir sus respectivas restricciones.



# Requerimientos

## Funcionales

Se indica exclusivamente los requerimientos de funcionamiento del sistema que apoyan y optimizan a los procesos del negocio.

## No funcionales

Requerimientos que no participan directamente en los procesos del negocio pero están involucrados. Por ejemplo la seguridad de accesos, equipamiento, patrones o metodologías de desarrollo a fin de tener un buen rendimiento, etc.



# Modelamiento del Sistema

Son las indicaciones gráficas de las interacciones de los sistemas informáticos (y procesos adjuntos) con los actores y entidades relacionados.

# Casos de Uso (CU)

## Actor

Un actor representa el rol jugado por una persona o cosa que actúa con el sistema.

*“Cliente, Administrador, Usuario no Registrado (Autenticado), Usuario Registrado (Autenticado), Jefe de Compras, Jefe de Personal, Moderador, Jefe de Departamento, Obrero de Planta, Supervisor...”*

**¿Actor o Rol?:** Sería mejor usar la palabra rol, pero algunos piensan que “Actor” fue usado debido a una mala traducción del Sueco

# Casos de Uso (CU)

## Actor

**NOTA:** NO TODOS los interesados en el sistema (stakeholders) son actores, sólo son actores aquellos que utilizarán el sistema

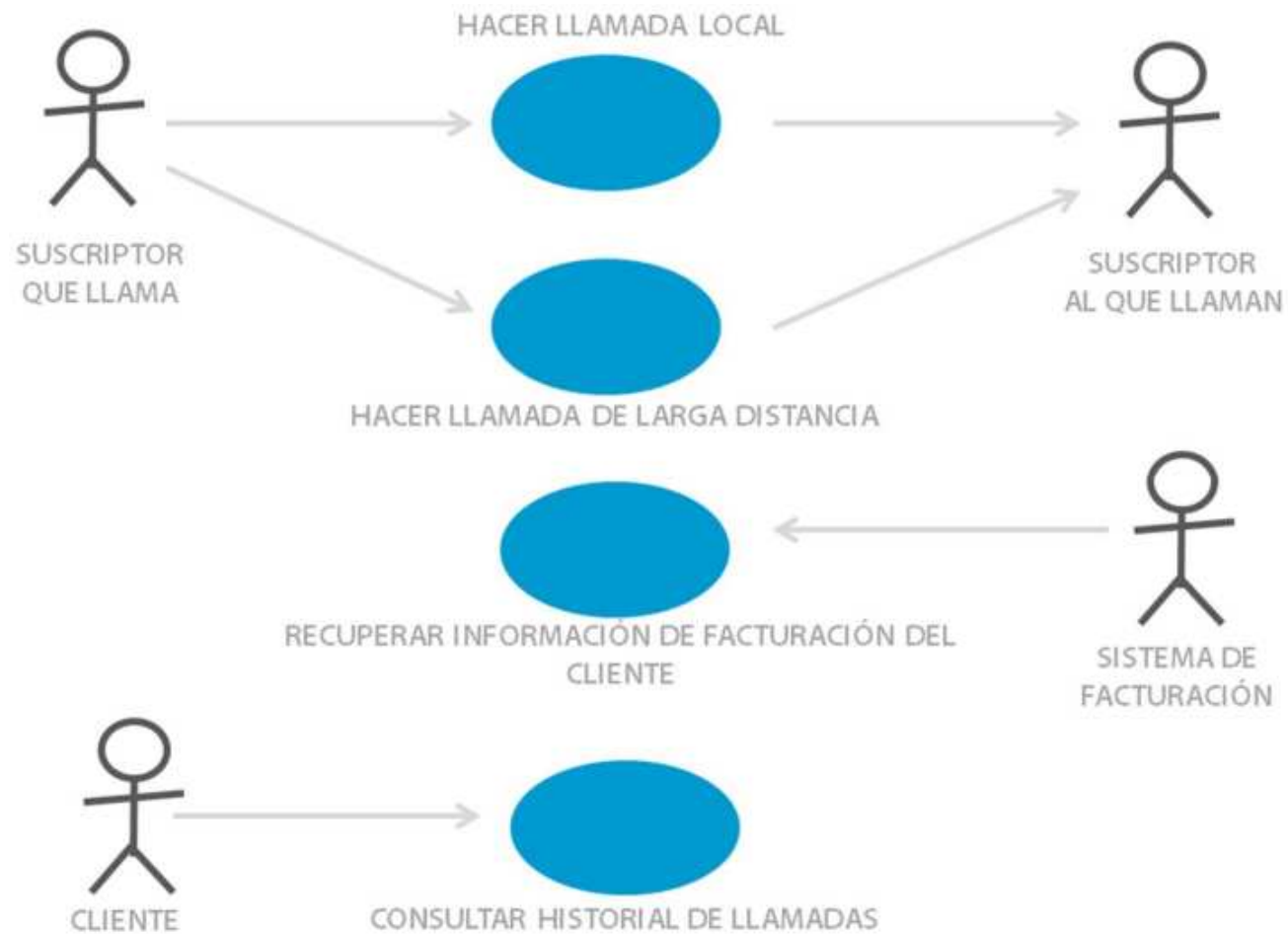
# Casos de Uso (CU)

## Actor

Nombre:	<i>&lt;nombre del actor&gt;</i>
Descripción:	<i>&lt;descripción del actor&gt;</i>
Nombre:	Usuario no Autenticado
Descripción:	Representa a un usuario que no se a identificado frente al sistema. Generalmente estos usuarios deberían poder registrarse (crear un nuevo usuario) o ingresar al sistema para transformarse en usuarios autenticados, en moderadores o en administradores del sistema



# Casos de Uso (CU)



# Casos de Uso

## ¿Por qué casos de uso?

- Proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales.
- Dirigen todo el proceso de desarrollo debido a que la mayoría de las actividades como el análisis, diseño y prueba se llevan a cabo partiendo de los casos de uso. El diseño y la prueba pueden también planificarse y coordinarse en términos de casos de usos.

# Casos de Uso

Que un diagrama de casos de uso sea fácil de crear es un elogio implícito para el UML. Hallar los casos de uso correctos y registrar sus responsabilidades en forma correcta es la decepción. Hallar los casos de uso correctos y describirlos de manera adecuada es el proceso crítico que impide que los listos ingenieros de software pasen por alto necesidades críticas y que inventen de manera innecesaria. En pocas palabras, los diagramas de casos de uso constituyen un macrorregistro de lo que usted quiere estructurar.

En el párrafo anterior, usé el prefijo *macro*. Macro en este contexto sencillamente significa “grande”. Los grandes objetivos, o macroobjetivos, son los que se mencionan como los argumentos, o razones, poderosos de la empresa para hacer algo. En los diagramas de casos de uso se captan los objetivos grandes, poderosos. En el texto de esos casos se captan los detalles de apoyo.

# Casos de Uso

## Documentando el CU

Nombre:	<i>&lt;nombre del caso de uso&gt;</i>
Autor:	<i>&lt;nombre del autor (o autores) del caso de uso&gt;</i>
Fecha:	<i>&lt;fecha de creación del caso de uso&gt;</i>
Descripción:	<i>&lt;breve descripción del caso de uso&gt;</i>
Actores:	<i>&lt;actores participantes en el caso de uso&gt;</i>
Precondiciones:	<i>&lt;condiciones que deben cumplirse para poder ejecutar el caso de uso&gt;</i>
Flujo Normal:	<i>&lt;flujo normal (feliz) de ejecución del caso de uso&gt;</i>
Flujo Alternativo:	<i>&lt;flujos alternativos de ejecución del caso de uso&gt;</i>
Poscondiciones:	<i>&lt;condiciones que deben cumplirse al finalizar la ejecución del caso de uso&gt;</i>

# Casos de Uso

## Documentando el CU

### FLUJO BÁSICO

1. Insertar Tarjeta
2. Validar Tarjeta
3. Seleccionar Retiro de Efectivo
4. Seleccionar Cuenta
5. Confirmar Disponibilidad de Fondos
6. Retornar Tarjeta
7. Entregar Efectivo

### FLUJOS ALTERNATIVOS

- A1 Tarjeta Inválida
- A2 Cantidad No-Estándar
- A3 Recibo Requerido
- A4 Fondos Insuficientes en ATM
- A5 Fondos Insuficientes en cuenta
- A6 Causaría Sobregiro
- A7 Tarjeta Atascada
- A8 Efectivo dejado atrás
- etc..

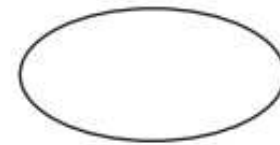
# Casos de Uso



Actor



Conector



Caso de uso

# Casos de Uso

El símbolo del caso de uso se utiliza para representar capacidades.

Al caso de uso se le da un nombre y una descripción mediante un texto.

Este último debe describir cómo inicia y finaliza el caso de uso, e incluye una descripción de la capacidad descrita por el nombre de la misma, así como escenarios de apoyo y requisitos no funcionales.



# Casos de Uso

## Conectores

Dado que los diagramas de casos de uso tienen múltiples actores y en virtud de que los casos de uso pueden estar asociados con los actores y con otros casos de uso, se utilizan **los conectores para indicar la manera en que ambos están asociados**. Además, los estilos de conectores pueden cambiar para transmitir más información acerca de la relación entre los actores y los casos de uso. Por último, los conectores pueden tener adornos y anotaciones que suministran incluso más información.



# Casos de Uso

## Conectores-Asociación

Un conector de línea simple se llama asociación y se usa para mostrar cuáles actores están relacionados con cuáles casos de uso.



# Casos de Uso

## Conectores-Dependencia

Un segundo estilo de conector es una línea punteada con una flecha direccional. Este estilo de conector se conoce como dependencia. La flecha apunta hacia el caso de uso del que depende.

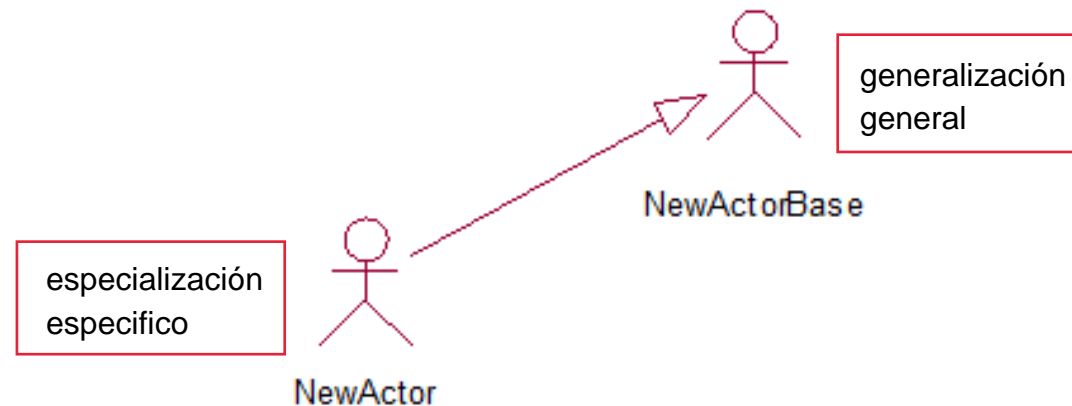


# Casos de Uso

## Conectores-Generalización

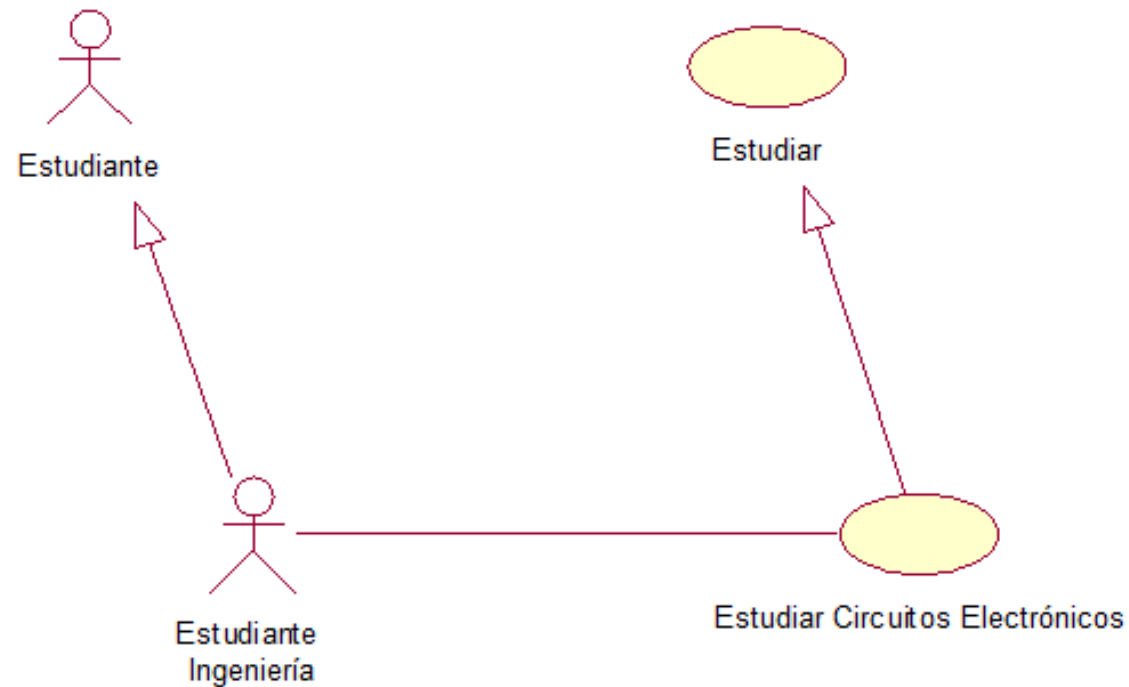
Un tercer estilo de conector es una línea dirigida con un triángulo hueco, al cual se le conoce como generalización. La palabra generalización en el UML significa “herencia”.

Cuando mostramos una relación de generalización entre dos actores o dos casos de uso, estamos indicando que el actor o el caso de uso “hijos” son un caso del actor o uso básico y algo más.



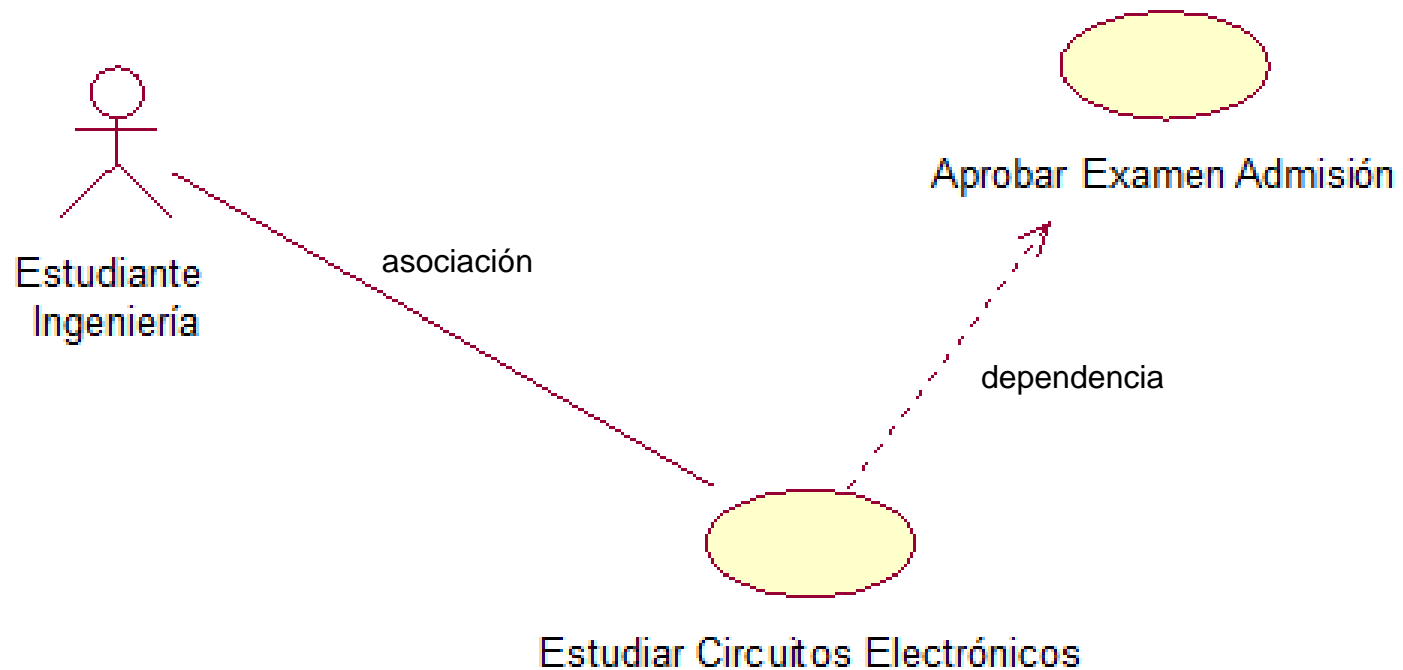
# Casos de Uso

## Conectores-Ejemplos



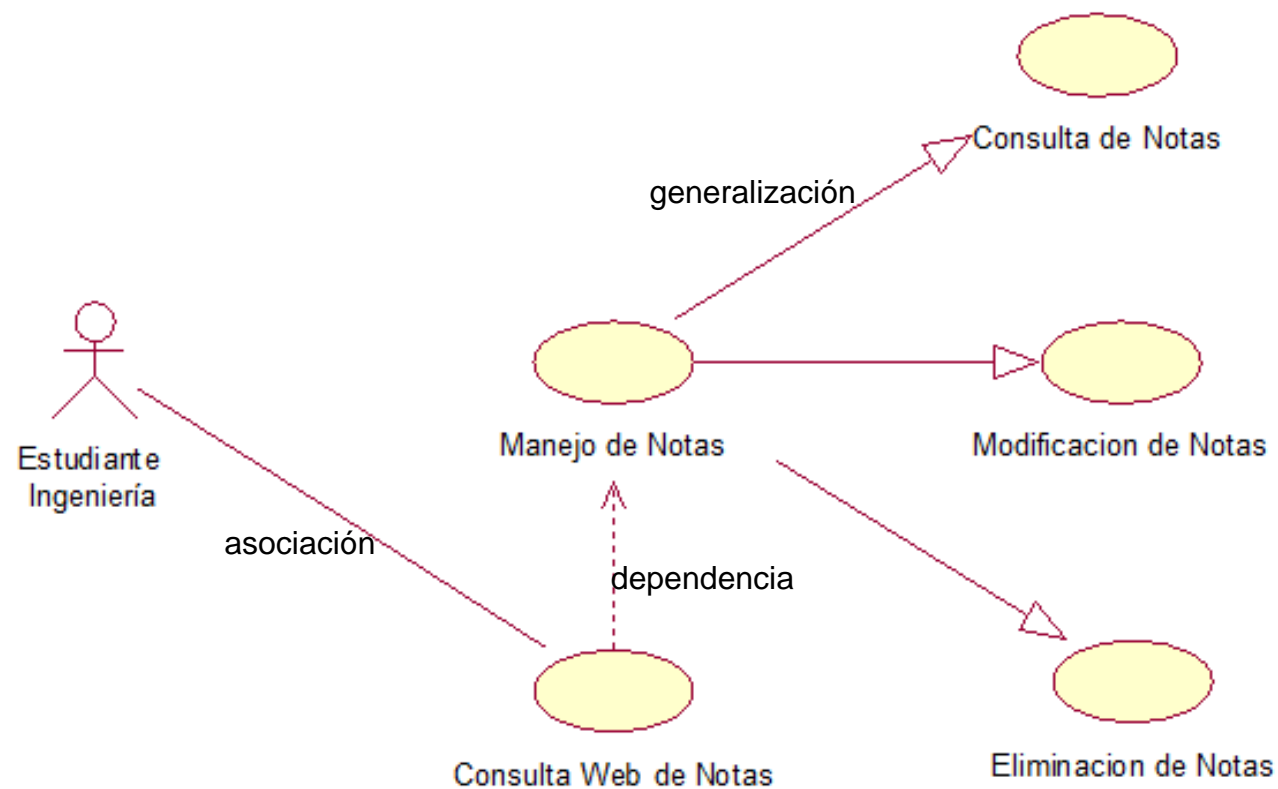
# Casos de Uso

## Conectores-Ejemplos



# Casos de Uso

## Conectores-Ejemplos



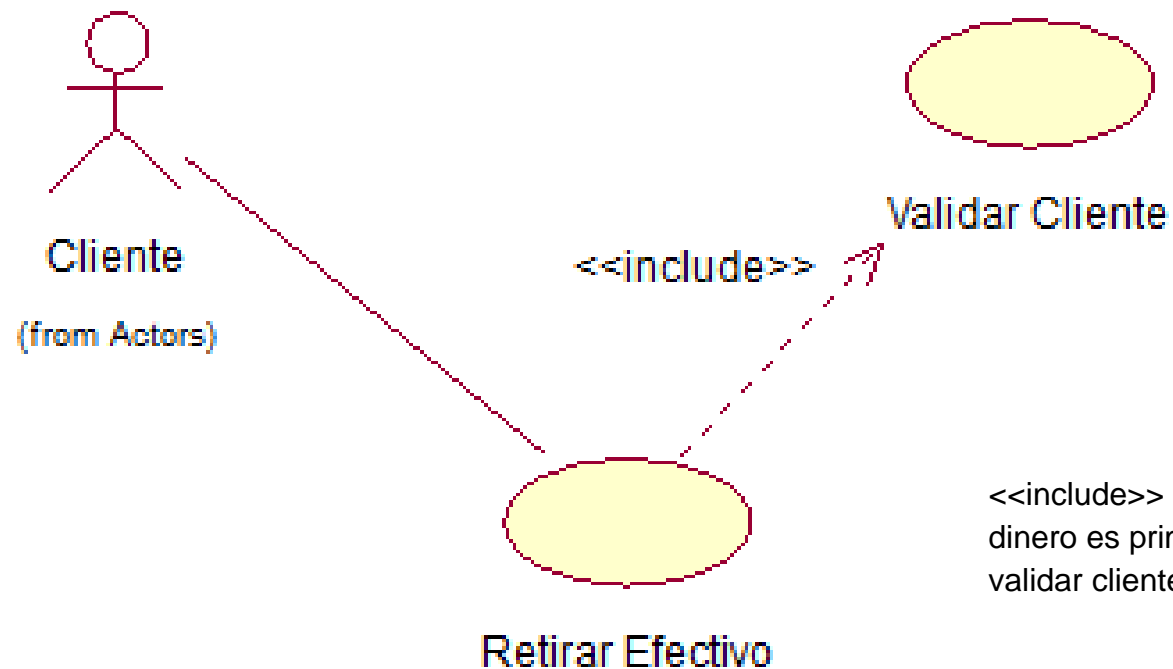
# Casos de Uso

## Conectores-Estereotipo <<include>>

Cuando un caso de uso necesita los servicios de otro caso de uso entonces se dice que el caso de uso dependiente incluye el caso de uso del que depende. (En código, una relación include (de inclusión) se implementa como reutilización de código). Include también se conoce como "uses".

# Casos de Uso

## Conectores-Estereotipo <<include>>

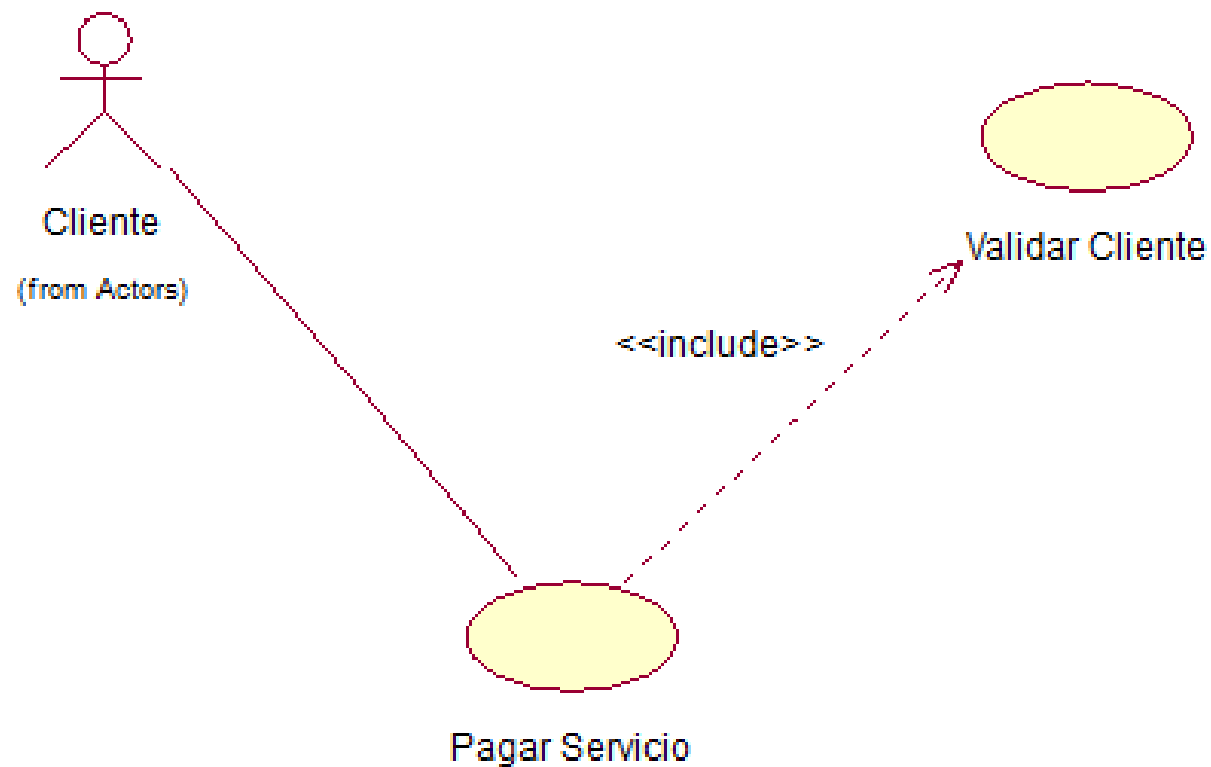


<<include>> hace referencia que Retirar dinero es primero o mas importante que validar cliente y se da como segunda



# Casos de Uso

## Conectores-Estereotipo <<include>>



# Casos de Uso

## Conectores-Estereotipo <<extend>>

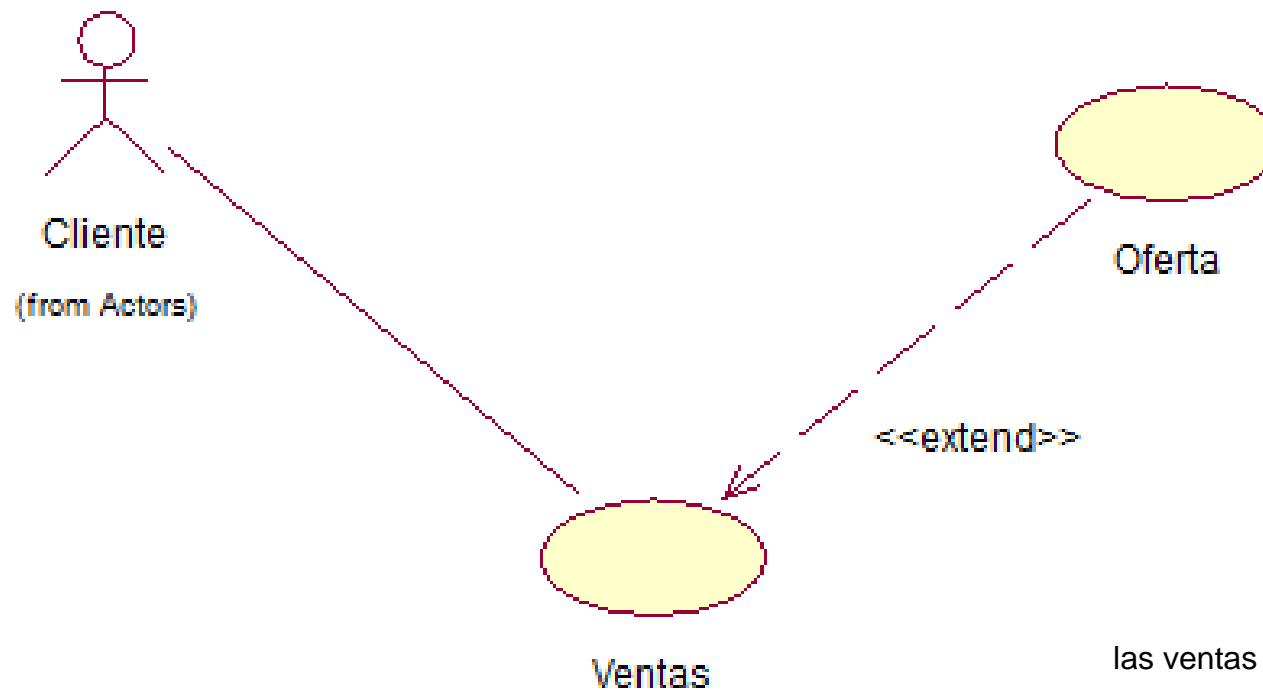
Es si el caso de uso que se extiende realiza acciones mas allá del caso de uso extendido, cumpliendo las siguientes condiciones:

- Denota un “punto de extensión”,
- Sirve para representar las condiciones de error y situaciones poco usuales.

*"El CU base contiene lo necesario para funcionar, mientras que el CU que lo extiende le añade funcionalidad".*

# Casos de Uso

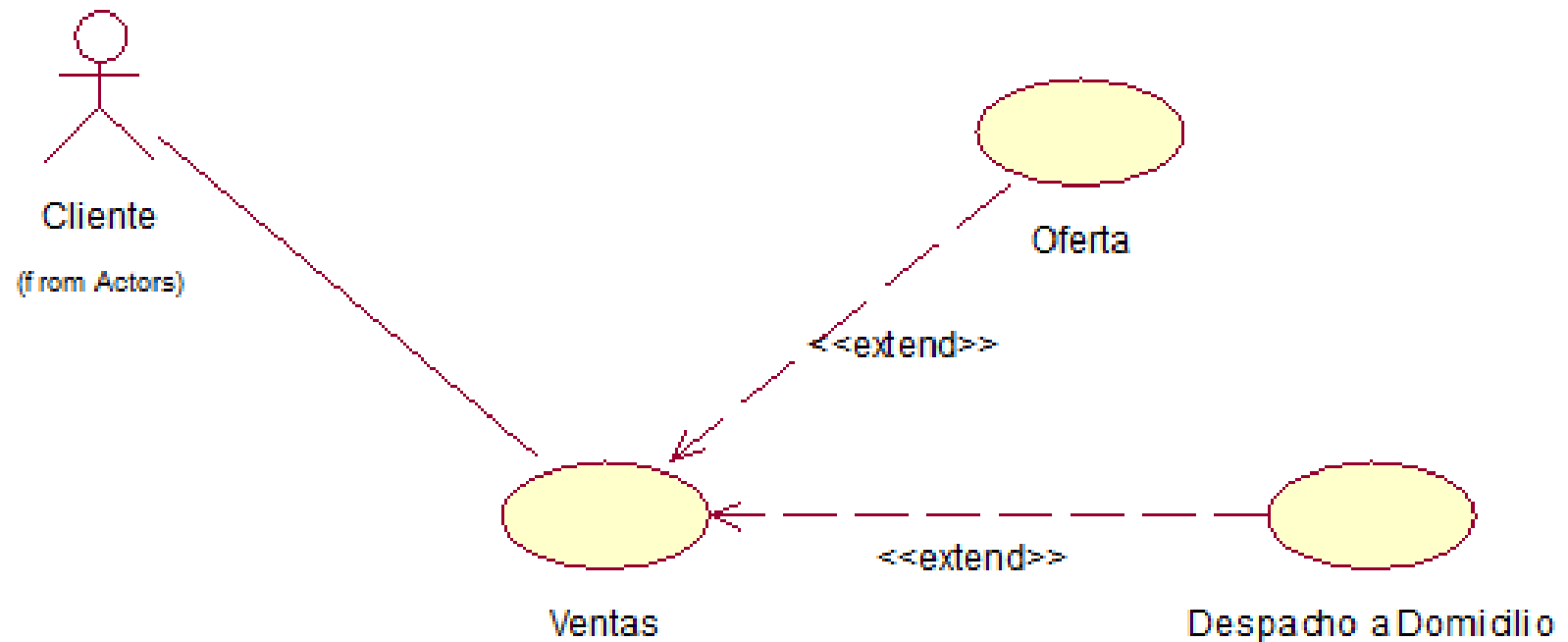
## Conectores-Estereotipo <<extend>>



las ventas son mas importantes que las  
ventas  
solo si hay necesidad se va a las ofertas

# Casos de Uso

## Conectores-Estereotipo <<extend>>



el despacho a domicilio solo se da si hay una venta, lo mismo la oferta.

# Casos de Uso

## Práctica

