



# Administración de Requerimientos con Casos de Uso

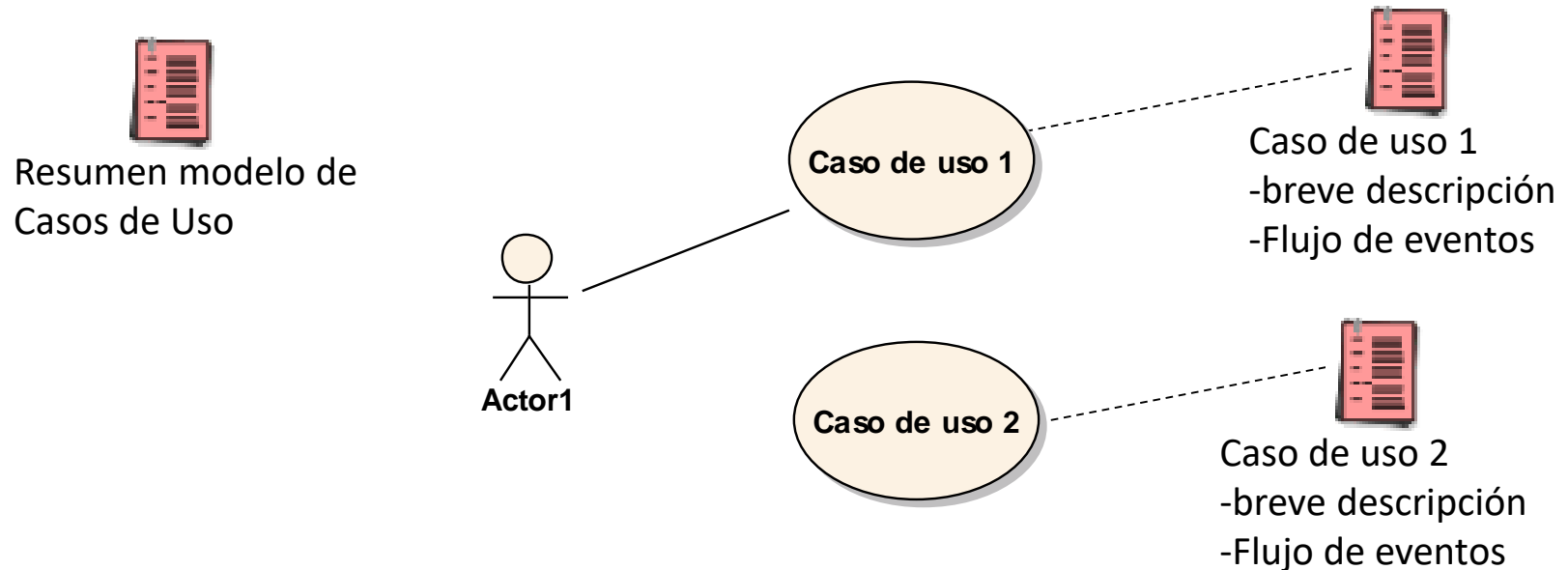
- Ilustran por qué el sistema es necesario
  - Casos de uso: para que usen los usuarios el sistema
  - Actores: quiénes y qué necesitan para interactuar con el sistema
- Da contexto para los requerimientos
- Facilitan el entendimiento
- Facilitan el acuerdo con los clientes
- Casos de uso son el camino para organizar requerimientos desde el punto de vista del usuario

## ■ ■ ¿Quiénes leen los casos de uso?

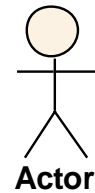
- Grupo de clientes
  - Clientes: aprueban lo que el sistema realizará
  - Usuarios: entender lo que el sistema realizará
- Grupo de desarrollo
  - Especificadores de requerimientos y casos de uso: refinan los requerimientos del sistema
  - Diseñadores: detectan clases en la etapa diseño
  - Testers: los usan como base para los casos de test
  - Administrador de proyecto: planificación y administración de recursos
  - Documentadores: para escribir la documentación

# ■ ■ Un modelo de casos de uso contiene diagramas y documentación

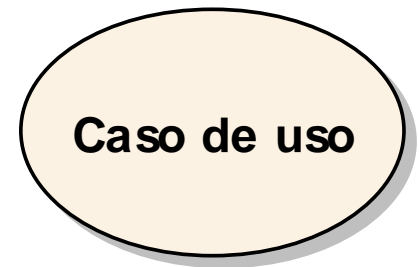
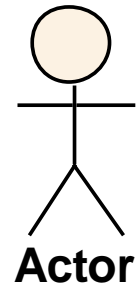
- Los diagramas dan una percepción visual del sistema.
- El texto da la descripción de los casos de uso y actores
- Lo más importante de un modelo de casos de uso, es el texto.  
La descripción es llamada flujo de eventos

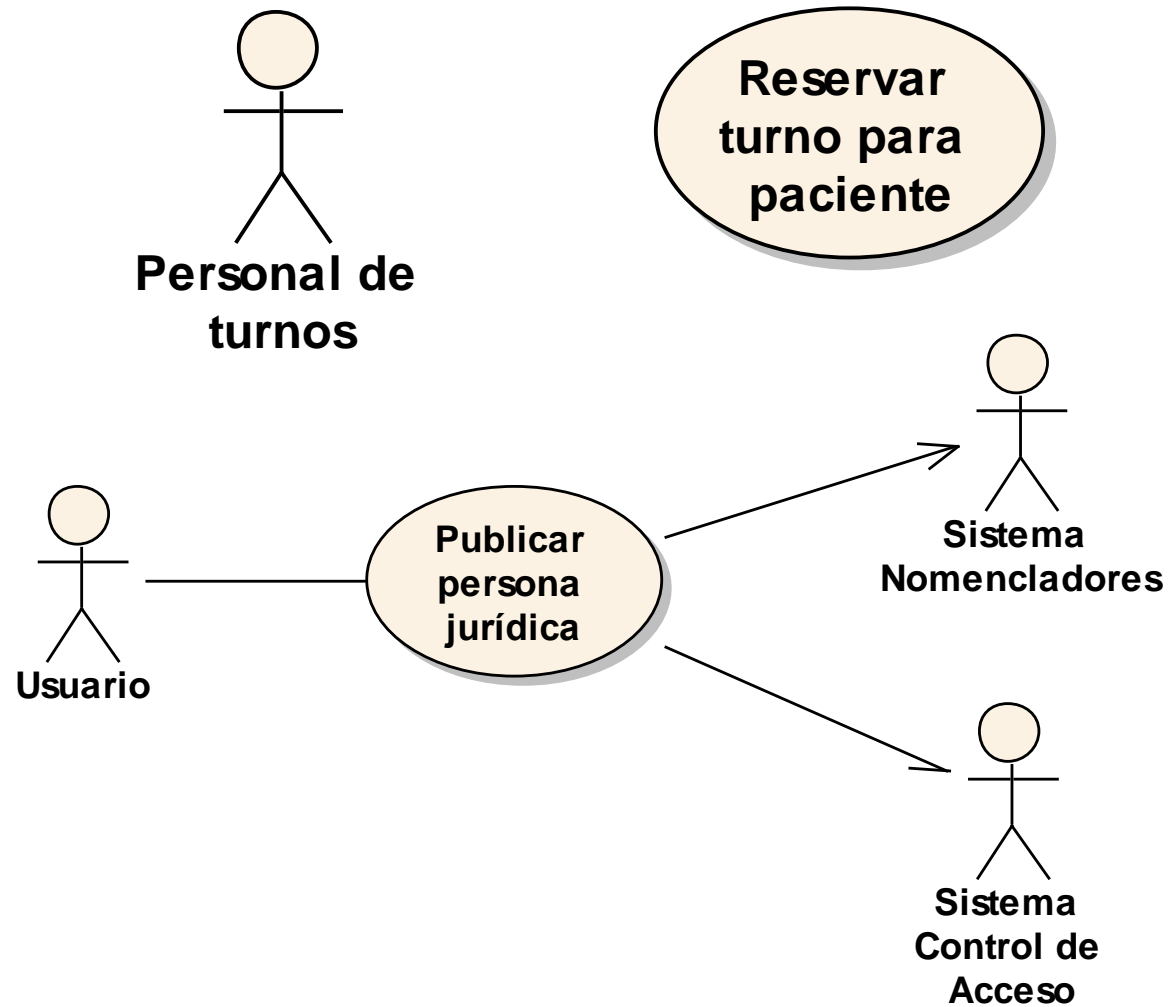


- Actor: alguien o alguna cosa fuera del sistema que interactúa con el sistema
- Caso de uso: es lo que un actor necesita usar del sistema para hacer algo

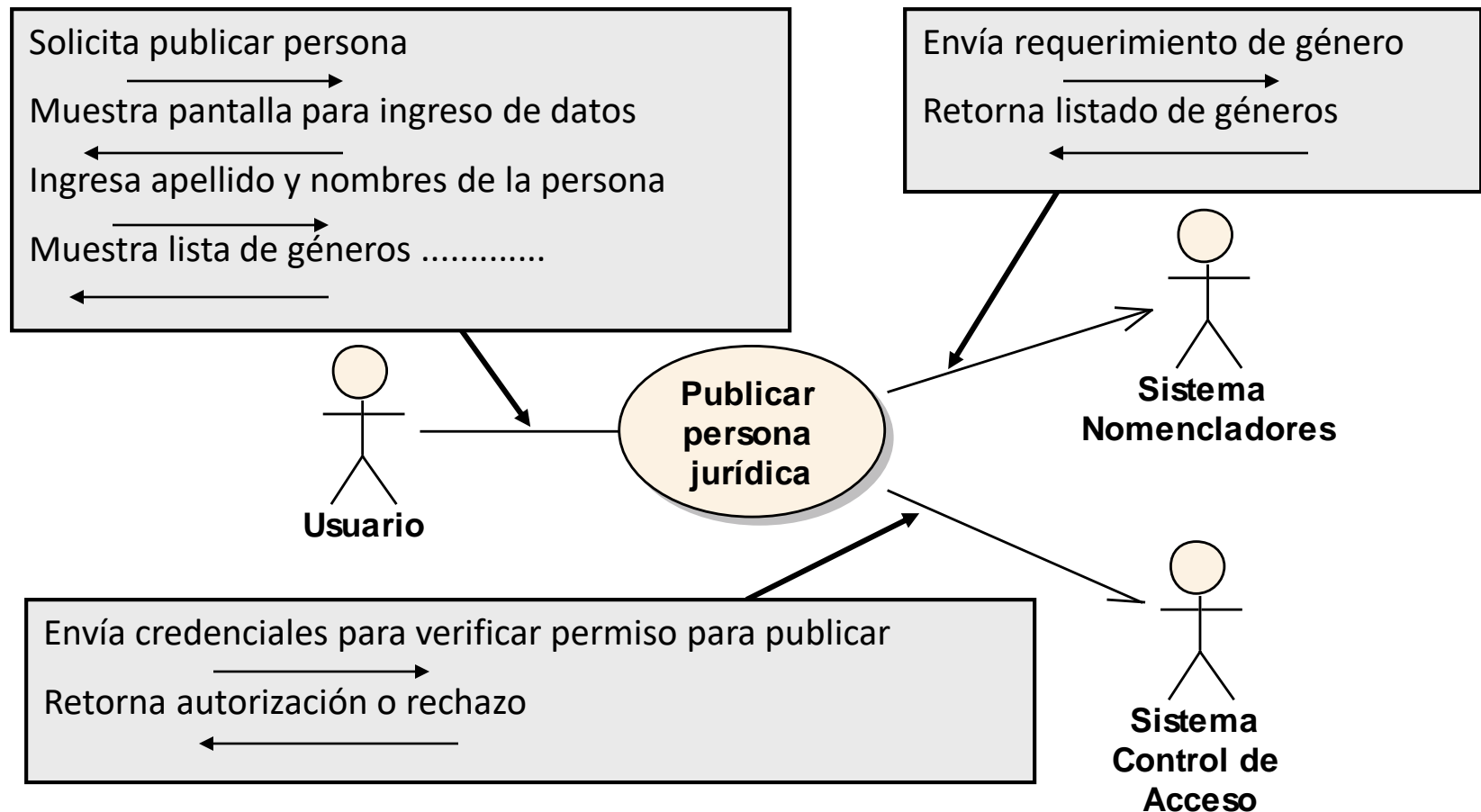


- Son un canal de comunicación entre un actor y un caso de uso
- Se utiliza una línea para su representación
- Opcionalmente se puede utilizar una flecha para indicar quién inicia la comunicación





## ■ ■ Cada asociación es un diálogo en sí misma



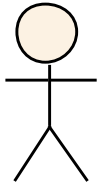
*Estos diálogos, también llamados escenarios, representan los detalles de un caso de uso*



## ■ ■ Un escenario es una instancia de un caso de uso

- Una instancia de un caso de uso es un flujo particular de eventos en el sistema, una secuencia específica de acciones que ilustran el comportamiento del sistema.
- Un caso de uso define un conjunto de escenarios relacionados que representan todas las secuencias posibles que pueden suceder, hasta que el valor resultante es alcanzado (o hasta que el sistema termina todos los intentos posibles).

# ■ ■ Ejemplos de escenarios



Personal de  
turnos



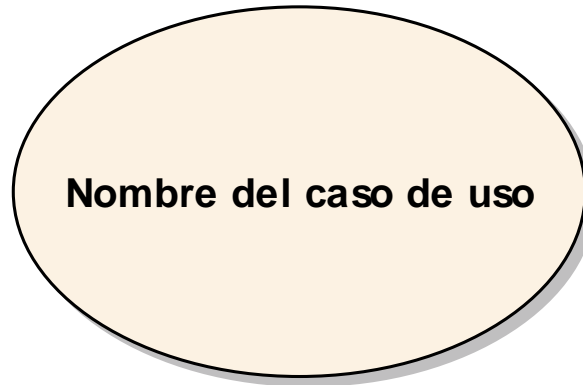
Reservar  
turno para  
paciente

## Escenario 1

Ingresar al sistema  
Aprobado el ingreso  
Ingresa el paciente a buscar  
Busca paciente en el sistema  
Retorna información del paciente  
Selecciona el paciente  
Busca turnos disponibles en el sistema  
Retorna turnos disponibles  
Asocia paciente a turno  
Fin del flujo.

## Escenario 2

Ingresar al sistema  
Aprobado el ingreso  
Ingresa el paciente a buscar  
Busca paciente en el sistema  
Paciente no encontrado  
Realiza una búsqueda extendida de pacientes  
Retorna lista de pacientes  
Selecciona el paciente  
Busca turnos disponibles en el sistema  
Retorna turnos disponibles  
Asocia paciente a turno  
Fin del flujo.



Un caso de uso define una **secuencia de acciones** **ejecutadas por un sistema** que producen un **resultado de valor observable para un actor**

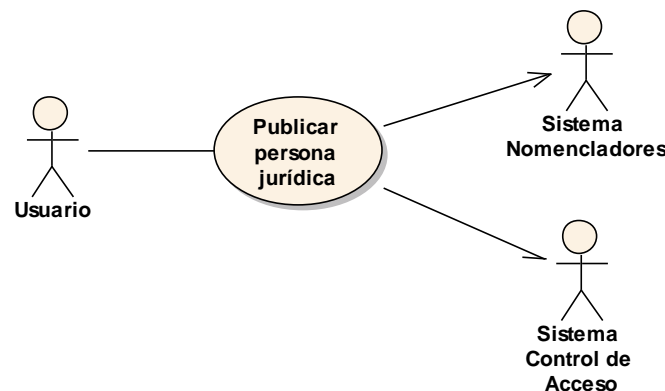
## ■ ■ ¿Qué es un caso de uso?

- Secuencia de acciones: son actividades **atómicas**, decisiones y requerimientos. Atómica significa que cada acción es ejecutada totalmente o no es ejecutada.
- Ejecutadas por el sistema: las acciones ejecutadas por el sistema son requerimientos funcionales.
- Resultado de valor observable: Si nadie obtiene valor desde el caso de uso luego el caso de uso es probablemente muy pequeño. Esto implica que el mismo debe ser combinado con otros casos de uso para proveer un conjunto completo de pasos que obtengan un resultado de valor para el actor

## ■ ■ ¿Qué es un caso de uso?

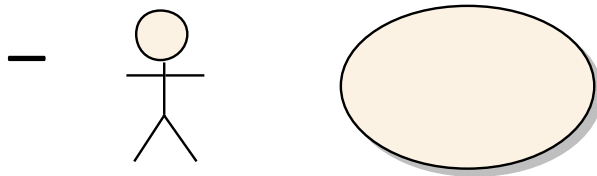
- Para un actor: indicando cual actor en particular recibe el valor, ayuda a evitar casos de uso muy grandes. Si más que un actor recibe valor desde un caso de uso, esto puede indicar que el caso de uso es muy grande, lo que se entiende que trata de hacer mucho para muchos actores.

Frecuentemente la secuencia de interacciones en un caso de uso involucra varios actores pero un actor recibe el resultado de valor. El actor que recibe el resultado de valor es llamado el **actor primario**. Este actor es que usualmente inicia el caso de uso.



- ¿Cuáles son los objetivos de cada actor?
  - ¿Porqué el actor necesita usar el sistema?
  - ¿Puede el actor crear, almacenar, modificar, borrar o leer datos en el sistema?. Si la respuesta es afirmativa ¿por qué?
  - ¿El actor necesita informarle al sistema a cerca de eventos externos o cambios?
  - ¿El actor necesita ser informado a cerca de ciertos eventos en el sistema?
- ¿El sistema provee al negocio el comportamiento correcto?

- Descripción textual de un caso de uso
  - Nombre
  - Breve descripción
  - Relación con actores
- Ejemplo
  - Reservar turno para paciente
  - Este caso de uso selecciona un paciente y luego le otorga un turno.



- Indicar el valor u objetivo
- Usar la forma activa: comenzar con un verbo
- Imaginar una lista de acciones a realizar
- Ejemplos de variaciones
  - Reservar turno para paciente
  - Reserva de turnos
  - Turnos
  - Usar el sistema de reserva de turnos



Desarrollar un CU es un proceso iterativo que involucra una secuencia de acciones.

1. Desarrollar un bosquejo del CU sin formato definido
2. Escribir la descripción general del CU
3. Bosquejar los flujos básicos que muestren los pasos que realizan el objetivo del CU
4. Bosquejar los flujos alternativos que incluyen los pasos involucrados en un comportamiento erróneo u opcional

## ■ ■ Porqué bosquejar los casos de uso

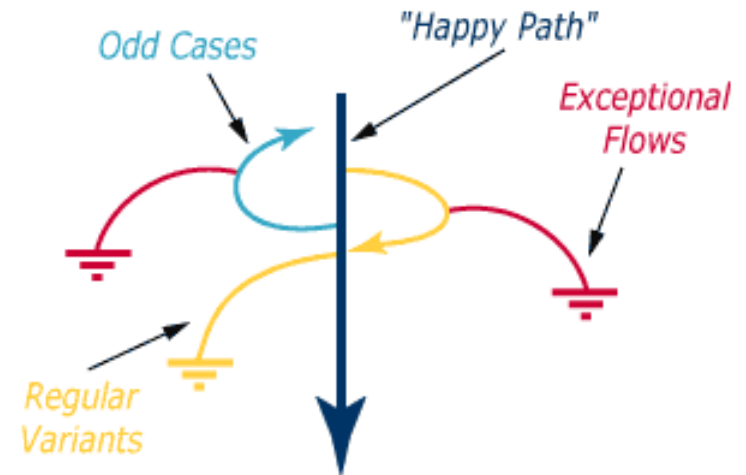
- No conocemos todo desde un primer momento. Bosquejar el comportamiento nos ayuda a descubrir qué es lo que no conocemos.
- El bosquejo de los CU crea un borrador para especificar totalmente el CU.
- Bosquejar nos ayuda a determinar si el CU es muy pequeño o muy grande para ser justo un CU.
- Una vez bosquejados los pasos, nos permite detectar si el CU no pertenece a otro CU, detectando que el CU no tiene a cargo una tarea relevante y por lo tanto puede no ser un CU
- Bosquejar agrega valor para identificar todos los posibles escenarios.

La estructura de flujos facilita seguir los diferentes escenarios y entender que sucede cuando ocurren las alternativas y donde ellas tienen lugar o finalizan.

- El flujo básico muestra los pasos necesarios para lograr el objetivo principal del CU
- Los flujos alternativos se “desvían” del flujo básico.
  - En algunos casos retorna al flujo básico
  - En otros casos finaliza la ejecución del CU

## ■ ■ Clasificación de los flujos de evento

- Un flujo básico
  - El escenario deseado
  - El escenario exitoso desde el comienzo al fin
- Varios flujos alternativos
  - Variantes regulares
  - Casos particulares
  - Flujos excepcionales (error)



## ■ ■ Guía para el bosquejo de flujos de evento

- Describa como el CU comienza y finaliza.
- Describa el intercambio de datos entre el actor y el CU
- No describa los detalles de las interfaces de usuario a menos que sean necesarias para entender el comportamiento del sistema.
- Describa los *flujos* de evento, no solo la funcionalidad. Comenzar cada acción con “Cuando el actor....”
- Describa solo los eventos que pertenecen al CU y no que sucede en otros CU o fuera del sistema.
- Evite terminología ambigua como “por ejemplo”, “etc.” e “información”
- Pormenorizar todos los flujos de evento. Todos los “qué” deben ser respondidos. Recuerde que el diseñador de casos de test utilizará este texto para identificar casos de test

- Clarificar detalles importantes en flujos de evento
  - Qué hace el actor
  - Qué hace el sistema en respuesta
  - Qué información es intercambiada
- Describir información adicional
  - Pre condiciones
  - Post condiciones
- Especificar requerimientos de software

- Cada CU
  - Describe acciones que el sistema realiza para generar algo de valor para el actor
  - Muestra como un actor utiliza la funcionalidad del sistema
  - Modela un diálogo entre el sistema y los actores
  - Es un completo y significativo flujo de eventos desde la perspectiva de un actor en particular

[Ver definición de escenario](#)

- Estructurar los flujos en pasos
- Numerar cada paso
- Si resulta necesario, describir el paso
- Describir cada paso teniendo en cuenta que los eventos “viajen” de ida y vuelta



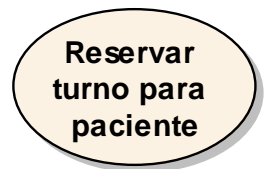
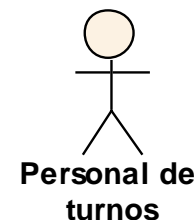
Se recomienda que cada paso en el flujo de eventos muestre la interacción entre el actor y el sistema de la siguiente forma:

- Qué hace el actor: <Actor> mensajes <Sistema>, y
- Qué hace el sistema en respuesta: <Sistema> mensajes <Actor>...

Notar que algunas interacciones son controladas por el sistema. Luego que el actor genera el requerimiento inicial que inicia el CU, el sistema controla la interacción: el sistema solicita información y el actor provee la información, y así siguiendo. En estos casos, una ida y vuelta dentro de un paso puede comenzar con acciones del sistema, y luego genera la respuesta al usuario

### Reserva de turno

1. El sistema muestra formulario de búsqueda de turnos.
2. Si el actor elige buscar por médico, se ejecuta el caso de uso Buscar persona de salud.
3. El actor completa formulario de búsqueda de turnos.
4. El sistema muestra turnos encontrados.
5. El actor encontró turno deseado.
6. El actor selecciona turno para el paciente.
7. El actor selecciona el motivo del turno
8. El sistema registra turno otorgado.
9. El sistema pregunta si desea imprimir documentos.
10. El actor desea imprimir documentos.
11. El sistema imprime comprobante de atención y recordatorio
12. Fin del flujo.



- Describir que sucede teniendo en cuenta los siguientes puntos:
  - Cuándo comienza
  - Qué condiciones deben satisfacerse
  - Qué acciones se realizarán
  - Definir el punto donde se reasume el flujo básico

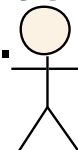
Para cada flujo alternativo específico, detallar la siguiente información:

- El punto de inicio en el flujo básico u otro subflujo donde el flujo alternativo es disparado (sugerencia: “En el paso  $n$  del flujo básico”)
- La condición de dispara su inicio (sugerencia: “si el sistema determina...”)
- Las acciones tomadas en el flujo alternativo
- Dónde el flujo básico u otro subflujo es reasumido después que el flujo alternativo finaliza. (sugerencia: “el caso de uso continúa en el paso  $n$  del flujo básico”). Si el CU finaliza en el flujo alternativo, colocar el texto “el caso de uso finaliza”.

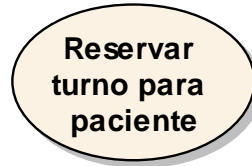
### Cancelación de turnos

Del paso 1, si el actor desea cancelar turnos del paciente se deriva en:

- 1.1. El actor selecciona la opción de ver los turnos del paciente seleccionado.
- 1.2. El sistema muestra los turnos del paciente
- 1.3. El actor selecciona el o los turnos que desea cancelar y le indica al sistema que proceda con la cancelación.
- 1.4. El sistema solicita que se ingrese el motivo de la cancelación del o los turnos (si se cancela más un turno a la vez, el motivo especificado será el mismo para todos).
- 1.5. El actor especifica el motivo de la cancelación del o los turnos (debe seleccionar el motivo de la lista de motivos de cancelación).
- 1.6. El sistema registra la cancelación del o los turnos.
- 1.7. El flujo vuelve al paso 1 del flujo básico.



Personal de  
turnos



Reservar  
turno para  
paciente

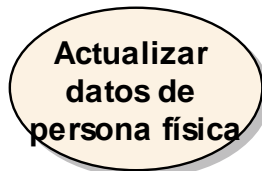
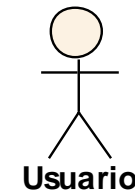
## ■ ■ Describir flujos alternativos generales

- Los flujos alternativos generales se especifican como flujos alternativos, excepto que no tienen un punto de comienzo definido ya que pueden comenzar en cualquier lugar. La siguiente información debe detallarse:
  - La condición que dispara su inicio (por ejemplo: “el usuario puede cancelar...”)
  - Las acciones llevadas a cabo en el flujo alternativo
  - Dónde el flujo básico u otro subflujo es reasumido después que el flujo alternativo finaliza. (sugerencia: “el caso de uso continúa en el paso *n* del flujo básico”). Si el CU finaliza en el flujo alternativo, colocar el texto “el caso de uso finaliza”.

- Cuando el flujo de eventos comience a resultar confuso debido a un comportamiento complejo, o donde un sólo flujo exceda una página impresa de longitud, los subflujos pueden ayudar a mejorar la claridad y a manejar la complejidad.
- Los subflujos se escriben moviendo un grupo lógico y auto-contenido de comportamiento detallado a un subflujo, y referenciando este comportamiento desde adentro del flujo de eventos.

### Actualizar instancia de persona

1. El sistema solicita ejecutar el CU Recuperar Persona Física
2. El sistema solicita al Sistema Administración Control de Acceso el indicador de calidad.
3. El sistema recupera el indicador de calidad y muestra al usuario la calidad propuesta con la cual se realizará la publicación del dato
- 4 . El usuario selecciona el nivel de calidad con el cual va a realizar la publicación. Si el usuario desea incorporar un nuevo documento para la persona, continua en el subflujo "Incorporar documento"
5. El sistema solicita al Sistema Nomencladores Comunes los tipos de documento y géneros.
6. El sistema muestra los tipos de documento y géneros
7. El usuario ingresa los datos actualizados de la persona
8. El sistema verifica que los datos mínimos requeridos hayan sido completados por el usuario
9. El sistema ejecuta validaciones sobre los datos a publicar con el objeto de detectar duplicaciones. Si las validaciones fueron correctas, continua con el subflujo ["Mecanismos de publicación"](#)
10. El sistema publica la actualización de Persona Física
11. El sistema retorna la instancia de publicación





## Incorporar documento

1. El sistema verifica el indicador de calidad con el cual la persona a actualizar fue incorporada en la base de Personas Físicas.

En caso de Incorporar un nuevo documento, si el tipo y número de documento se encuentra ingresado con un indicador de calidad menor o igual al actual vaya al paso 2

En caso de actualizar datos de la persona, si se encuentra ingresado con un indicador de calidad menor o igual al actual vaya al paso 2

2. Retorna al paso 10 del flujo básico

### Calidad inferior

Del paso 1 en el subflujo mecanismos de publicación, el indicador de calidad es menor, al que se encuentra ingresado, se deriva en:

1.1 El sistema informa al usuario que la actualización no puede ser publicada

1.2 Finaliza el CU

- Son condiciones que se deben cumplimentar para que el CU pueda comenzar
- No son eventos que puedan comenzar el CU
- Opcional: utilizar solo si el estado del sistema es importante para comenzar el CU
- Se pueden definir pre-condiciones a nivel de subflujo

Ejemplo:

Datos necesarios

Pre-condición

Para el correcto funcionamiento de este caso de uso deben estar cargados en el sistema las personas o pacientes y las agendas de los profesionales.

- Garantiza la veracidad cuando el CU finaliza
- Puede contener variantes
- Opcional: utilizar solo si el estado del sistema es importante para finalizar el CU
- Se pueden definir post-condiciones a nivel de subflujo

Observación: si algo falla en el CU, se puede utilizar la post-condición para cubrirse diciendo “la acción es completada, o, sí algo ha fallado, la acción no es ejecutada”

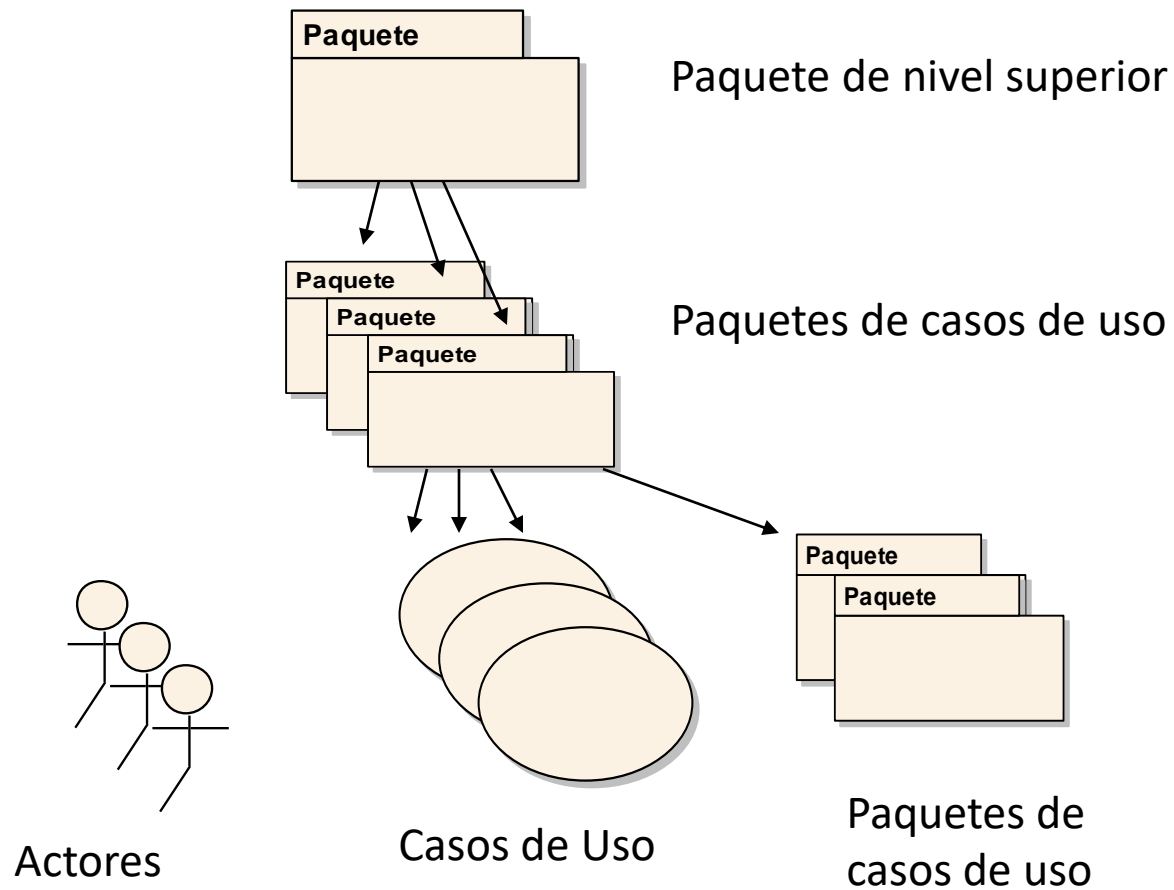
Turnos otorgados  
Post-condición

Luego de la ejecución de este caso de uso queda registrado un nuevo turno y/o cancelado un turno existente para el paciente seleccionado.

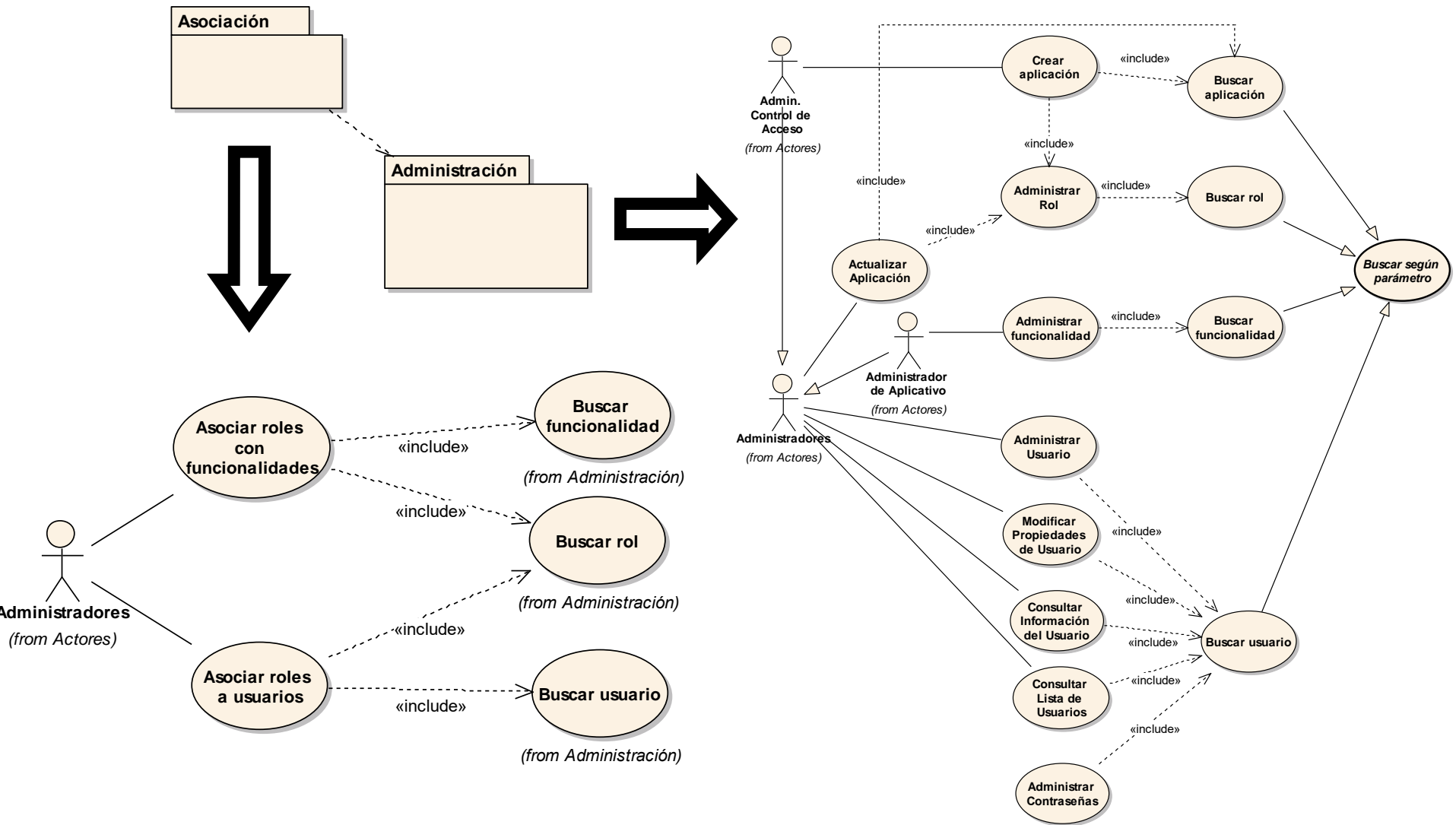
- Síntomas
  - CU muy pequeños
  - Muchos CU
  - CU sin resultados de valor
  - Nombres que incluyen operaciones de bajo nivel
    - “Operación”+”Objeto”
    - “Función”+”Datos”
    - Ejemplo: “Insertar tarjeta”
  - Dificulta el entendimiento del modelo completo
- Acciones correctivas
  - Buscar un mayor contexto
    - ¿Por qué está insertando tarjeta?
  - Colocarse en el rol del usuario
    - ¿Qué quiere lograr el usuario?
    - ¿El objetivo de quién, satisface este CU?
  - ¿Qué valor agrega este CU?
    - ¿Cuál es la historia detrás de este CU?

- Un paquete es un mecanismo de agrupamiento general utilizado en UML
  - Si el modelo es muy grande para visualizarlo en una simple unidad puede dividirlo en paquetes.
  - Un paquete en un modelo de CU puede contener CU, actores, relaciones y otros paquetes contenidos

# Visualización de paquetes en el modelo de casos de uso



# Ejemplo de paquetes





- Las RN enuncian características propias del dominio que deben ser consideradas a la hora de plantear usos del sistema. Surgen como resultado de las actividades de relevamiento: no se infieren, se capturan y si no hay información suficiente para enunciarlas completas debe hacerse la consulta a un referente del negocio.
- Las RN aparecen fuertemente vinculadas a los casos de uso y a los objetos del dominio de la aplicación indicando restricciones a sus valores, relaciones y comportamiento

## ■ ■ Tipos de Reglas del Negocio

- **Reglas de estímulo-respuesta**, son las que indican cuando y en qué condiciones debe ser realizada una transacción en el dominio.

*Ej: Cuando un cliente solicita un servicio debe verificarse que sus pagos se encuentren al día antes de permitirle el llenado del formulario correspondiente.*

- **Reglas de restricción de los objetos (también aplicable a rangos de valores)**, definen condiciones y políticas que afectan a las entidades del dominio y que no pueden ser violadas.

*Ej: Cada socio tiene derecho a N prestaciones anuales gratuitas.*

## ■ ■ Tipos de Reglas del Negocio (continuación)

- **Reglas de restricción de la aplicación**, definen condiciones y políticas que restringen los servicios y que deben ser consideradas parte de la lógica de la aplicación.

*Ej: Los criterios de búsqueda de material son: por autor o por tema y son mutuamente excluyentes.*

- **Reglas de inferencia**, indican relaciones asociativas y transitivas entre los objetos.

*Ej: Si un cliente no hizo uso de los servicios contratados durante el período de vigencia anterior, entonces se lo beneficiará con un 10% de descuento durante el siguiente período.*

- **Reglas de cálculo**, delinean un algoritmo o ecuación.

*Ej: punto de reposición = Stock\_Actual +  
Indice\_de\_Demanda/Indice\_Estacional*

- **Reglas de composición de los objetos**, indican qué elementos intervienen en la definición y descripción de otros más complejas.

*Ej: la solicitud de vacante incluye tales y tales  
datos.*

## ■ ■ ¿Por qué cambian los requerimientos?

- Porque hemos fallado en consultar a la persona correcta, con las preguntas correctas y en el tiempo correcto
- Porque el problema que está siendo resuelto ha cambiado
- Porque los usuarios han modificado sus intenciones o percepciones
- Porque el ambiente externo ha cambiado
- Porque hemos fallado al crear un proceso que ayude a manejar los cambios
- Porque nuestro entendimiento del problema ha mejorado

# ■ ■ Anatomía de un proyecto



How the customer explained it



How the Project Leader understood it



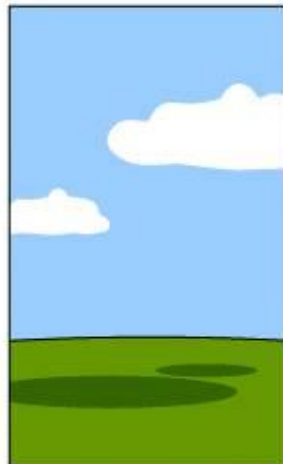
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



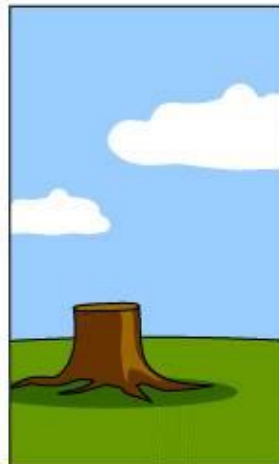
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

## ■ Métricas que nos ayudan a manejar los requerimientos

- ¿Cuántos CU tenemos?
- ¿Qué características críticas aún no han sido aprobadas?
- ¿Cuál es el costo estimado de los cambios propuestos?
- ¿Cuántos cambios se produjeron desde la última revisión?
  - ¿Quién ha autorizado los cambios?
  - ¿Cuál es el impacto de los cambios en la planificación del testing?
  - ¿Hay menos cambios esta semana que la anterior?

Ayuda a asegurar que estamos construyendo el sistema correcto.

Tracear un requerimiento a requerimientos relacionados nos ayuda a asegurar que si un requerimiento ha cambiado, los requerimientos relacionados también deberán reflejar los cambios

Una de las claves para una efectiva administración de cambios en los requerimientos es la trazabilidad de los requerimientos a otros elementos del proyecto



Necesidades del negocio

**manejan**

necesidades del cliente

**las cuales manejan**

necesidades del usuario

**los cuales demandan**

características del producto

**que manejan**

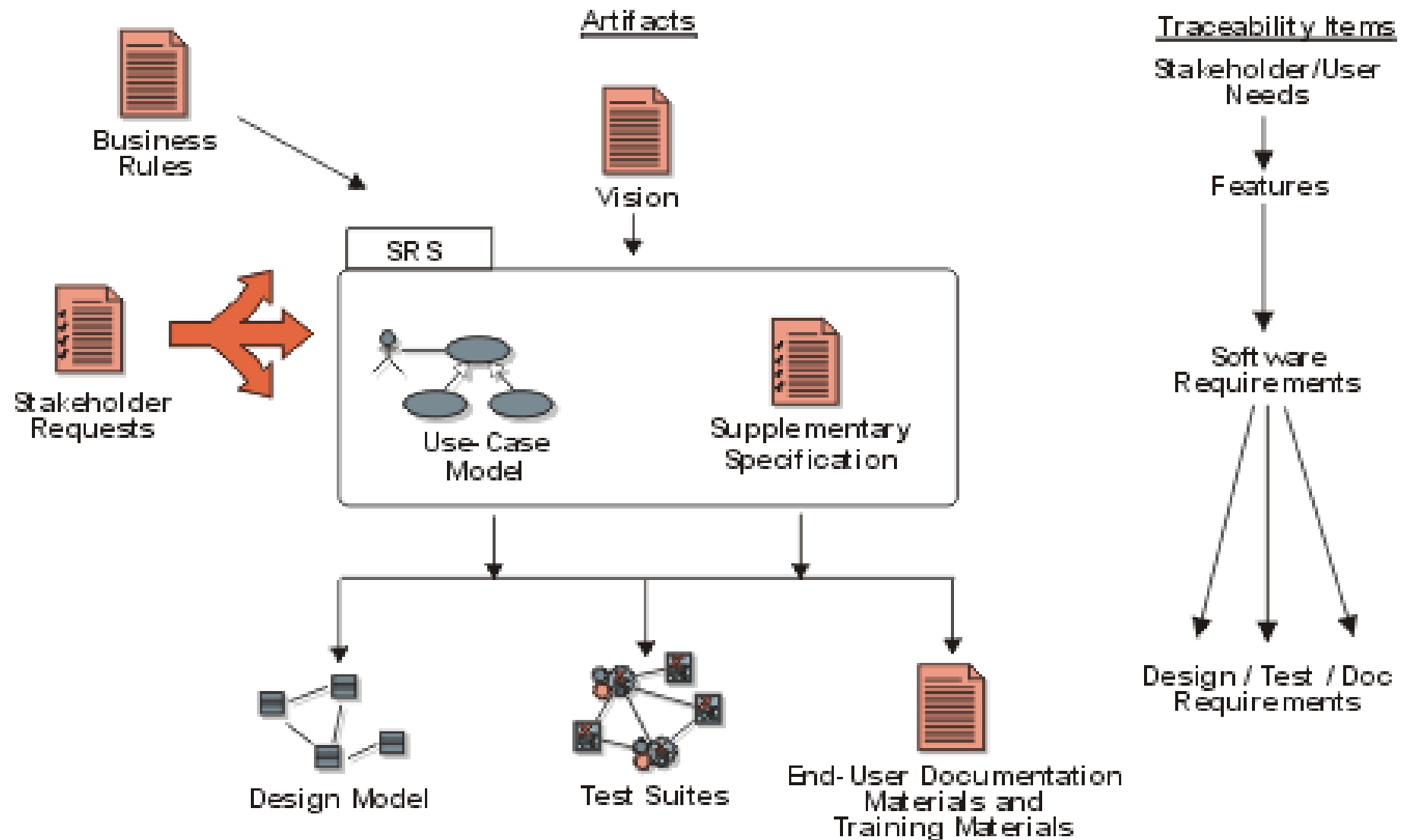
requerimientos de software

**que los desarrolladores**

Implementan, testean y documentan

- Asegurar calidad
  - Verificar que la implementación cumplimenta todos los requerimientos
  - Verificar que la aplicación hace solo lo que fue entendido
- Analizar el impacto de los cambios
  - Detectar requerimientos relacionados
  - Inspeccionar requerimientos relacionados

# ■ ■ Determinar la estrategia de trazabilidad de requerimientos



- Linkear una característica a un CU. Permite verificar que cada CU es derivado de una característica que se encuentra directamente relacionada a una necesidad del usuario
- Una característica puede ser linkeada a un subflujo de un CU (tal como un flujo alternativo)
- Una característica puede linkearse a requerimientos no funcionales que apliquen a todo el sistema o a un CU en particular

Asegurarse que cada requerimiento puede ser verificado y que cada requerimiento fue considerado

- ¿Qué es estructurar?
  - Separar partes de los CU para realizar nuevos CU
- ¿Por qué estructurar el modelo de CU?
  - Para simplificar el modelo de CU original
    - Facilitar el entendimiento
    - Facilitar el mantenimiento
  - Rehusar el comportamiento
    - Compartir entre varios CU

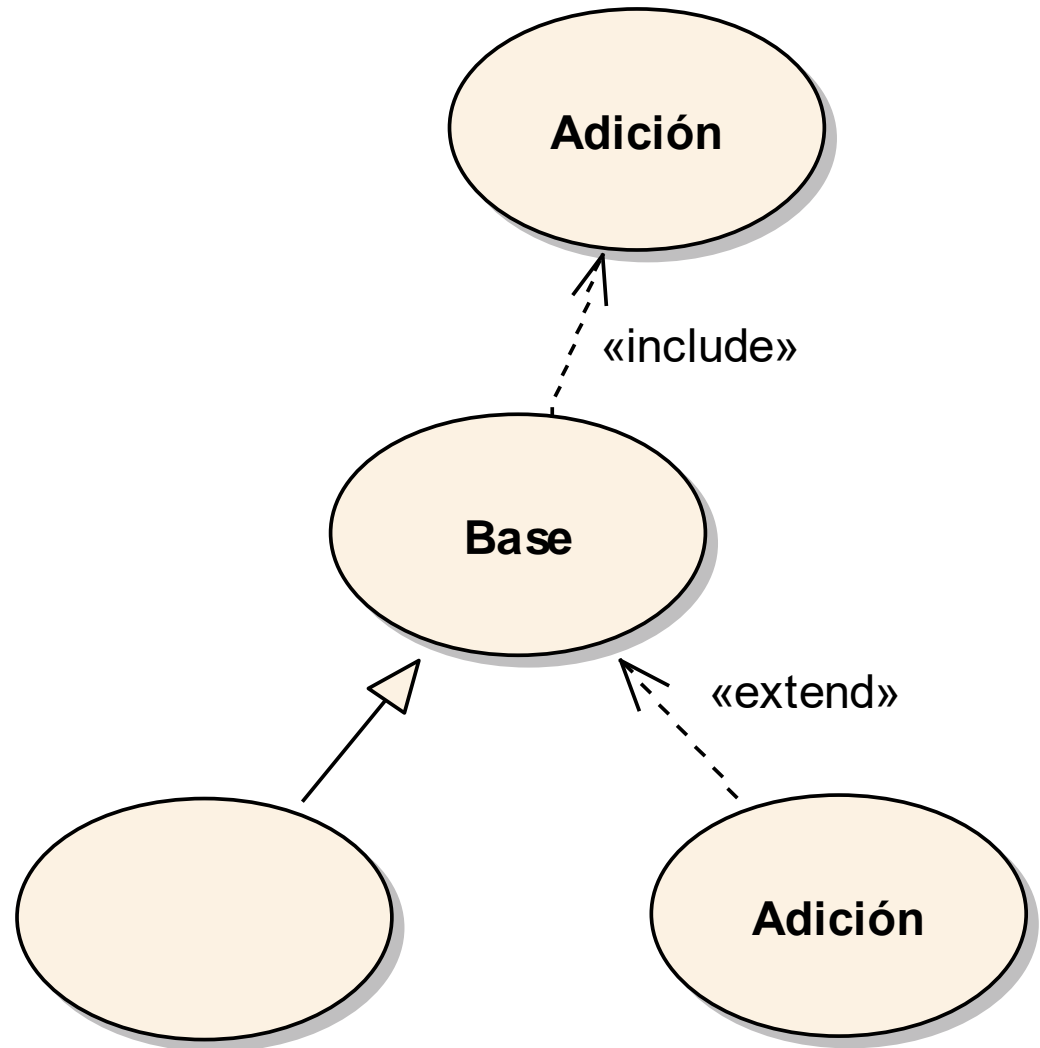
Es importante no comenzar a estructurar el modelo de CU hasta que el usuario haya acordado y entendido el conjunto de requerimientos

Advertencia: no comience esta actividad demasiado pronto ya que puede confundir a su cliente (o a Ud. mismo)

- El propósito de estructurar el modelo de CU es extraer comportamiento de un CU que puede ser representado en un CU separado.
- Ejemplos de comportamiento que pueden ser descomponerse :
  - Comportamiento común;
  - Comportamiento opcional;
  - Comportamiento excepcional y
  - Comportamiento que será desarrollado en iteraciones posteriores

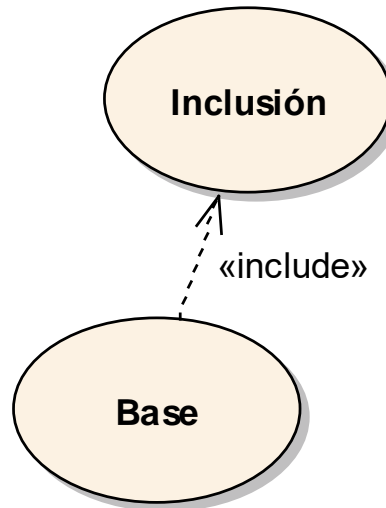
El CU que representa la modificación es llamado el CU adicionado. El CU original que es modificado es llamado el CU base

- Inclusión
- Extensión
- Generalización



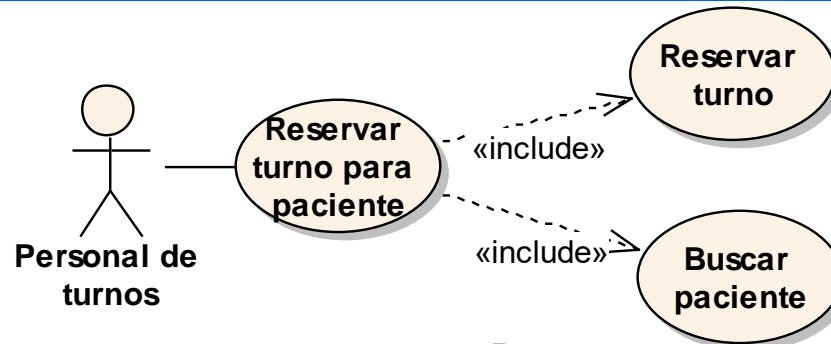
## ¿Qué es un caso de uso de Inclusión?

- Una relación entre un CU base a un CU de inclusión
- El comportamiento definido en el CU de inclusión está **explícitamente insertado** en el CU base





## ■ ■ Ejemplo de Inclusión



### Reservar turnos para paciente

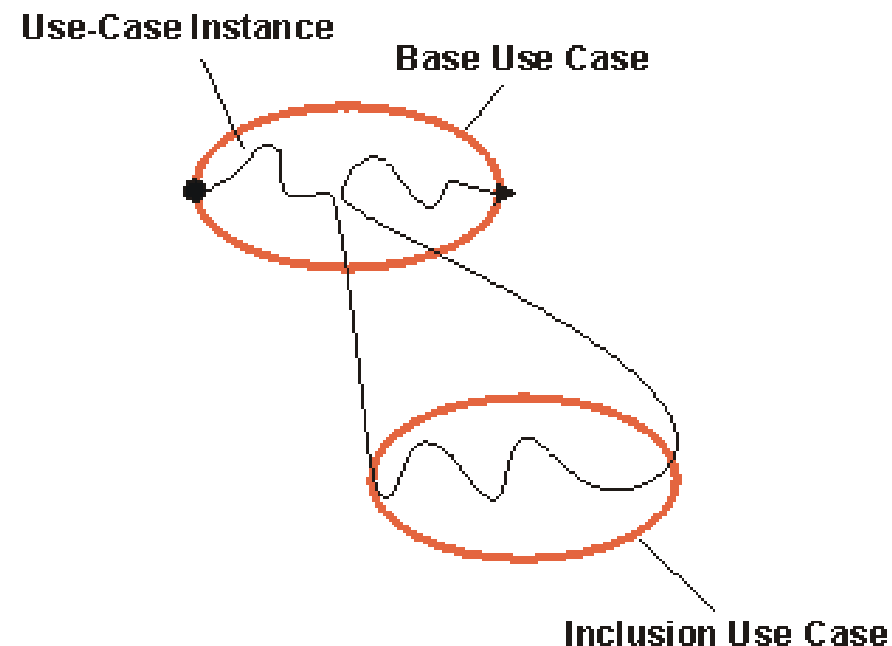
1. Se ejecuta el caso de uso buscar paciente.
2. El actor selecciona un paciente.
3. Se ejecuta el caso de uso reservar turno.
4. Fin del flujo.

### Reserva de turnos

1. El sistema muestra formulario de búsqueda de turnos.
2. Si el actor elige buscar por médico, se ejecuta el caso de uso Buscar persona de salud.
3. El actor completa formulario de búsqueda de turnos.
4. El sistema muestra turnos encontrados.
5. El actor encontró turno deseado.
6. El actor selecciona turno para el paciente. (continua)

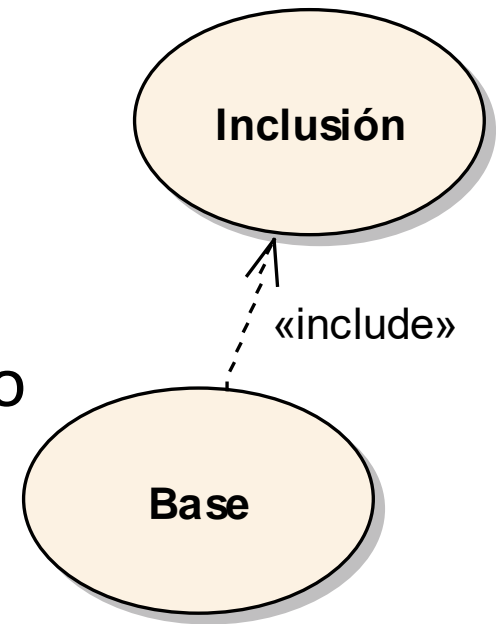
## ■ ■ Ejecutar la Inclusión

- El comportamiento de la inclusión es insertada en alguna ubicación en el CU base.
- La relación de inclusión no es condicional.
- Si una instancia de un CU contiene la inclusión en el CU base, el CU de inclusión es ejecutado.
- Si una instancia de un CU nunca alcanza el paso, en el CU base, donde la relación de inclusión es definida, el CU de inclusión nunca es ejecutado
- El CU base puede depender del resultado de la ejecución del CU de inclusión, pero ni el CU base ni el de inclusión pueden acceder cada uno de ellos a atributos del otro



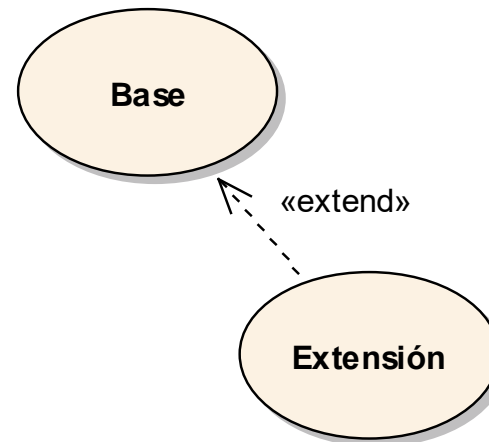
## ■ ■ ¿Para qué usar una relación de inclusión?

- Separar comportamiento común a dos o más CU
  - Evitar describir el mismo comportamiento múltiples veces
  - Asegurar que el comportamiento común permanece consistente
- Separar y encapsular comportamiento desde un CU base
  - Simplificar flujos de evento complejos
  - Separar comportamiento que no es parte del propósito primario



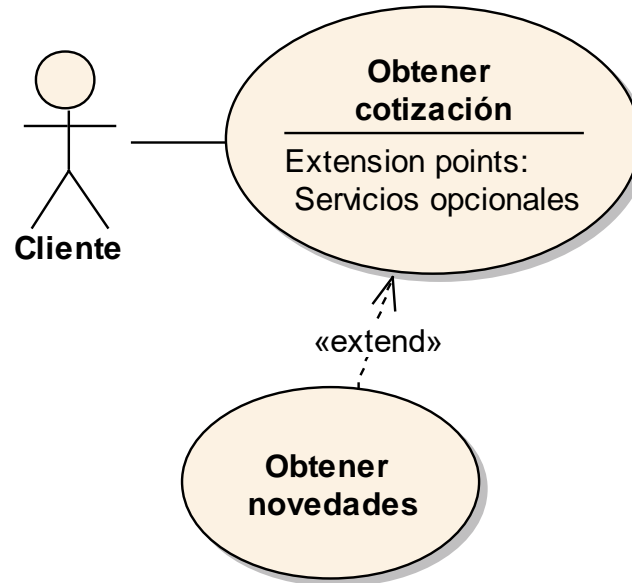
## ■ ■ ¿Qué es un caso de uso de extensión?

- Asociación desde un CU de extensión a un CU base
  - El comportamiento del CU de extensión es insertado en el CU base
  - Se inserta si la condición de extensión es verdadera
  - La inserción en el CU base es denominada punto de extensión



El CU de extensión puede ser abstracto, pero no tiene que serlo necesariamente

## ■ ■ Ejemplo de relación de extensión



El CU Obtener cotización ha sido extendido con otro servicio, tal como Obtener novedades. Esta extensión puede ser haber sido un flujo alternativo en el CU base.

Factorizando este comportamiento en un CU nos permite rehusar el comportamiento y enfatizar que son comportamientos separados y opcionales

## ■ ■ Ejemplo de relación de extensión (continuación)

### Obtener Cotización

1. Ejecutar el CU Identificar cliente para verificar la identidad del mismo
2. El sistema muestra las opciones y el cliente selecciona “obtener cotización”
3. El cliente obtiene la cotización
4. El cliente obtiene otras cotizaciones
5. ....

Punto de extensión:

Servicios opcionales: ocurre después del paso 3 en el flujo básico

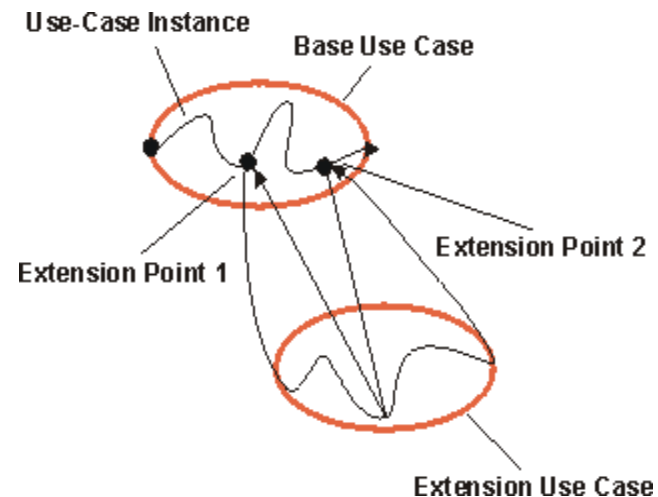
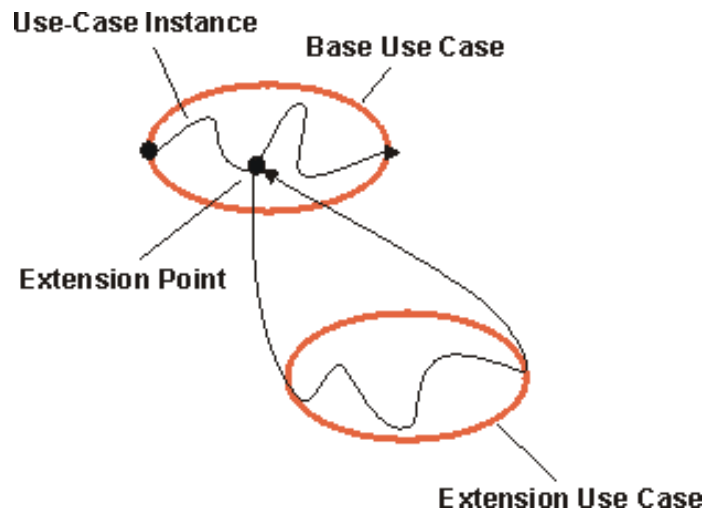
### Obtener novedades

Este CU extiende el CU Obtener Cotización, en el punto de extensión “Servicios opcionales”

- Si el cliente selecciona “Obtener novedades”, el sistema consulta al cliente el período de tiempo y el número de ítems
- El cliente ingresa el período de tiempo y el número de ítems. El sistema envía la solicitud al Sistema de Novedades, recibe la respuesta y muestra las novedades al cliente.
- Continúa el CU Obtener Cotización

## ■ ■ Ejecutar la extensión

- Cuando una instancia del CU base alcanza una ubicación en la cual el CU base tiene un punto de extensión definido, la condición en la correspondiente relación de extensión es evaluada.
  - Si la condición es verdadera o ausente, la instancia del CU continúa en el CU extendido.
  - Si la condición de la relación de extensión es falsa, la extensión no es ejecutada



## ■ ■ ¿Para qué utilizar una relación de extensión?

- Separar comportamiento opcional o excepcional
  - Ejecutado solo bajo ciertas condiciones
  - Separar flujos para simplificar el CU base
  - Ejemplo: accionar una alarma
- Agregar comportamiento extendido
  - Comportamiento desarrollado separadamente, posiblemente en versiones posteriores
  - Ejemplo: CU avisar por e-mail



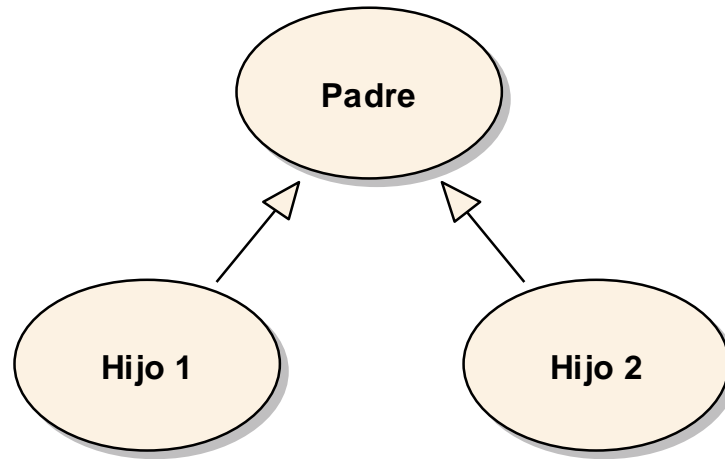
## ■ ■ ¿Para qué utilizar una relación de extensión? (continuación)

- El CU base no controla la condición para la ejecución de la extensión. Las condiciones son escritas dentro de la extensión
  - El CU de extensión puede acceder y modificar propiedades del CU base.
  - El CU base no puede ver las extensiones y no puede acceder a sus propiedades
  - El CU base es implícitamente modificado por la extensión pero no tiene visibilidad de la especificación de la extensión
  - El CU base debe ser completo en sí mismo. Sin embargo el CU no es independiente de las extensiones, porque este no puede ser ejecutado sin la posibilidad de seguir las extensiones

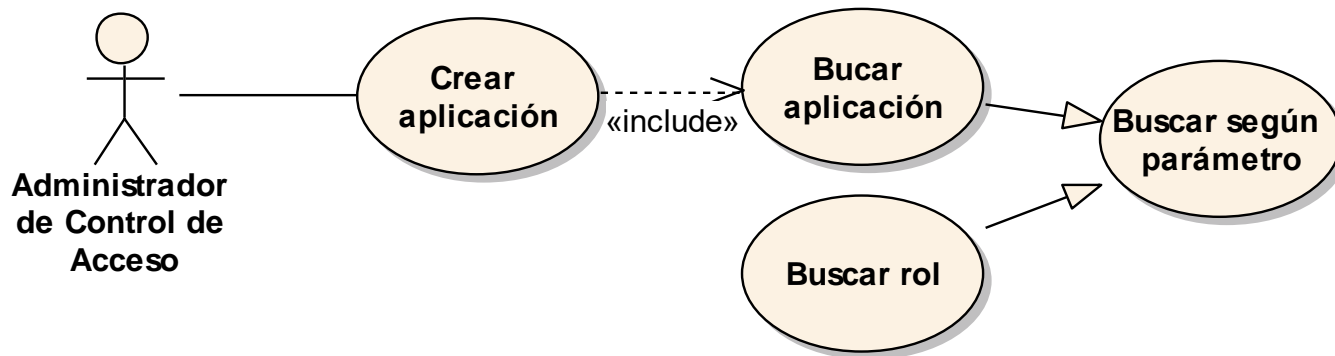
- El CU base no necesita ser modificado con el conocimiento de la existencia del CU que extiende a este.
- La llamada es colocada en un espacio especial para extensiones.
- Los puntos de extensión identificados, es un buen indicio para planificar la dinámica de los requerimientos
- El CU base no necesita conocer que está siendo extendido

## ■ ■ ¿Qué es un caso de uso de generalización?

- Una relación de un CU hijo a un padre
  - Describe un comportamiento compartido en el padre
  - Describe un comportamiento especializado en el hijo
  - Comparten un objetivo común



## ■ ■ Ejemplo de caso de uso de extensión



### Buscar según criterio de búsqueda

1. El usuario ingresa el criterio de búsqueda
2. El sistema realiza la búsqueda acorde a los criterios ingresados
3. El sistema muestra los datos coincidentes con el criterio ingresado.

### Buscar aplicativo

Se redefine el paso 1 del flujo básico del caso de uso abstracto buscar según parámetro de la siguiente forma:

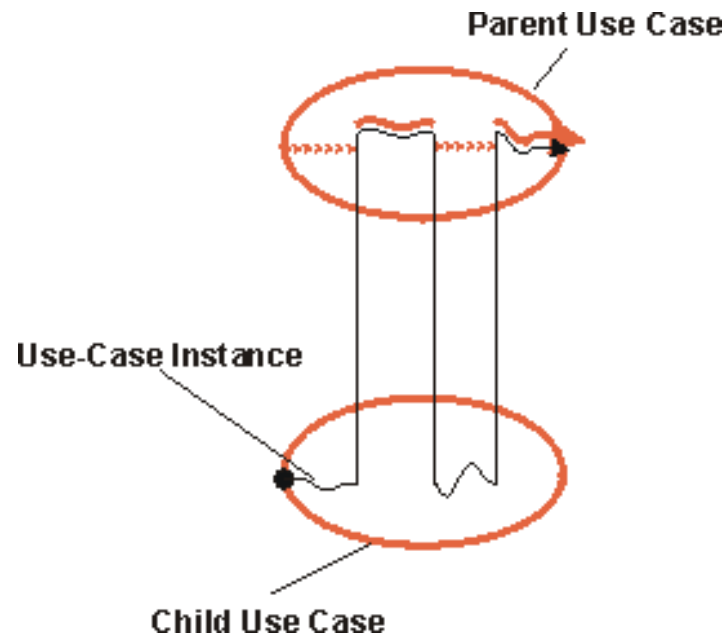
- El usuario selecciona buscar aplicativo
- El sistema identifica el criterio de búsqueda de aplicativos en el sistema nomencladores comunes de BU para dar la orden de buscar

Se redefine el paso 3 del flujo básico del caso de uso abstracto buscar según parámetro de la siguiente forma:

3. El sistema muestra el nombre y la descripción de los aplicativos recuperados

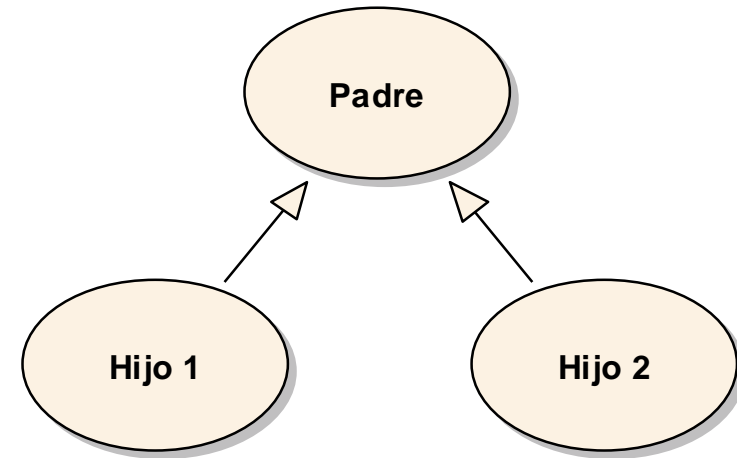
## ■ ■ Ejecutar un caso de uso de generalización

- Una instancia de un CU ejecutando un caso de uso hijo sigue el flujo de eventos descritos por el caso de uso padre, insertando comportamiento adicional y modificando comportamiento como es definido en el flujo de eventos del CU hijo



## ■ ■ ¿Para qué utilizar casos de uso de generalización?

- Mostrar comportamiento común, estructura y propósito en dos o más CU
  - Muestra que los hijos son parte de una familia de CU
  - Evitar describir el mismo comportamiento múltiples veces
  - Asegura que el comportamiento común permanezca consistente
  - Permite el rechazo de requerimientos

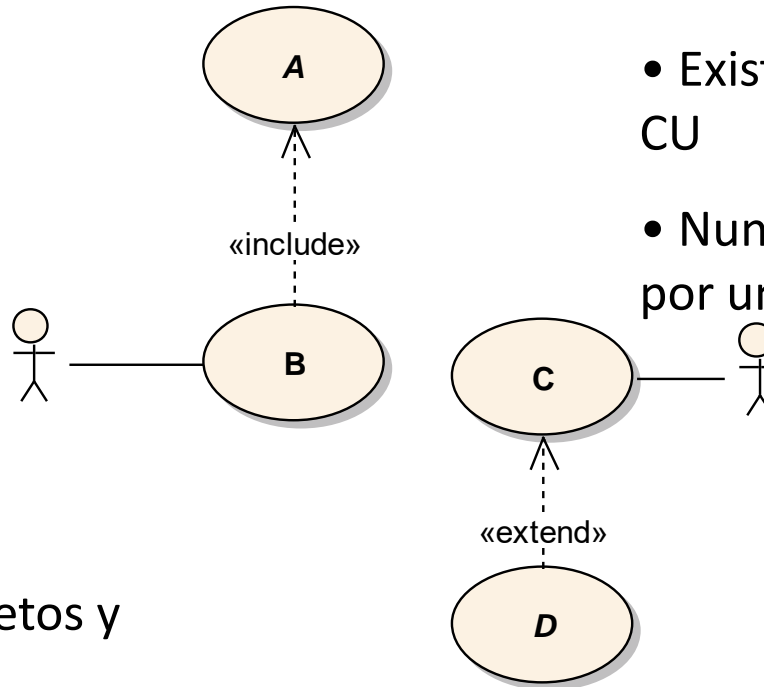


Un CU padre puede ser especializado en uno o más CU que representan formas más específicas de un padre. Ni el padre, ni los hijos son necesariamente abstractos, aunque el padre en muchos casos es abstracto

- Cuando extraemos comportamiento en un CU incluido o extendido, estos nuevos CU, nunca serán ejecutados solos, ellos existen como parte de otro CU. Estos CU frecuentemente son llamados abstractos.
- Tener en cuenta que:
  - Que un CU concreto puede incluir otro CU concreto
  - Que un CU concreto puede ser extendido por otro CU concreto

## ■ ■ Ejemplo caso de uso abstracto

Un CU es o concreto o abstracto



### CU concretos (B y C):

- Tienen que ser completos y significativos
- Pueden ser instanciados por un actor

### CU abstractos (A y D):

- No tienen que ser completos
- Existen solo para otros CU
- Nunca son instanciados por un actor