INGENIERÍA DE SOFTWARE I

Tema 4: Ingeniería de Requisitos

Grado en Ingeniería Informática
Fecha de última modificación: 9-2-2026

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es
Dr. Miguel Ángel Conde González / mconde@usal.es

Departamento de Informática y Automática
Universidad de Salamanca

Resumen

Resumen	Es el punto de partida para un proyecto *software* y la parte más importante del proceso de desarrollo. Si los desarrolladores no conocen de forma precisa el problema a resolver, no es probable que se obtenga una solución correcta y útil. Así pues, la correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto *software*, independientemente del tipo de proyecto que se trate, dado que su mala captura es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida. La ingeniería de requisitos es la parte de la Ingeniería del Software que aborda el problema de la definición de los servicios que el sistema ha de proporcionar y de establecer las restricciones operativas del mismo. Los casos de uso se han convertido en una de las técnicas de modelado más utilizadas para la determinación y documentación de los requisitos funcionales de un sistema *software*. En este tema se presentan los conceptos y principios básicos de la ingeniería de requisitos. Así, se dará una visión global de los diferentes tipos de requisitos, para posteriormente presentar con detalle la notación que propone UML para la técnica de los casos de uso
Descriptores	Ingeniería de requisitos; requisito; restricción; obtención (elicitación) de requisitos; análisis de requisitos; especificación de requisitos *software* (ERS); modelo de casos de uso; caso de uso; actor; relaciones entre casos de uso; especificación de casos de uso
Bibliografía	[Booch et al., 2007] Capítulo 16 [Durán y Bernárdez, 2002] [Larman, 2003] Capítulos 4, 5, 6 y 7 [OMG, 2017] [Pressman, 2010] Capítulo 5 [Rumbaugh et al., 2007] [Sommerville, 2011] Capítulo 4

Esquema

- Introducción
- Ingeniería de requisitos
- Requisitos
- Especificación de requisitos del *software*
- MDB: Una metodología de elicitación de requisitos
- Vista de casos de uso en UML
- Requisitos en el Proceso Unificado
- Caso de estudio
- Aportaciones principales del tema
- Ejercicios
- Lecturas complementarias
- Referencias

https://unsplash.com/photos/4XOuAqQSj-Y

1. Introducción

Problemática (i)

"*Nunca sopla viento favorable para el que no sabe a dónde va*"
Seneca (55a.C - 39d.C)

"*La correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto software, independientemente del tipo de proyecto que se trate, dado que una mala captura de los mismos es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida*"
[Johnson, 1995]

Problemas en la obtención de requisitos

⇓

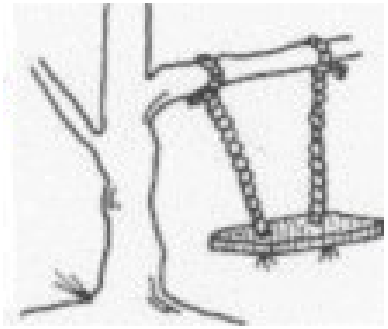
Crisis del software

"*La parte más difícil de construir de un sistema software es decidir qué construir. [...] Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después*"
[Brooks, 1995]

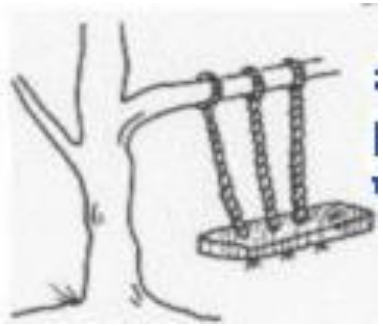
"*El coste de un cambio en los requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al coste que hubiera representado el mismo cambio durante las fases iniciales de desarrollo*"
[Pressman, 2002]

Problemática (ii)

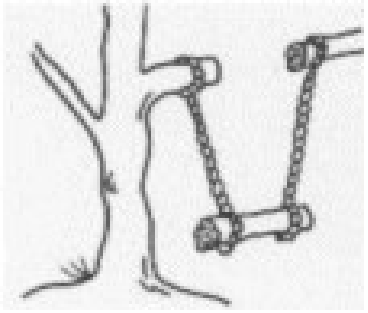
Esto es lo que pidió el usuario



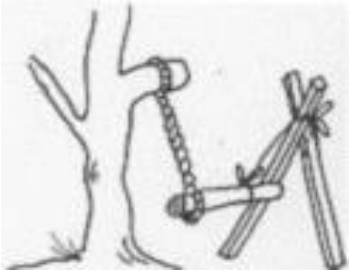
El programador lo escribió así



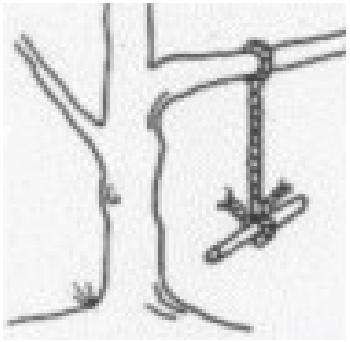
El analista lo vio de esta forma



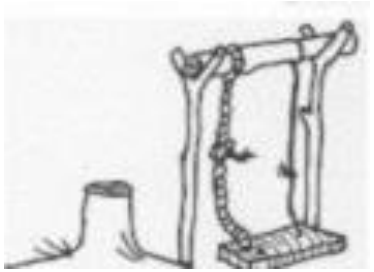
Esto es lo que quería el usuario



Así se diseñó el sistema



Así funciona el sistema en la actualidad

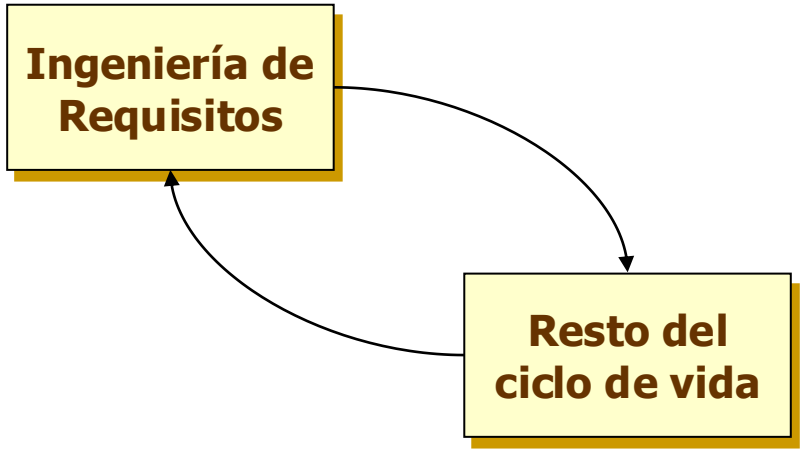


https://unsplash.com/photos/JaoVGh5aJ3E

2. Ingeniería de requisitos

Visión global

- Primera fase del ciclo de vida del *software* en la que se produce una especificación a partir de ideas informales
- Deben obtenerse y documentarse
  - Los requisitos de información
  - Los requisitos funcionales
  - Los requisitos no funcionales
  - Los criterios para medir el grado de su consecución



- El proceso de desarrollo de dicha especificación de requisitos es lo que se conoce como **ingeniería de requisitos**
- Importancia creciente de
  - El correcto entendimiento (obtención), documentación (especificación) y validación de las necesidades de los usuarios y clientes
  - La medida de la calidad de los sistemas en función del grado de satisfacción de los usuarios

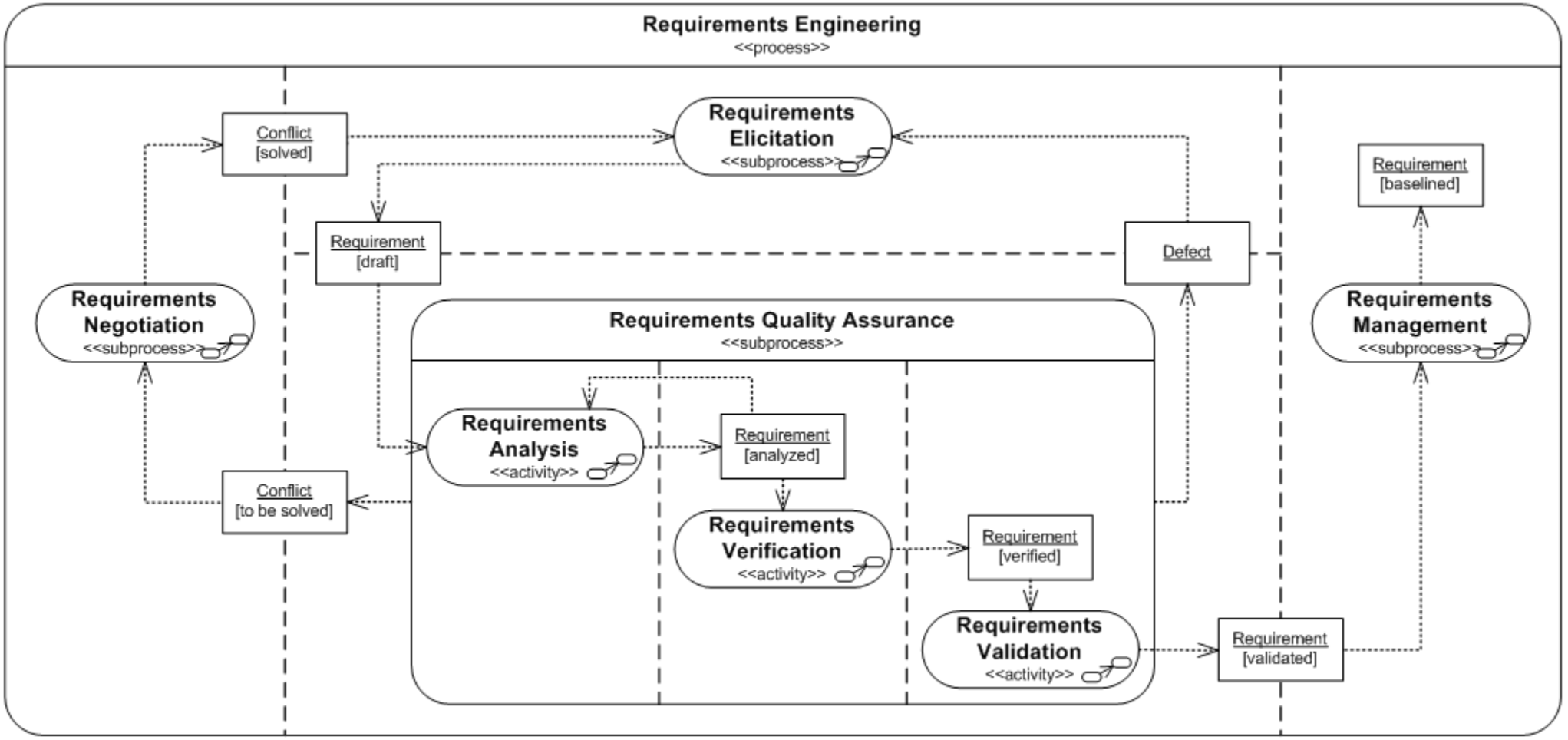
Definición (i)

- Todas las actividades relacionadas con: **(a)** identificación y documentación de las necesidades de clientes y usuarios; **(b)** creación de un documento que describe la conducta externa y las restricciones asociadas [de un sistema] que satisfará dichas necesidades; **(c)** análisis y validación del documento de requisitos para asegurar consistencia, compleción y viabilidad; **(d)** evolución de las necesidades [Hsia et al., 1993]
- … el uso sistemático de procedimientos técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo de un sistema que satisfaga las necesidades del usuario. Téngase en cuenta que todas las disciplinas de la ingeniería son semejantes, la ingeniería de requisitos no se guía por conductas esporádicas, aleatorias o por modas pasajeras, si no que se debe basar en el uso sistemático de aproximaciones contrastadas [Reifer, 1994]

Definición (ii)

- Aplicación disciplinada de principios científicos y técnicas para desarrollar, comunicar y gestionar requisitos [Christel y Kang 1992]
- El proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del conocimiento obtenido [Christel y Kang 1992]
- Un proceso sistemático de desarrollo de requisitos mediante un proceso cooperativo consistente en analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación, y comprobar la exactitud de la comprensión conseguida [Loucopoulus y Karakostas, 1995]
- Un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades [Durán, 2000]

Proceso de ingeniería de requisitos (i)



Proceso de Ingeniería de Requisitos (solo se muestran algunos productos)

Proceso de ingeniería de requisitos (ii)

- **Obtención (elicitación) de requisitos**
  - Este subproceso, probablemente el más crítico y el más difícil de realizar, tiene como objetivos buscar, investigar y ayudar a los clientes y usuarios a documentar sus necesidades
  - La documentación de los requisitos deberá hacerse siempre usando el vocabulario de clientes y usuarios, de forma que estos puedan entenderlos, siendo lo más habitual emplear lenguaje natural
  - Las técnicas más comunes son las entrevistas, reuniones en grupo, estudio *in situ*...
- **Análisis de requisitos**
  - Se denomina análisis a la distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos; Estudio, mediante técnicas informáticas, de los límites, características y posibles soluciones de un problema al que se aplica un tratamiento por ordenador [RAE, 2025]
  - Esta actividad es parte del subproceso de aseguramiento de la calidad
  - Tiene como objetivo principal detectar conflictos en los requisitos obtenidos, normalmente mediante técnicas de modelado conceptual y de prototipado de interfaz de usuario
  - Los modelos generados son también una importante herramienta de comunicación con diseñadores y programadores

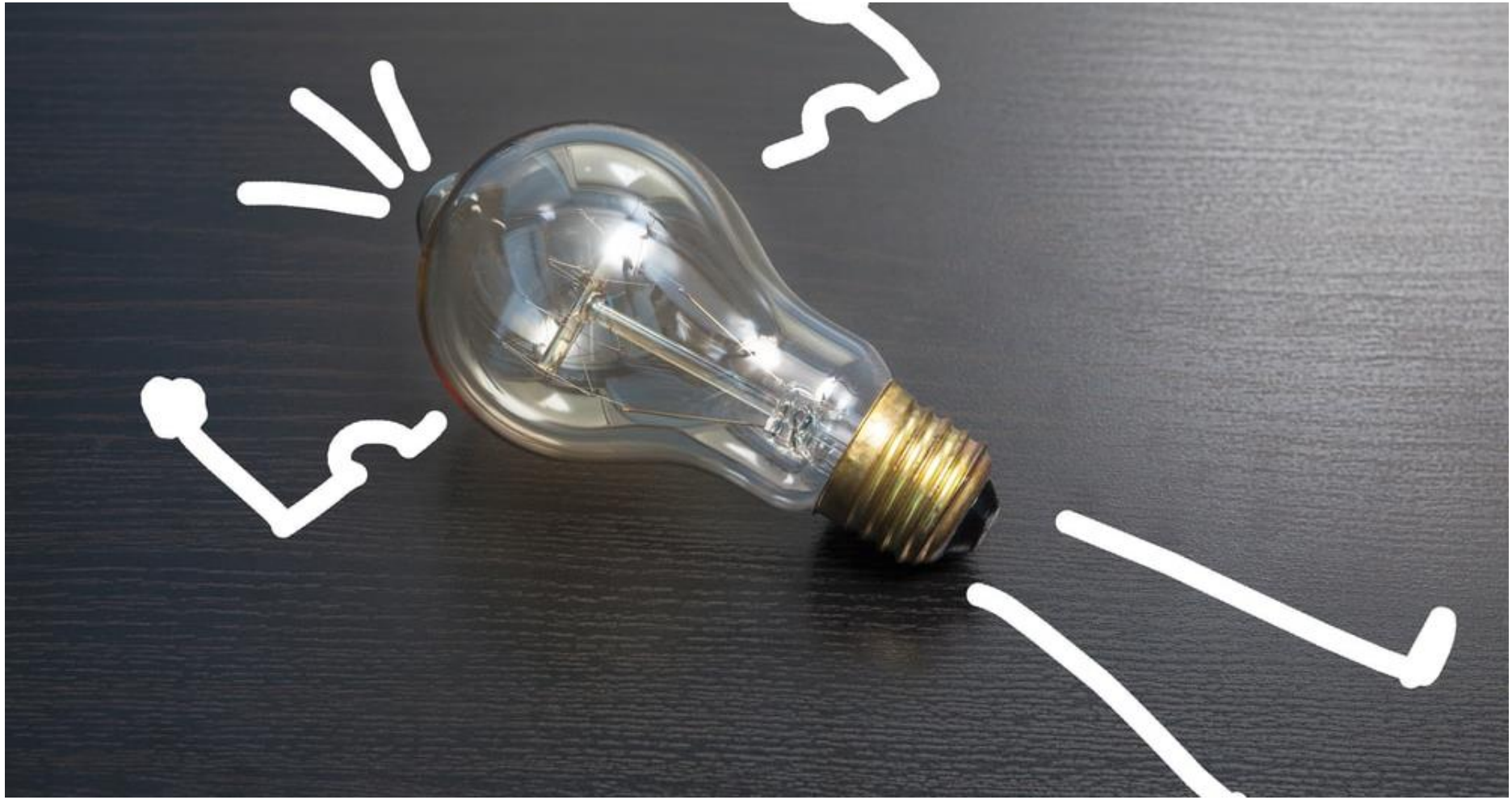
Proceso de ingeniería de requisitos (iii)

- **Verificación de requisitos**
  - Esta actividad de calidad tiene como objetivo detectar defectos en los requisitos previamente analizados, normalmente mediante técnicas como revisiones formales, listas de comprobación (*checklists*)…
- **Validación de requisitos**
  - Esta tercera actividad de calidad intenta asegurar que los requisitos verificados reflejan realmente las necesidades de clientes y usuarios
  - Las técnicas empleadas suelen ser reuniones en las que se revisan los requisitos mediante el apoyo de prototipos de interfaz de usuario
- **Negociación de requisitos**
  - El objetivo de este subproceso es buscar soluciones a los conflictos detectados que satisfagan a los distintos *stakeholders*
- **Gestión de requisitos**
  - Este subproceso gestiona todo el proceso, en especial las peticiones de cambios en los requisitos, el impacto de dichas peticiones, las distintas versiones de los requisitos…

El factor humano en la ingeniería de requisitos

- La comunicación es uno de los aspectos más destacables en la ingeniería de requisitos

- Esta característica hace de la ingeniería de requisitos una disciplina especialmente compleja al intervenir el factor humano
  - Este factor es el responsable de que la Ingeniería de Requisitos tenga aspectos sociales y culturales y no solo técnicos [Goguen, 1994]



https://unsplash.com/photos/9upRLjfKP8

3. Requisitos

¿Qué describe un requisito?

- Una utilidad para el usuario
  - *"El tratamiento de textos ha de incluir la comprobación y corrección gramatical"*
- Una propiedad general del sistema
  - *"El sistema ha de garantizar que la información personal solamente será accesible mediante autorización explícita"*
- Una restricción general del sistema
  - *"El sensor ha de muestrearse 10 veces por segundo"*
- Cómo llevar a cabo cierto cálculo
  - *"Calificación final = nota examen + 2*nota trabajo + 2/3 nota ejercicios"*
- Una restricción sobre el desarrollo del sistema
  - *"El sistema ha de implementarse en C#"*

Concepto de requisito (i)

- Condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado [Piattini et al., 1996]
- **(a)** Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo. **(b)** Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal. **(c)** Una representación en forma de documento de una condición o capacidad como las expresadas en **(a)** o en **(b)** [IEEE, 1999a]
- Una característica del sistema que es una condición para su aceptación [DoD, 1994]
- Una propiedad que debe exhibirse para solucionar algún problema del mundo real [Sawyer y Kontoya, 2001]

Concepto de requisito (ii)

- Aparente simplicidad del concepto
- Es frecuente encontrar el término **requisito** calificado con adjetivos que pueden resultar confusos en un primer momento
  - de sistema
  - *hardware*
  - *software*
  - de usuario
  - de cliente
  - funcional
  - no funcional
  - ...

Dimensiones de los requisitos (i)

- La gran cantidad de calificativos que se aplican al término requisito muestra distintos aspectos ortogonales que a menudo se consideran aisladamente
- Para clarificar la situación es frecuente identificar dimensiones para clasificar los requisitos
  - Ámbito
  - Característica que define
  - Audiencia
  - Representación

Dimensiones de los requisitos (ii)

- **Ámbito**
  - Indica en qué contexto se debe entender el requisito
    - *Sistema*, *Software*, *Hardware*
- **Característica que define**
  - Los requisitos se clasifican en función de la naturaleza de la característica del sistema que se especifica
    - *Requisitos funcionales, Requisitos no funcionales*
- **Audiencia**
  - Indica a quién está dirigido el requisito, es decir, las personas que deben ser capaces de entenderlo. Implica un nivel de descripción determinado
    - *Requisitos-C, Requisitos-D* [Rombach, 1990; Brackett, 1990]
    - *Requisitos de usuario* o *Requisitos de cliente, Requisitos software* [Mazza et al., 1994]
    - *Requisitos de usuario, Requisitos de sistema, Especificaciones de diseño* [Sommerville, 2002]
- **Representación**
  - Establece la forma cómo se definen los requisitos
    - *Formal, Semiformal, No formal*

Dimensiones de los requisitos (iii)

Definición de requisitos de usuario

1. El *software* debe proporcionar un medio para la representación y acceso a ficheros externos creados por otras herramientas

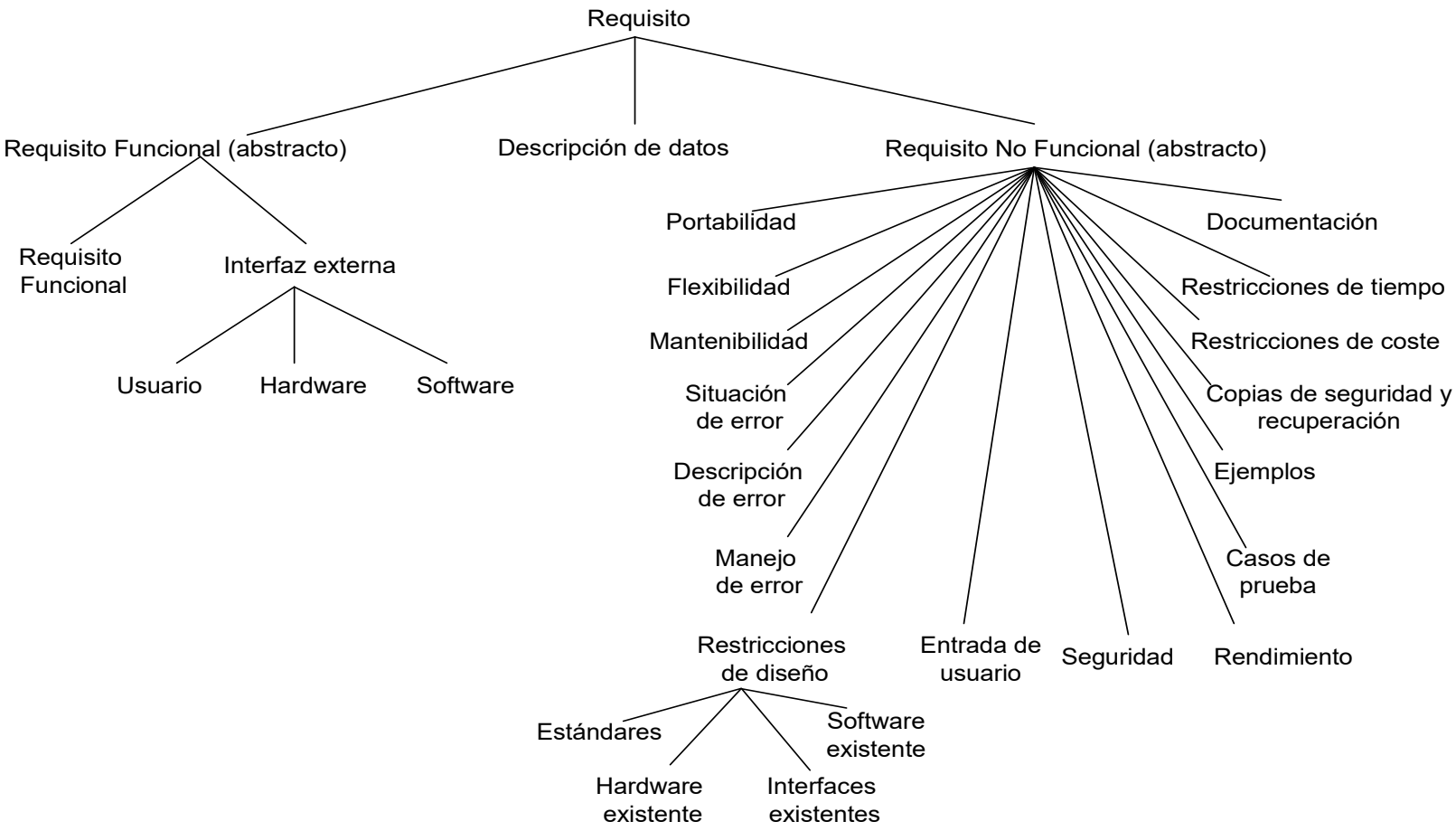
Especificación de requisitos del sistema

1.1 Hay que proporcionar al usuario utilidades para definir el tipo de los ficheros externos
1.2 Cada tipo de fichero externo tendrá asociado una herramienta que podrá ser aplicada al fichero
1.3 Cada tipo de fichero externo podrá estar representado mediante un icono específico en la interfaz del usuario
1.4 Se proporcionarán utilidades para que el usuario pueda definirse el tipo de icono que asociará a cada tipo de fichero
1.5 Cuando un usuario seleccione un icono que representa un tipo de fichero externo, el efecto de la selección será la aplicación de la herramienta asociada con el tipo de fichero externo al fichero representado por el icono seleccionado

Dimensiones de los requisitos (iv)

- K. Pohl (1997) establece una completa clasificación de requisitos, RSM (*Requirements Specification Model*)
  - Las principales categorías son
    - Requisitos funcionales
      - Describen la funcionalidad o los servicios que se espera que este proveerá
        - De usuario: descripción general
        - De sistema: descripción detallada (función, entradas, salidas...)
    - Requisitos de datos
    - Requisitos no funcionales

Dimensiones de los requisitos (v)



Jerarquía de especialización de RSM – adaptado de [Pohl, 1997]

Dimensiones de los requisitos (vi)

- *"El sistema ha de mantener el registro de todo el material de la biblioteca incluyendo libros, revistas, vídeos, informes, CD-Roms..."*
  - Requisito general expresado en términos generales. Qué tiene que hacer el sistema
- *"El sistema debe permitir a los usuarios buscar un ejemplar por título, autor o ISBN"*
  - Requisito funcional que define una parte de funcionalidad del sistema
- *"La interfaz del usuario ha de implementarse mediante un navegador web"*
  - Requisito de implementación, establece cómo ha de implementarse el sistema
- *"El sistema ha de soportar al menos 20 transacciones por segundo"*
  - Requisito de rendimiento que especifica el rendimiento mínimo aceptable para ese sistema

Problemas con los requisitos

- Los requisitos no reflejan las necesidades reales del cliente
- Requisitos inconsistentes o incompletos
- El cambio de requisitos, una vez acordados, es muy costoso
- Problemas de comunicación
  - Incomprensiones entre los clientes, los que desarrollan los requisitos y los ingenieros de *software* que desarrollan o mantienen el sistema

Requisito vs. restricción (i)

- Dada una lista de deseos y necesidades, ¿cuáles son los requisitos y cuáles son las restricciones?
  - Las restricciones limitan la forma en la que se puede resolver un problema

El programa tiene que imprimir transparencias de color en blanco y negro estableciendo la correspondencia de forma automática e imprimir sobre una impresora postscript

Requisito vs. restricción (ii)

Rq1. Imprimir transparencias de color en una impresora de blanco y negro

Rq2. Establecer la correspondencia al blanco y negro de forma automática

El programa tiene que imprimir transparencias de color en blanco y negro estableciendo la correspondencia de forma automática e imprimir sobre una impresora postscript

Requisito vs. restricción (iii)

Restricción: Utilizar el formato *postscript*

El programa tiene que imprimir transparencias de color en blanco y negro estableciendo la correspondencia de forma automática e imprimir sobre una impresora postscript

Requisito vs. restricción (iv)

- Indicación: Aquello que es tangible o visible para el usuario es normalmente un requisito
  - Los dos primeros son visibles para el usuario el tercero no

El programa tiene que imprimir transparencias de color en blanco y negro estableciendo la correspondencia de forma automática e imprimir sobre una impresora postscript

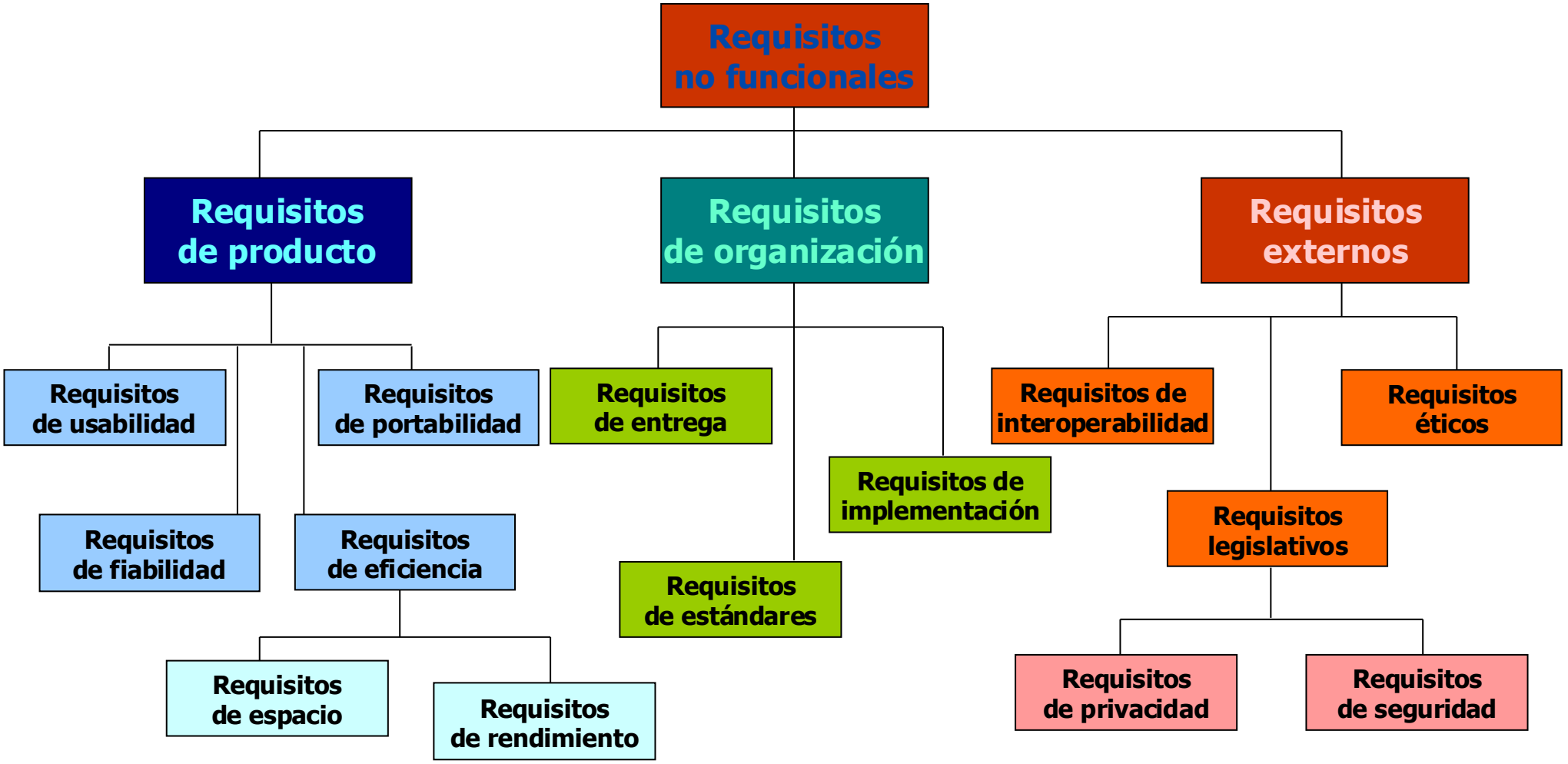
Requisitos no funcionales (i)

- Requisitos no relacionados directamente con la funcionalidad del sistema
- Pueden estar relacionados con propiedades emergentes del sistema
- Pueden describir restricciones al producto a desarrollar
- Pueden describir restricciones externas del sistema
- Definen las cualidades globales que el sistema ha de exhibir
- Suelen hacer referencia al sistema considerado de forma global
- Suelen ser requisitos más críticos que los requisitos funcionales
- Suelen ser difíciles de verificar

Requisitos no funcionales (ii)

- Clasificación de los requisitos no funcionales [Sommerville, 2002]
  - **Requisitos de producto**
    - Especifican el comportamiento del producto
      - Tiempo de respuesta, memoria requerida, fiabilidad, portabilidad, usabilidad…
  - **Requisitos de organización**
    - Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador
      - Estándares de proceso, lenguajes de programación, métodos de diseño, estándares de documentación...
  - **Requisitos externos**
    - Factores externos al sistema y de su proceso de desarrollo
      - Interoperabilidad, éticos, legislativos, privacidad, seguridad...

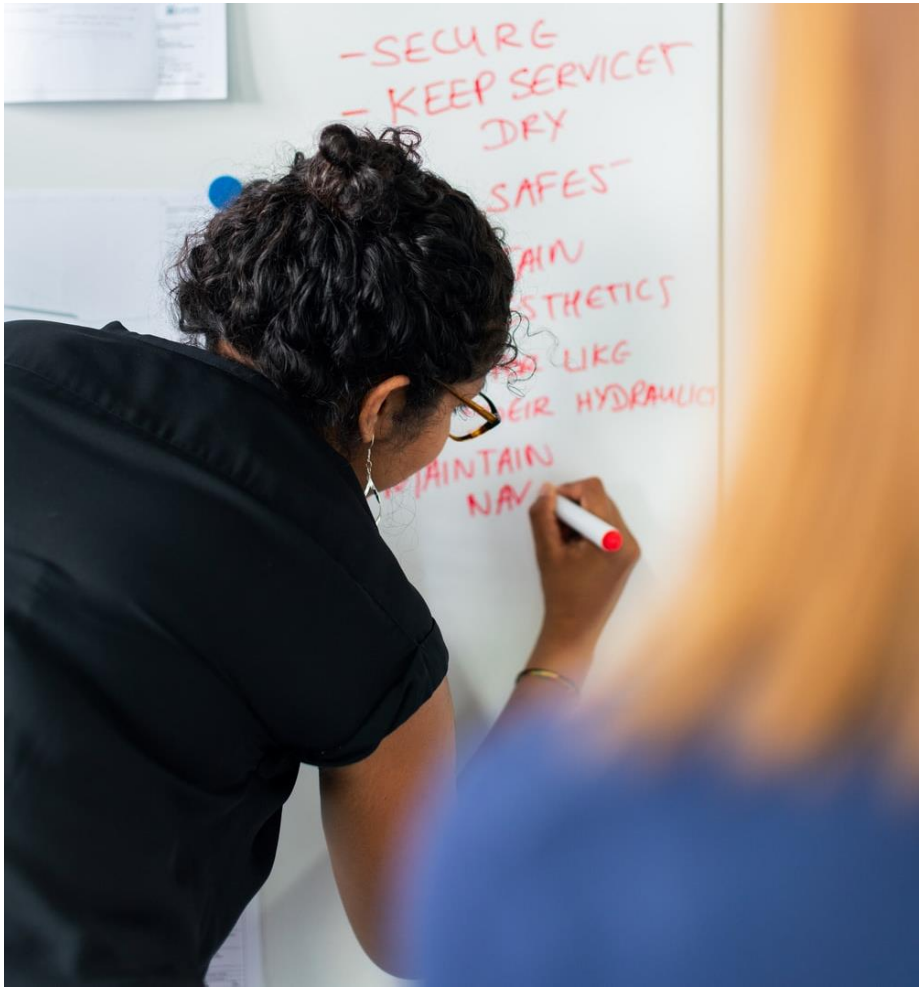
Requisitos no funcionales (iii)



[Sommerville, 2002]

Requisitos no funcionales (iv)

- *"El máximo espacio de almacenamiento ocupado por el sistema debe ser de 8 MB porque el sistema debe alojarse completamente en una memoria de solo lectura e instalarse en el coche"*
  - Requisito de producto que define una restricción en el tamaño del producto
- *"El proceso software y los documentos a realizar deben conformar el proceso y los estándares de documentación recogidos en la norma TELMo-ES-2003"*
  - Requisito de organización que especifica que el sistema debe desarrollarse de acuerdo a un proceso estándar dentro de la compañía
- *"El sistema no debe revelar ninguna información personal sobre los clientes excepto su nombre y su número de referencia"*
  - Requisito externo se deriva de la necesidad del sistema de cumplir la legislación vigente sobre protección de datos



https://unsplash.com/photos/-68PdYVLh3U

4. Especificación de requisitos del software

Especificación de requisitos (i)

- Los requisitos se recogen en documentos técnicos que reciben el nombre genérico de ERS (Especificación de Requisitos del *Software*)
- Este documento debe contemplar tanto los requisitos–C como los requisitos–D
- Hay metodologías que abogan por la separación de estas representaciones en dos documentos diferentes
  - DRS (Documento de Requisitos del Sistema), también denominado catálogo de requisitos, donde se recogen los requisitos–C
  - ERS propiamente dicha, donde se recogen los requisitos–D
- En la realización de una ERS participan
  - Ingenieros de *software* (analistas)
  - Clientes y usuarios

Especificación de requisitos (ii)

Una especificación es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema [IEEE, 1999a]

La ERS es la documentación de requisitos esenciales (funciones, rendimiento, diseño, restricciones y atributos) del *software* y de sus interfaces [IEEE, 1999a]

Propiedades deseables de los requisitos (i)

- **Comprensible por clientes y usuarios**
  - Una especificación debe servir como canal de comunicación entre los participantes en el proceso de ingeniería de requisitos
  - La mejor forma de lograr esta comunicación es pensar en la audiencia a la que van dirigidos los requisitos
- **No ambigua**
  - Cada requisito solo tiene una interpretación
- **Completa**
  - Incluye todos los requisitos significativos
  - Define la respuesta a todo tipos de entradas
  - Es conforme con el estándar de especificación a cumplir
  - Están etiquetadas y referenciadas todas las figuras, tablas...

Propiedades deseables de los requisitos (ii)

- **Consistente**
  - No hay conflictos ni contradicciones
  - Es consistente externamente si y solo si todo requisito contenido en ella no está en conflicto con otros documentos de nivel superior
  - Es consistente internamente si y solo si no existen conflictos entre los requisitos que contiene
  - Tipos de conflictos entre requisitos [Davis, 1993]
    - **Conflictos de conducta.** Dos o más requisitos especifican conductas distintas del sistema para las mismas condiciones y el mismo estímulo externo
    - **Conflictos de términos.** Se utilizan términos distintos para referirse al mismo concepto
    - **Conflictos de característica.** Dos o más requisitos especifican aspectos contradictorios para la misma característica del sistema
    - **Conflictos temporales.** Dos o más requisitos exigen características temporales contradictorias al sistema

Propiedades deseables de los requisitos (iii)

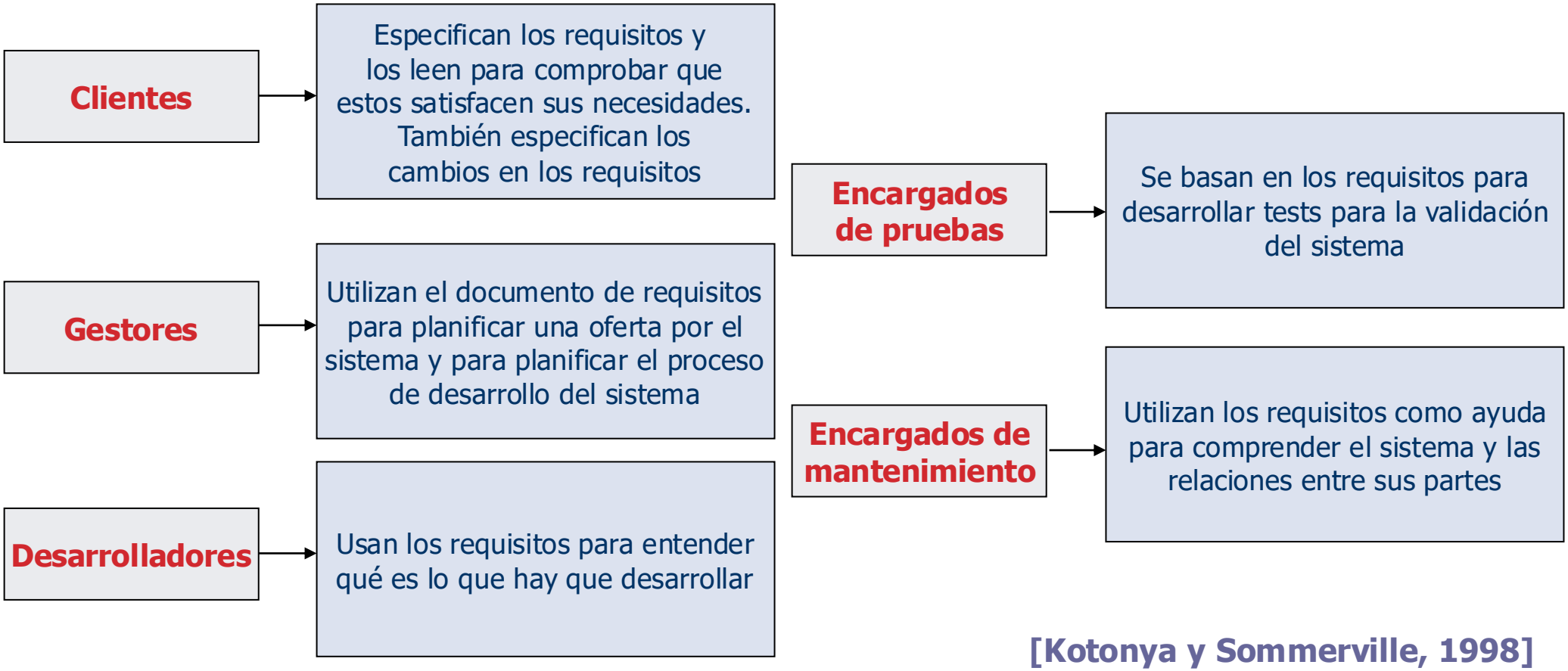
- **Verificable**
  - Todo requisito contenido en ella es verificable. Existe un proceso finito y de coste razonable por el que una persona o una máquina pueda comprobar que el sistema final cumple el requisito
  - Una condición absolutamente necesaria para que un requisito sea verificable es que no sea ambiguo y que se defina de forma mensurable
  - Los procedimientos de observación para comprobar que el sistema cumple los requisitos son la base para las pruebas de aceptación por parte del cliente [Wieringa, 1996]
- **Modificable**
  - Su estructura y estilo de redacción permiten que los cambios se puedan realizar fácil, completa y consistentemente
  - Debe tener una organización coherente
  - No debe ser redundante
  - Los requisitos deben expresarse individualmente y no de forma conjunta

Propiedades deseables de los requisitos (iv)

- **Trazable**
  - Para cada requisito contenido en ella se conoce su origen y puede referenciarse como origen en posteriores documentos
  - Cada requisito puede seguirse hacia atrás y hacia delante
- **Anotada con importancia y estabilidad**
  - Cada requisito contenido en ella está anotado con la importancia que tiene su cumplimiento para clientes y usuarios y la estabilidad que se espera del requisito, es decir, la probabilidad de que cambie durante el desarrollo
- **Independiente del diseño y de la implementación**
  - No especifica una determinada descomposición del sistema (arquitectura) ni ningún aspecto de su posible implementación
  - Solo deben admitirse requisitos que limiten la libertad de los diseñadores y programadores en el caso de que el cliente lo solicite explícitamente

Usuarios de una ERS

- El documento de requisitos tiene un conjunto de diversos usuarios que requieren un uso diferente del mismo



[Kotonya y Sommerville, 1998]

Estructura de una ERS (i)

1. Introducción
- 1.1. Objetivo
- 1.2. Ámbito
- 1.3. Definiciones, acrónimos y abreviaturas
- 1.4. Referencias
- 1.5. Descripción del resto del documento

2. Descripción general
- 2.1. Perspectiva del producto
- 2.2. Funciones del producto
- 2.3. Características del usuario
- 2.4. Limitaciones generales
- 2.5. Supuestos y dependencias

3. Requisitos específicos
- 3.1. Requisitos funcionales
- 3.2. Requisitos de interfaz externa
- 3.3. Requisitos de ejecución
- 3.4. Restricciones de diseño
- 3.5. Atributos de calidad
- 3.6. Otros requisitos

Apéndices
Índice

IEEE Std. 830-1998 [IEEE, 1999b]

Estructura de una ERS (ii)

3. Requisitos específicos
   - 3.1. Requisitos funcionales
     - 3.1.1. Requisito funcional 1
       - 3.1.1.1. Introducción
       - 3.1.1.2. Entradas
       - 3.1.1.3. Salidas
     - 3.1.2. Requisito funcional 2
     - ……
     - 3.1.n. Requisito funcional n
   - 3.2. Requisitos de interfaz externa
     - 3.2.1. Interfaces de usuario
     - 3.2.2. Interfaces *hardware*
     - 3.2.3. Interfaces *software*
     - 3.2.4. Interfaces de comunicaciones
   - 3.3. Requisitos de ejecución
   - 3.4. Restricciones de diseño
     - 3.4.1. Acatamiento de estándares
     - 3.4.2. Limitaciones *hardware*
     - ……
   - 3.5. Requisitos no funcionales (atributos de calidad)
     - 3.5.1. Seguridad
     - 3.5.2. Mantenimiento
     - ……
   - 3.6. Otros requisitos
     - 3.6.1. Base de datos
     - 3.6.2. Operaciones
     - 3.6.3. Adaptación de situación

IEEE Std. 830-1998 [IEEE, 1999b]

Estructura de una ERS (iii)

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas

- **1. Introducción**
- **2. Participantes en el proyecto**
- **3. Descripción del sistema actual**
- **4. Objetivos del sistema**
- **5. Catálogo de requisitos del sistema**
  - 5.1 Requisitos de información
  - 5.2 Requisitos funcionales
    - 5.2.1 Diagramas de casos de uso
    - 5.2.2 Definición de actores
    - 5.2.3 Casos de uso del sistema
  - 5.3 Requisitos no funcionales
- **6. Matriz de rastreabilidad objetivos/requisitos**
- **7. Glosario de términos**
- **8. Conflictos pendientes de resolución *[opcional, pueden ir en un documento aparte]***

Apéndices *[opcionales]*

[Durán y Bernárdez, 2002]

5. MDB: Una metodología de elicitación de requisitos

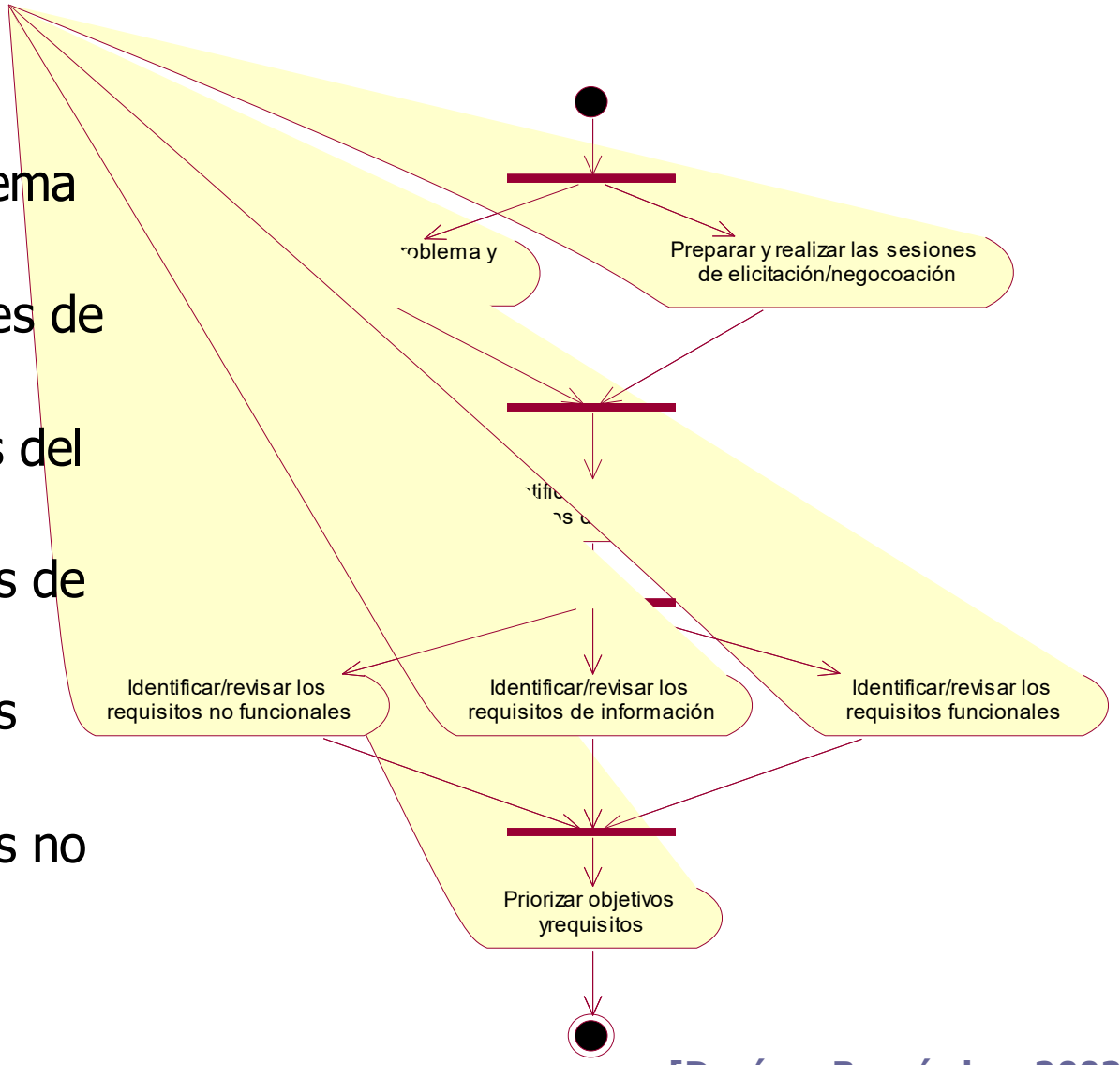
https://unsplash.com/photos/jfR5wu2hMI0

Introducción

- MDB – Método de Durán y Bernárdez
- Desarrollada en el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla
- Metodología para la obtención y documentación de los requisitos de sistemas de información
- Principales referencias [Durán, 2000; Durán y Bernárdez, 2002]
- Soportada por herramienta CASE
  - REM (*REquirements Manager*) [Durán et al., 2002]
  - Existe una versión más actualizada que se denomina REMUS y está accesible en https://bit.ly/330O9w9

Tareas

- Obtener información sobre el dominio del problema y el sistema actual
- Preparar y realizar las reuniones de obtención/negociación
- Identificar/revisar los objetivos del sistema
- Identificar/revisar los requisitos de información
- Identificar/revisar los requisitos funcionales
- Identificar/revisar los requisitos no funcionales
- Priorizar objetivos y requisitos



[Durán y Bernárdez, 2002]

Técnicas

- Se recomiendan diversas técnicas para llevar a cabo las tareas anteriores
- Las más importantes son los casos de uso, las plantillas y los patrones lingüísticos
- Para la correcta descripción de los casos de uso (que son una forma de expresar requisitos funcionales), los requisitos no funcionales, los requisitos de información, así como de otros requisitos del sistema, se recurre a la utilización de diversos tipos de plantillas
  - Objetivos del sistema
  - Requisitos de información
  - Requisitos de restricción (reglas de negocio)
  - Actores
  - Casos de uso
  - Requisitos funcionales (expresados de forma tradicional, como texto libre)
  - Requisitos no funcionales
- Cada plantilla presenta los campos de información necesarios para especificar el concepto que está representando

Estructura del documento de requisitos del sistema

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas

1. Introducción
2. Participantes en el proyecto
3. Descripción del sistema actual
4. Objetivos del sistema
5. Catálogo de requisitos del sistema
   - 5.1 Requisitos de información
   - 5.2 Requisitos funcionales
     - 5.2.1 Diagramas de casos de uso
     - 5.2.2 Definición de actores
     - 5.2.3 Casos de uso del sistema
   - 5.3 Requisitos no funcionales
6. Matriz de rastreabilidad objetivos/requisitos
7. Glosario de términos
8. Conflictos pendientes de resolución *[opcional, pueden ir en un documento aparte]*

Apéndices *[opcionales]*

[Durán y Bernárdez, 2002]

Plantilla para los objetivos del sistema

- Los objetivos del sistema pueden considerarse como **requisitos de alto nivel** [Sawyer y Kontoya, 2001], de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos

OBJ-<id>	**<nombre descriptivo>**
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual (<organización del autor>)
Fuentes	• <fuente de la versión actual> (<organización de la fuente>)
Descripción	El sistema deberá <objetivo a cumplir por el sistema>
Subobjetivos	• OBJ-x <nombre del subobjetivo> • …
Importancia	<importancia del objetivo>
Urgencia	<urgencia del objetivo>
Estado	<estado del objetivo>
Estabilidad	<estabilidad del objetivo>
Comentarios	<comentarios adicionales sobre el objetivo>

[Durán y Bernárdez, 2002]

Plantillas para requisitos de información (i)

IRQ-<id>	**<nombre descriptivo>**	
Versión	<nº de la versión actual> (<fecha de la versión actual>)	
Autores	• <autor de la versión actual (<organización del autor>)	
Fuentes	• <fuente de la versión actual> (<organización de la fuente>)	
Objetivos asociados	• OBJ-x <nombre del objetivo> • …	
Requisitos asociados	• Rx-y <nombre del requisito> • …	
Descripción	• El sistema deberá <capacidad del sistema>	
Datos específicos	• <datos específicos sobre el concepto relevante> • …	
Tiempo de vida	Medio	Máximo
	<tiempo medio de vida>	<tiempo máximo de vida>
Ocurrencias simultáneas	Medio	Máximo
	<nº medio de ocurr. Simul.>	<nº máximo de ocurr. Simul.>
Importancia	<importancia del objetivo>	
Urgencia	<urgencia del objetivo>	
Estado	<estado del objetivo>	
Estabilidad	<estabilidad del objetivo>	
Comentarios	<comentarios adicionales sobre el objetivo>	

[Durán y Bernárdez, 2002]

Plantillas para requisitos de información (ii)

CRQ-<id>	**<nombre descriptivo>**
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual (<organización del autor>)
Fuentes	• <fuente de la versión actual> (<organización de la fuente>)
Objetivos asociados	• OBJ-x <nombre del objetivo> • …
Requisitos asociados	• Rx-y <nombre del requisito> • …
Descripción	• El sistema deberá <capacidad del sistema>
Importancia	<importancia del objetivo>
Urgencia	<urgencia del objetivo>
Estado	<estado del objetivo>
Estabilidad	<estabilidad del objetivo>
Comentarios	<comentarios adicionales sobre el objetivo>

[Durán y Bernárdez, 2002]

Plantilla para actores

- Aunque los actores de los casos de uso no son requisitos, por homogeneidad con el estilo de definición del resto de los elementos que componen el catálogo de requisitos se ha descrito la plantilla para definirlos

ACT-<id>	**<nombre descriptivo>**
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual (<organización del autor>)
Fuentes	• <fuente de la versión actual> (<organización de la fuente>)
Descripción	El sistema deberá <objetivo a cumplir por el sistema>
Comentarios	<comentarios adicionales sobre el objetivo>

[Durán y Bernárdez, 2002]

Plantilla para requisitos funcionales

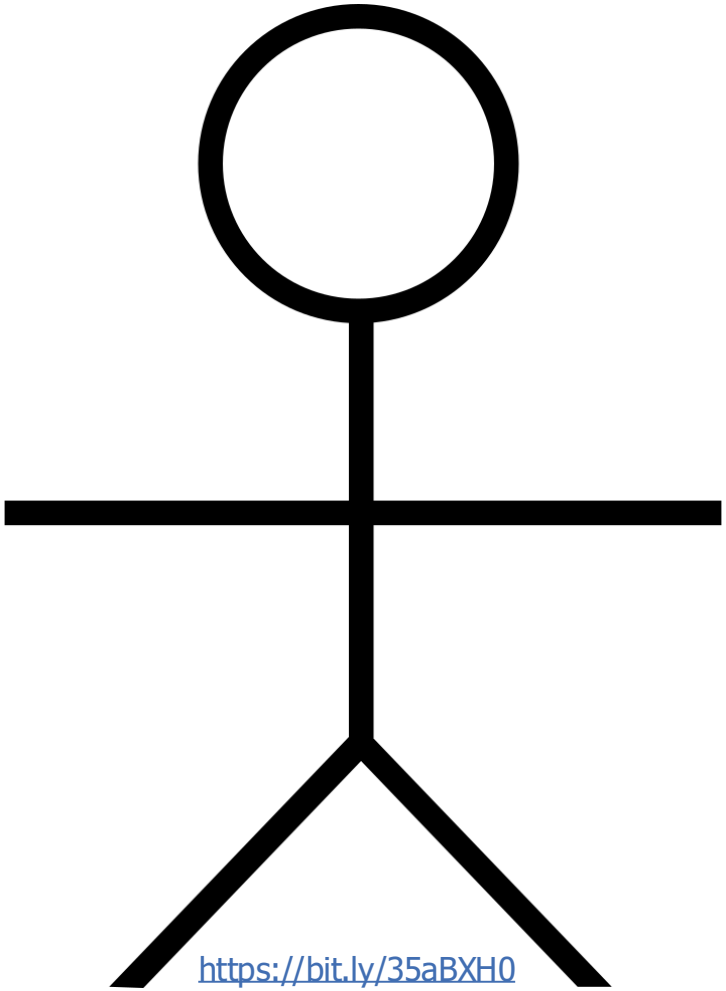
CU-<*id*>	***<nombre descriptivo>***	
Versión	*<nº de la versión actual>* (*<fecha de la versión actual>*)	
Autores	• *<autor de la versión actual>* (*<organización del autor>*) ...	
Fuentes	• *<fuente de la versión actual>* (<organización de la fuente>) ...	
Objetivos asociados	• OBJ-*x* *<nombre del objetivo>* ...	
Requisitos asociados	• Rx-y *<nombre del requisito>* ...	
Descripción	El sistema debe comportarse tal como se describe en el siguiente caso de uso {abstracto durante la realización de los siguientes casos de uso: *<lista de casos de uso>*, cuando *<evento de activación>* [o durante la realización de los siguientes casos de uso: *<lista de casos de uso>*]}	
Precondición	*<precondición del caso de uso>*	
Secuencia normal	**Paso**	**Acción**
	p_1	{El actor *<actor>*, El sistema} *<acción/es realizada/s por actor/sistema>*
	p_2	Se realiza el caso de uso *<caso de uso (RF-x)>*
	p_3	Si *<condición>*, {el actor *<actor>*, el sistema} *<acción/es realizada/s por actor/sistema>*
	p_4	Si *<condición>*, se realiza el caso de uso *<caso de uso (RF-x)>*
	...	...
Poscondición	*<poscondición del caso de uso>*	
Excepciones	**Paso**	**Acción**
	p_i	Si *<condición excepción>*, {el actor *<actor>*, el sistema} *<acción/es realizada/s por actor/sistema>*, a continuación este caso de uso {continúa, queda sin efecto}
	p_j	Si *<condición excepción>*, se realiza el caso de uso *<caso de uso (RF-x)>*, a continuación este caso de uso {continúa, queda sin efecto}
	...	...
Rendimiento	**Paso**	**Acción**
	q	*m <unidad de tiempo>*
	...	...
Frecuencia	*<nº de veces>* veces / *<unidad de tiempo>*	
Importancia	*<importancia del requisito>*	
Urgencia	*<urgencia del requisito>*	
Estado	*<estado del requisito>*	
Estabilidad	*<estabilidad del requisito>*	
Comentarios	*<comentarios adicionales sobre el requisito>*	

[Durán y Bernárdez, 2002]

Plantilla para requisitos no funcionales

NFR-<id>	**<nombre descriptivo>**
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual (<organización del autor>)
Fuentes	• <fuente de la versión actual> (<organización de la fuente>)
Objetivos asociados	• OBJ-x <nombre del objetivo> • …
Requisitos asociados	• Rx-y <nombre del requisito> • …
Descripción	• El sistema deberá <capacidad del sistema>
Importancia	<importancia del objetivo>
Urgencia	<urgencia del objetivo>
Estado	<estado del objetivo>
Estabilidad	<estabilidad del objetivo>
Comentarios	<comentarios adicionales sobre el objetivo>

[Durán y Bernárdez, 2002]



https://bit.ly/35aBXH0

6. Vista de casos de uso

Introducción

- La vista de casos de uso captura la funcionalidad de un sistema, de un subsistema, o de una clase, tal como se muestra a un usuario exterior
- Reparte la funcionalidad del sistema en transacciones significativas para los usuarios ideales de un sistema
- Los usuarios del sistema se denominan **actores** y las particiones funcionales se conocen con el nombre de **casos de uso**
- La técnica que se utiliza para modelar esta vista es el diagrama de casos de uso

Generalidades sobre los casos de uso

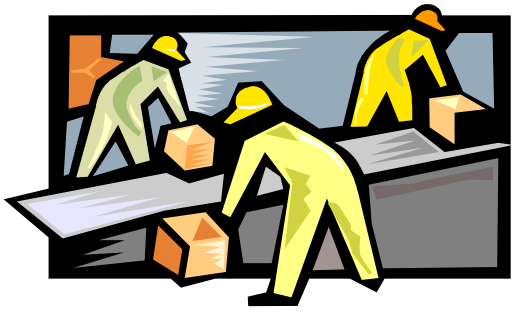
- Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por **Ivar Jacobson** [Jacobson, 1987; Jacobson et al., 1992] e incorporadas actualmente en UML [OMG, 2017]
- No pertenecen estrictamente al enfoque orientado a objetos
  - Técnica de captura de requisitos
  - Los casos de uso cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos
  - Están basados en el lenguaje natural
    - Accesibles a los usuarios
  - Presentan ventajas sobre la descripción meramente textual de los requisitos funcionales

Elementos de un diagrama de casos de uso (i)

Un diagrama de casos de uso es un grafo de actores, un conjunto de casos de uso encerrados por los límites de un sistema (un rectángulo), asociaciones entre los actores y los casos de uso y relaciones de generalización entre los actores

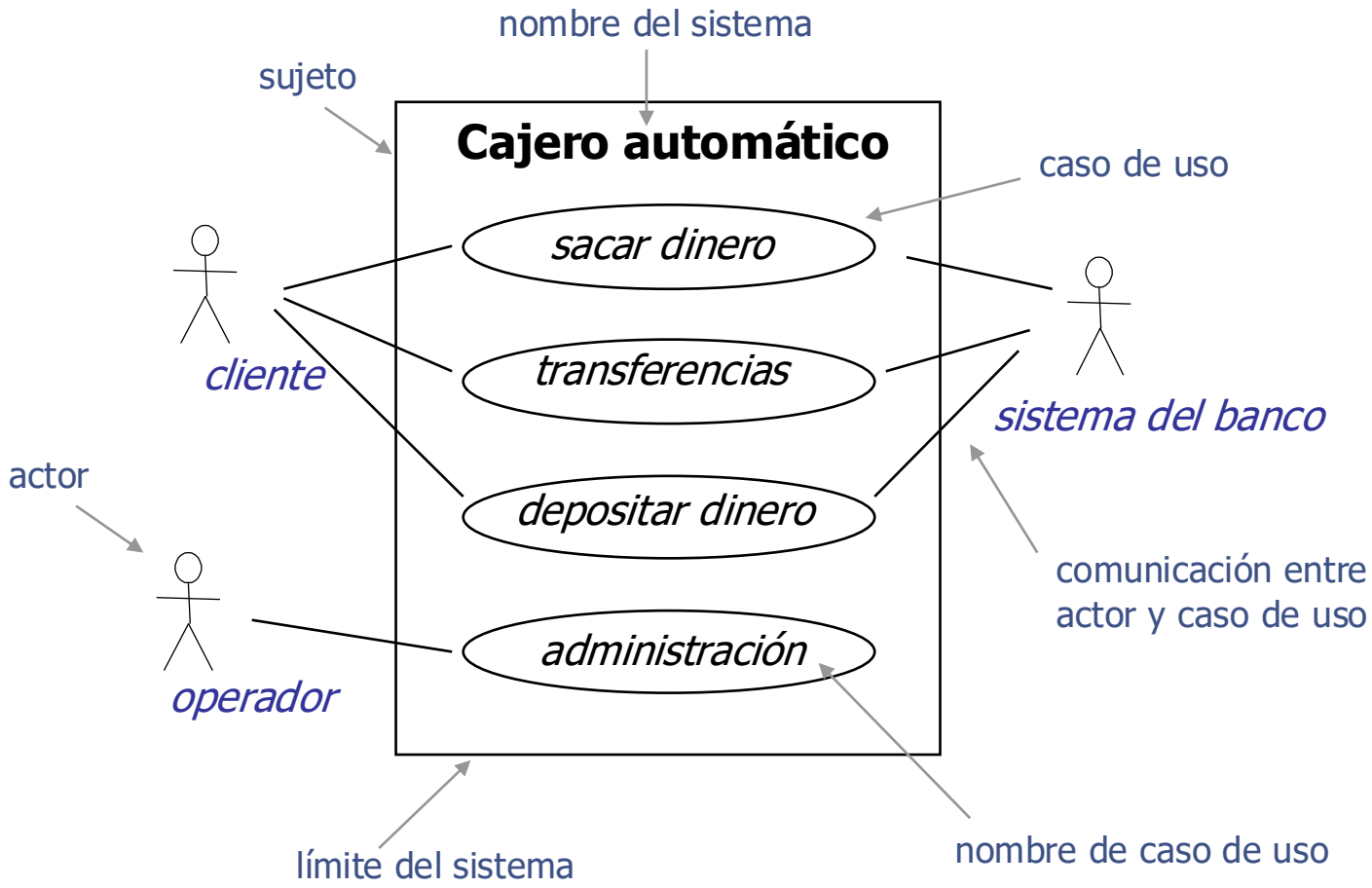
[Rumbaugh et al., 1999]

Actores



Casos de uso

Elementos de un diagrama de casos de uso (ii)



Ejemplo de diagrama de casos de uso

Actores (i)

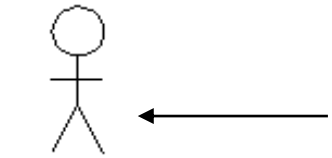
- Un actor es una idealización de una persona, un proceso o una entidad externa que interacciona con un sistema, subsistema o clase
  - Se incluye el propio sistema que se está estudiando cuando solicita los servicios de otros sistemas
- Un actor abstrae y caracteriza a un usuario externo o a un conjunto de usuarios externos relacionados que interactúan con el sistema o clasificador
- Cada actor define un conjunto de roles que los usuarios de un sistema asumen cuando interactúan con el mismo
  - El conjunto completo de actores describe todas las diferentes formas de comunicación entre los usuarios externos y el sistema
- La misma persona física puede interpretar varios papeles como actores distintos
- El nombre del actor describe el papel desempeñado

Actores (ii)

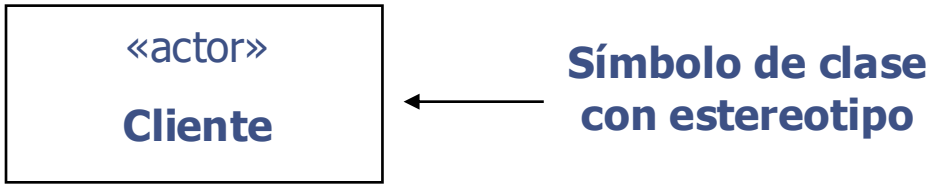
- Tipos de actores [Larman, 2002]
  - Principales
    - Tiene objetivos de usuario que se satisfacen mediante el uso de los servicios del sistema
    - Se identifican para encontrar los objetivos de usuario, los cuales dirigen los casos de uso
  - De apoyo
    - Proporcionan un servicio al sistema
    - Normalmente se trata de un sistema informático, pero podría ser una organización o una persona
    - Se identifican para clarificar las interfaces externas y los protocolos
  - Pasivos
    - Está interesado en el comportamiento del caso de uso, pero no es principal ni de apoyo
    - Se identifican para asegurar que todos los intereses necesarios se han identificado y satisfecho
    - Los intereses de los actores pasivos algunas veces son sutiles o es fácil no tenerlos en cuenta, a menos que estos actores sean identificados explícitamente

Actores (iii)

- Los actores se representan en UML con el icono estándar de los casos de uso que es el "*stick man*" o "monigote" con el nombre del actor al pie de la figura

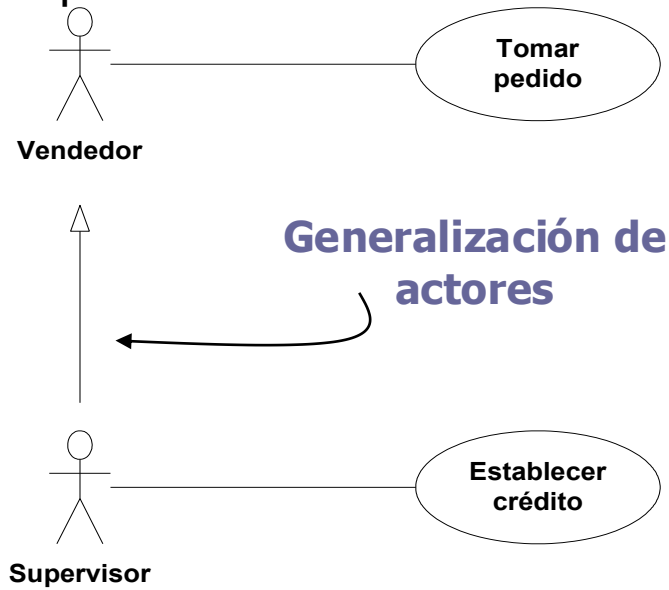


- Un actor también se puede representar mediante un símbolo de clase con el estereotipo **«actor»**

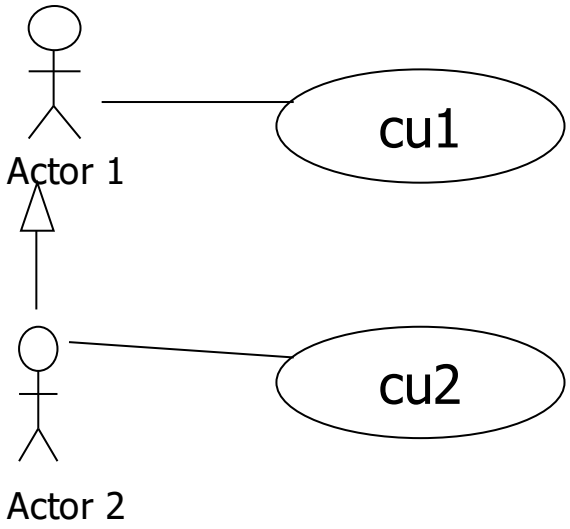


Actores (iv)

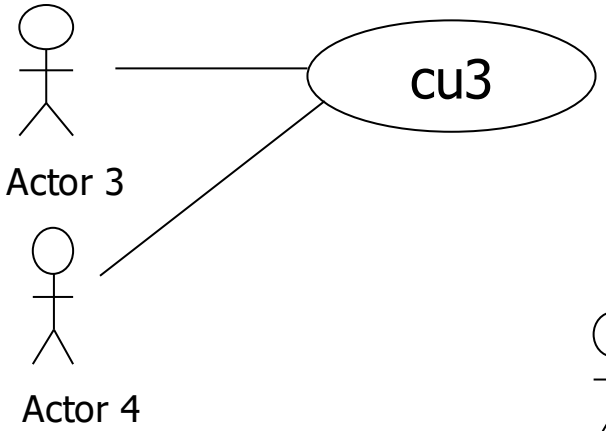
- Puede definirse en una jerarquía de generalización de actores
  - Indica que un actor desempeña el mismo papel que otro actor, pero además puede desempeñar roles adicionales
  - Los actores descendientes heredan los roles y las relaciones con los casos de uso del actor antecesor
  - Principio de capacidad de sustitución
    - Un actor **A** que hereda de otro actor **B** es capaz de comunicarse con los mismos casos de uso que **B**
  - Puede existir generalización múltiple



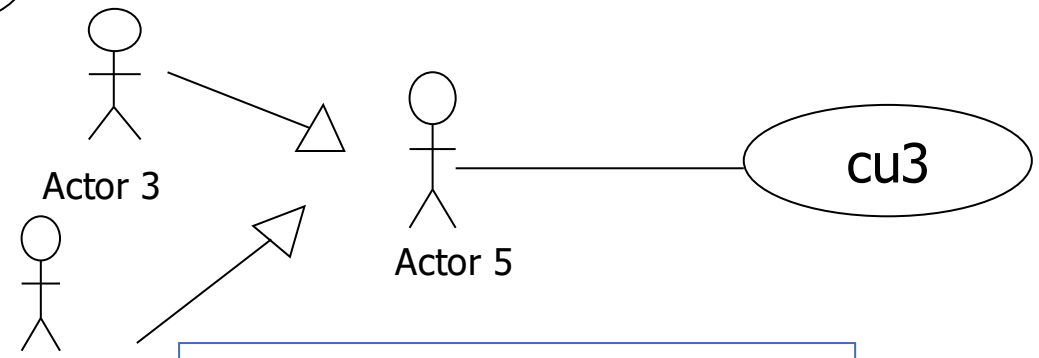
Actores (v)



El actor 1 o el actor 2 se comunica con el caso de uso cu1, el actor 2, además se comunica con el caso de uso cu2



Los actores 3 y 4 se comunican con el caso de uso cu3



El actor 3 o el actor 4 se pueden comunicar con el caso de uso cu3

Casos de uso (i)

Un caso de uso es una forma, patrón o ejemplo concreto de utilización, un escenario que comienza con algún usuario del sistema que inicia alguna transacción o secuencia de eventos interrelacionados

[Jacobson et al., 1992]

Un caso de uso es una unidad coherente de funcionalidad externamente visible proporcionada por un clasificador (denominado sistema) y expresada mediante secuencias de mensajes intercambiados por el sistema y uno o más actores de la unidad del sistema

[Rumbaugh et al., 2005]

Casos de uso (ii)

- En general, las diferentes definiciones pueden catalogarse en cuatro dimensiones [Cockburn, 1997a; Cockburn, 1997b]
  - Propósito
    - Hace referencia a la utilidad que se le da al caso de uso
    - Se distingue entre
      - Casos de uso de negocio, que describen las operaciones de un negocio
      - Casos de uso de sistema, escritos por un equipo de desarrollo para documentar el *software* que están desarrollando
  - Contenido
    - Pueden ser contradictorios o consistentes
    - Los contenidos consistentes pueden expresarse mediante un texto consistente o con contenidos formales
  - Pluralidad
    - Indica si el caso de uso no es más que otro nombre para expresar un escenario o si un caso de uso contiene uno o más escenarios
  - Estructura
    - Indica si el caso de uso tiene una estructura, ya sea formal o informal, o si por el contrario es una colección no estructurada de secuencias de interacción

Casos de uso (iii)

Dimensión	Valor
Propósito	Requisitos funcionales
Contenidos	Texto consistente
Pluralidad	Múltiples escenarios por caso de uso
Estructura	Semiformal

Enfoque de los casos de uso en UML

Casos de uso (iv)

- El propósito de un caso de uso es definir un cierto comportamiento de un clasificador sin revelar la estructura interna del clasificador
- Cada caso de uso especifica un servicio que proporciona el clasificador a sus usuarios
  - Describe una secuencia completa en términos de interacción entre los usuarios y el clasificador, así como las respuestas ofrecidas por el clasificador
- Los casos de uso incluyen el comportamiento normal que se tiene como respuesta a una solicitud del usuario
  - También incluyen las posibles variantes de la secuencia normal
    - Secuencias alternativas, comportamiento frente a excepciones y manejo de errores
- Un caso de uso es un descriptor que describe un comportamiento potencial
  - La ejecución de un caso de uso es una instancia de un caso de uso

Casos de uso (v)

- Una instanciación de un caso de uso se denomina escenario [Regnal et al., 1996; Booch et al., 1999]
- Un escenario es una secuencia específica de acciones que ilustra un comportamiento
  - Un camino particular a través de la descripción abstracta y general proporcionada por el caso de uso
- Los escenarios atraviesan el caso de uso, siguiendo el camino nominal, así como todo camino alternativo y excepcional [Muller, 1997]
- Hay un efecto de expansión de los casos de uso a los escenarios
  - Cada caso de uso puede expandirse en varias decenas de escenarios, ya que para cada caso de uso puede haber escenarios principales y secundarios
- Cada instancia de un caso de uso se inicia por un mensaje que proviene de una instancia de un actor
  - Como respuesta al mensaje, la instancia de caso de uso ejecuta una secuencia de acciones especificadas por el caso de uso

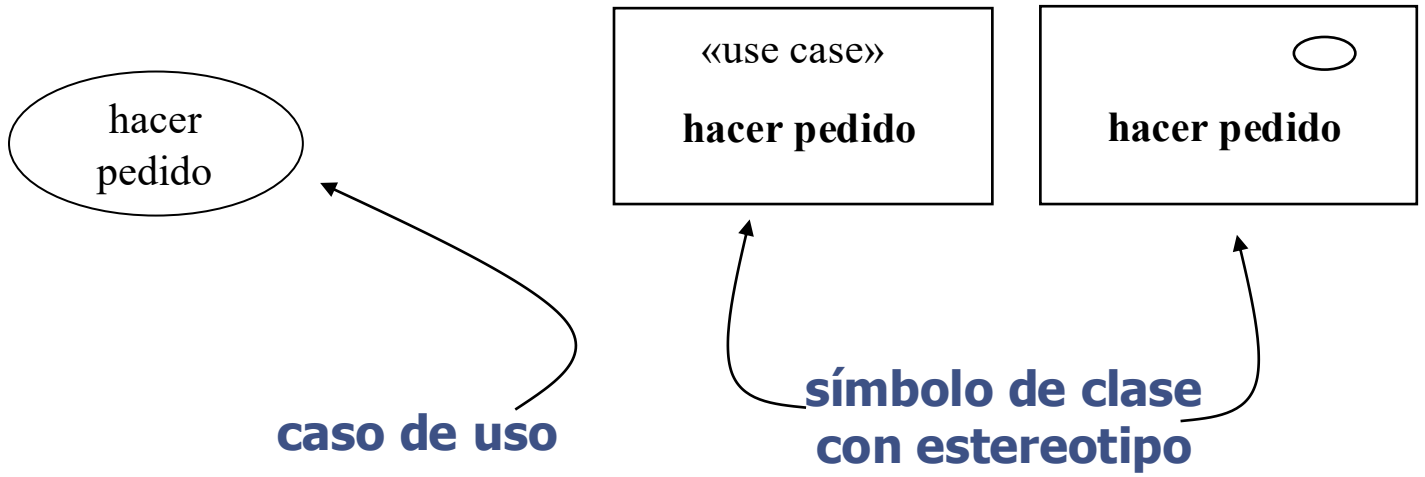
Niveles de descripción

Casos de uso (vi)

- División del trabajo
  - Conjunto de casos de uso que describe los procesos de la actividad relevante del sistema desarrollada por los usuarios. Se centra en la definición de los límites del sistema
- Funciones del sistema específicas de la aplicación
  - Conjunto de casos de uso que describe las funciones que el sistema proporciona y relacionadas con el dominio de la aplicación
- Funciones del sistema específicas de la actividad
  - Conjunto de casos de uso que describe las funciones de soporte del sistema no relacionadas directamente con el dominio de la aplicación. Incluyen funciones de gestión de ficheros, actualización...
- Diálogos
  - Conjunto de casos de uso que describe las interacciones entre los usuarios y la interfaz de usuario del sistema. Se centran en los flujos de control y elementos de presentación

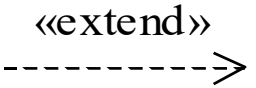
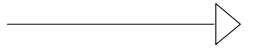
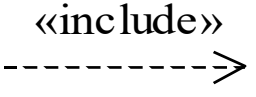
Casos de uso (vii)

- Los casos de uso se representan en UML por una elipse conteniendo el nombre, aunque también puede dibujarse como un rectángulo clasificador estereotipado con la palabra reservada **«use case»** o con el símbolo de una elipse



Relaciones entre los casos de uso

- Un caso de uso puede participar en varias relaciones, además de poderse asociar con actores

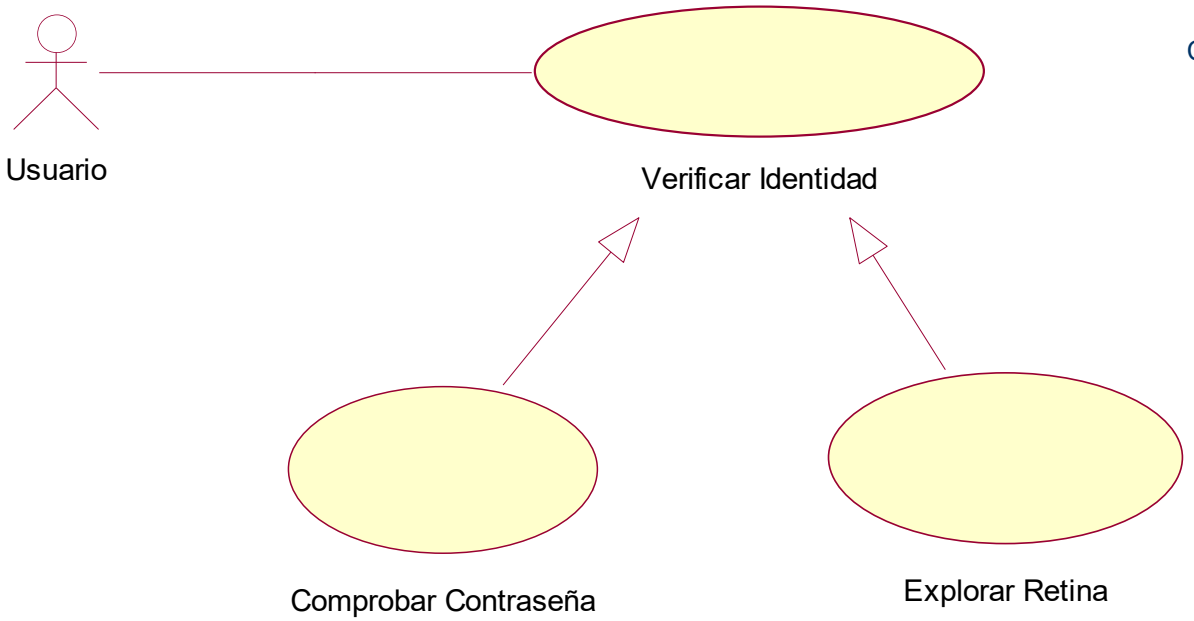
Relación	Función	Notación
Asociación	Línea de comunicación entre un actor y un caso de uso en el que participa	
Extensión	Inserción de comportamiento adicional en un caso de uso base que no tiene conocimiento sobre él	
Generalización de casos de uso	Una relación entre un caso de uso general y un caso de uso más específico, que hereda y añade propiedades al caso de uso base	
Inclusión	Inserción de comportamiento adicional en un caso de uso base, que describe explícitamente la inserción	

Tipos de relaciones en los diagramas de casos de uso

Generalización de casos de uso (i)

- Una relación de generalización relaciona un caso de uso especializado con otro caso de uso más general
- El hijo hereda los atributos, operaciones y secuencias de comportamiento del padre, pudiendo agregar atributos y operaciones propios
- El caso de uso hijo añade comportamiento al caso de uso padre insertando secuencias de acción adicionales en la secuencia del padre en puntos arbitrarios
- La capacidad de sustitución para los casos de uso significa que la secuencia de comportamiento de un caso de uso hijo tiene que incluir la secuencia de comportamiento de su padre
- La notación empleada es la del símbolo normal de generalización

Generalización de casos de uso (ii)



Comportamiento de caso de uso para el padre, **Verificar Identidad**

El padre es abstracto, no hay secuencia de comportamiento

Un descendiente concreto tiene que proporcionar el comportamiento

Comportamiento de caso de uso para el hijo, **Comprobar Contraseña**

- Obtener la contraseña en BD maestra
- Pedir contraseña al usuario
- El usuario proporciona la contraseña
- Comparar contraseña con entrada del usuario

Comportamiento de caso de uso para el hijo, **Explorar Retina**

- Obtener signatura de la retina en BD maestra
- Explorar la retina del usuario y obtener su signatura
- Comparar la signatura maestra con la signatura explorada

Extensión de casos de uso (i)

- La relación de extensión es una especie de dependencia
  - El caso de uso cliente añade un comportamiento incremental al caso de uso base mediante la inserción de secuencias de acción adicionales a la secuencia base
- El caso de uso cliente contiene uno o más segmentos separados de secuencia de comportamiento
- La relación de extensión contiene una lista de nombres de puntos de extensión del caso de uso base
- Un punto de extensión representa una localización dentro del caso de uso base donde se puede insertar la extensión
- Una relación de extensión también puede tener asociada una condición

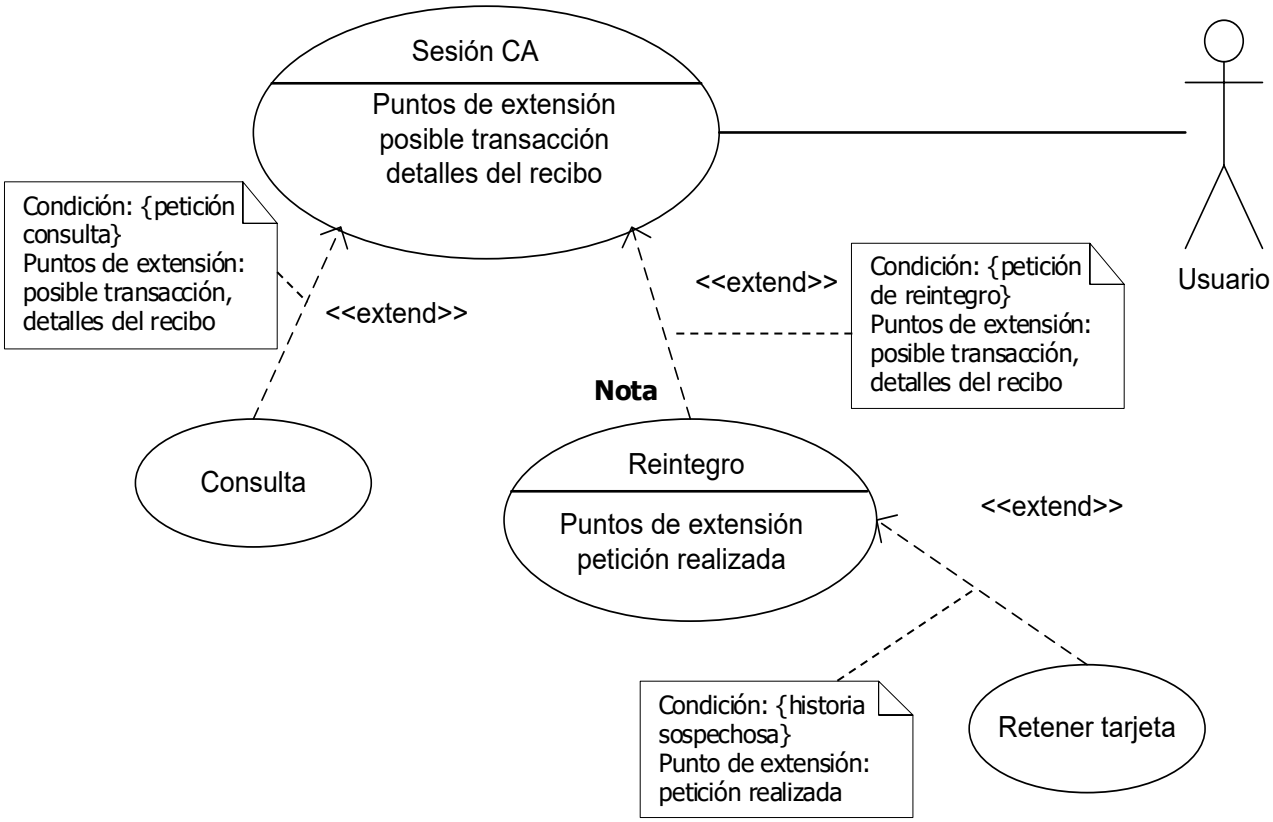
Extensión de casos de uso (ii)

- El caso de uso extensor no es necesariamente instanciable por separado
  - Consiste en uno o más segmentos que describen las secuencias adicionales de comportamiento
  - Cada segmento en un caso de uso extensor se puede insertar en una localización separada en el caso de uso base
- Cuando la ejecución de una instancia de un caso de uso alcanza una localización en el caso de uso base referenciada por el punto de extensión y se cumple cualquier condición en la extensión, entonces la ejecución de la instancia se puede transferir a la secuencia de comportamiento del segmento correspondiente del caso de uso extensor

Extensión de casos de uso (iii)

- Se pueden aplicar múltiples relaciones de extensión al mismo caso de uso base
- Un caso de uso extensor puede extender más de un caso de uso base
- Las extensiones pueden ampliar otras extensiones de una manera jerarquizada
  - Un caso de uso extensor puede ser él mismo la base en una relación de extensión, inclusión o generalización
- La notación de la relación de extensión es una flecha discontinua desde el caso de uso extensor al símbolo del caso de uso base con una punta de flecha apuntando a la base, a la que se añade la palabra clave **«extend»**
  - ~~Puede aparecer una lista de los nombres de los puntos de extensión entre paréntesis después de la palabra clave~~

Extensión de casos de uso (iv)



Caso de uso base para la sesión CA

mostrar el anuncio del día	
incluye (identificar cliente)	inclusión
incluye (validar cuenta)	inclusión
<posible transacción>	punto de extensión
imprimir cabecera del recibo	
<detalles del recibo>	punto de extensión
desconexión	

Caso de uso de extensión para la consulta

segmento	primer segmento
recibir petición de consulta	
mostrar información de la consulta	
segmento	segundo segmento
imprimir la información sobre el reintegro	

Caso de uso de extensión para el reintegro

segmento	primer segmento
recibir petición de retirada de efectivo	
especificar cantidad	
<petición realizada>	punto de extensión
segmento	segundo segmento
desembolsar efectivo	

Caso de uso de extensión para la retención de tarjetas

segmento	segmento único
retener tarjeta	
terminar la sesión	

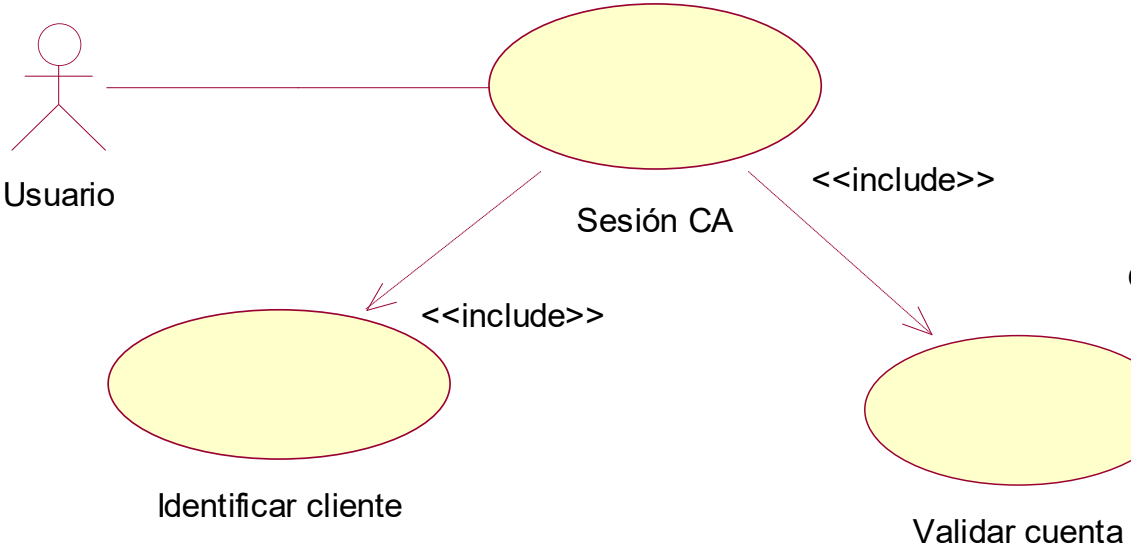
Inclusión de casos de uso (i)

- Esta relación denota la inclusión de la secuencia de comportamiento del caso de uso proveedor en la secuencia de interacción de un caso de uso cliente, bajo el control del caso de uso cliente, en una localización que especifique el cliente en su descripción
- Se trata de una dependencia, no de una generalización
- La instancia de caso de uso está ejecutando el caso de uso cliente
  - Cuando llega al punto de inclusión, comienza a ejecutar el caso de uso proveedor hasta que este finaliza
  - Después sigue ejecutando el caso de uso cliente más allá de la localización de la inclusión
- El caso de uso incluido no es necesariamente un clasificador instanciable independientemente

Inclusión de casos de uso (ii)

- A un mismo caso de uso se le pueden aplicar múltiples relaciones de inclusión
- El mismo caso de uso proveedor puede incluirse en múltiples casos base
- Las inclusiones se pueden anidar
  - Una inclusión puede servir como base para otra inclusión, extensión o generalización posterior
- La inclusión representa un comportamiento encapsulado que, potencialmente, puede ser reutilizado en múltiples casos de uso base
- La relación de inclusión posee la propiedad de localización, que se define como una situación, dentro del cuerpo de la secuencia de comportamiento del caso de uso base, en que se debe insertar la inclusión
  - La inclusión es una sentencia explícita situada dentro de la secuencia de comportamiento del caso de uso base
- La notación de la relación de inclusión es una flecha discontinua desde el caso de uso base hasta el símbolo del caso de uso incluido o proveedor, con una cabeza de flecha abierta en la inclusión, sobre la que se pone la palabra reservada **«include»**

Inclusión de casos de uso (iii)



Caso de uso base para la sesión CA

mostrar el anuncio del día
incluye (identificar cliente) inclusión
incluye (validar cuenta) inclusión
imprimir cabecera del recibo
desconexión

Caso de uso incluido para Identificar cliente

obtener nombre cliente
incluye (verificar identidad)
si fracasa la verificación **entonces** abortar la sesión
obtener los números de cuenta del cliente

Caso de uso incluido para Validar cuenta

establecer conexión con la BD de cuentas
obtener estado de cuenta y límites

Comparación de las relaciones de casos de uso (i)

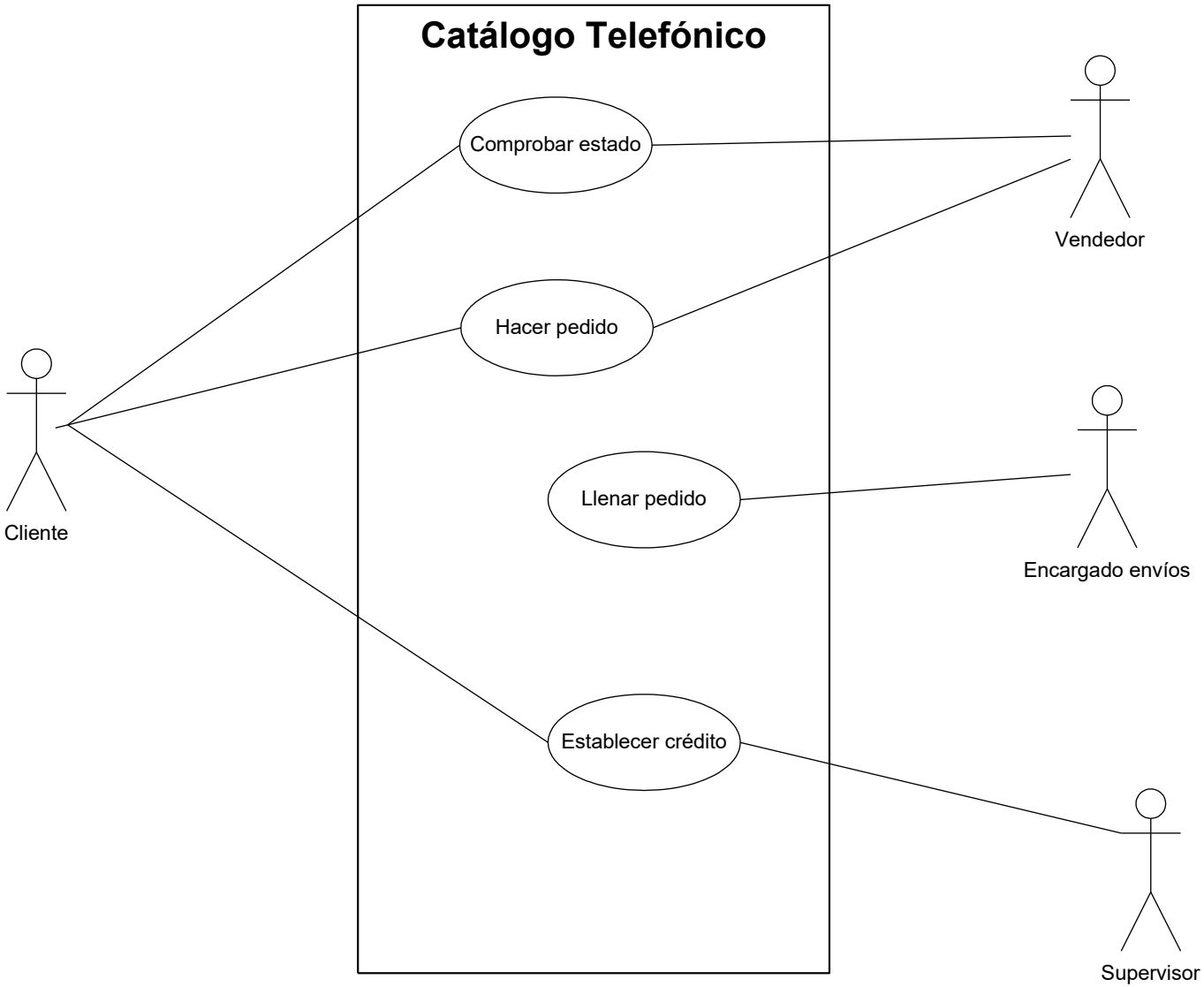
- Las relaciones **«include»** y **«extend»** son constructores similares, por lo que cuándo usar uno u otro puede no estar claro
- La distinción principal entre los dos es la dirección de la relación
  - **«include»**
    - Las condiciones bajo las que el caso de uso destino se inicia son descritas en el caso de uso iniciados, como un evento en el flujo de eventos
  - **«extend»**
    - Las condiciones bajo las que la extensión se inicia se describen en la extensión en la condición de iniciación (condición de entrada al caso de uso)
- Utilizar la relación **«extend»** para especificar comportamiento excepcional, opcional o que ocurra raramente
- Utilizar la relación **«include»** para el comportamiento que sea compartido por dos o más casos de uso

Comparación de las relaciones de casos de uso (ii)

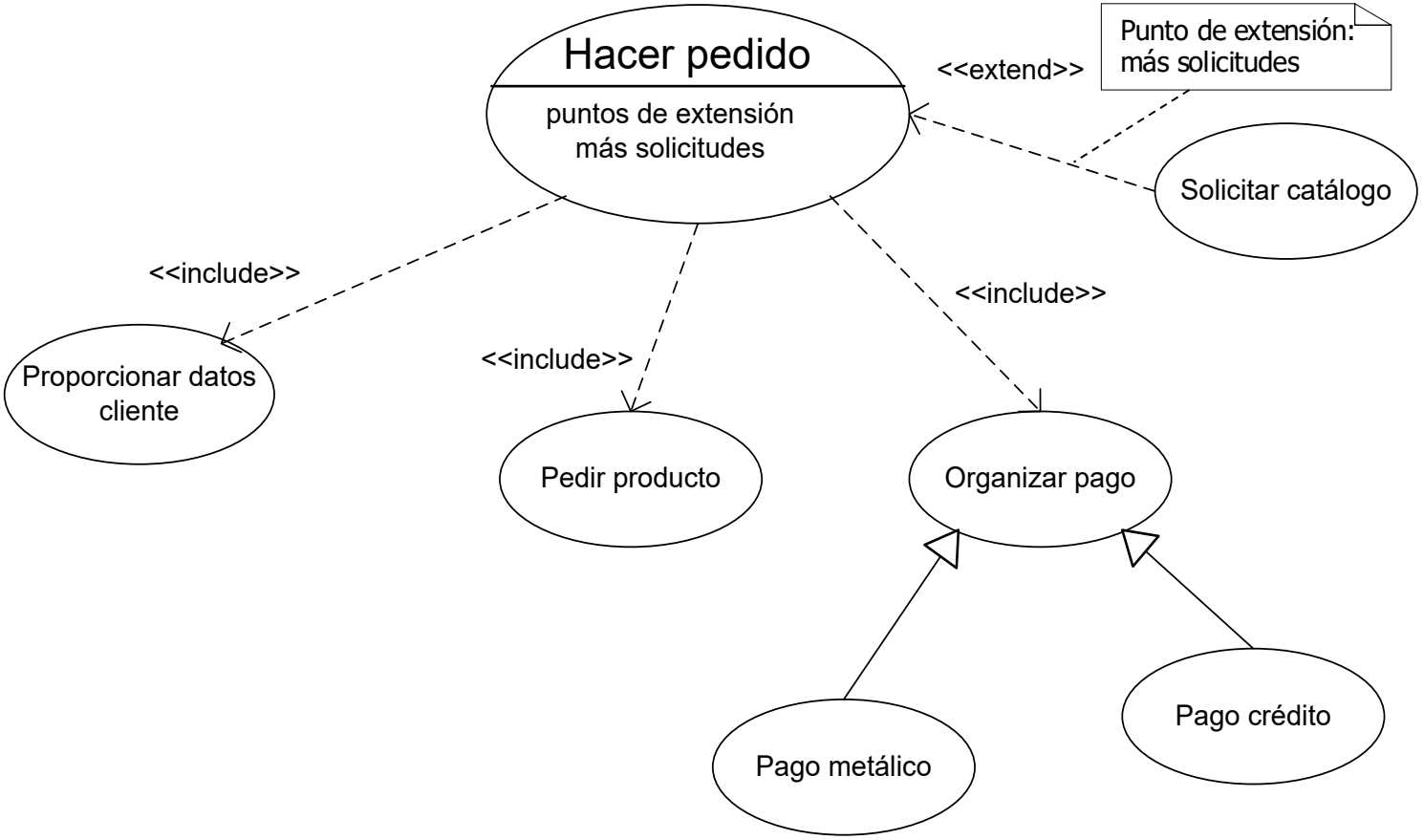
Propiedad	*Extensión*	*Inclusión*	*Generalización*
Comportamiento base Comportamiento añadido	Caso de uso base Caso de uso de extensión	Caso de uso base Caso de uso incluido	Caso de uso padre Caso de uso hijo
Dirección de referencia	El caso de uso extensor referencia al caso de uso base	El caso de uso base referencia al caso de uso incluido	El caso de uso hijo referencia al caso uso padre
¿Base modificada?	La extensión modifica implícitamente el comportamiento de la base. La base debe estar formada sin extensión, pero si la extensión está presente, una instancia de la base puede ejecutar la extensión	La inclusión modifica explícitamente el efecto de la base. La base puede estar bien formada o no sin la inclusión, pero una instanciación de la base ejecuta la inclusión	El efecto de ejecutar el padre no se modifica por el hijo. Para obtener efectos distintos, se debe instanciar el hijo y no el padre
¿Instanciable?	La extensión no es necesariamente instanciable. Puede ser un fragmento	La inclusión no es necesariamente instanciable, puede ser un fragmento	El hijo no es necesariamente instanciable. Puede ser abstracto
¿Puede acceder a los atributos de la base?	Las extensiones pueden acceder y modificar el estado de la base	La inclusión puede acceder al estado de la base. La base debe proveer los atributos apropiados esperados por la inclusión	El hijo puede acceder y modificar el estado de la base (por los mecanismos usuales de la herencia)
¿Puede la base ver al otro?	La base no puede ver las extensiones y debe estar bien formada en su ausencia	La base ve la inclusión y puede depender de sus efectos, pero no puede acceder a sus atributos	El padre no puede ver al hijo, y debe estar bien formado en su ausencia
Repetición	Depende de la condición	Exactamente una repetición	El hijo controla su propia ejecución

[Rumbaugh et al., 1999]

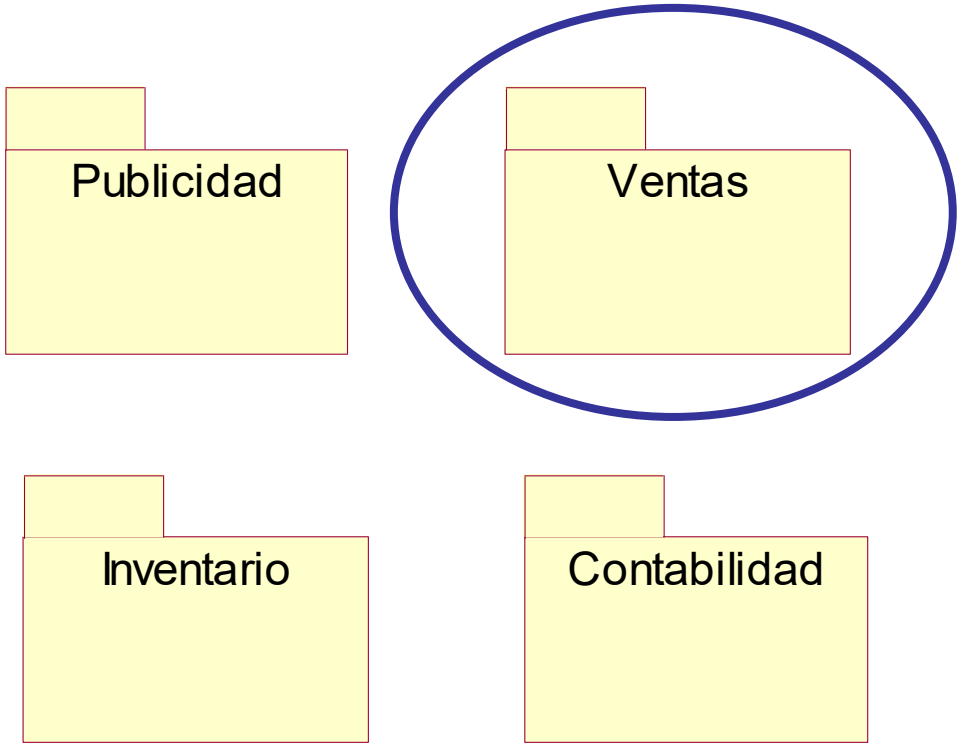
Ejemplo 1 (i)


Catálogo Telefónico
Comprobar estado
Hacer pedido
Llenar pedido
Establecer crédito
Cliente
Vendedor
Encargado envíos
Supervisor

Ejemplo 1 (ii)


Hacer pedido
puntos de extensión
más solicitudes
<<extend>>
Punto de extensión:
más solicitudes
Solicitar catálogo
<<include>>
Proporcionar datos
cliente
<<include>>
Pedir producto
<<include>>
Organizar pago
Pago metálico
Pago crédito

Ejemplo 2 (i)



Paquetes de casos de uso del sistema

Ejemplo 2 (ii)

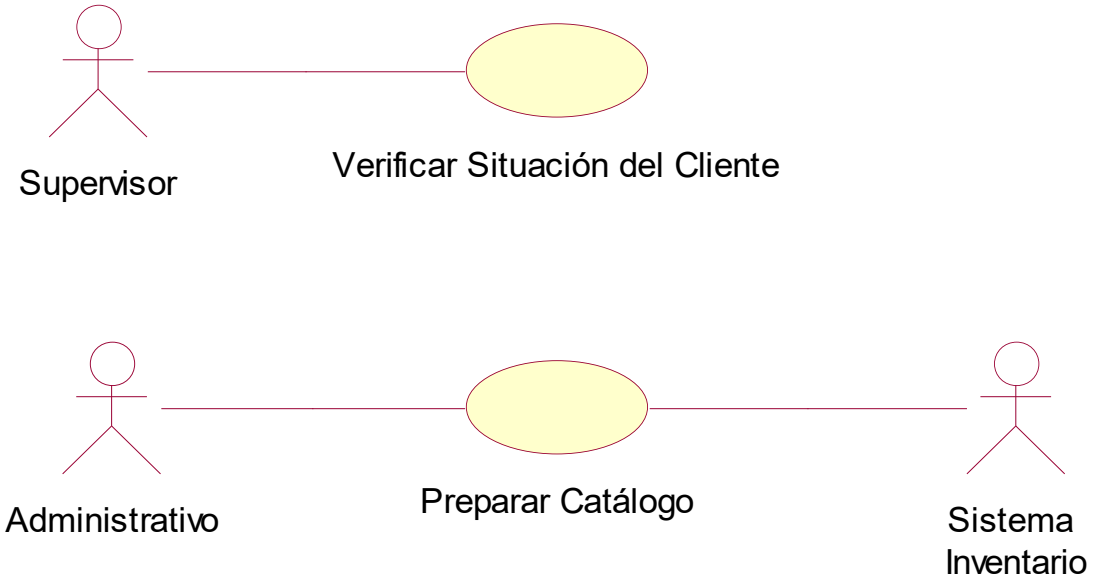


Diagrama Ventas

Ejemplo 2 (iii)

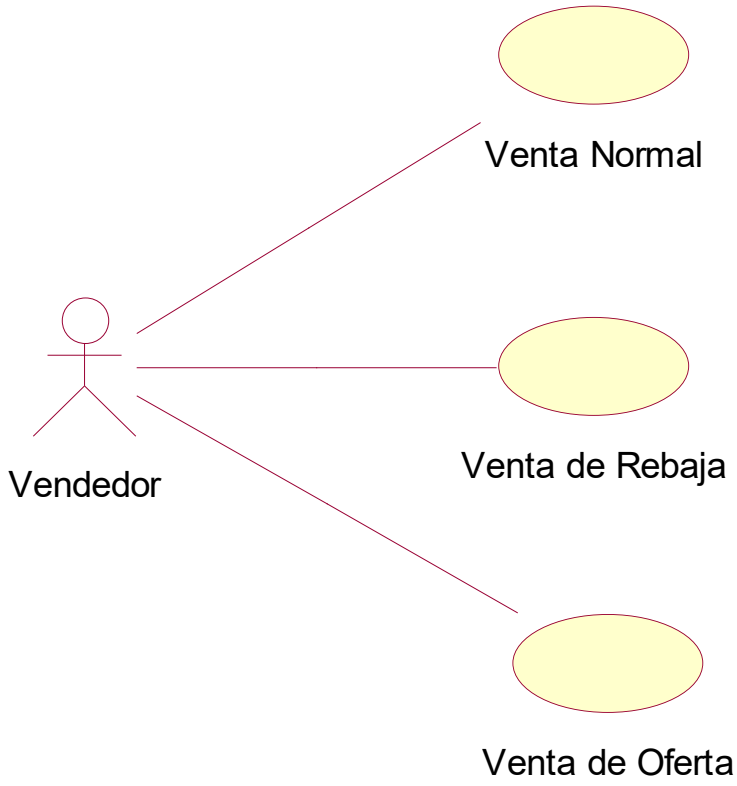
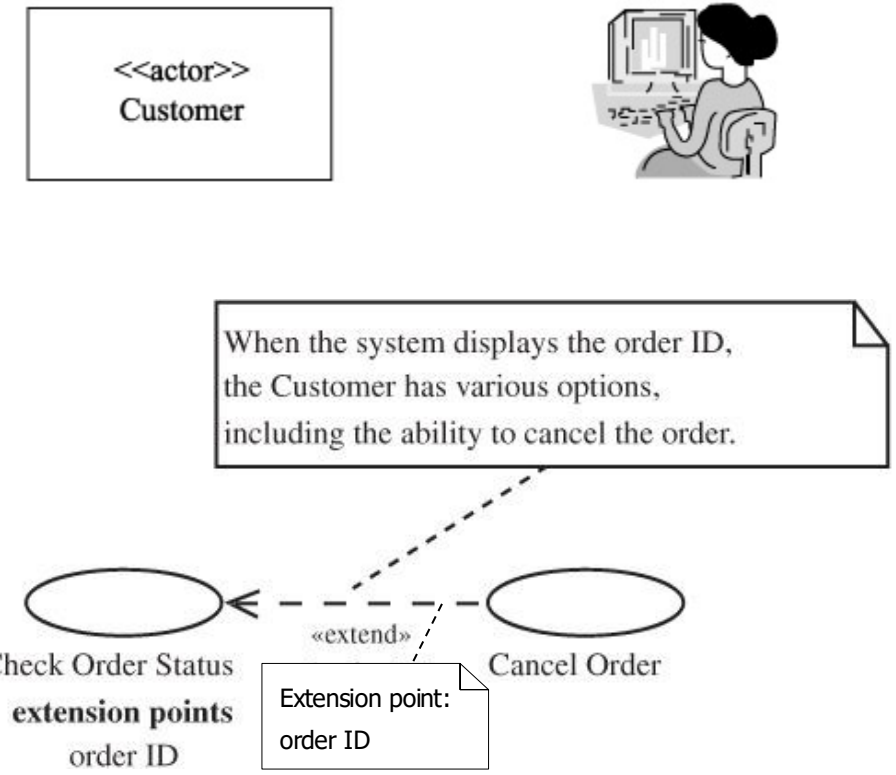
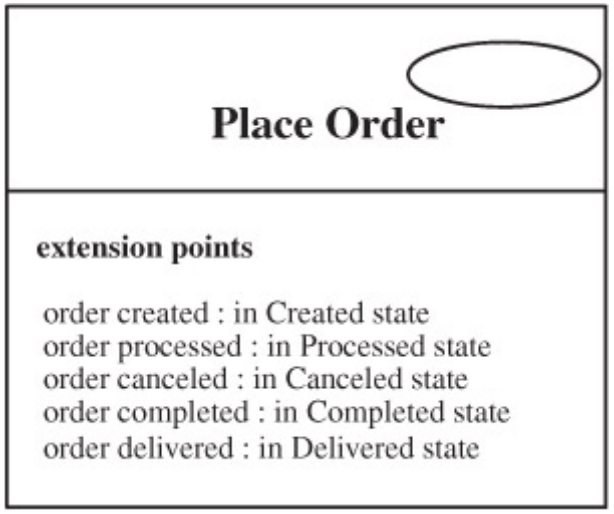


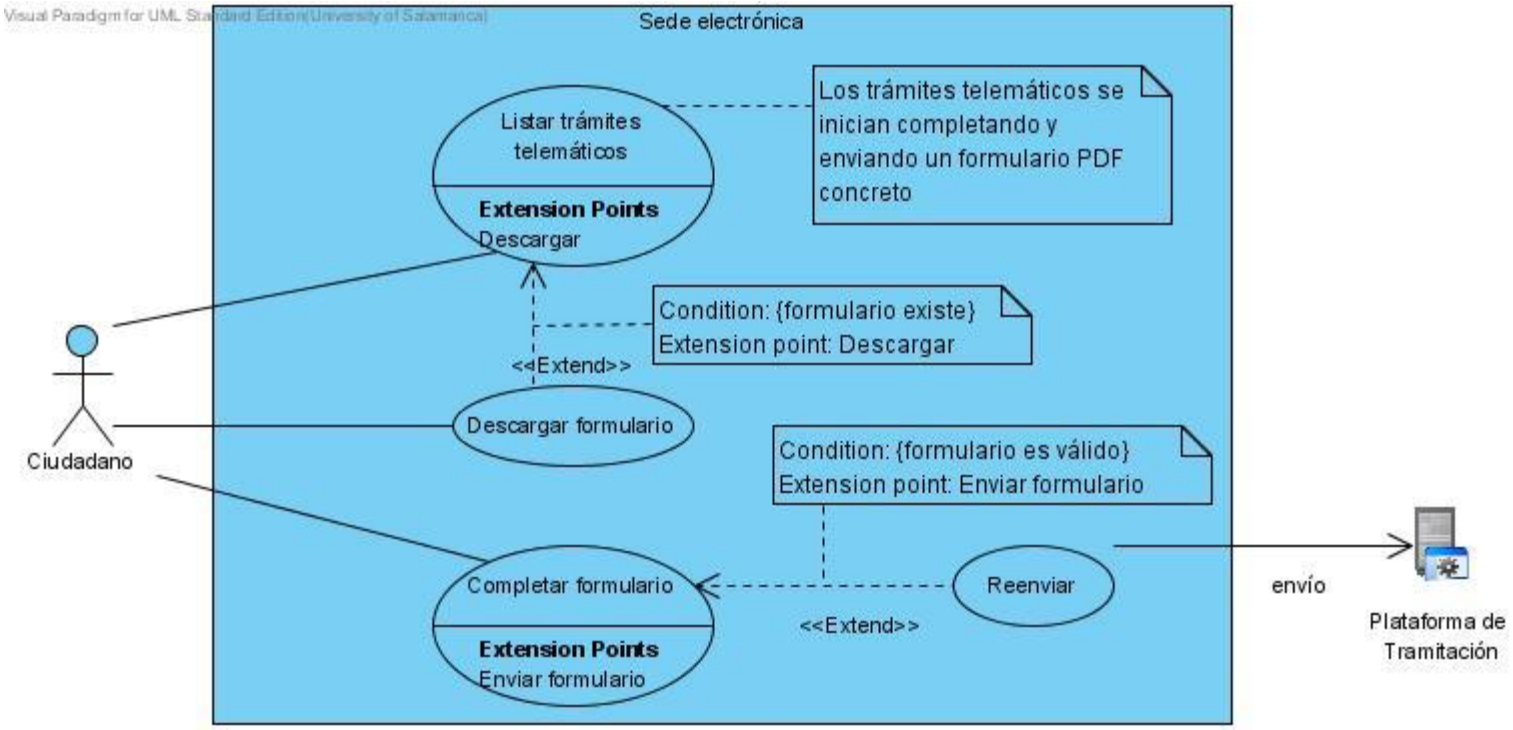
Diagrama Realizar Venta

Cambios importantes en UML 2 (i)



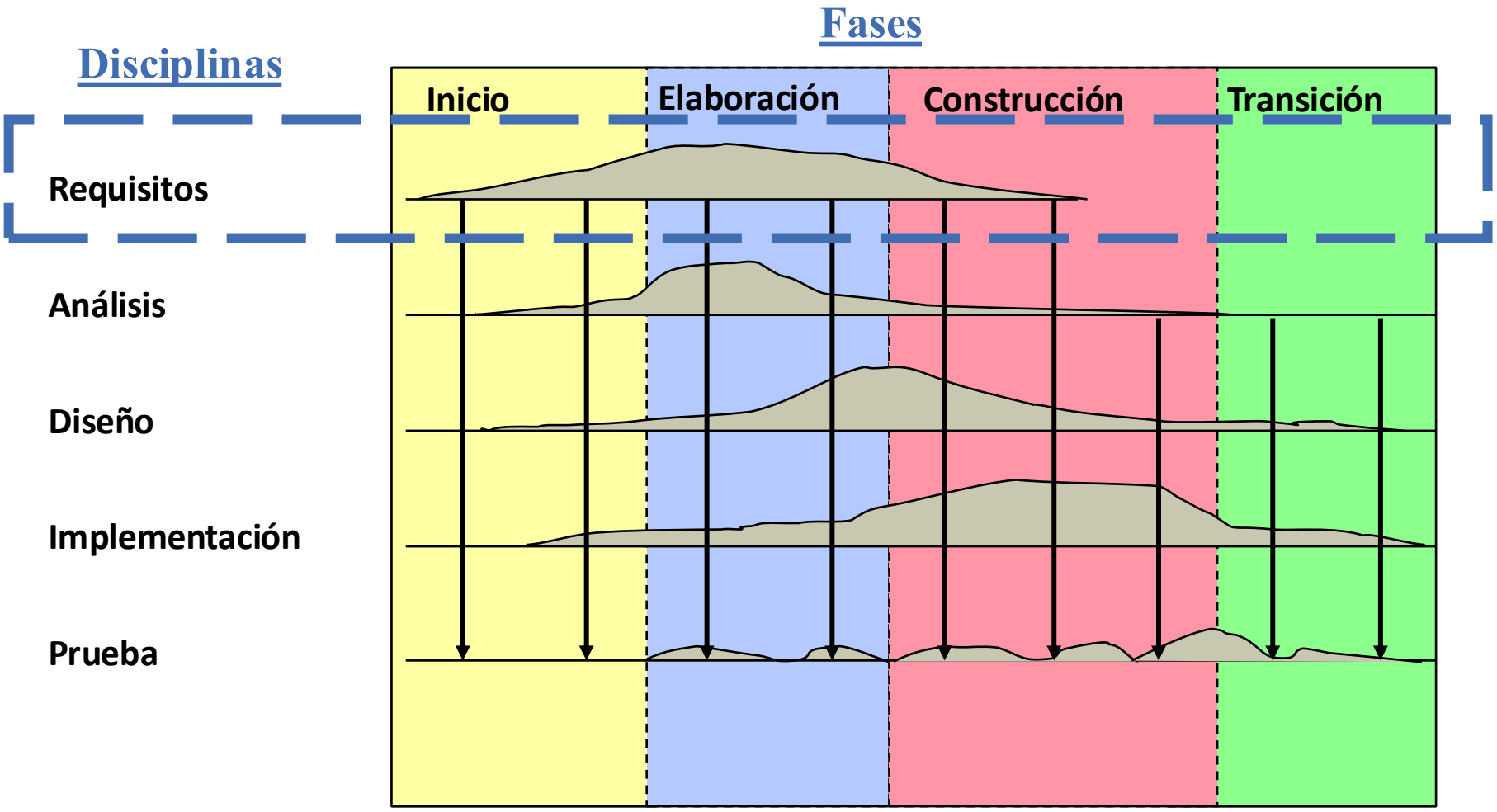


Cambios importantes en UML 2 (ii)



7. Requisitos en el Proceso Unificado

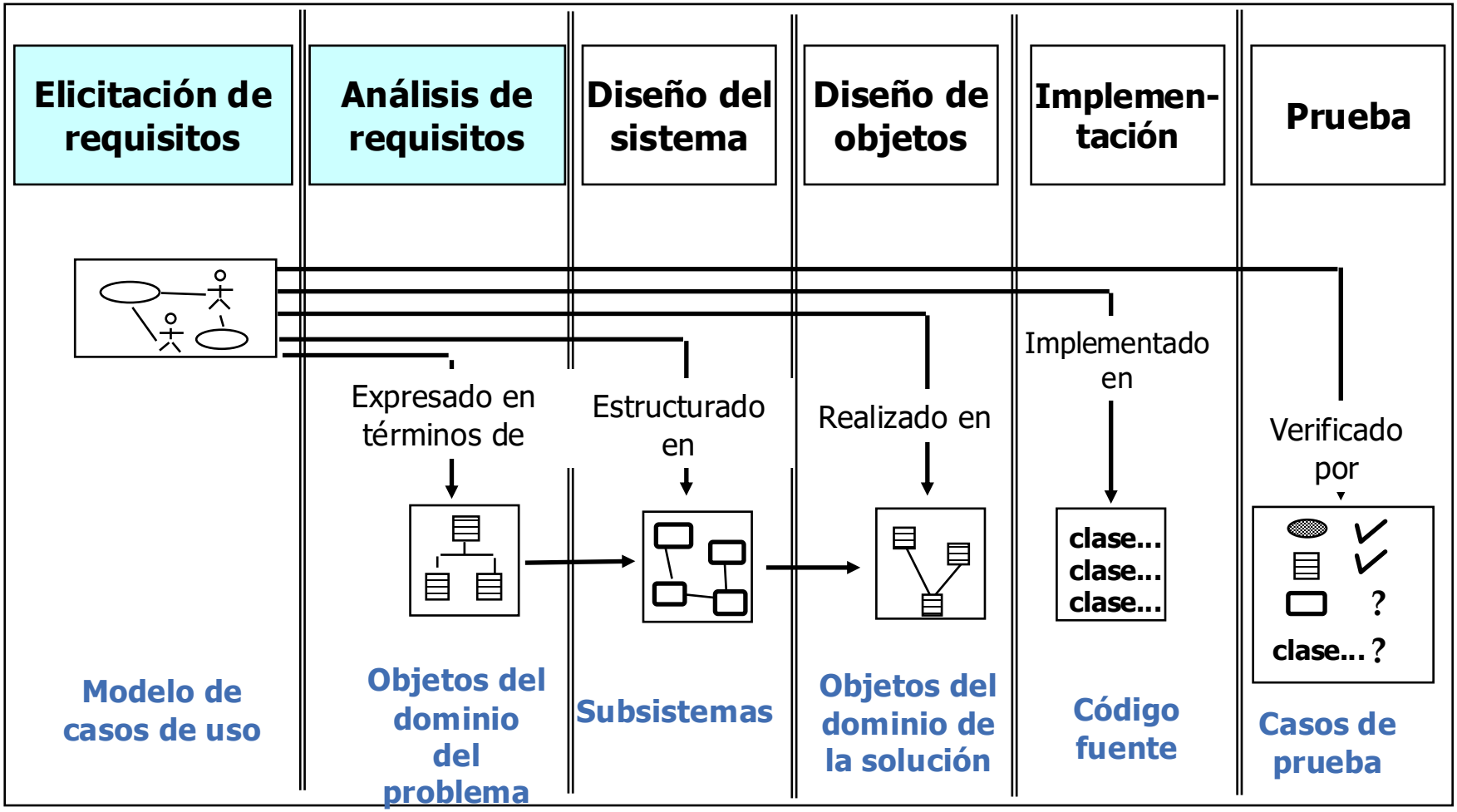
Ciclo de vida del Proceso Unificado



Requisitos en el Proceso Unificado

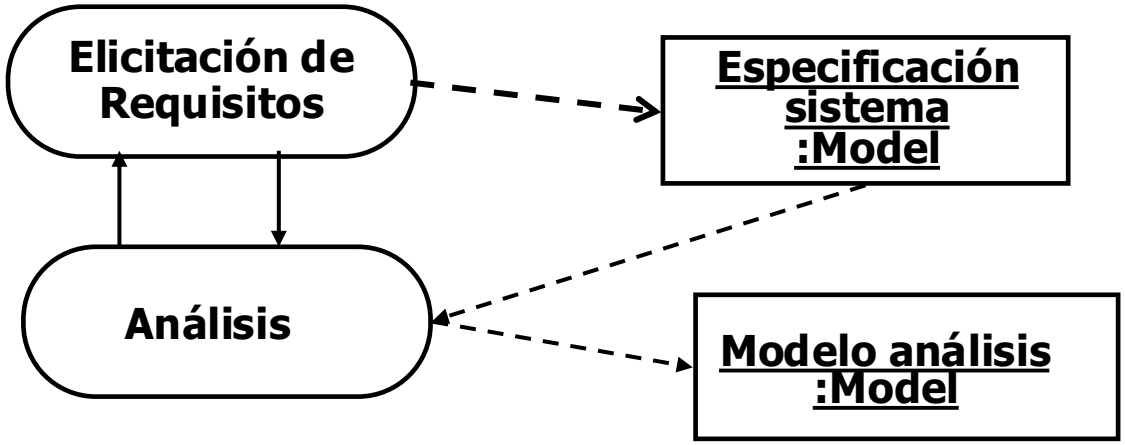
- Construye dos modelos principales
  - **Modelo del dominio**
    - Describe los requisitos de datos estáticos de la aplicación
  - **Modelo de casos de uso**
    - Describe los requisitos de procesamiento dinámico de la aplicación
- Estos modelos se desarrollan de forma incremental durante el desarrollo (principalmente en las fases de inicio y elaboración)
  - **Inicio**: Identifica la mayoría de los elementos del dominio y de los casos de uso para delimitar el sistema y el alcance del proyecto. Se detallan los casos de uso críticos
  - **Elaboración**: Se capturan los restantes requisitos de forma que se pueda estimar el tamaño y el esfuerzo de desarrollo
  - **Construcción**: Se capturan los requisitos residuales
  - **Transición**: No se capturan requisitos salvo que alguno haya cambiado

Proceso dirigido por casos de uso



[Bruegge y Dutoit, 2000]

Captura y recogida de requisitos



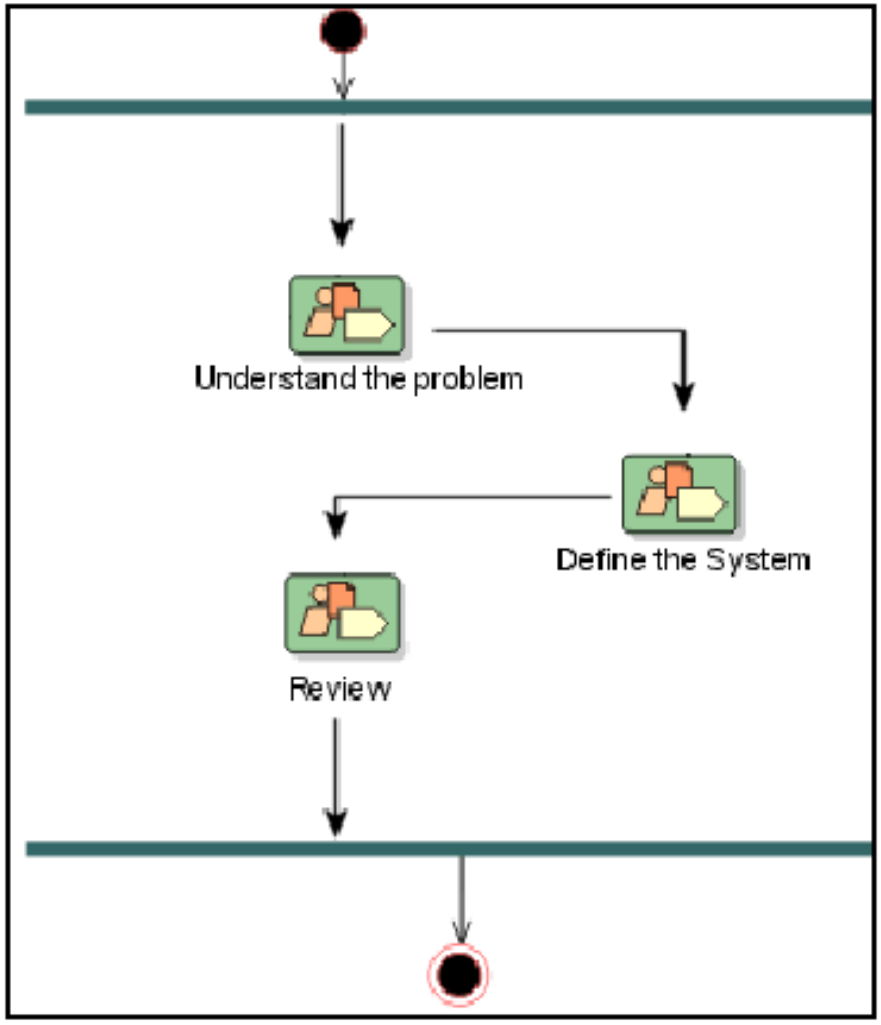
- **Elicitación de requisitos**
  - Especificación del sistema comprensible por el usuario
- **Análisis**
  - Modelo de análisis que el equipo de desarrollo puede interpretar sin ambigüedades
- La especificación del sistema y el modelo de análisis representan la misma información, uno en lenguaje natural y el otro en notación formal o semiformal
- Se centra en la visión que el usuario tiene del sistema

Identificación de las necesidades del usuario

- Recoger información del dominio de la aplicación
  - **Investigación** de la documentación existente
  - **Observación** de cómo se realiza el trabajo diario
  - **Entrevistas** personales y mediante cuestionarios
  - **Prototipado** de las interfaces y las funciones

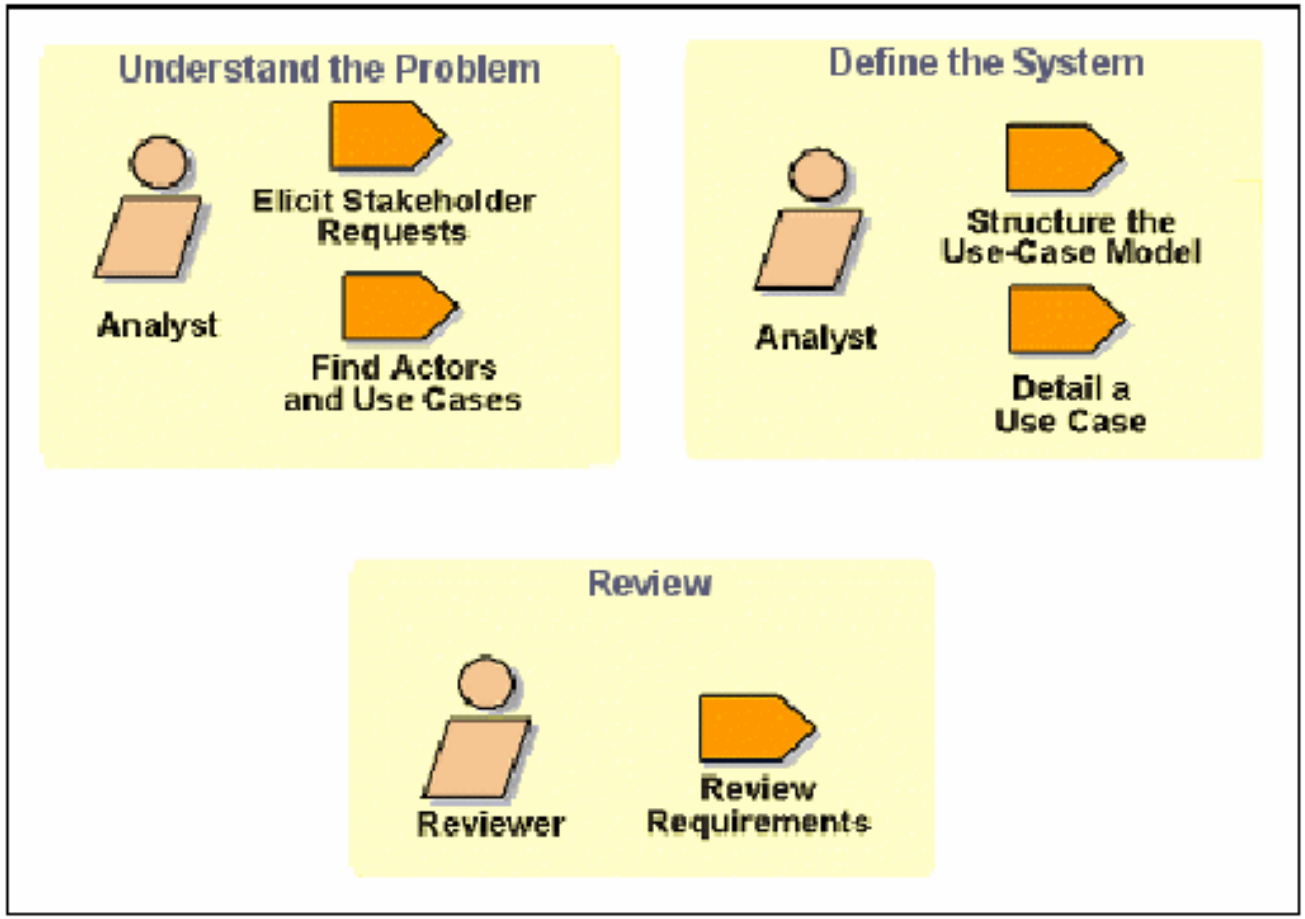
- Distinguir las necesidades de los deseos
  - **Necesidades**
    - Características críticas para el buen funcionamiento del sistema
  - **Deseos**
    - Características deseables pero no esenciales

Flujos de trabajo en la disciplina de requisitos



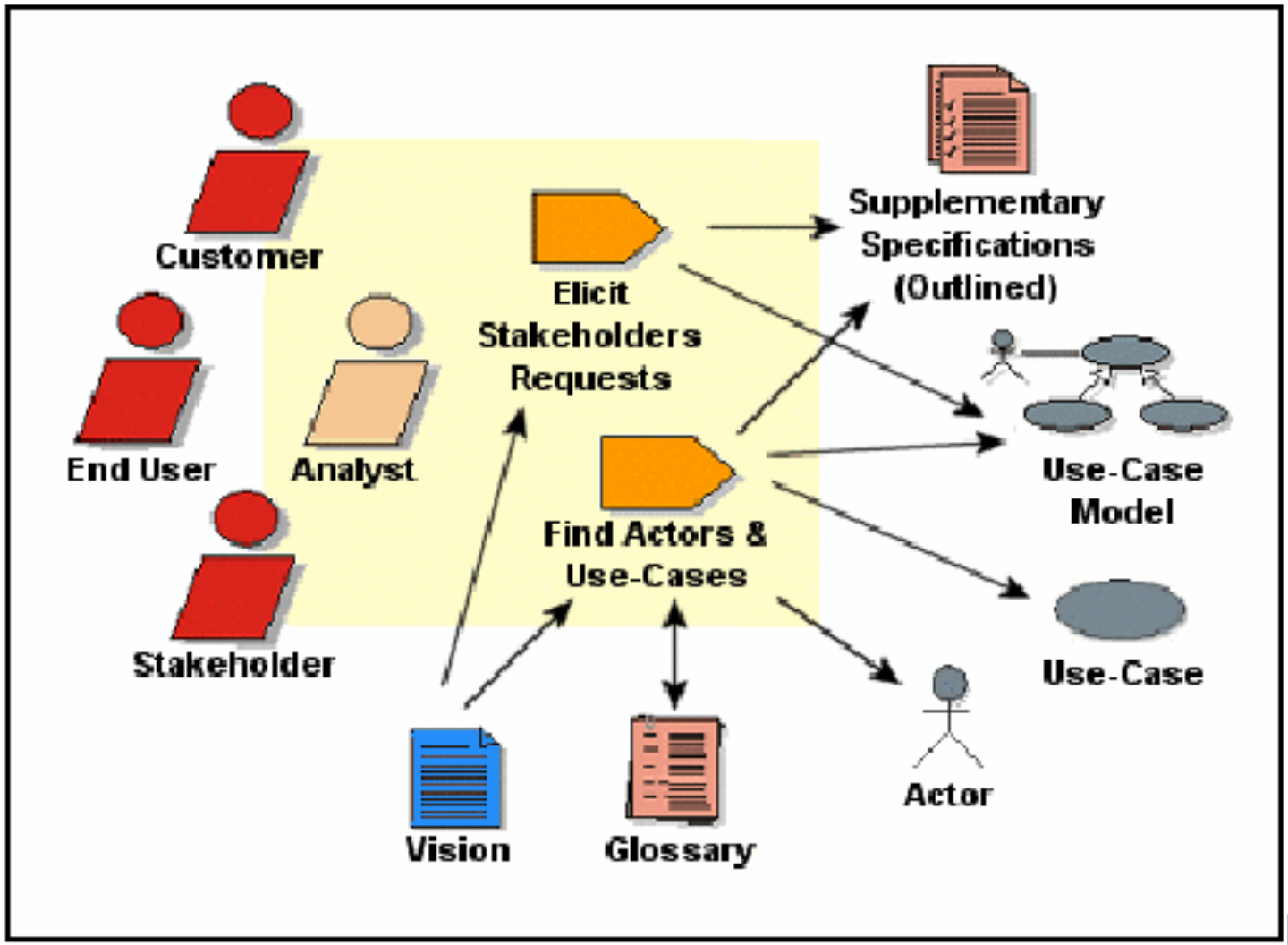
[Jacobson et al., 1999]

Actividades en la disciplina de requisitos



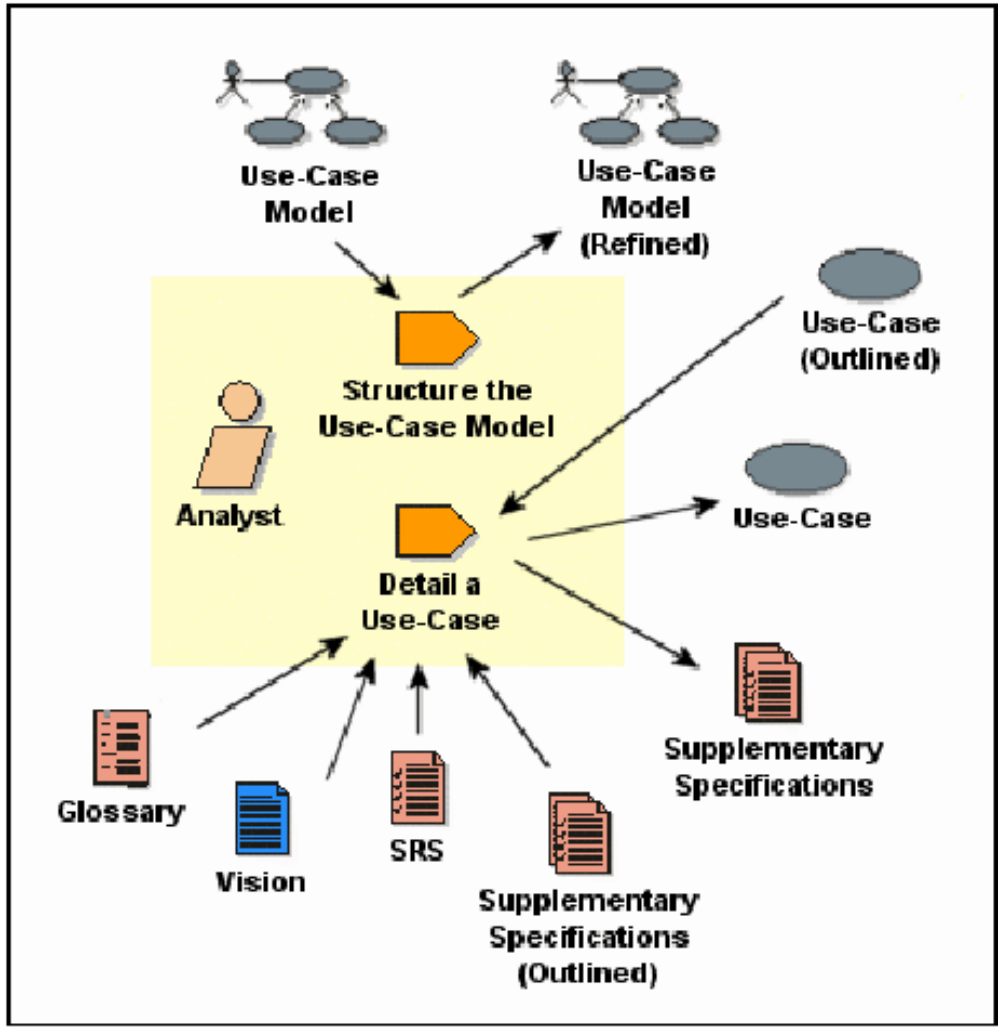
[Jacobson et al., 1999]

Comprensión del problema



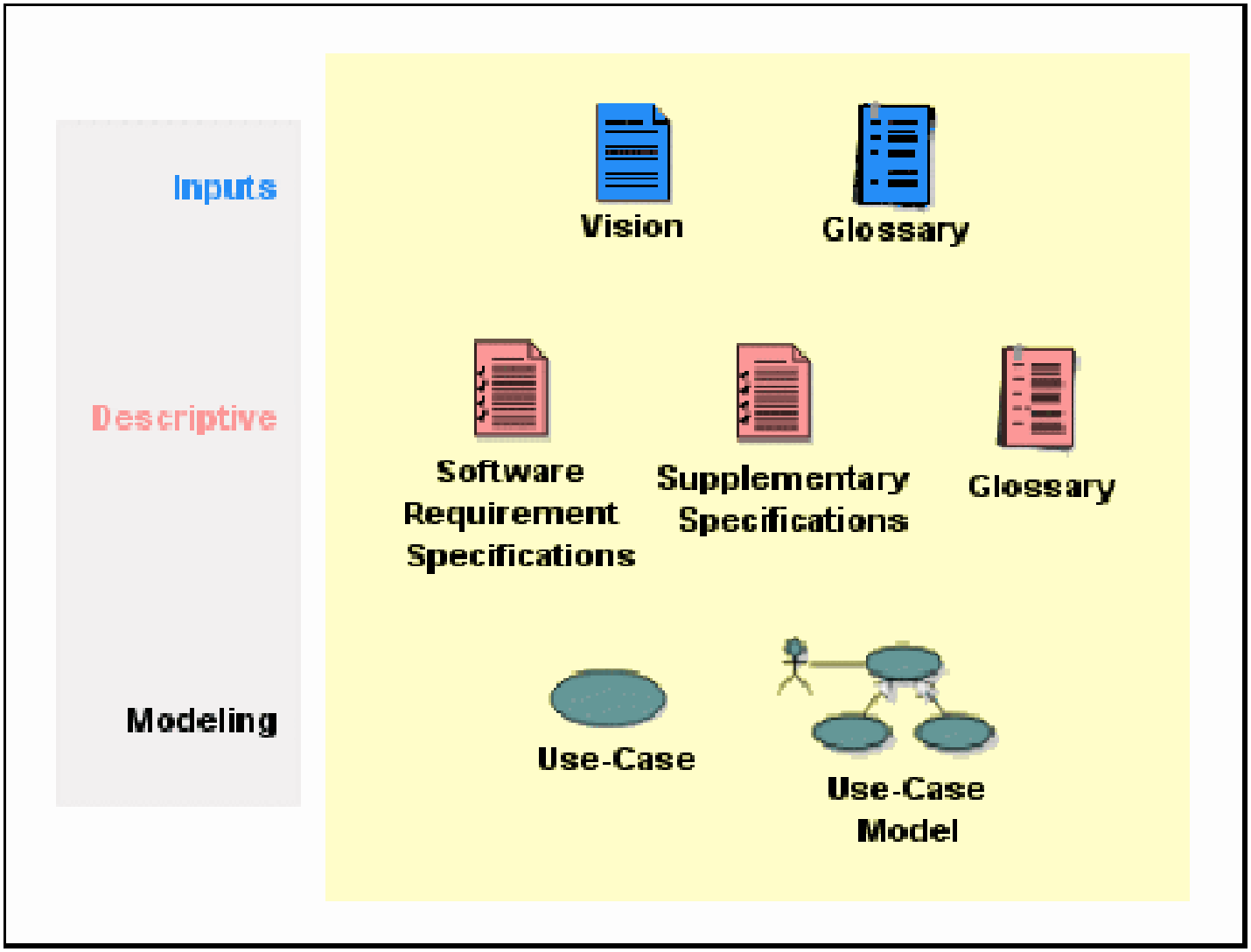
[Jacobson et al., 1999]

Definición del sistema



[Jacobson et al., 1999]

Artefactos en la disciplina de requisitos



[Jacobson et al., 1999]

Modelo de requisitos - Modelo de casos de uso

- El objetivo es **delimitar el sistema** y **definir qué funcionalidad** tiene que afectar al sistema
- El primer modelo a crear del sistema es el modelo de casos de uso
- Son una representación orientada a la funcionalidad del sistema
- Son la base para la identificación de las clases semánticas del sistema en el análisis orientado a objetos
- Describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario
  - Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno

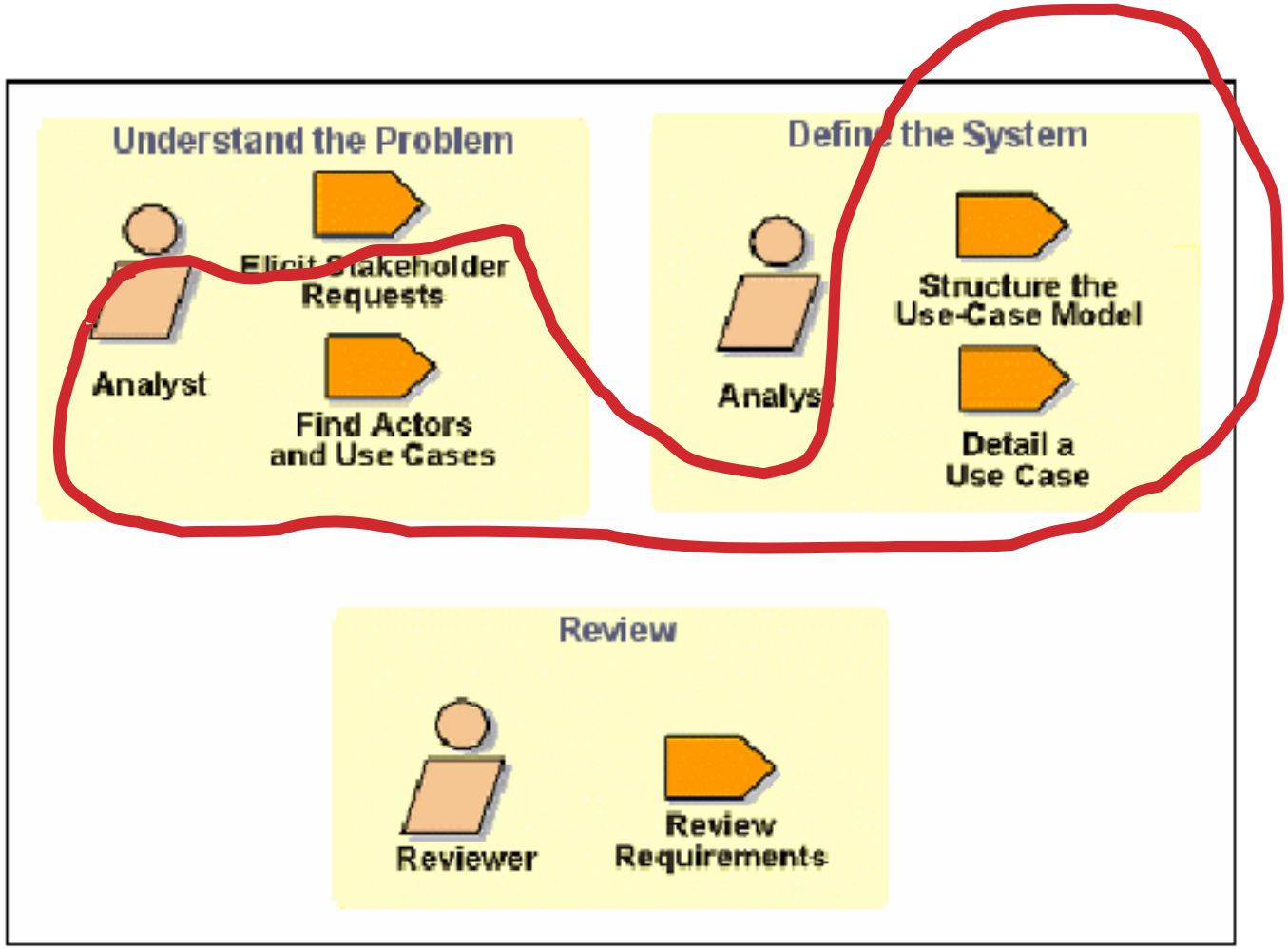
Conceptos a manejar (i)

- El concepto principal a manejar será el de **escenario** que vendrá descrito diagramáticamente por diagramas de interacción y que, en la mayoría de las ocasiones, se denominarán en un abuso del lenguaje como "casos de uso"
- Escenario
  - Iteración estructurada entre entidades en la que se invocan a las responsabilidades de estas
  - Secuencia de pasos que suceden en ciertas condiciones para conseguir el objetivo primario del actor y con un resultado determinado en relación a este objetivo
  - Hechos que describen un sistema existente y su entorno incluyendo el comportamiento de los agentes con la suficiente información de contexto para permitir el descubrimiento y la validación de los requisitos del sistema
  - Son instancias de la experiencia actual con un sistema según se percibe por sus usuarios
  - Una descripción narrativa de lo que la gente hace y experimenta cuando intenta utilizar sistemas de computación y aplicaciones

Conceptos a manejar (ii)

- Paso de escenario
  - Una interacción con el sistema
  - Una responsabilidad a ser realizada por un actor
  - Un caso de uso referenciado
- Casos de uso
  - Colección de posibles escenarios entre el sistema en estudio y actores externos, caracterizados por el objetivo que el actor primario tiene en relación con las responsabilidades del sistema
- Casos de uso externos
  - Tratan al sistema como una "caja negra" y muestran cómo las entidades externas y su entorno interaccionan con él
  - Muestran cómo las entidades externas "utilizan" el sistema para hacer algo
- Casos de uso internos
  - Muestran cómo interaccionan las entidades internas del sistema
  - Puede interpretarse como una muestra de cómo las entidades "utilizan" a otras entidades para conseguir "hacer cosas"

Ingeniería de requisitos en el Proceso Unificado (i)



Actividades

Proceso Iterativo

- Identificar actores
  - Se identifican los diferentes tipos de usuarios a los que el sistema ha de dar soporte
- Identificar escenarios
  - Se desarrolla el conjunto de escenarios para la funcionalidad proporcionada por el sistema
- Identificar casos de uso
  - Obtención de los casos de uso que representan el sistema
- Refinar casos de uso
  - Garantizar que la especificación del sistema esté completa. Se describe el comportamiento del sistema en presencia de errores o condiciones de excepción
- Identificar relaciones entre casos de uso
  - Se consolida el modelo de casos de uso eliminando redundancias
- Identificar requisitos no funcionales
  - Aspectos relacionados con la funcionalidad: restricciones de rendimiento, documentación, recursos, seguridad, calidad

Actividad: Identificar actores (i)

- Representan entidades externas que interaccionan con el sistema
  - Cualquier cosa que está "enlazada" al sistema: persona, sistema *software*, dispositivos *hardware*, almacenes de datos, redes de comunicaciones
- Son abstracciones de rol y no necesariamente se corresponden con personas
- Cada entidad externa puede estar representada por varios actores
  - Si una persona física desempeña diferentes roles respecto al sistema se representa por varios actores
- Sirven para definir los límites del sistema y para buscar todas las perspectivas desde las que los desarrolladores tienen que considerar el sistema

Actividad: Identificar actores (ii)

- Guía
  - ¿Quién usa el sistema?
    - ¿Qué grupo de usuarios están soportados por el sistema para llevar a cabo su trabajo?
    - ¿Qué grupo de usuarios ejecutan las principales funciones del sistema?
  - ¿Qué grupo de usuarios realiza las funciones auxiliares, tales como mantenimiento y administración?
    - ¿Quién inicial el sistema?
    - ¿Quién instala el sistema?
    - ¿Quién apaga el sistema?
  - ¿Interaccionará el sistema con algún sistema *software* o *hardware* externo?
    - ¿Existen otros sistemas que utilicen el sistema?
    - ¿Quién obtiene información del sistema?
    - ¿Quién proporciona información a nuestro sistema?
  - ¿Ocurre algo de forma automática en un tiempo predeterminado?

Actividad: Identificar escenarios (i)

- Descripción informal, concreta y orientada de una característica del sistema desde el punto de vista de un actor
- Los escenarios pueden tener diferentes utilidades a lo largo del ciclo de vida. Existen los siguientes tipos de escenarios
  - ***As-is scenarios:*** Describen la situación actual
  - ***Visionary scenarios:*** Describen un sistema futuro. Pueden utilizarse en la obtención de requisitos y en la representación de diseño. Pueden considerarse como prototipos baratos
  - ***Evaluation scenarios:*** Describen las tareas de usuario contra las que se evaluará el sistema
  - ***Training scenarios:*** Son tutoriales utilizados para introducir a los nuevos usuarios al sistema. Son instrucciones paso a paso diseñados para llevar de la mano al usuario a través de las tareas comunes
- En la elicitación de requisitos los desarrolladores y los usuarios escriben y refinan una serie de escenarios para conseguir un entendimiento común acerca de lo que debe ser el sistema

Actividad: Identificar escenarios (ii)

- Guía
  - ¿Cuáles son las tareas que el actor quiere que realice el sistema?
  - ¿A qué información quiere acceder el actor?
  - ¿Quién crea los datos?
  - ¿Puede ser modificada o eliminada? ¿Por quién?
  - ¿Acerca de que cambios externos tiene el actor que informar al sistema? ¿Con qué frecuencia? ¿Cuándo?
  - ¿Acerca de qué eventos ha de ser informado el actor por parte del sistema? ¿Con qué periodicidad?
  - La respuesta a estas preguntas se obtiene de la documentación del dominio de la aplicación
  - Los escenarios han de escribirse utilizando el lenguaje del dominio
  - Los escenarios se formalizan en los casos de uso

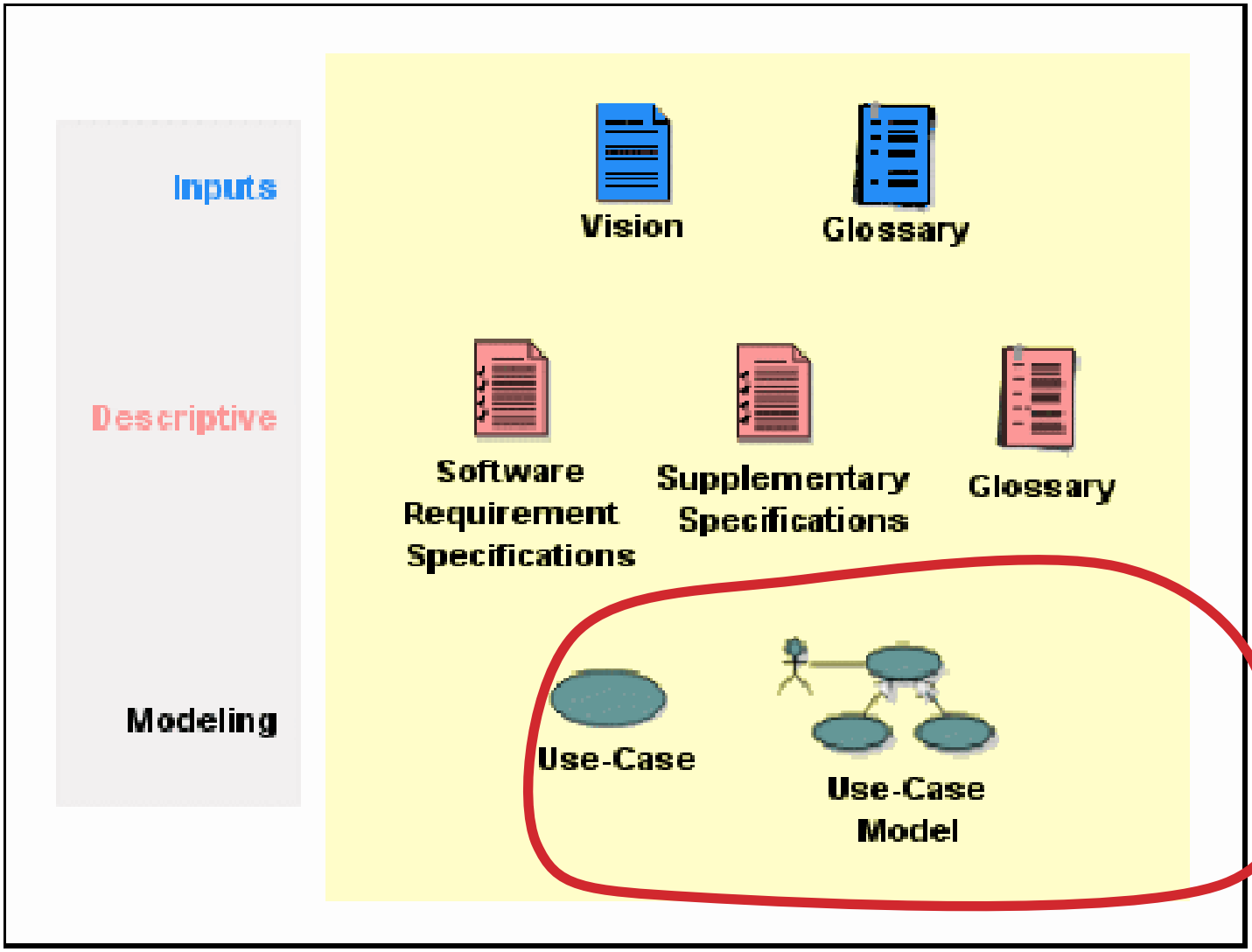
Actividad: Identificar casos de uso (i)

- La identificación de actores y escenarios permiten la comprensión del dominio de la aplicación y la definición del sistema correcto
- El siguiente paso es la formalización de los escenarios en casos de uso
- Un caso de uso especifica todas los escenarios posibles para una funcionalidad dada
- Un caso de uso se inicia por un actor
- Una vez iniciado el caso de uso puede interaccionar con otros actores
- Cada caso de uso es una secuencia completa de eventos iniciada por un actor y especifica la interacción que tiene lugar entre un actor y el sistema
- Un caso de uso es una forma concreta de utilizar un sistema haciendo uso de alguna parte de su funcionalidad
- Agrupa una familia de escenarios de uso según un criterio funcional
- Es un proceso iterativo en el que se debe incluir, refinar, rescribir, eliminar los casos de uso
- No hay que olvidar los casos "raros" o el manejo de excepciones

Actividad: Identificar casos de uso (ii)

- Se identifican los casos de uso de cada actor
  - Un caso de uso es un comportamiento del sistema que produce un resultado mensurable y valioso para un actor
  - Describe las cosas que los actores quieren del sistema
  - Debe ser una tarea completa desde la perspectiva del actor
  - En UML un caso de uso siempre se inicializa por un actor. Pueden existir situaciones en las que se inicie de forma interna al sistema
- Guía para encontrar casos de uso
  - ¿Qué funciones desea el actor del sistema?
  - ¿Almacena el sistema información? ¿Qué actores la crean, leen, actualizan o borran?
  - ¿Necesita el sistema comunicar los cambios en su estado interno a algún actor?
  - ¿Existen eventos externos que el sistema tenga que conocer? ¿Qué actores notifican estos eventos al sistema?
  - ¿Cuáles son las operaciones de mantenimiento del sistema?
- Los casos de uso se determinan observando y precisando, actor por actor, las secuencias de interacción (los escenarios) desde el punto de vista del usuario

Ingeniería de requisitos en el Proceso Unificado (ii)



Actividad: Elaborar el modelo de casos de uso inicial

- Una vez identificados los actores, escenarios y casos de uso se construye el **modelo inicial de casos de uso**
  - Se establecen las relaciones de comunicación entre los actores y los casos de uso
    - Estas relaciones representan el flujo de información del caso de uso
    - El actor que inicia el caso de uso se distingue del resto de actores con los que se puede comunicar el caso de uso
    - Estas relaciones se van identificando a medida que se identifican los casos de uso
  - Se construye el Diagrama de Casos de Uso
  - Se describen los casos de uso
    - Descripción del escenario básico

Actividad: Documentar los casos de uso (i)

- Cada caso de uso tiene que describir **qué** hace un sistema
- Un caso de uso debe ser simple, inteligible, claro y conciso
- Hay que considerar
  - Funcionalidad básica – Flujo de eventos principal (escenario principal)
  - Alternativas – Flujo de eventos excepcional (escenario alternativo)
  - Condiciones de error – Flujo de eventos excepcional
  - Precondiciones y poscondiciones – Qué tiene que ser cierto antes y después del caso de uso
- La descripción caso de uso puede contener condiciones, bifurcaciones e iteraciones
- El flujo de eventos se describe inicialmente de forma textual
- Cuando la comprensión de los requisitos ha avanzado se utilizan diagramas para especificar los escenarios. Diagramas de interacción
- Conviene separar el flujo principal del flujo alternativo
- Se comienza describiendo el flujo principal (escenario básico) al que se le irán añadiendo alternativas y excepciones

Actividad: Documentar los casos de uso (ii)

- Escenario básico
  - Se escribe como si todo transcurriese de forma correcta
  - Debe existir un escenario básico para cada caso de uso
  - Serie de sentencias declarativas sin ramificaciones ni alternativas
- Escenarios alternativos
  - Describen secuencias distintas a la que fue utilizada en el camino básico
  - Documentan las alternativas, situaciones de error o cancelación
  - Dos métodos de localización de caminos alternativos
    - Revisión de cada sentencia del escenario básico
      - ¿Hay alguna acción adicional que pueda hacerse en este punto?
      - ¿Hay algo que pueda fallar en este punto?
      - ¿Hay algún comportamiento que pueda suceder en cualquier momento?
    - Utilización de categorías
      - Un actor [aborta la aplicación / cancela una operación particular / solicita ayuda / proporciona mal los datos]
      - El sistema [falla / no está disponible]
  - Cada camino alternativo necesita un nombre y/o una descripción breve
  - Se documenta en una sección de caminos alternativos o en documentación aparte

Actividad: Documentar los casos de uso (iii)

- Flujo de eventos
  - Es una serie de sentencias declarativas que "relatan" los pasos de un escenario desde la perspectiva del actor
  - Ha de indicar cómo se inicia
    - Sentencia del tipo "El caso de uso comienza cuando..."
  - Ha de indicar cómo finaliza
    - Sentencia del tipo " El caso de uso finaliza con..."
  - Las alternativas se muestran con una sentencia `if`
  - Se pueden indicar repeticiones
    - Hay que indicar claramente donde empieza y finaliza la repetición
    - Hay que indicar la condición de finalización
    - Utilizar `for` o `while`
  - Cada paso tiene que ser una sentencia declarativa simple
  - Por defecto, los pasos deberán estar ordenados temporalmente. En caso contrario ha de especificarse
  - No debe incluirse demasiado detalle. Se están recogiendo los requisitos, no realizando análisis y diseño
- Los requisitos de tipo no funcional o se ponen en un documento aparte o se añaden como coletilla al final de la descripción

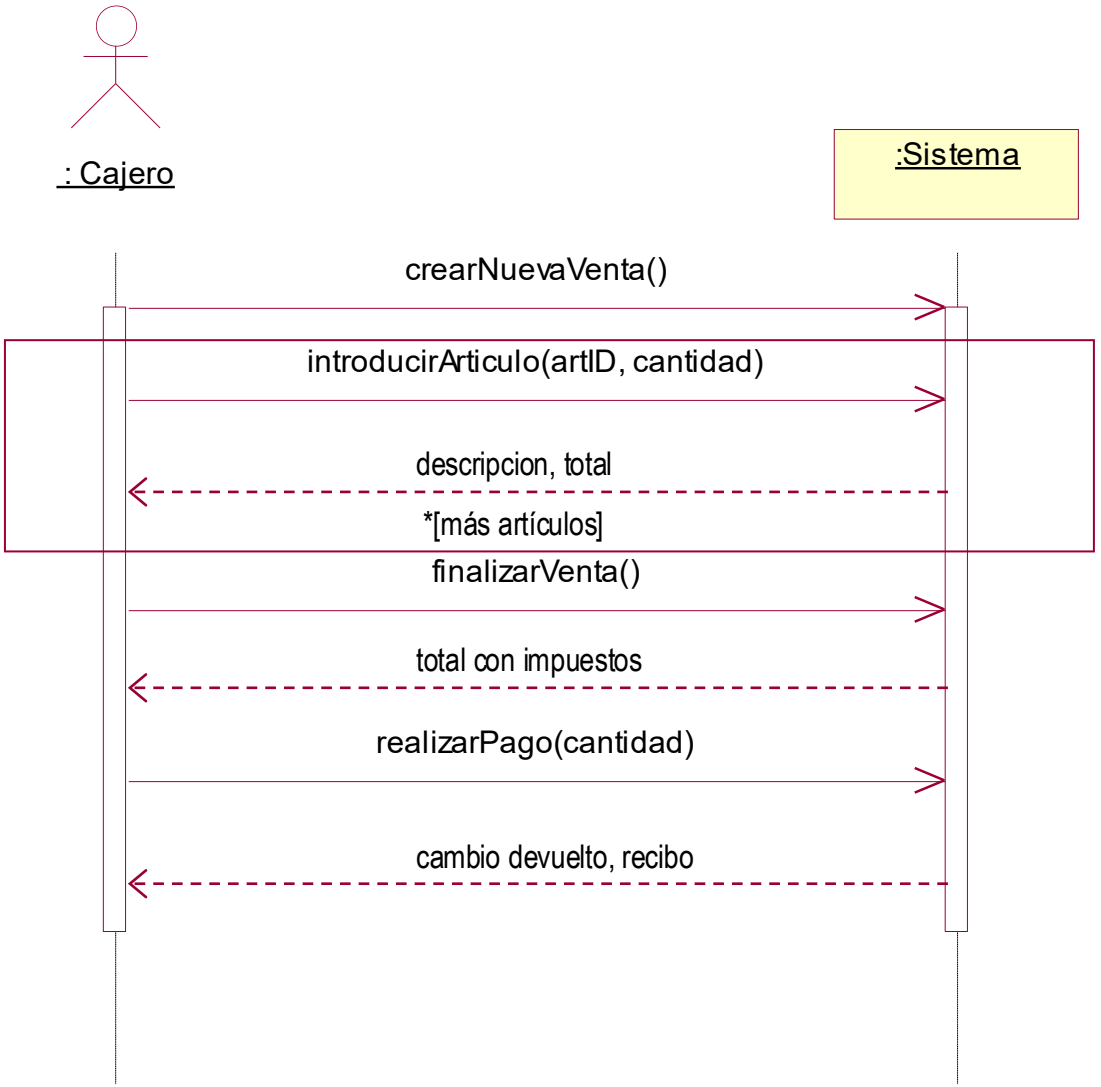
Actividad: Documentar los casos de uso (iv)

- Existen múltiples propuestas para la utilización concreta de los casos de uso como técnica tanto de obtención como de especificación de los requisitos funcionales del sistema
- Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo diversas propuestas y se describen usando lenguaje natural
  - Algunas de estas propuestas son
    - [Schneider y Winters, 2001]
    - [Cockburn, 2000]
    - [Durán y Bernárdez, 2002]

Actividad: Documentar los casos de uso (v)

- Los casos de uso describen cómo interactúan los actores externos con el sistema *software* a crear
- Sería deseable aislar e ilustrar las operaciones que un actor externo solicita a un sistema
- Un **diagrama de secuencia del sistema** (DSS) muestra, para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas
- Todos los sistemas se tratan como cajas negras
- Los diagramas destacan los eventos que cruzan los límites del sistema desde los actores a los sistemas
- Debería hacerse un DSS para el escenario principal del caso de uso, así como para los escenarios alternativos complejos o frecuentes [Larman, 2002]

Actividad: Documentar los casos de uso (vi)



[Larman, 2002]

Actividad: Identificar relaciones entre casos de uso (i)

- Incluso en sistemas de tamaño medio pueden aparecer un número elevado de casos de uso
- Una vez elaborado el modelo inicial de casos de uso hay que organizarlos
- Pueden existir similitudes en varios casos de uso que se pueden abstraer en un caso de uso común
  - Se utilizará la relación **«include»** para reducir la redundancia entre casos de uso
- Se puede desear extender un caso de uso sin cambiar la descripción original
  - Se utilizará la relación **«extend»** para separar los flujos de eventos comunes de los excepcionales
- También se pueden encontrar similitudes en actores
- En resumen, se utilizan las relaciones de generalización, inclusión y extensión para factorizar el comportamiento común y las variantes

Actividad: Identificar relaciones entre casos de uso (ii)

- Si se repiten bloques de comportamiento entre casos de uso, puede indicar que existe algo genérico que puede reutilizarse
- Se puede abstraer el comportamiento común en una relación de inclusión
- Factorizar el comportamiento común de los casos de uso presenta grandes ventajas, incluyendo descripciones más cortas y menor redundancia
- El comportamiento **únicamente** puede ser factorizado en un caso de uso separado si es compartido por dos o más casos de uso
-  La fragmentación excesiva de la especificación del sistema puede conducir a una especificación confusa
- Procedimiento
  - Identificar los pasos de los escenarios que se quieren utilizar en diferentes sitios
  - Agruparlos en un caso de uso y darle nombre
-  El caso de uso incluido **no puede** tener dependencias con respecto a ningún caso de uso que lo incluya

Actividad: Identificar relaciones entre casos de uso (iii)

- Un caso de uso extiende a otro caso de uso si el caso de uso extendido puede incluir el comportamiento de la extensión bajo ciertas condiciones
- Modela la parte de un caso de uso que el usuario puede ver como un comportamiento opcional del sistema
- Se utiliza para extender de **forma condicionada** el comportamiento de un caso de uso existente
- Es una forma de añadir comportamiento a un caso de uso sin modificarlo
- Se utiliza cuando se trabajo con una versión posterior de un producto existente o para indicar los sitios donde un producto puede adaptarse o personalizarse
- El caso de uso se actualiza **añadiendo** puntos de extensión
- Las extensiones se describen al igual que los casos de uso. Debe existir una condición de entrada (inicio) y de salida
- El caso de uso extendido **no cambia**

Actividad: Identificar relaciones entre casos de uso (iv)

- La generalización indica que un caso de uso es una versión especializada de otro caso de uso
- El caso de uso general puede ser únicamente una descripción, los flujos se especifican en los casos de uso especializados
- Es posible agregar comportamiento al caso de uso hijo añadiendo pasos a la secuencia de comportamiento heredada del padre, así como declarando relaciones de extensión y de inclusión para el hijo
- Si el padre es abstracto, su secuencia de comportamiento puede tener secciones que sean explícitamente incompletas en el padre y que debe proporcionar el hijo
- El hijo puede redefinir pasos heredados del padre
- En la secuencia heredada del padre se pueden intercalar pasos adicionales
- El uso de la generalización múltiple en casos de uso requiere la especificación explícita de la forma en la que se intercalan las secuencias de comportamiento de los padres para crear la secuencia correspondiente al hijo
- El caso de uso base (**A**) es autosuficiente. El caso de uso hijo (**B**) necesita al caso de uso base (obtiene la funcionalidad base de **A**) y controla qué es ejecutado desde **A** y qué cambia

Construcción del Modelo de casos de uso – Resumen (i)

- Cada caso de uso debe representar un comportamiento distinto e identificable del sistema o de una parte del mismo
- Un caso de uso bien estructurado cumple que [Booch et al., 1999]
  - Nombra un comportamiento simple, identificable y razonablemente atómico del sistema o parte del sistema
  - Factoriza el comportamiento común, incorporando ese comportamiento desde otros casos de uso que incluye
  - Factoriza variantes, colocando ese comportamiento en otros casos de uso que lo extienden o lo especializan
  - Describe el flujo de eventos de forma suficientemente clara para que alguien externo al sistema lo entienda fácilmente
  - Se describe por un conjunto mínimo de escenarios que especifican la semántica normal y las variantes del caso de uso

Construcción del Modelo de casos de uso – Resumen (ii)

- A la hora de realizar los casos de usos más que preguntarse si un caso de uso es válido o no, habría que preguntarse cuál es el nivel útil para expresar los casos de uso en el análisis de requisitos de una aplicación
- Para el análisis de requisitos de una aplicación informática, hay que centrarse en los casos de uso al nivel de **procesos del negocio elementales** o **EBPs** (*Elementary Business Processes*)
  - Un EBP es un término que procede del campo de la ingeniería de procesos del negocio
    - Es una tarea realizada por una persona en un lugar, en un instante, como respuesta a un evento de negocio, que añade un valor cuantificable para el negocio y deja los datos en un estado consistente [Larman, 2002]
- Un caso de uso de nivel EBP se denomina caso de uso de nivel de objetivo de usuario, para remarcar que sirve (o debería servir) para satisfacer un objetivo de un usuario del sistema o actor principal
  - Procedimiento
    - Encontrar los objetivos de usuario
    - Definir un caso de uso para cada objetivo

Construcción del Modelo de casos de uso – Resumen (iii)

- Por lo general se define un caso de uso de nivel EBP por cada objetivo de usuario
- Se nombra cada caso de uso de forma similar al objetivo de usuario
- Los casos de uso se suelen nombrar comenzando por un verbo
- Una excepción típica a crear un caso de uso por objetivo es agrupar objetivos separados en un caso de uso CRUD (*Create-Retrieve-Update-Delete* – Crear-Recuperar-Actualizar-Eliminar)
  - Por convención se denomina a este caso de uso **Gestionar<X>**

https://unsplash.com/photos/fJLyQ81u80Y

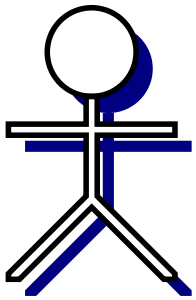
8. Caso de estudio

Sistema de procesamiento de pedidos

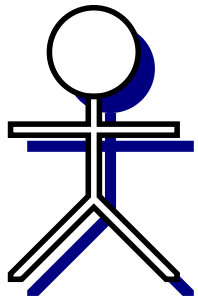
- Descripción del problema
  - Se desea desarrollar un *software* para el procesamiento de pedidos para una empresa de venta por correo llamada NW, que actúa como revendedora de productos adquiridos a varios proveedores
  - Dos veces al año la compañía publica un catálogo de productos que se envía por correo a los clientes y otras personas interesadas
  - Los clientes adquieren los productos enviando una lista de productos con el pago a NW. Esta empresa despacha el pedido y envía los productos a la dirección del cliente
  - El *software* de procesamiento de pedidos ha de hacer un seguimiento desde el momento en el que se solicita hasta el momento en que el producto es enviado
  - NW tiene que proporcionar un servicio rápido. Debe ser capaz de enviar un pedido de cliente por el medio más rápido y eficiente posible
  - Los clientes pueden devolver elementos, que se incorporan de nuevo al almacén, pero en ocasiones han de pagar una tasa

Ejemplo tomado de [Schneider y Winters, 2001]

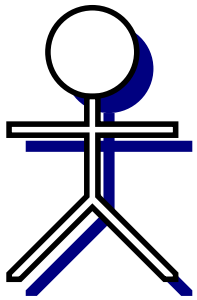
Sistema de procesamiento de pedidos – Actores (i)



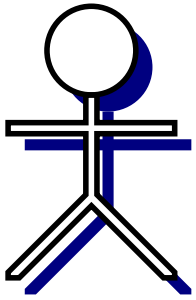
Cliente



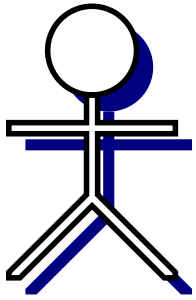
Comercial



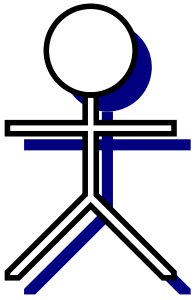
Almacenista



Servicio de entrega



Sistema de contabilidad



Sistema de inventario

Sistema de procesamiento de pedidos – Actores (ii)

- Cliente
  - Una persona que solicita productos a NW
- Comercial
  - Un empleado de NW que procesa las peticiones de los clientes
- Servicio de entrega
  - Los que se encargan de hacer la distribución del pedido
- Almacenista
  - Empleado de NW que empaqueta, etiqueta y prepara pedidos
- Sistema de inventario
  - Sistema software que mantiene el control del inventario de la compañía
- Sistema de contabilidad
  - Sistema software que mantiene los libros de contabilidad de la compañía

Sistema de procesamiento de pedidos – Escenarios

- ¿A qué información quiere acceder el actor? Al catálogo de productos
- Escenario: Obtener el catálogo de productos
- Descripción: Este escenario describe cómo un cliente puede solicitar un catálogo
  1. El cliente selecciona "Obtener Catálogo"
  2. Se muestra la pantalla de "Obtener Catálogo"
  3. El usuario introduce el nombre y su dirección postal
  4. El usuario selecciona enviar
  5. El sistema crea un pedido para un catálogo de productos con un coste de 0€
  6. El sistema graba el pedido

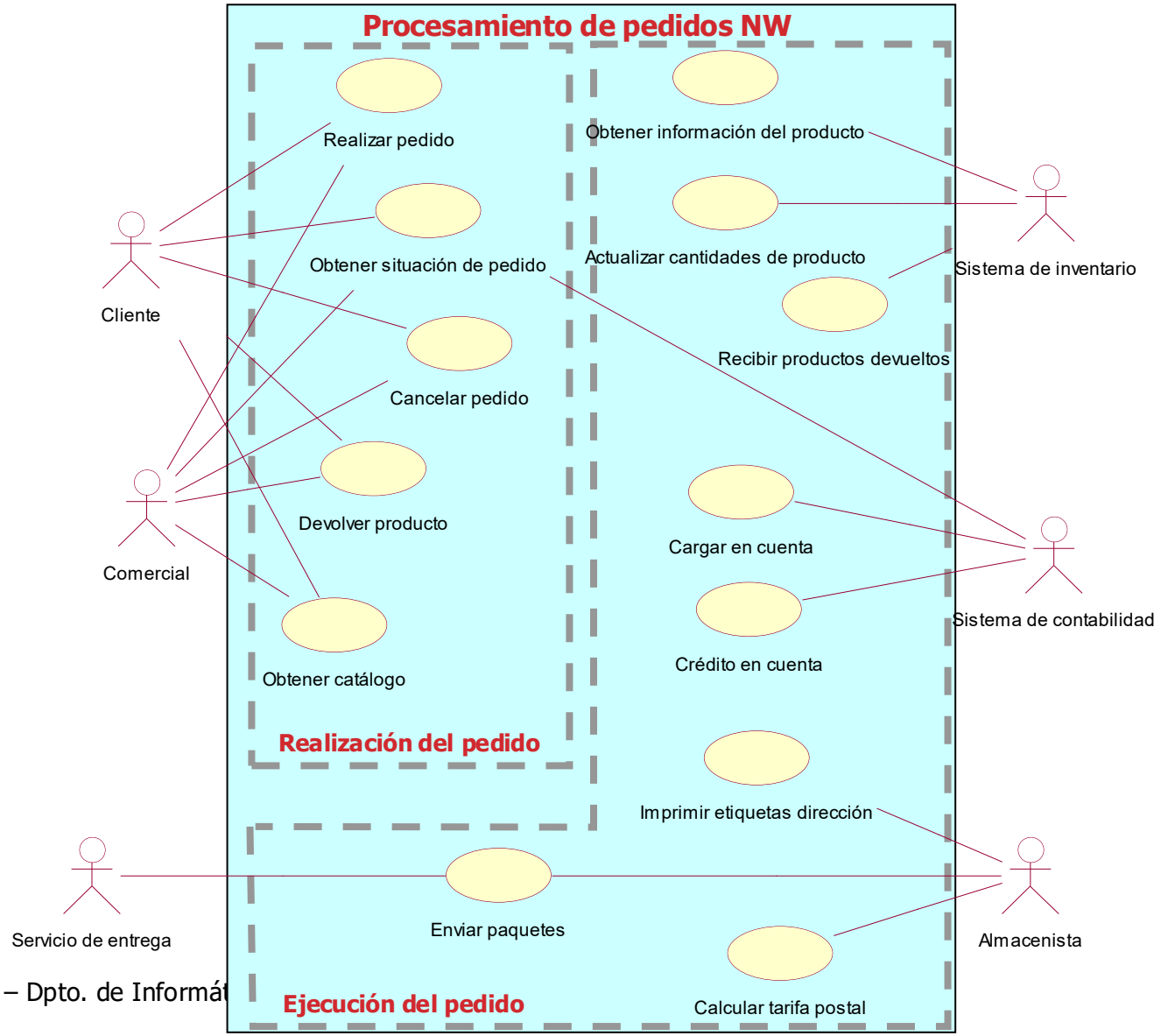
Sistema de procesamiento de pedidos – Casos de uso (i)

- Realizar pedido
  - Un cliente crea un pedido nuevo para solicitar productos y proporciona el medio de pago de esos productos
- Obtener catálogo
  - Un cliente solicita un catálogo
- Obtener situación pedido
  - Un cliente obtiene información acerca del estado de un pedido realizado
- Devolver producto
  - Un cliente devuelve un pedido y recupera el importe
- Cancelar pedido
  - Un cliente cancela un pedido existente
- Enviar paquetes
  - La empresa realiza la entrega de los productos a los clientes
- Calcular tarifa postal
  - Determinar el importe del envío al cliente

Sistema de procesamiento de pedidos – Casos de uso (ii)

- Imprimir etiquetas dirección
  - Generar las etiquetas con la dirección postal de los clientes
- Obtener información del producto
  - Recuperar información acerca de un producto, su precio y cantidad en almacén
- Actualizar cantidades de producto
  - Actualizar la cantidad de producto en almacén
- Recibir productos devueltos
  - Procesamiento necesario que hay que realizar cuando un producto es devuelto por un cliente
- Cargar en cuenta
  - Cargar el importe del pedido en la cuenta de cliente
- Crédito en cuenta
  - Establecer el crédito en la cuenta del cliente

Sistema de procesamiento de pedidos – Modelo de casos de uso inicial



Sistema de procesamiento de pedidos – Documentación de CU (i)

- **Ingeniero del software**
  - ¿Cómo se realizan los pedidos por parte de los clientes?
- **Cliente**
  - El cliente se conecta al sistema y selecciona "Realizar Pedido", e introduce su nombre y dirección. Si el cliente introduce solamente su código postal, el sistema proporciona la ciudad y provincia
  - Después el cliente introduce los códigos de los productos que desea y el sistema le proporciona la descripción y el precio para cada artículo. A medida que va solicitando artículos el sistema va controlando el total de artículos solicitados en el orden en el que se han introducido
  - Cuando ha acabado tiene que proporcionar la información de la tarjeta de crédito para el pago y pulsa botón de "Enviar"
  - Es entonces cuando el sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido

Sistema de procesamiento de pedidos – Documentación de CU (ii)

- Escenario base
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal, el sistema proporciona la ciudad y provincia
  4. El cliente introduce los códigos de los productos que desea
  5. El sistema proporciona la descripción y el precio para cada artículo
  6. El sistema mantiene el control del total de artículos solicitados en el orden en el que se han introducido
  7. El cliente introduce la información de la tarjeta de crédito para el pago
  8. El cliente selecciona "Enviar"
  9. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (iii)

- Se señalan las alternativas
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal
     1. El sistema proporciona la ciudad y provincia
  4. El cliente introduce los códigos de los productos que desea
  5. El sistema proporciona la descripción y el precio para cada artículo
  6. El sistema mantiene el control del total de artículos solicitados en el orden en el que se han introducido
  7. El cliente introduce la información de la tarjeta de crédito para el pago
  8. El cliente selecciona "Enviar"
  9. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (iv)

- Se señalan las iteraciones, por ejemplo con una construcción `for`
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal
     1. El sistema proporciona la ciudad y provincia
  4. El cliente introduce los códigos de los productos que desea
  5. El sistema proporciona la descripción y el precio para cada artículo
  6. El sistema mantiene el control del total de artículos solicitados en el orden en el que se han introducido
  7. El cliente introduce la información de la tarjeta de crédito para el pago
  8. El cliente selecciona "Enviar"
  9. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (v)

- Quedando…
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal
     1. El sistema proporciona la ciudad y provincia
  4. El cliente introduce los códigos de los productos que desea
  5. Para cada código de producto introducido
     1. El sistema proporciona la descripción y el precio para cada artículo
     2. El sistema añade el precio del artículo al total

     fin para
  6. El cliente introduce la información de la tarjeta de crédito para el pago
  7. El cliente selecciona "Enviar"
  8. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  9. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (vi)

- También se podría haber utilizado una construcción `while`
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal
     1. El sistema proporciona la ciudad y provincia
  4. ~~El cliente introduce los códigos de los productos que desea~~
  5. Mientras el cliente introduzca códigos de producto
     1. El sistema proporciona la descripción y el precio para cada artículo
     2. El sistema añade el precio del artículo al total

     fin mientras
  6. El cliente introduce la información de la tarjeta de crédito para el pago
  7. El cliente selecciona "Enviar"
  8. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  9. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (vii)

- Quedando con la construcción `while` como sigue

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección
3. Si el cliente introduce solamente su código postal
    1. El sistema proporciona la ciudad y provincia
4. Mientras el cliente introduzca códigos de producto
    1. El sistema proporciona la descripción y el precio para cada artículo
    2. El sistema añade el precio del artículo al total

    fin mientras
5. El cliente introduce la información de la tarjeta de crédito para el pago
6. El cliente selecciona "Enviar"
7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (viii)

- Se puede introducir un requisito no funcional

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección
3. Si el cliente introduce solamente su código postal
   1. El sistema proporciona la ciudad y provincia
4. Mientras el cliente introduzca códigos de producto
   1. El sistema proporciona la descripción y el precio para cada artículo
   2. El sistema añade el precio del artículo al total

   fin mientras
5. El cliente introduce la información de la tarjeta de crédito para el pago
6. El cliente selecciona "Enviar"
7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Requisito no funcional

El sistema ha de responder siempre a la entrada del usuario en menos de un segundo

Sistema de procesamiento de pedidos – Documentación de CU (ix)

- Se pueden especificar los actores implicados

Cliente

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección
3. Si el cliente introduce solamente su código postal
   1. El sistema proporciona la ciudad y provincia
4. Mientras el cliente introduzca códigos de producto
   1. El sistema proporciona la descripción y el precio para cada artículo
   2. El sistema añade el precio del artículo al total

   fin mientras
5. El cliente introduce la información de la tarjeta de crédito para el pago
6. El cliente selecciona "Enviar"

Sistema de contabilidad

7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Requisito no funcional

El sistema ha de responder siempre a la entrada del usuario en menos de un segundo

Sistema de procesamiento de pedidos – Documentación de CU (x)

- Ahora habría que centrarse en las alternativas
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. Si el cliente introduce solamente su código postal
     1. El sistema proporciona la ciudad y provincia
  4. Mientras el cliente introduzca códigos de producto
     1. El sistema proporciona la descripción y el precio para cada artículo
     2. El sistema añade el precio del artículo al total

     fin mientras
  5. El cliente introduce la información de la tarjeta de crédito para el pago
  6. El cliente selecciona "Enviar"
  7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad. Si alguna información no es correcta el sistema, solicita al cliente que la corrija
  8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza. Si el pago no se confirma, el sistema solicita al cliente que corrija la información de pago o que cancele. Si el cliente elige corregir la información volver al paso 5, si el cliente elige cancelar el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (xi)

- Se tiene el caso de uso con el escenario base y las alternativas señaladas

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección
3. Si el cliente introduce solamente su código postal
   1. El sistema proporciona la ciudad y provincia
4. Mientras el cliente introduzca códigos de producto
   1. El sistema proporciona la descripción y el precio para cada artículo
   2. El sistema añade el precio del artículo al total

   fin mientras
5. El cliente introduce la información de la tarjeta de crédito para el pago
6. El cliente selecciona "Enviar"
7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Caminos alternativos

- En cualquier momento, antes de seleccionar "Enviar", el cliente puede seleccionar "Cancelar". El pedido no se guarda y el caso de uso finaliza
- En el paso 7, si alguna información no es correcta, el sistema solicita al cliente que corrija la información
- En el paso 8, si el pago no se confirma, el sistema solicita al cliente que corrija la información de pago o que cancele. Si el cliente elige corregir la información, volver al paso 5 del escenario básico. Si el cliente elige cancelar, el caso de uso finaliza

Sistema de procesamiento de pedidos – Documentación de CU (xii)

- Se documentan los caminos (escenarios) alternativos

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección

⋮

6. El cliente selecciona "Enviar"
7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Caminos alternativos

Alternativa 1: Cancelar

1. En cualquier punto del escenario base del caso de uso, el cliente puede seleccionar "Cancelar"
2. El sistema solicita al cliente que verifique la cancelación
3. El cliente selecciona "Ok" y el caso de uso finaliza

Alternativa 2: Datos incorrectos

⋮

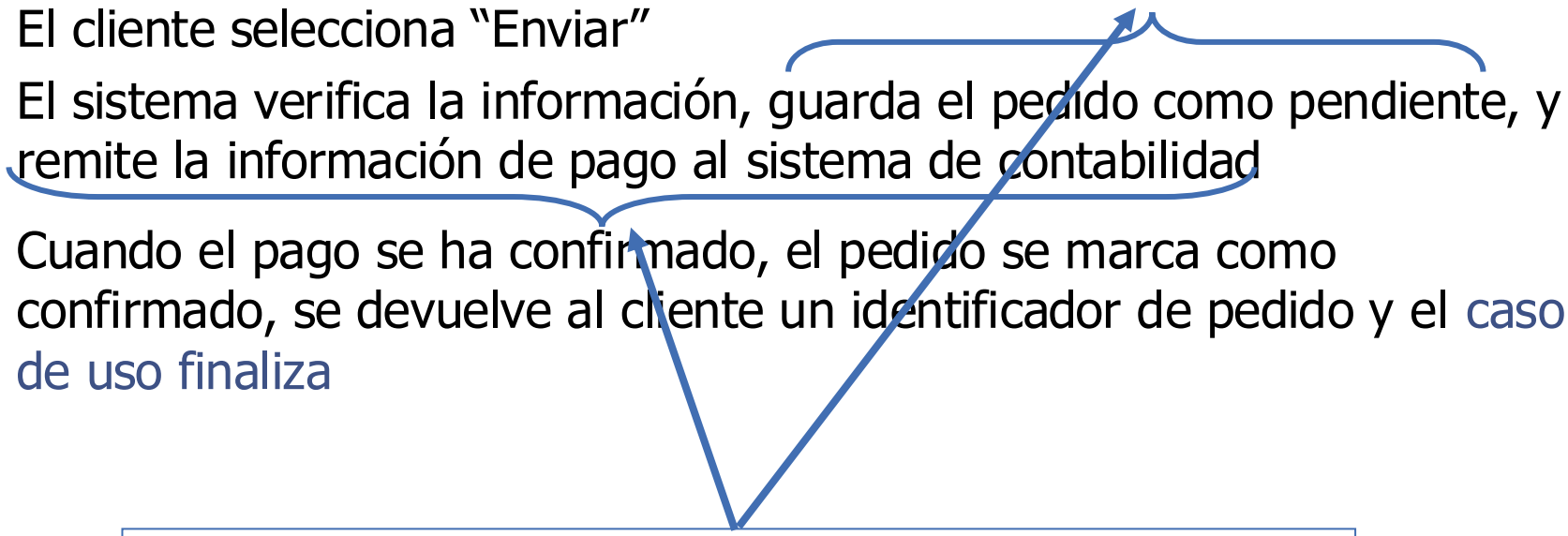
Alternativa 3: Pago no confirmado

⋮


Y ahora, ¿qué tengo que hacer?
Pero... ¡si acabas de empezar!
¡Ya he acabado el análisis de requisitos!
Refinar

Sistema de procesamiento de pedidos – Refinar los CU (i)

- Se llegó a este punto
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  ⋮
  6. El cliente selecciona "Enviar"
  7. El sistema verifica la información, guarda el pedido como pendiente, y remite la información de pago al sistema de contabilidad
  8. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza



Pero ahora se va a mejorar la descripción del escenario en el punto 7

Sistema de procesamiento de pedidos – Refinar los CU (ii)

- Quedando así más claro

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección

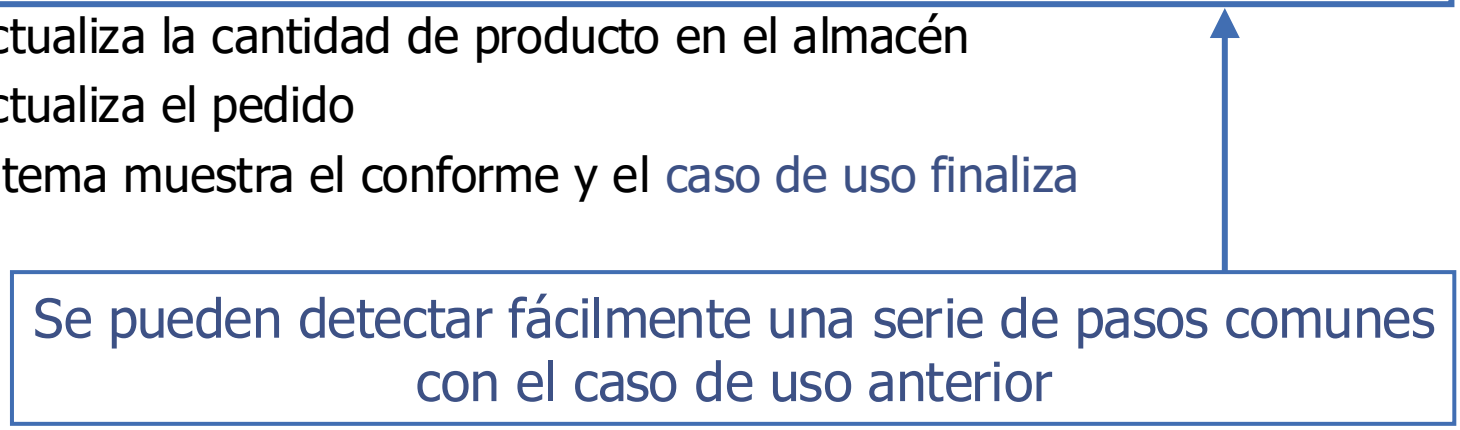
⋮

6. El cliente selecciona "Enviar"
7. El sistema verifica la información
8. Guarda el pedido como pendiente
9. Solicita una autorización de pago al sistema de contabilidad
10. Cuando la pago se ha confirmado, el pedido se marca como confirmado,
11. El sistema de contabilidad envía el conforme
12. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (iii)

- Se va a considerar ahora el caso de uso **Devolver producto**
  1. El caso de uso se inicia cuando el comercial selecciona "Devolver producto"
  2. El comercial busca el pedido
  3. El sistema muestra el pedido seleccionado
  4. El comercial selecciona el producto a devolver
  5. El comercial selecciona "Devolver"
  6. Se solicita una actualización de cuenta
  7. Se envía la información de la tarjeta de crédito y la cantidad a abonar
  8. El sistema de contabilidad envía el conforme
  9. Se actualiza la cantidad de producto en el almacén
  10. Se actualiza el pedido
  11. El sistema muestra el conforme y el caso de uso finaliza



Sistema de procesamiento de pedidos – Refinar los CU (iv)

- Esto provoca la identificación de un nuevo caso de uso
  - Caso de uso: **Actualizar cuenta**
    - El caso de uso interacciona con el sistema de contabilidad para aplicar los cargos o los débitos a una cuenta de cliente
  - Flujo de eventos
    - Escenario básico
      1. El caso de uso comienza cuando se recibe una petición de actualizar una cuenta
      2. El sistema envía la información de la tarjeta de crédito y la cantidad de crédito o débito al sistema de contabilidad
      3. El sistema de contabilidad envía el conforme
      4. El caso de uso finaliza
    - Escenarios alternativos
      - La cuenta no existe
      - La cuenta está en números rojos
      - El sistema de contabilidad no está disponible

Sistema de procesamiento de pedidos – Refinar los CU (v)

- Así los casos de uso **Realizar pedido** y **Devolver producto** se ven modificados

Caso de uso: Devolver producto

1. El caso de uso se inicia cuando el comercial selecciona " Devolver producto"
2. El comercial busca el pedido
3. El sistema muestra el pedido seleccionado
4. El comercial selecciona el producto a devolver
5. El comercial selecciona "Devolver"
6. **Incluir Actualizar Cuenta** ←
7. Se actualiza la cantidad de producto en el almacén
8. Se actualiza el pedido
9. El sistema muestra el conforme y el caso de uso finaliza

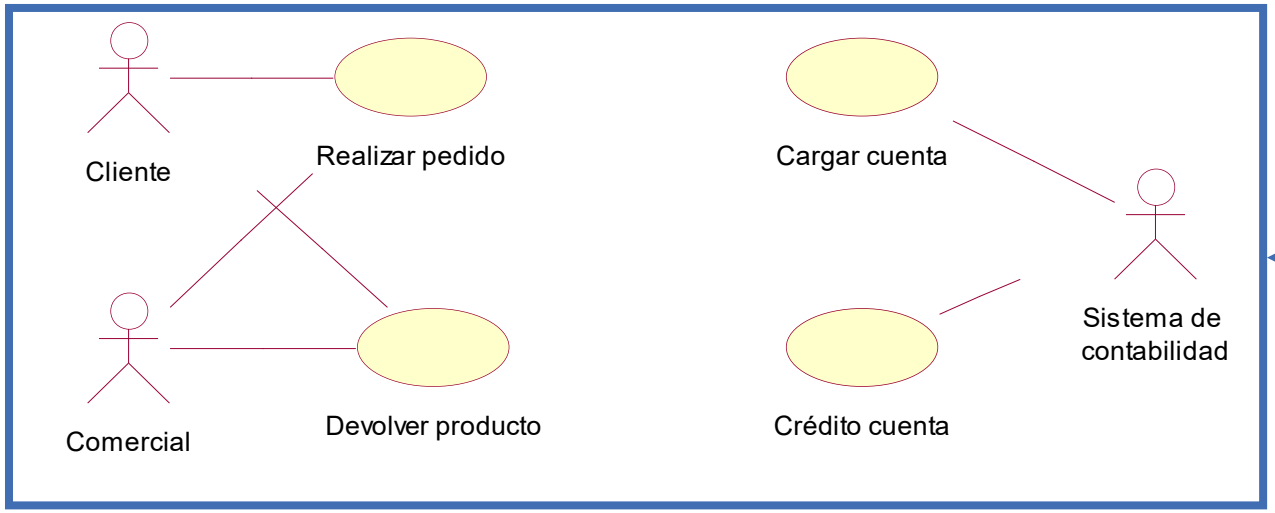
Caso de uso: Realizar pedido

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección

⋮

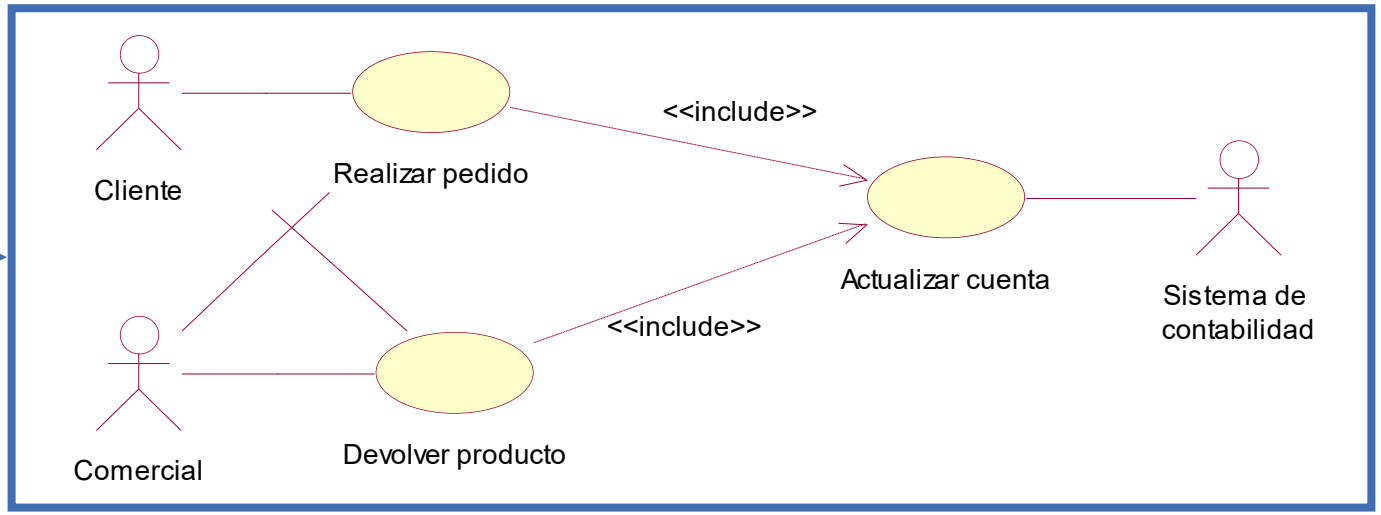
6. El cliente selecciona "Enviar"
7. El sistema verifica la información
8. Guarda el pedido como pendiente
9. **Incluir Actualizar Cuenta** ←
10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (vi)



De este diagrama

Se pasa a este otro



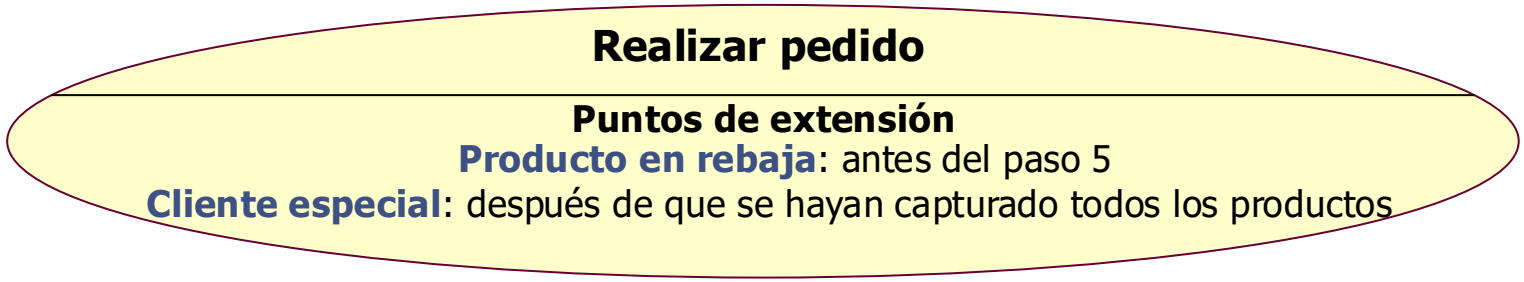
Sistema de procesamiento de pedidos – Refinar los CU (vii)

- El usuario acaba de acordarse de que hay que tener en cuenta las rebajas y los descuentos para los mejores clientes

- Se va a utilizar la relación **«extend»** y así reutilizar la especificación existente

Sistema de procesamiento de pedidos – Refinar los CU (viii)

- Se incluyen los puntos de extensión



- La descripción del caso de uso no cambia

Caso de uso: Realizar pedido

1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
2. El cliente introduce su nombre y dirección

⋮

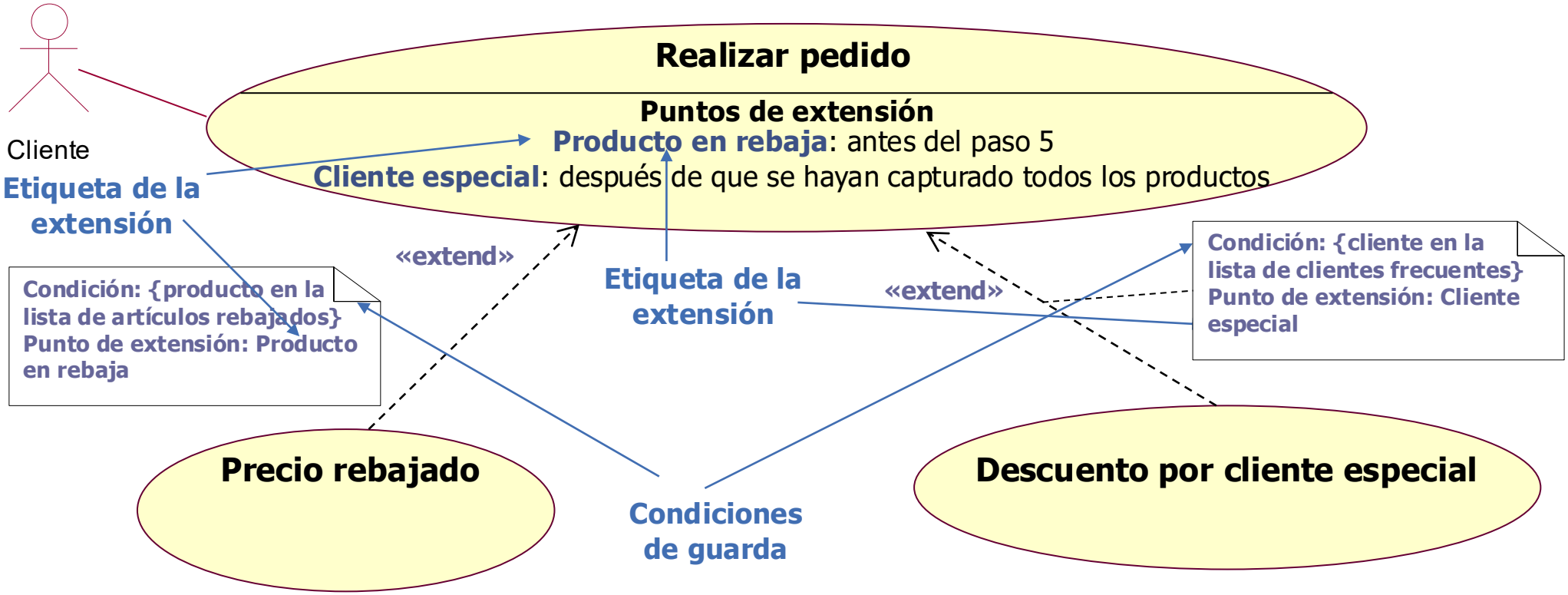
6. El cliente selecciona "Enviar"
7. El sistema verifica la información
8. Guarda el pedido como pendiente
9. **Incluir Actualizar Cuenta**
10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

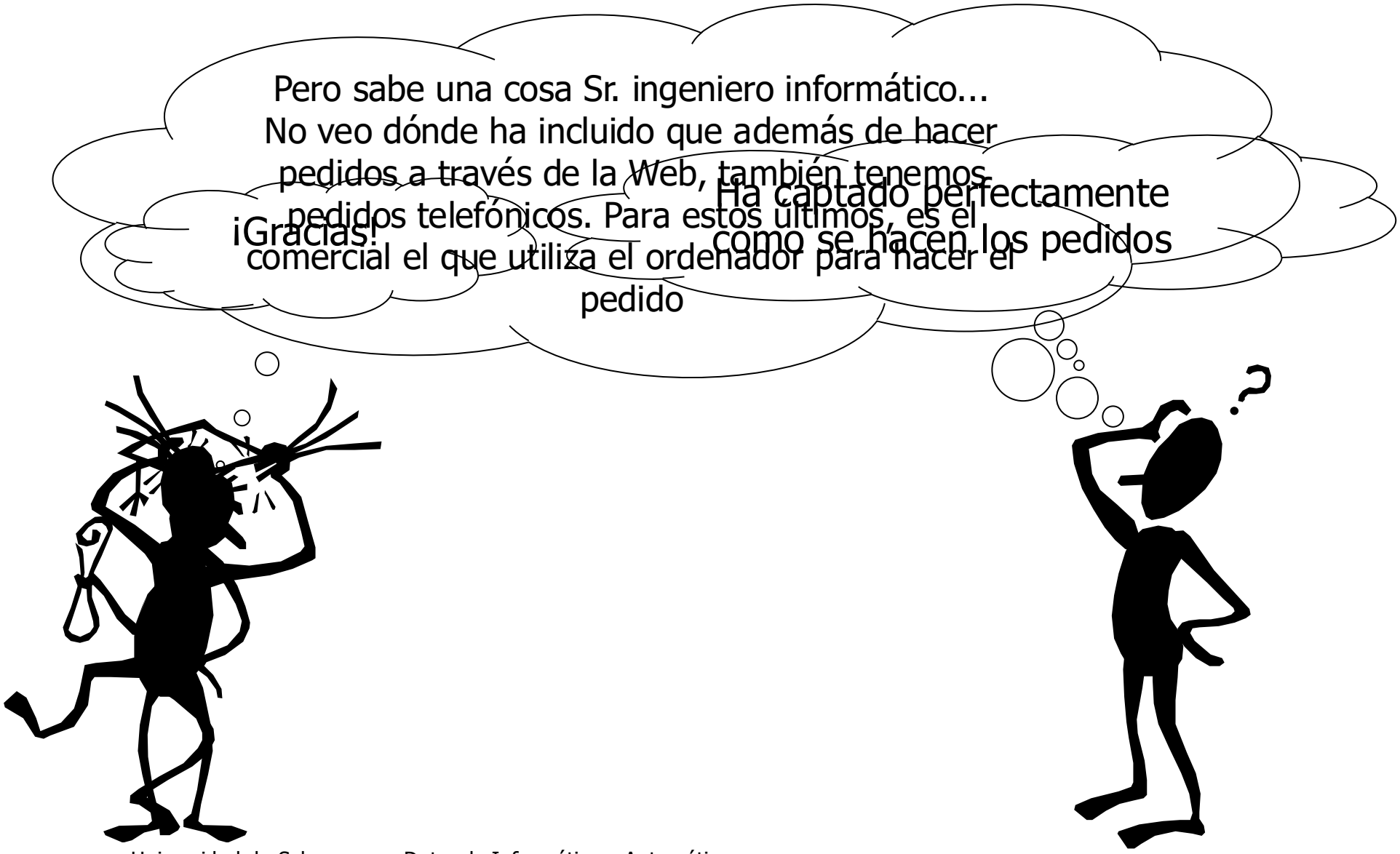
Sistema de procesamiento de pedidos – Refinar los CU (ix)

- Ya solo queda describir los casos de uso extensión
  - Caso de uso: **Precio rebajado**
    - El caso de uso comienza cuando el sistema detecta el descuento por rebaja del producto
    - El sistema muestra el porcentaje de descuento en el pedido
    - El sistema calcula una cantidad de descuento multiplicando el precio original por el descuento de rebaja
    - El sistema resta la cantidad a descontar del total del pedido y el caso de uso finaliza
  - Caso de uso: **Descuento por cliente especial (cliente frecuente)**
    - El caso de uso comienza cuando el sistema detecta el descuento al cliente
    - El sistema muestra el porcentaje de descuento en el pedido
    - El sistema calcula una cantidad de descuento multiplicando el precio original por el descuento de rebaja
    - El sistema resta la cantidad a descontar del total del pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (x)

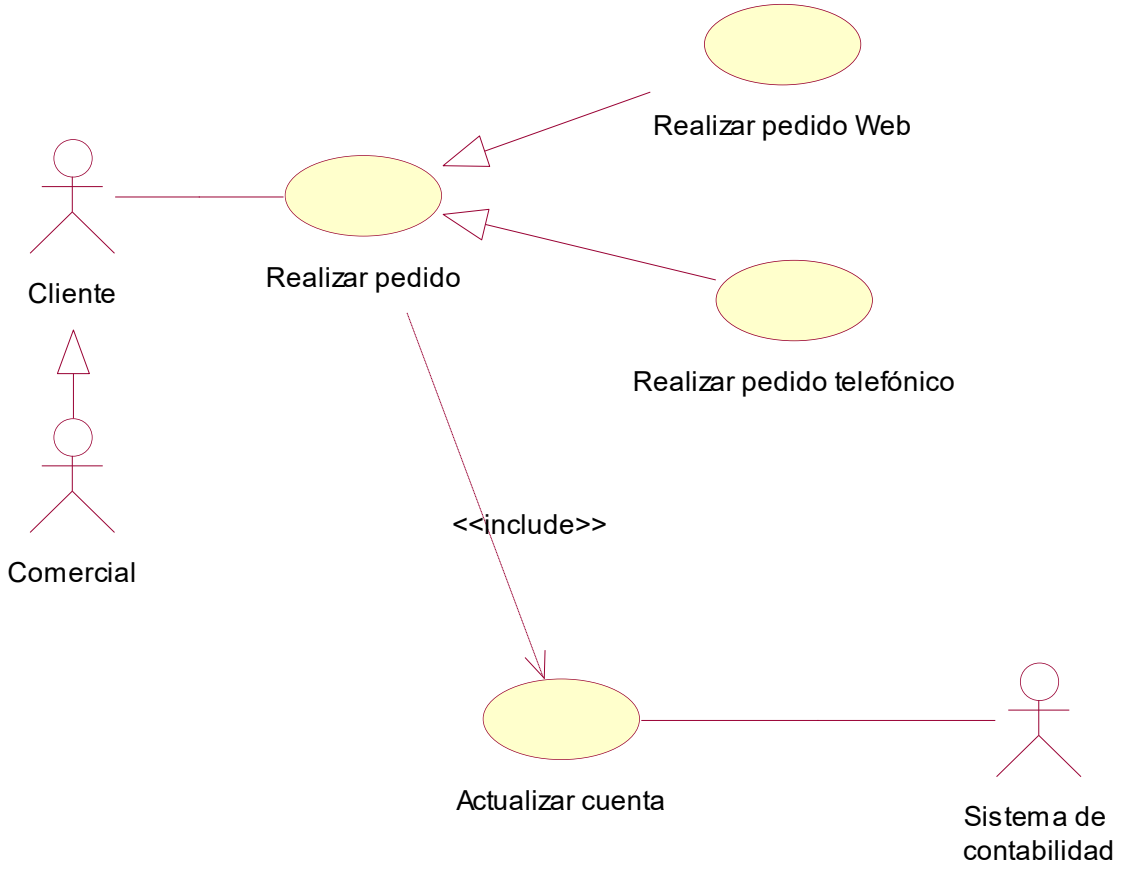
- Se actualiza la parte del diagrama correspondiente




Pero sabe una cosa Sr. ingeniero informático...
No veo dónde ha incluido que además de hacer pedidos a través de la Web, también tenemos pedidos telefónicos. Para estos últimos, es el comercial el que utiliza el ordenador para hacer el pedido
¡Gracias!
Ha captado perfectamente como se hacen los pedidos

Sistema de procesamiento de pedidos – Refinar los CU (xi)

- Se rehace en diagrama de casos de uso



Sistema de procesamiento de pedidos – Refinar los CU (xii)

- Se documentan los casos de uso
  - Se tienen dos posibles alternativas
    - En la primera, se especifica el caso de uso **Realizar pedido**, como un caso de uso abstracto, y se difiere a los casos de uso **Realizar pedido telefónico** y **Realizar pedido web** la descripción concreta de la funcionalidad
    - En el segundo supuesto, en el caso de uso **Realizar pedido** se describe la funcionalidad común a los dos casos de uso, y en los casos de uso **Realizar pedido telefónico** y **Realizar pedido web** se describen las extensiones correspondientes

Sistema de procesamiento de pedidos – Refinar los CU (xiii)

- Alternativa 1: se describe **Realizar pedido** como un caso de uso abstracto

Caso de uso: **Realizar Pedido**

Este caso de uso permite a un cliente realizar pedidos de productos a NW. Los datos requeridos para este caso de uso incluyen la dirección de facturación del cliente, información de pago y la lista de productos seleccionados. Opcionalmente, el cliente puede especificar una dirección de entrega diferente a la dirección de facturación

Sistema de procesamiento de pedidos – Refinar los CU (xiv)

- Caso de uso: **Realizar pedido telefónico** (Realizar pedido abstracto)
  1. El caso de uso comienza cuando el cliente llama al comercial de NW
  2. El comercial obtiene un identificador de catálogo del cliente y lo introduce en el sistema
  3. El sistema obtiene el nombre y dirección del cliente de la base de datos
  4. El comercial verifica esta información con el cliente
  5. El comercial le solicita los códigos de productos al cliente y los introduce en el sistema
  6. Para cada código de producto introducido
     1. El sistema proporciona una descripción de producto y su precio
     2. El sistema añade el precio del producto al total

     fin para
  7. El comercial le solicita la información de pago al cliente y la introduce en el sistema
  8. El comercial envía la información de pago al sistema
  9. El sistema almacena el pedido como pendiente
  10. Incluir Actualizar Cuenta
  11. Cuando el pago está confirmado, el pedido se marca como confirmado, se envía un identificador de pedido al cliente, y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (xv)

- Caso de uso: **Realizar pedido web** (Realizar pedido abstracto)
  1. El caso de uso comienza cuando el cliente selecciona "Realizar pedido" en la página principal de NW
  2. El sistema muestra la página principal del catálogo en línea
  3. El cliente navega por el catálogo en línea y selecciona los productos a adquirir
  4. Para cada producto seleccionado
     1. El sistema proporciona una descripción de producto y su precio.
     2. El cliente selecciona añadir el producto a la carta de compras.
     3. El sistema añade el precio del producto al total de la carta de compras.

     fin para
  5. El cliente selecciona "Compra"
  6. El sistema solicita al cliente su nombre de usuario y su clave
  7. El cliente introduce un nombre de usuario y una clave y selecciona "Enviar"
  8. El sistema muestra la dirección de envío y método de pago para esta cuenta de cliente
  9. El cliente selecciona "Ok"
  10. El sistema almacena el pedido como pendiente
  11. Incluir Actualizar Cuenta
  12. Cuando el pago está confirmado, el pedido se marca como confirmado, se envía un identificador de pedido al cliente, y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (xvi)

- Alternativa 2: Se realiza la descripción de los casos de uso **Realizar pedido**, **Realizar pedido telefónico** y **Realizar pedido web** partiendo de la descripción del caso de uso **Realizar pedido**, pero incorporando las variaciones en los casos de uso hijos

Sistema de procesamiento de pedidos – Refinar los CU (xvii)

- Se parte de la especificación conocida de **Realizar pedido**
  1. El caso de uso se inicia cuando el cliente selecciona "Realizar Pedido"
  2. El cliente introduce su nombre y dirección
  3. El cliente introduce los códigos de los productos que desea
  4. Para cada código de producto introducido
     1. El sistema proporciona la descripción y el precio para cada artículo.
     2. El sistema añade el precio del artículo al total.

     fin para
  5. El cliente introduce la información de la tarjeta de crédito para el pago
  6. El cliente selecciona "Enviar"
  7. El sistema verifica la información
  8. Guarda el pedido como pendiente
  9. Incluir Actualizar Cuenta
  10. Cuando el pago se ha confirmado, el pedido se marca como confirmado, se devuelve al cliente un identificador de pedido y el caso de uso finaliza

Sistema de procesamiento de pedidos – Refinar los CU (xviii)

- Y se especifican los dos casos de uso hijos
  - **Realizar pedido telefónico**
    - Este caso de uso es igual al de **Realizar pedido** excepto en que
      - El cliente proporciona toda la información al comercial, que será quién la introduzca en el sistema
      - En el paso 2, el cliente proporciona un identificador de catálogo que será utilizado para obtener la información de la cuenta del cliente de la base de datos. El comercial verifica la información con el cliente
      - En el paso 7, la dirección de envío no hay que verificarla
  - **Realizar pedido web**
    - Este caso de uso es igual al de **Realizar pedido** excepto en que
      - El paso 2 se suprime
      - En el paso 3, el cliente selecciona los productos navegando por un catálogo en línea en lugar de introducir los códigos de productos
      - En el paso 4.1, el sistema muestra la información, y el cliente elige añadir un artículo a la carta de compras
      - En el paso 4.2, el total se asocia a la carta de compras
      - En los pasos 5 y 6, el sistema se identifica en el sistema, y este proporciona la dirección de envío y la información de pago para la cuenta del cliente
      - En el paso 7, la información de envío no se tiene que verificar

https://bit.ly/3pT7tRW

9. Aportaciones principales del tema

Aportaciones principales (i)

- Un requisito es una declaración de un servicio o una restricción de un sistema
- La ingeniería de requisitos es el proceso implicado en el desarrollo de los requisitos del sistema. Proceso de descubrir, analizar, documentar y verificar los servicios y restricciones del sistema
- Una especificación de requisitos del software (ERS) es la declaración formalizada de los requisitos de un sistema
- Un *stakeholder* es cualquier persona afectada o involucrada en el sistema de alguna forma
- La gestión de requisitos es el proceso implicado en la gestión de cambios de los requisitos

Aportaciones principales (ii)

- El producto es el "sistema" a "entregar"
  - Normalmente consiste en *software* y en algunas ocasiones también en *hardware*
- El dominio es el producto, sus usuarios directos y otros "elementos" del entorno
  - Dominio (interno): El producto, los usuarios, sus actividades y los sistemas con los que el producto ha de comunicarse
  - Dominio (externo): Usuarios de segundo nivel, se comunican con los usuarios de primer nivel
- El término sistema se suele utilizar como sinónimo del producto a entregar más el hardware que lo soporta. Desde un punto de vista abstracto el dominio también "pertenece" al sistema
  - Estas consideraciones suelen dar lugar a la distinción entre requisitos de sistema y requisitos *software*

Aportaciones principales (iii)

- El concepto de requisito se ve calificado en función de distintos aspectos ortogonales (su ámbito, característica definida por el requisito, su audiencia, y su representación)
- Los requisitos de sistema se suelen clasificar en funcionales y no funcionales
- Los requisitos funcionales son afirmaciones sobre los servicios que el sistema debe ofrecer
- Los requisitos no funcionales son restricciones sobre los servicios o las funciones que el sistema ofrece
- Los requisitos no funcionales suelen ser requisitos más críticos que los requisitos funcionales
- El documento de requisitos del *software* es la declaración de acuerdo de los requisitos del sistema a construir
  - Debe estar organizar de tal forma que pueda ser utilizado tanto por los clientes del sistema como por los ingenieros del software

Aportaciones principales (iv)

- Los casos de uso
  - Como elementos fundamentales en la ingeniería de requisitos
  - Como directores del proceso *software*
- Difusión a través de su inclusión en UML
- Un caso de uso describe una interacción con los actores como consecuencia de mensajes entre el sistema y uno o más actores
  - Se considera al sistema como una caja negra
  - Los actores obtienen resultados observables
- Relaciones entre casos de uso para factorizar el comportamiento común y las variantes
- Los casos de uso representan una vista funcional del sistema, además sirven para facilitar la determinación y documentación de los requisitos funcionales del sistema *software* a construir
  - Es necesario el paso a la aproximación mediante objetos

Aportaciones principales (v)

- Se ha de estar seguro de que cada caso de uso describe una porción significativa de la utilización del sistema que sea comprensible por los expertos del dominio y los programadores
- Cuando se definan los casos de uso de forma textual, utilizar nombres y verbos que sean precisos y consistentes para que ayuden en la obtención de los objetos y de los mensajes que se utilizarán en los diagramas de interacción
- Un diagrama de casos de uso debe
  - Contener solamente casos de uso del mismo nivel de abstracción
  - Incluir solamente los actores requeridos
- Cuando se tiene un gran número de casos de uso, estos han de organizarse en paquetes

Aportaciones principales (vi)

- Resumen de la documentación de un caso de uso
  - **Nombre del caso de uso**
  - **Actores**
    - Descripción de los actores implicados en el caso de uso
  - **Inicio del caso de uso (condición de entrada)**
    - Utilizar una frase del tipo: " El caso de uso se inicia cuando..."
  - **Flujo de eventos**
    - Formato libre, sentencias del lenguaje natural
  - **Condición de salida**
    - Frase del tipo " El caso de uso finaliza cuando..."
  - **Excepciones**
    - Describir qué pasa si algo va mal
  - **Requisitos especiales**
    - Lista de los requisitos no funcionales y de las restricciones

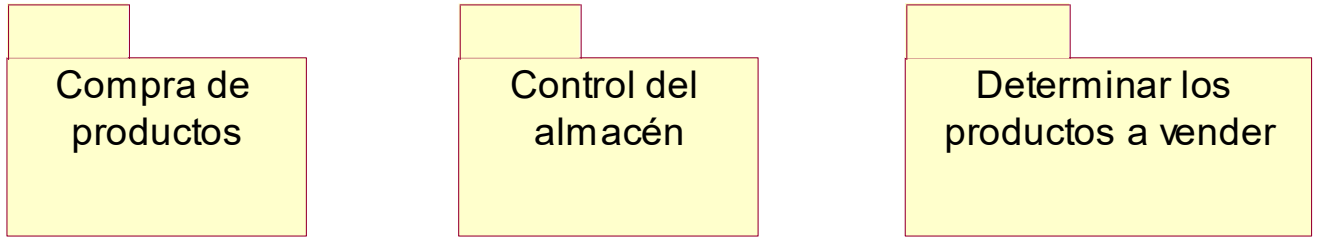
Aportaciones principales (vii)

- La construcción de un caso de uso es un proceso iterativo en el que se llevan a cabo las siguientes tareas [Booch et al., 1999]
  - Identificación de los actores principales
  - Reorganización de los actores identificando tanto los roles más generales como los más especializados
  - Identificación de las formas más importantes que tiene cada actor de interaccionar con el elemento a modelar
  - Se deben considerar las formas excepcionales en las que cada actor puede interactuar con el elemento
  - Organización de los comportamientos como casos de uso
  - Conforme crecen los modelos, los casos de uso tienden a juntarse en grupos relacionados conceptual y semánticamente

10. Cuestiones y ejercicios

Ejercicio resuelto (i)

- Crear un diagrama de casos de uso para una tienda en línea. Centrarse en las siguientes funciones
  - Compra de productos
  - Control del almacén
  - Determinar los productos a vender



Ejercicio resuelto (ii)

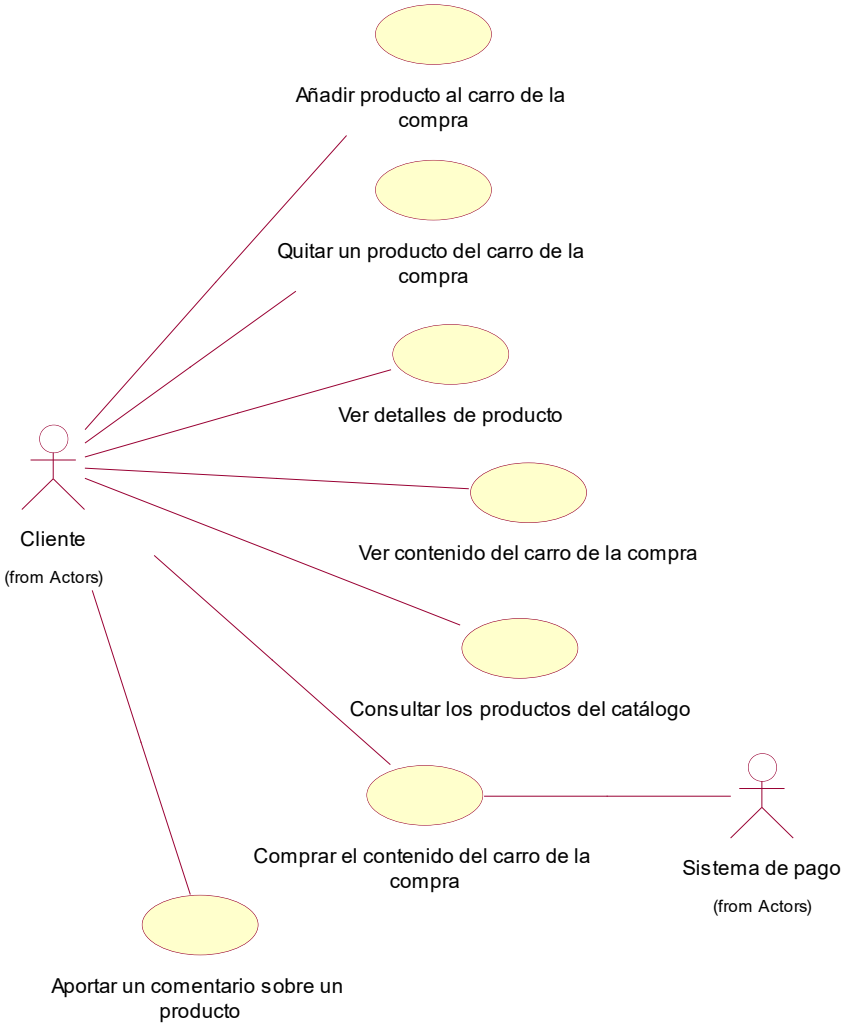


Diagrama Compra de productos

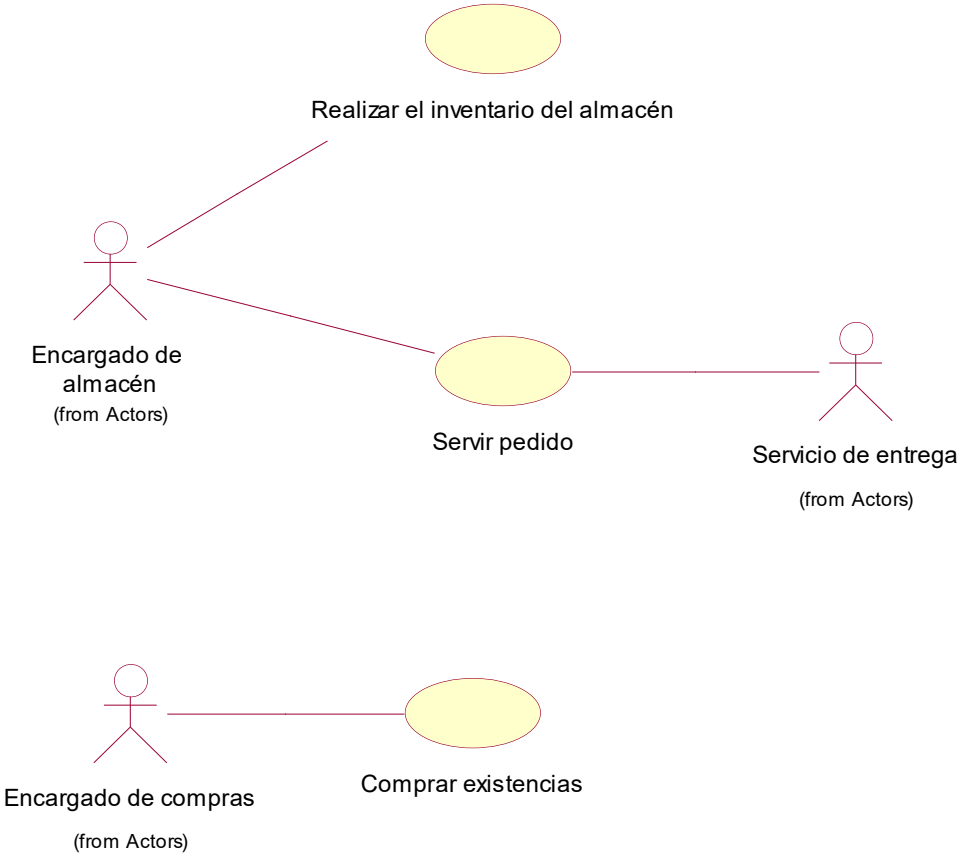


Diagrama Control del almacén

Ejercicio resuelto (iii)

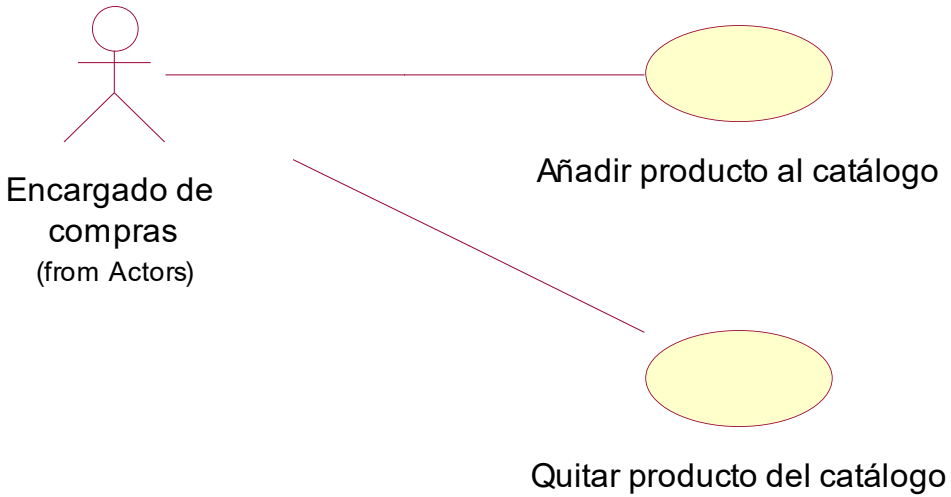
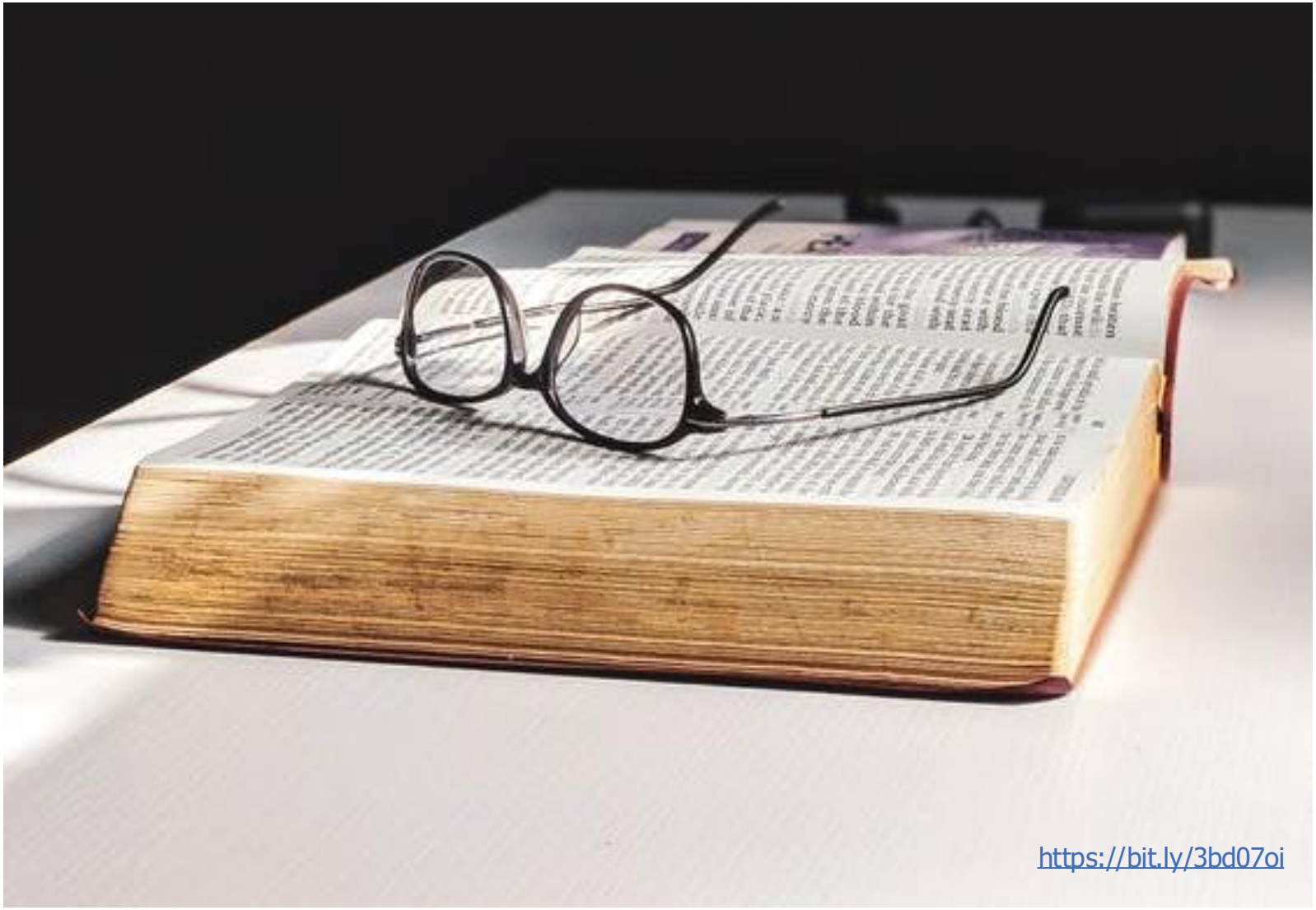


Diagrama Determinar los productos a vender

Cuestiones y ejercicios

- Discutir el problema de utilizar lenguaje natural para la definición de los requisitos
- Completar el ejercicio de la tienda en línea incluyendo relaciones entre casos de uso
- Realizar la descripción narrativa de los casos de uso del ejercicio de la tienda en línea
- Realizar el catálogo de requisitos de un vídeo club, utilizando casos de uso para ello
- Realizar el catálogo de requisitos de un sistema de seguimiento de cursos para una organización, utilizando casos de uso para ello
- Utilizar la experiencia personal sobre el uso de cajeros automáticos para desarrollar los casos de uso derivados de los requisitos de uno de estos sistemas



11. Lecturas complementarias

Lecturas complementarias (i)

- S. Blake, "*Use Cases vs. User Stories: How They Differ and When to Use Them*". En: EasyAgile. 2021. Disponible en: https://d66z.short.gy/WdX6m8
  - Comparativa entre los casos de uso y las historias de usuario
- A. Casamayor, D. Godoy y M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology,* vol. 52, no. 4, pp. 436-445, 2010. doi: 10.1016/j.infsof.2009.10.010
  - Artículo que aboga por la detección temprana de los requisitos no funcionales utilizando técnicas automáticas
- C. Ebert, "Putting requirement management into praxis: Dealing with nonfunctional requirements," *Information and Software Technology,* vol. 40, no. 3, pp. 175-185, 1998. doi: 10.1016/S0950-5849(98)00049-4
  - Artículo que se centra en la gestión práctica de los requisitos no funcionales
- D. J. Grimshaw y G. W. Draper, "Non-functional requirements analysis: deficiencies in structured methods," *Information and Software Technology,* vol. 43, no. 11, pp. 629-634, 2001. doi: 10.1016/S0950-5849(01)00171-9
  - Artículo que examina las deficiencias de los métodos estructurados a la hora de gestionar los requisitos no funcionales

Lecturas complementarias (ii)

- A. M. Hickey y A. M. Davis, "The Role of Requirements Elicitation Techniques in Achieving Software Quality," presentado en Eighth International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ'2002 (September 09-10th, 2002), Essen, Germany, 2002
  - Artículo que pone de manifiesto la relación existente entre las técnicas de obtención de requisitos y la calidad del producto resultante
- L. A. Maciaszek, Requirements analysis and system design: Developing information systems with UML. Essex, UK: Addison-Wesley Longman Ltd., 2001
  - Cabe destacar los capítulos 3 "*Requirements Determination*" y 4 "*Requirements Specification*"
- N. Power, "Variety and quality in requirements documentation," presentado en Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ'2001 (June 4-5, 2001), Interlaken, Switzerland, 2001
  - Artículo que discute diferentes formas en las que los documentos de requisitos varían sus contenidos, sus propósitos o sus formas de uso

12. Referencias

https://unsplash.com/photos/tZzYZzO_CX0

Referencias (i)

[Booch et al., 1999] Booch, G., Rumbaugh, J., Jacobson, I. "*The Unified Modeling Language User Guide*". Object Technology Series. Addison-Wesley, 1999

[Brackett, 1990] Brackett, J. W. "*Software Requirements*". SEI Curriculum Module SEI-CM-19-1.2. Software Engineering Institute. Carnegie Mellon University, Pittsburgh, PA 15213 (USA). 1990

[Brooks, 1995] Brooks, F. P. Jr. "The Mythical Man–Month: Essays on Software Engineering". Anniversary Edition. Addison–Wesley, 1995

[Bruegge y Dutoit, 2000] Bruegge, B., Dutoit, A. "*Object-Oriented Software Engineering*". Prentice-Hall, 2000

[Cockburn, 1997a] Cockburn, A. "*Structuring Use Cases with Goals. Part I*". Journal of Object-Oriented, A. Programming (JOOP), 10(5):35-40. September 1997

[Cockburn, 1997b] Cockburn, A. "*Structuring Use Cases with Goals. Part II*". Journal of Object-Oriented Programming (JOOP), 10(7). November/December 1997

[Cockburn, 2000] Cockburn, A. "*Writing Effective Use Cases*". Addison-Wesley, 2000

[Christel y Kang 1992] Christel, M. G., Kang, K. C. "*Issues in Requirements Elicitation*". Technical Report CMU/SEI-92-TR-12 (ESC-TR-92-012). Software Engineering Institute. Carnegie Mellon University, Pittsburgh, PA 15213 (USA). September 1992

[Davis, 1993] Davis, A. M. "*Software Requirements: Objects, Functions and States*". 2nd Edition. Prentice–Hall, 1993

[DoD, 1994] DoD. "*Military Standard 498: Software Development and Documentation*". Department of Defense of the United States of America. 1994

[Durán, 2000] Durán Toro, A. "*Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*". Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla. Septiembre de 2000

Referencias (ii)

[Durán et al., 2002] Durán, A., Ruiz, A., Corchuelo, R., Toro, M. "*Supporting Requirements Verification Using XSLT*". En Proceedings of the IEEE Joint International Conference on Requirements Engineering. Essen, Alemania. 2002

[Durán y Bernárdez, 2002] Durán, A., Bernárdez, B. "*Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3)*". Informe Técnico LSI-2000-10, Universidad de Sevilla. https://d66z.short.gy/gEMQlZ [Última vez visitado, 9-2-2026]. Abril 2002

[Hsia et al., 1993] Hsia, P., Davis, A., Kung, D. "*Status Report: Requirements Engineering*". IEEE Software, 10(6):75-79. 1993

[IEEE, 1999a] IEEE. "*IEEE Software Engineering Standards Collection 1999 Edition. Volume 1: Customer and Terminology Standards*". IEEE Computer Society Press, 1999

[IEEE, 1999b] IEEE. "*IEEE Software Engineering Standards Collection 1999 Edition. Volume 4: Resource and Technique Standards*". IEEE Computer Society Press, 1999

[Jacobson, 1987] Jacobson, I. "*Object Oriented Development in an Industrial Environment*". En Proceedings of the 1987 OOPSLA - Conference proceedings on Object-Oriented Programming Systems, Languages and Applications. (October 4-8, 1987, Orlando, FL USA). Pages 183-191. ACM, 1987

[Jacobson et al., 1992] Jacobson, I., Christerson, M., Jonsson, P., Övergaad, G. "*Object Oriented Software Engineering: A Use Case Driven Approach*". Addison-Wesley, 1992

[Jacobson et al., 1999] Jacobson, I., Booch, G., Rumbaugh, J. "*The Unified Software Development Process*". Object Technology Series. Addison-Wesley, 1999

Referencias (iii)

[Johnson, 1995] Johnson, J. "*Chaos: The Dollar Drain of IT Project Failures*". Application Development Trends, 2(1):41-47. 1995

[Kotonya y Sommerville, 1998] Kotonya, G., Sommerville, I. "Requirements Engineering: Processes and Techniques". John Wiley and Sons, 1998

[Larman, 2002] Larman, C. "*Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*". 2nd Ed. Prentice Hall, 2002

[Loucopoulus y Karakostas, 1995] Loucopoulus, P., Karakostas, V. "*System Requirements Engineering*". McGraw-Hill, 1995

[Mazza et al., 1994] Mazza, C., Fairclough, J., Melton, B., Pablo, D. de, Scheffer, A., Stevens, R. "*Software Engineering Standards*". Prentice–Hall, 1994

[Muller, 1997] Muller, P.-A. "*Modélisation Objet avec UML*". Eyrolles, 1997

[OMG, 2017] OMG. "*OMG Unified Modeling Language Specification. Version 2.5.1*" Object Management Group Inc. December 2017. https://www.omg.org/spec/UML/2.5.1/ [Última vez visitado, 9-2-2026]

[Piattini et al., 1996] Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. "*Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*". Ra-ma, 1996

[Pohl, 1997] Pohl, K. "*Requirements Engineering: An Overview*". En A. Kent y J. Williams, (Eds.), *Encyclopedia of Computer Science and Technology*, 36 (suppl. 21). New York, USA: Marcel Dekker, 1997. Disponible en: https://d66z.short.gy/v5XTqK [Última vez visitado, 9-2-2026]

[Pressman, 2002] Pressman, R. S. "*Ingeniería del Software: Un Enfoque Práctico*". 5ª Edición. McGraw-Hill. 2002

[RAE, 2025] Real Academia Española "*Diccionario de la Lengua Española*". Versión electrónica 23.8.1. http://www.rae.es. [Última vez visitado, 9-2-2026]. 2025

Referencias (iv)

[Regnal et al., 1996] Regnal, B., Anderson, M., Bergstand, J. "*A Hierarchical Use Case Model with Graphical Representations*". En Proceedings of ECBS'96, IEEE Computer Society Press, New York 1996

[Reifer, 1994] Reifer, D. J. "*Requirements Engineering*". En Marciniak, J. J. (Ed.) Encyclopedia of Software Engineering. Páginas 1043-1054. Wiley, 1994

[Rombach , 1990] Rombach, H. D. "*Software Specifications: A Framework*". Curriculum Module SEI–CM–11–2.1, Software Engineering Institute, Carnegie Mellon University. 1990

[Rumbaugh et al., 1999] Rumbaugh, J., Jacobson, I., Booch, G. "*The Unified Modeling Language. Reference Manual*". Addison-Wesley, Object Technology Series, 1999

[Rumbaugh et al., 2005] Rumbaugh, J., Jacobson, I., Booch, G. "*The Unified Modeling Language. Reference Manual*". 2nd ed. Pearson-Addison-Wesley, Object Technology Series, 2005

[Sawyer y Kotonya, 2001] Sawyer, P., Kotonya, G. "*Software Requirements*". Capítulo 2 en Abran, A. (Co-Excutive Editor), Moore, J. W. (Co-Excutive Editor), Bourque, P. (Editor), Dupuis, R. (Editor), Tripp, L. L. (Chair), Guide to the Software Engineering Body of Knowledge SWEBOK. IEEE-CS Press, 2001

[Schneider y Winters, 2001] Schneider, G., Winters, J. P. "*Applying Use Cases: A Practical Guide*". 2nd Ed. Addison-Wesley, Object Technology Series, 2001

[Sommerville, 2002] Sommerville, I. "*Ingeniería del Software*". 6ª Edición, Addison-Wesley. 2002

[Wieringa, 1996] Wieringa, R. J. " *Requirements Engineering: Frameworks for Understanding*". John Wiley & Sons, 1996

INGENIERÍA DE SOFTWARE I

Tema 4: Ingeniería de Requisitos

Grado en Ingeniería Informática
Fecha de última modificación: 9-2-2026

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es
Dr. Miguel Ángel Conde González / mconde@usal.es

Departamento de Informática y Automática
Universidad de Salamanca