

Análisis de Sistemas CIF 5555 2022-1







Resultados de aprendizaje

- El alumno debería:
 - Aplicar el proceso de Ingeniería de Requisitos para la elicitación de las necesidades de los clientes y usuarios



Lenguaje de modelado unificado

INTRODUCCIÓN A UML



Qué es un modelo

- Abstracciones del sistema (o del dominio) utilizadas para especificar su estructura y comportamiento
- Un modelo captura aspectos importantes de la cosa a modelar desde un cierto punto de vista y simplifica u omite el resto:
- ejemplo planta de una casa
- Un modelo se expresa con un formalismo que lo hace fácil de usar y comprender



Propiedades de un modelo

El modelo proporciona una representación del sistema que se reduce al mínimo para minimizar la complejidad de la gestión.

Un sistema se describe no mediante una vista única, sino mediante un conjunto de vistas que, en su conjunto, nos brindan una vista completa del modelo.



Más sobre los modelos

- Contiene conocimientos sobre el "qué" y el "cómo" de un sistema.
 - Herramienta de comunicación y discusión
 - Documentación
- Fundamental para un proceso de diseño colaborativo, en el corazón del desarrollo de software moderno
- ¿Cómo se modela un sistema?
- ¿Cómo se representa un modelo?



Cómo usar un modelo

- Un proyecto se puede realizar como
 - Bosquejo (sketch)
- el modelo no está completo, pueden hacer visibles soluciones alternativas
 - Proyecto detallado (blueprint)
- contienen suficientes detalles para permitir a los desarrolladores crear un sistema listo para ejecutar sin tomar decisiones de diseño
- Los modelos como borrador o blueprint se pueden usar para ingeniería hacia adelante o hacia atrás
 - Ejecutable (UML como lenguaje de programación)
- tan completo y preciso que el código se puede generar automáticamente a partir del modelo





Cómo describir el modelo

- Necesidad de un lenguaje común
- Semi-formal

- Que se puede utilizar para describir diferentes aspectos del proyecto:
 - Descripción estructural vs descripción de un comportamiento
 - Borrador del proyecto vs proyecto final
 - Descripción de dominio vs descripción de la estructura del código



UML

Unified Modeling Language: lenguaje de modelado unificado

- Familia de notaciones gráficas basadas en un único meta-modelo
- Se utiliza para respaldar la descripción y el diseño de sistemas de software, en particular aplicaciones sw OO, pero no solo....
- Los diagramas son generalmente fáciles de entender por los usuarios con un mínimo de conocimientos en el campo.
- UP (Proceso Unificado) describe actividades y "recomienda" el uso de UML. Viven bien en simbiosis pero uno no implica al otro



Breve historia

- In 1994: muchos lenguajes y métodos de modelización oo. Sin embargo, hubo algunas primicias en la clase -Booch y Rumbaugh para los lenguajes y Jacobson para las metodologías
- 1994: primer intento de unificación – el lenguaje Fusion
- 1994: Booch y Rumbaugh se unen en la Rational Corporation para crear lo que llaman UML

- 1996: El Object Management Group (OMG) lanza una primera propuesta de estandarización de UML
- 1997: OMG aprueba el estándar UML 1.0
- 2000: UML 1..4
- 2006: UML 2.0
- 2006 ... Arquitectura dirigida por modelos (MDA)
- Actualmente UML 2.5



Generalidades

- Unificación a nivel de lenguaje, no de método
- Modelado de cada etapa del proceso de desarrollo
- Aplicable a múltiples tipos de proyectos y dominios
- Independiente
 - del lenguaje de desarrollo
 - del modelo de ciclo de vida



Objetivos

- Visualización
 - Comunicación y comprensión
- Especificación y documentación
 - Descripción del sistema en todos sus niveles
- Realización
 - Soporte para la automatización de codificación



sobre UML

- El principio básico de UML es que un sistema de software puede verse como un conjunto de objetos colaborativos
- Se consideran dos aspectos fundamentales del sistema
 - Estructura estática: objetos necesarios y relaciones entre ellos.
 - Comportamiento dinámico: cómo los objetos trabajan juntos para lograr el objetivo.

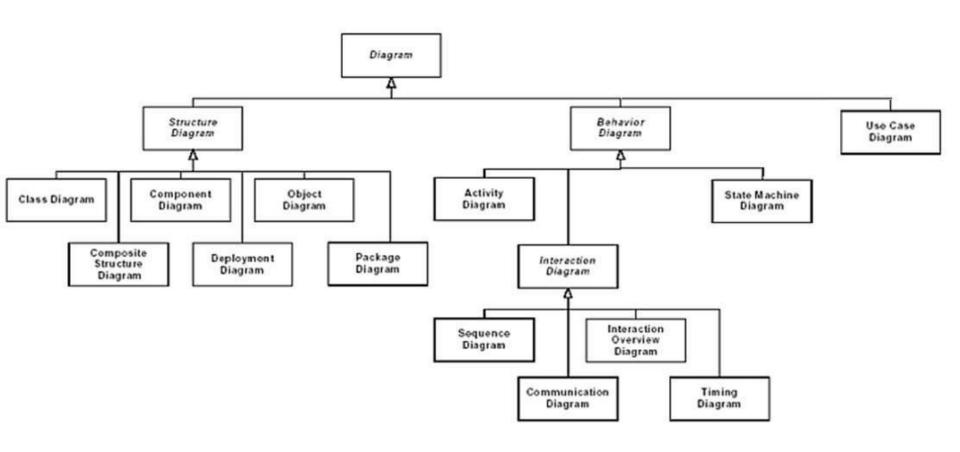


Diagramas en UML

- En UML, un modelo se representa gráficamente como un diagrama.
- Un diagrama nos da una vista de la parte de la realidad descrita por un modelo.
- Modelamos, mediante diagramas adecuados, las funcionalidades, procesos y arquitecturas software de un proyecto.
- UML ofrece una multiplicidad de diagramas: multiplicidad de vistas para cada modelo (por ejemplo, dominio y estructura del código) que es de interés para describir una aplicación.
- Un conjunto de diagramas describe un modelo.



Diagramas UML: 14 diferentes





Tipos de modelos

Modelo estático

- Los conceptos del dominio
 - Modelados por entidades y relaciones
- Implementación del sistema
 - clases de construcción, componentes....
- Ausencia de aspectos dependientes del tiempo

Modelo Dinámico

 Modela el comportamiento de las entidades descritas en el modelo estático



Diagramas que veremos en el curso

- ... de clases
- ... de objetos
- ... de componentes
- ... de estructura compuesta
- ... de máquina de estado
- de casos de uso
- ... de actividad
- ... de secuencia
- ... de despliegue
- ... de paquetes

- Representación gráfica
 - de un conjunto de elementos del modelo
 - de acuerdo con una cierta vista
- Grafo
 - Vértices = elementos del modelo
 - Arcos = relaciones entre elementos



algunos conceptos de UML

- Modelo: abstracción de (parte de) un sistema
 - EJEMPLO: Auto
 - la maqueta
 - el diseño del auto
 - la fórmula para calcular la distancia de frenado
- Modelo estático:
 - describe los elementos del sistema y sus relaciones

- Modelo dinámico:
 - describe el comportamiento del sistema en el tiempo
- Diseño
 - Conjunto de modelos: las diferentes dimensiones del sistema
- Vista: descripción de un aspecto de un modelo



Elementos básicos de UML

- Diagramas
 - de clases, de casos de uso, de interacción, ...
- Entidad
 - clases, interfaces, componentes, casos de uso, ...
- Relaciones
 - asociaciones, generalizaciones, dependencias, ...



Clasificadores versus instancias

- Un clasificador modela un concepto que describe instancias.
 - Ej. Una clase modela objetos
- Los clasificadores en UML son
 - Clases
 - Actores
 - Casos de uso
 - Componentes
 - Nodos (ej., hw) ...
- Algunos diagramas pueden estar a nivel de clasificador o a nivel de instancia



CASOS DE USO



Requisitos

- ¿Qué se supone que "haga" el (futuro) sistema de software?
- Modelado de requisitos: consiste en recopilar y modelar requisitos
 - Modelo estático:
 - diagrama de casos de uso
 - Modelo dinámico:
 - narrativas asociadas con casos de uso



Casos de uso

- El diagrama de casos de uso describe los requisitos funcionales del sistema
- Este diagrama captura la funcionalidad que debe ofrecer un sistema, visto desde el exterior: las tareas que un usuario puede realizar con la ayuda del sistema
- Un actor es una entidad externa al sistema, que interactúa directamente con él en un rol determinado
 - usuario
 - otro sistema
- Un caso de uso es
 - una función o servicio ofrecido por el sistema a uno o más actores
 - formalmente: una tarea que un actor puede realizar con la ayuda del sistema
 - expresado como un conjunto de ESCENARIOS
- Un escenario es
 - Una secuencia de interacciones (intercambios de mensajes) entre el sistema y los actores





Modelado de los casos de uso

El modelado implica los pasos siguientes :

- Identificar un límite potencial del sistema
- Identificar los actores
- Identificar los casos de uso
- Identificar las relaciones entre actores y casos de uso
- Especificar el caso de uso
 - con una descripción textual (narrativa)



Modelado de los casos de uso

El diagrama consta de cuatro entidades:

- actores: los roles asumidos por las personas y las cosas (máquinas) que usan el sistema
- casos de uso: qué pueden hacer los actores con el sistema
- relaciones: Relaciones significativas entre actores y casos de uso
- límite del sistema: un rectángulo dibujado alrededor de los casos de uso para indicar el límite del sistema que se está

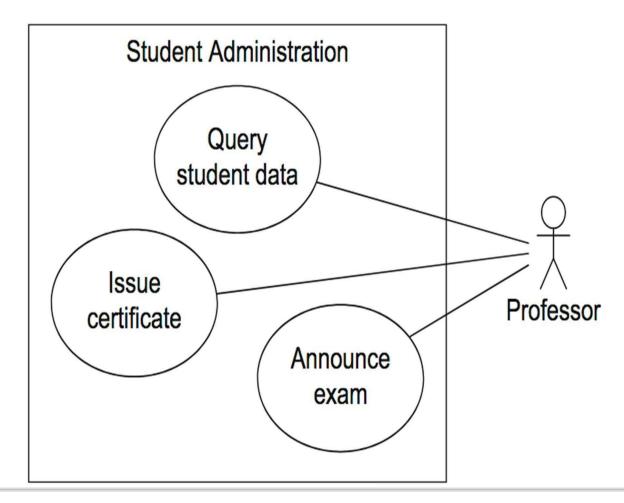


Modelado de casos de uso

- Modelo de caso de uso = una vista del sistema
 - enfatiza el comportamiento tal como aparece ante los actores externos
 - divide la funcionalidad del sistema en interacciones ("casos de uso")
 - casos de uso son significativos para los usuarios o sistemas externos ("actores")

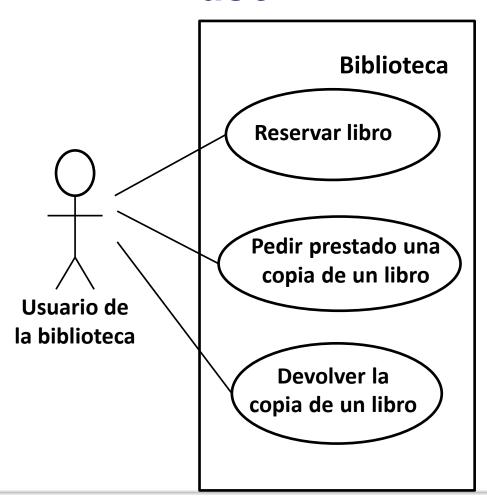


Primer ejemplo de diagrama de casos de uso





Otro ejemplo de diagrama de casos de uso

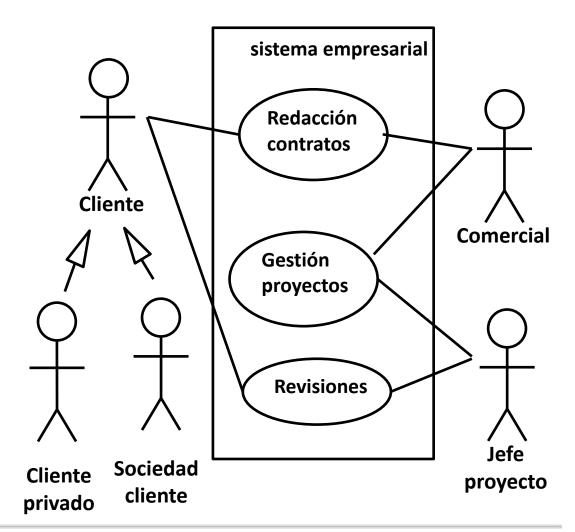




Tercer ejemplo de diagrama casos de uso

Ejemplo

- Los clientes pueden ser particulares o empresas
- Clientes y vendedores participan en la redacción de contratos
- Los clientes y los jefes de proyecto participan en las revisiones
- Comercial y los jefes de proyecto gestionan

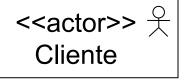




Actores

- Definición
 - Actor es una entidad externa que está involucrada en la interacción con el sistema descrito en un caso de uso.
- Descripción
 - actores = roles
 - actores pueden ser también diálogos, y sistemas externos
- Notación



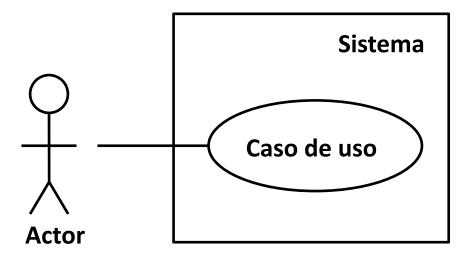




 Casos de uso sin actores solo si: incluidos (include), extensiones o sub-casos



Semántica del caso de uso



- Un actor es un usuario u otro sistema, en un rol particular
- Un caso de uso es una tarea que los actores realizan con la ayuda del sistema.
- La asociación actor-caso representa una interacción (secuencia de mensajes).
- Un caso de uso SOLO lo inicia un actor (principal: la narrativa define quién es el actor principal)
 - Posiblemente tiempo



Casos de uso: descripción narrativa

- Documento que describe el modelo dinámico
 - los escenarios relevantes de un caso de uso
 - desde el punto de vista de los actores
- Características
 - Inicio, fin, pasos intermedios
 - Condiciones excepcionales
 - Precondiciones, Postcondiciones



Casos de uso: descripción narrativa

- Cada caso de uso
 - Tiene una secuencia de transiciones normales
 - Puede tener varias secuencias alternativas
 - Tiene varias secuencias de transiciones excepcionales para la gestión de situaciones erróneas
- Un escenario es una instancia de un caso de uso, una historia, una secuencia de acciones que produce un resultado observable para uno o más actores.



Descripción narrativa

Nombre: Nombre de caso de uso

ID identificador

Breve descripción: Dos líneas de resumen

Actores primarios: Actores que inician el caso de uso

Actores secundarios: Otros actores que interactúan con el caso de uso

Precondiciones : Deben postularse antes de la ejecución del caso

de uso

Secuencia principal de Secuencia de pasos

eventos:

Postcondiciones De la secuencia principal

Secuencia alternativa de eventos:

Errores, ramificaciones e interrupciones en la secuencia principal



Ejemplo de descripción narrativa

Nombre:

Prorrogar el préstamo

Breve descripción:

Un usuario desea extender el período de préstamo.

Actores primarios:

Usuario de biblioteca

Actores secundarios:

Ninguno

Precondiciones:

El libro ya está asignado al usuario.

Secuencia principal de

Asegúrese de que nadie lo haya reservado

eventos:

Asegúrese de que el usuario sea elegible para el préstamo.

Postcondiciones

Actualizar el estado del usuario y del libro Préstamo extendido, estado urgente actualizado

Secuencia alternativa de eventos:

El libro ya está reservado, usuario no admitido al préstamo





Use Case 1 Buy Stocks over the Web 👭

Primary Actor: Purchaser

Scope: Personal Advisors / Finance package (PAF)

Level: User goal

Stakeholders and Interests:

Purchaser—wants to buy stocks and get them added to the PAF portfolio

automatically.

Stock agency-wants full purchase information.

Precondition: User already has PAF open.

Minimal Guarantee: Sufficient logging information will exist so that PAF can detect that something went wrong and ask the user to provide details.

Success Guarantee: Remote web site has acknowledged the purchase; the logs and the user's portfolio are updated.

Main Success Scenario:

- 1. Purchaser selects to buy stocks over the web.
- 2. PAF gets name of web site to use (E*Trade, Schwab, etc.) from user.
- 3. PAF opens web connection to the site, retaining control.
- 4. Purchaser browses and buys stock from the web site.
- 5. PAF intercepts responses from the web site and updates the purchaser's portfolio.
- PAF shows the user the new portfolio standing.

Extensions:

- 2a. Purchaser wants a web site PAF does not support:
 - 2a1. System gets new suggestion from purchaser, with option to cancel use case.
- 3a. Web failure of any sort during setup:
 - 3a1. System reports failure to purchaser with advice, backs up to previous step.
 - 3a2. Purchaser either backs out of this use case or tries again.
- 4a. Computer crashes or is switched off during purchase transaction:
 - 4a1. (What do we do here?)
- 4b. Web site does not acknowledge purchase, but puts it on delay:
 - 4b1. PAF logs the delay, sets a timer to ask the purchaser about the outcome.
- 5a. Web site does not return the needed information from the purchase:
 - 5a1. PAF logs the lack of information, has the purchaser <u>update questioned</u> <u>purchase</u>.





Elementos de la descripción narrativa

(meta de actor primario) (nivel de meta[resumen, actor, subfunción]) (actor primario) (pasos de acción: sentencias completas "Poner un pedido" mostrando (meta / Dependiente) quién toma la acción... 3 - 9 pasos de extensión.) Secuencia principal de eventos: 1. Dependiente identifica cliente, item y cantidad. 2. Sistema acepta y mete en la cola el pedido. (condición que causa diferentes acciones) (paso(s) de acción que manejan esas Secuencias alterntivas de eventos: condiciones) 1a. Crédito bajo y cliente es 'Preferencial': 1a1. Sistema da crédito de todos modos. 1b. Crédito bajo y cliente no es 'Preferencial' : 1b1. Dependiente realiza escenario <u>Inscripción de Cliente</u> (llamada a <u>Preferido</u> y acepta solamente prepago. otro caso de uso) 2a. Bajo en inventario: Cliente acepta: 2a1. Dependiente reduce pedido a nivel de inventario disponible.



Relación entre precondición, postcondición, secuencia principal y secuencia alternativa

- Paralelo con los triples de Hoare:
 - {Precondición} secuencia principal {Postcondición}
- Por lo tanto:
 - para cada estado σ que satisface la precondición, la ejecución del caso de uso a partir de σ termina produciendo un estado σ' que satisface la postcondición
- ... a menos que ocurra algún imprevisto como los enumerados en la secuencia alternativa, en este caso la postondición no está garantizada

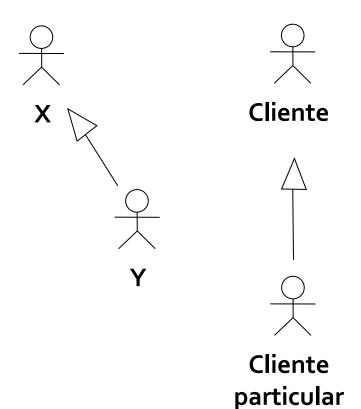


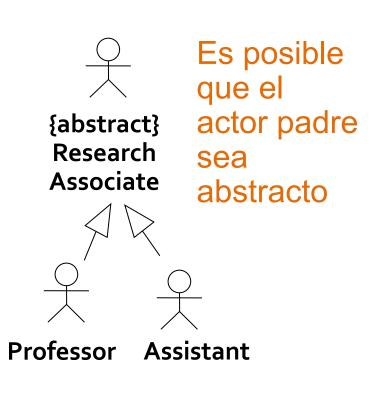
Sintaxis de la secuencia principal de eventos

- Hacer a
- Si (expresión booleana)
 Hacer b
- 3. De lo contrario
 - 3.1 Hacer c [OPCIONAL]
- 4. Para (expresión de iteración)
 - 4.1 Hacer d
- While (expresión booleana)
 - 5.1 Hacer e
 - 5.2 Hacer f



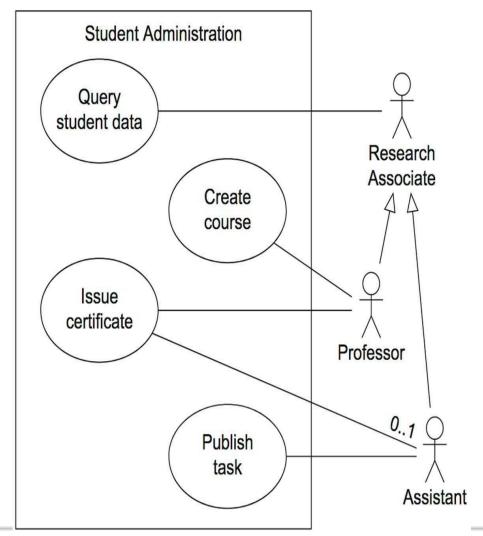
Casos de uso: generalización y especialización entre actores





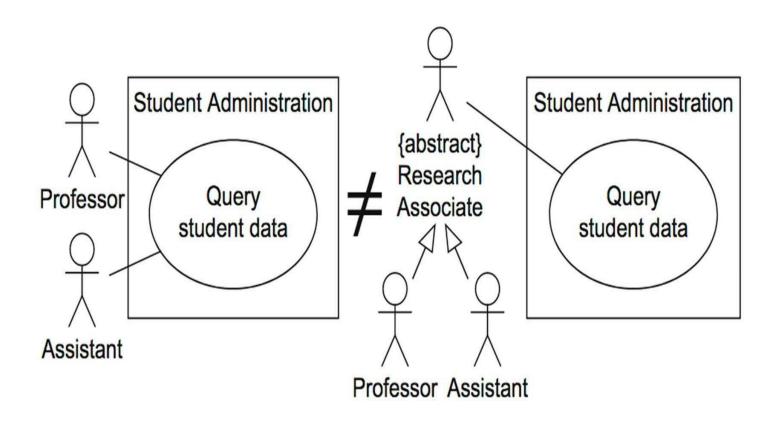


Ejemplos de generalización



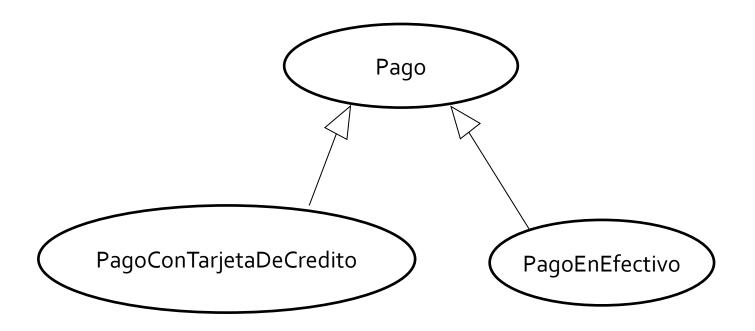


Ejemplos de generalización





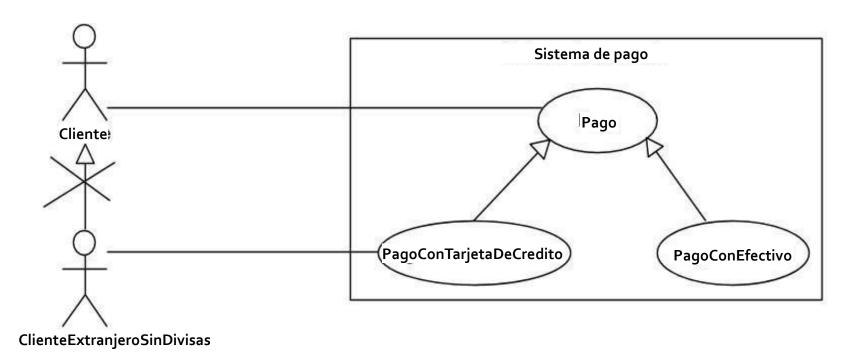
Generalización de los casos de uso





Cuidado con la generalización ...

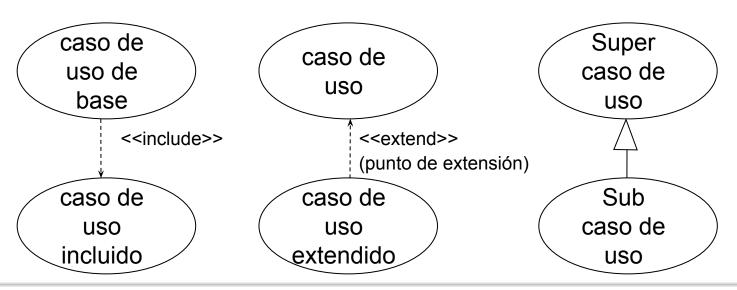
Usando la generalización, tenga cuidado de que el clasificador especializado herede todas las relaciones del clasificador padre (Liskov)





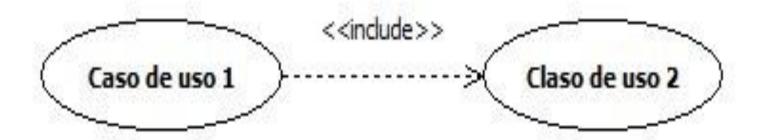
Relaciones entre casos de uso

- include: caso de uso de base incluye la funcionalidad del caso de uso incluido
- extend: un caso de uso es opcionalmente extendido por la funcionalidad de otro caso de uso
- generalización: sub caso de uso hereda comportamiento y semántica de super casos de uso





Inclusión de caso de uso



- La relación include apunta al caso de uso que se incluye
 - El caso de uso 1 incorpora la interacción descrita por el caso de uso 2 como parte de la interacción que (1) describe
- Similar a una llamada de función
 - La narrativa del caso de uso 1 debe invocar la ejecución del caso de uso incluido (2)
- <<include>> es un estereotipo
 - Los estereotipos son palabras clave que anotan elementos de un diagrama para aclarar su significado.
 - En este caso, se especializa la relación de dependencia - -> diciendo que se trata de una inclusión





Ejemplo de inclusión



 Tanto para pedir prestado un ejemplar de un libro como para extender el préstamo, es necesario reservar el libro

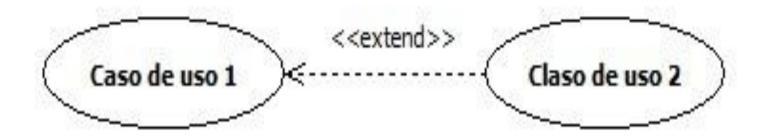


Secuencia principal de los eventos : include, caso básico

- 1. Hacer a
- Hacer b
- 3. Include (nombre del caso de uso incluido)
- 4. Hacer d
- El caso de uso incluido se especifica por separado
 - Puede ser instanciable (completo) (iniciado por un actor)
 -o no instanciable (ejecutado solo cuando se incluye)



Extensión de caso de uso

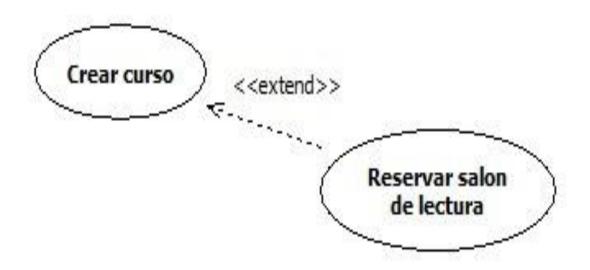


- La relación de extensión apunta al caso de uso que se está extendiendo
 - El caso de uso 1 puede incorporar la interacción descrita por el caso de uso 2
 - La narrativa del caso de uso 1 puede, en algunos casos, invocar la ejecución del caso de uso (2)
- <<extend>> es un estereotipo
 - Los estereotipos son palabras clave que anotan elementos de un diagrama para aclarar su significado.
 - En este caso, se especializa en la relación de dependencia - -> diciendo que se trata de una extensión
 - Preste atención a la dirección de la flecha opuesta a la inclusión



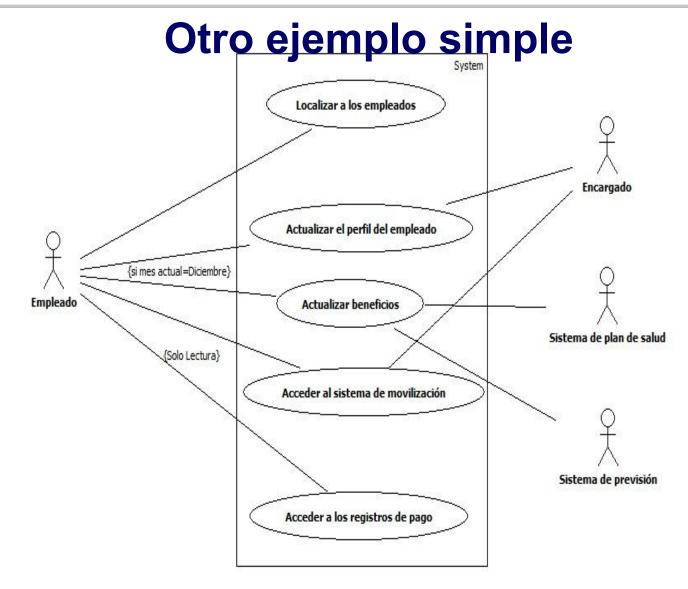


Ejemplo de Extensión



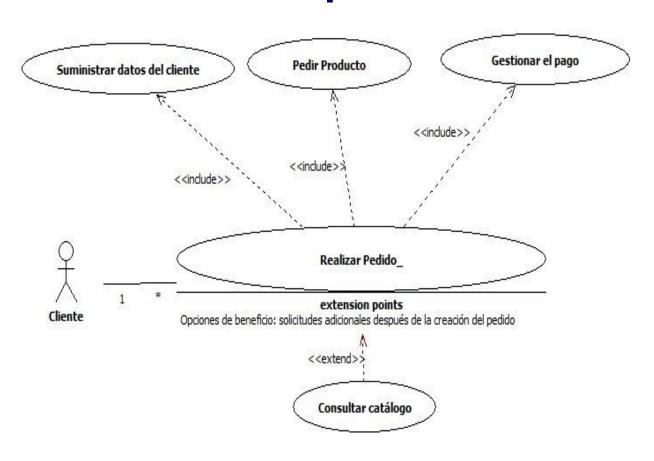
 Cuando se crea un curso nuevo se puede reservar una sala pero no es obligatorio





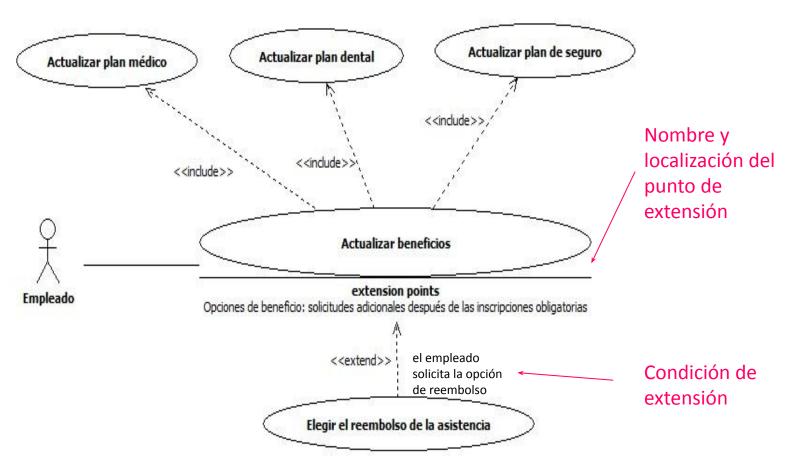


Relaciones de casos de uso no tan simples





Más relaciones de casos de uso no tan simples





¿Cómo identificar los casos de uso?

- Definir la lista de los actores luego tratar de desarrollar los casos de uso para cada actor
- Identificar eventos externos a los cuales reacciona el sistema/usuario.



HISTORIAS DE USUARIO



Historias de usuarios

- Escrito por los clientes (o usuarios finales) como cosas que el sistema necesita hacer por ellos
- Aproximadamente 3 frases en la terminología del cliente, sin detalles técnicos
- Escrito en tarjetas (!)
- Puede aumentarse con muestras (por ejemplo, Informe formateado)
- Priorizar (alto, medio, bajo)



Ejemplo de tarjeta de historia

111 Vista del instructor

La vista del instructor para un curso determinado incluye el número de curso, nombre y semestre; un botón para editar la introducción; un botón para designar las reservas de la Biblioteca y un botón para ajustar la configuración del curso.

De lo contrario, la vista del instructor es la misma que la vista del estudiante.

Prioridad alta



Continuación de Ejemplo

222 Vista del estudiante

La vista del estudiante para un curso determinado incluye el número de curso, nombre y semestre; información general del curso; e información del instructor.

Hay botones para seleccionar introducción, discusión, tablero, correo electrónico de la clase, diccionario, bloc de notas y ayuda.

Prioridad alta





Continuación de Ejemplo

123 Vista del instructor frente a la vista del alumno

Cuando un instructor selecciona uno de los cursos que está enseñando, se muestra la vista del instructor.

La vista del instructor debe incluir un botón para mostrar la vista del estudiante, y luego esa vista especial del estudiante debe incluir un botón para volver a la vista del instructor

Prioridad media



Continuación de Ejemplo

321 Acceso

Se le pide al usuario que ingrese un id y contraseña

Si la id y la contraseña no son correctos, al usuario se le muestra la pantalla predeterminada con su lista de cursos actuales

Prioridad alta



¿Por qué las historias de usuarios están en tarjetas?

- Unidad tangible de
 - Discusión
 - Estimación
 - Planificación
 - Pruebas
 - Terminación
- Usada para:
 - Construcción de tareas de ingeniería
 - Creación de pruebas de aceptación
 - Derivación de estimaciones de tiempo para la planificación



Historias de usuario para la priorización

M. Daneva et al. / The Journal of Systems and Software 86 (2013) 1333-1353

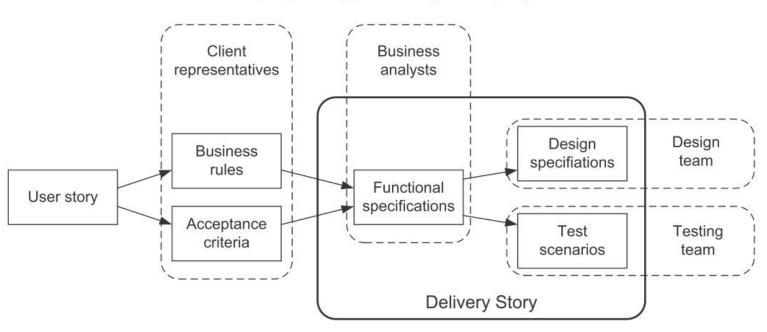


Fig. 1. The user story – delivery story translation process.

