Ingeniería de Requisitos

Tema 5. Especificación de Requisitos

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

Materiales cedidos para su uso docente por el prof. Emilio Insfran Pelozo. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia

Contenido

- Introducción
- Tipos de Especificación de Requisitos
  - informal
  - semi-formal
  - formal
- Modelos de Análisis
  - Modelo del Dominio
  - Modelo del Negocio
- Casos de Uso
- Detalle de Casos de Uso
  - Guiones
  - Interacciones

Introducción

- Siguiendo un proceso en espiral de IDR, el resultado de las fases de *Elicitación* e *Identificación* de requisitos debe ser especificado de forma precisa

  Especificación de Requisitos:

  – **Entrada**: sentencias acordadas/negociadas de diversos tipos:
    - objetivos
    - requisitos del sistema y del software de alto nivel
    - asunciones sobre el entorno/organización
    - propiedades relevantes del dominio
    - información sobre las tareas del negocio
    - conceptos y definiciones

  – **Salida**: primera versión de los documentos de ERS organizados de acuerdo a una estructura coherente de forma a cumplir las características deseables de una buena ERS:

    No ambigua, Completa, Fácil de verificar, Consistente, Fácil de modificar, Trazable, Identificable, Categorizada, Fácil de utilizar, etc.

Contenido

- Introducción
- Tipos de Especificación de Requisitos
  - informal
  - semi-formal
  - formal
- Modelos de Análisis
  - Modelo del Dominio
  - Modelo del Negocio
- Casos de Uso
- Detalle de Casos de Uso
  - Guiones
  - Interacciones

Tipos de Especificación

- Cada *sentencia* de la ERS (de acuerdo a su categoría) debe ser especificado de forma precisa en un *lenguaje de especificación* apropiado
- Este lenguaje debe ser apropiado para facilitar la *comunicación* con los *stakeholders* y con los *desarrolladores*
- Existen diferentes técnicas y lenguajes de especificación:
  - informales
  - semi-formales
  - formales

Tipos de Especificación (Informal)
Documentación Libre en Lenguaje Natural

- Sentencias en prosa
- Es la opción “más sencilla” y no tiene limitaciones de expresividad
- No tiene barreras de comunicación
  - el lenguaje natural puede ser entendido por stakeholders y desarrolladores
- No necesita entrenamiento especial
- Sin embargo, es *propenso* a una serie características no deseables:
  - **ambigüedad**: afirmaciones con varias interpretaciones
  - **ruido**: afirmaciones no relacionadas al dominio del problema
  - **estructuración pobre**: sentencias no organizadas
  - **modificabilidad pobre**: sentencias cuya modificación se propaga globalmente en la ERS
  - **referencias hacia delante**: referencias a aspectos aun no definidos
  - **identificabilidad pobre**: dificultad para localizar información específica
  - etc.

Tipos de Especificación (Informal)
Documentación Disciplinada en Lenguaje Natural Estructurado

- Una documentación *disciplinada* con *lenguaje natural estructurado*, debe seguir unas reglas locales y globales:

  - **Reglas locales**: cómo se escriben las sentencias en lenguaje natural, por ejemplo:
    - guías de estilo para sentencias
    - plantillas predefinidas para sentencias

  - **Reglas globales**: cómo el documento de requisitos debe ser estructurado, por ejemplo:
    - reglas de agrupación
    - plantillas de documentos de requisitos

Tipos de Especificación (Informal)
Documentación Disciplinada en Lenguaje Natural Estructurado

Reglas locales (1/2)

- Guías de estilo para sentencias
  - Principios estándar de escritura técnica adaptada a documentos de requisitos:
    - No incluir más de UN requisito, asunción o propiedad por cada sentencia
    - Mantener las sentencias cortas
    - Evitar uso innecesario de jerga y acrónimos
    - Usar listas de viñetas para detallar propiedades/características de sentencias
    - Usar diagramas para expresar relaciones complejas entre conceptos
    - Usar figuras para visualizar conceptos y enfatizar puntos importantes
    - Usar tablas
    - Usar ecuaciones para relacionar información cuantitativa
    - etc.

Tipos de Especificación (Informal)
Documentación Disciplinada en Lenguaje Natural Estructurado

Reglas locales (2/2)

- Plantillas predefinidas para sentencias
  - Las plantillas pueden ayudar para presentar distintos tipos de sentencias de una forma estandarizada y para gestionar trazabilidad
  - Una plantilla para sentencias provee campos como:
    - **identificador**: referencia única a la sentencia en toda la ERS
    - **categoría**: tipo de sentencia, por ej. requisito funcional o no funcional
    - **especificación**: es la especificación de la sentencia (siguiendo unas guías de estilo para sentencias)
    - **fuente**: origen de la sentencia (un stakeholder, un documento, etc.)
    - **prioridad**: para compararlo con otras sentencias
    - **estabilidad**: para la gestión del cambio
    - etc.

Tipos de Especificación (Informal)
Documentación Disciplinada en Lenguaje Natural Estructurado

Reglas globales (1/2)

- Reglas de agrupación
  - Utilizadas para aumentar la cohesión de la ERS
  - Definir secciones en la ERS para agrupar sentencias consideradas similares:
    - objetivos del sistema
    - requisitos funcionales
    - requisitos no funcionales
    - etc.

Tipos de Especificación (Informal)
Documentación Disciplinada en Lenguaje

Reglas globales (2/2)

- Plantillas de documentos de requisitos
  - Existen varios estándares que definen plantillas para organizar las sentencias en una ERS: IEEE 830, ESA PSS-05, VOLERE, etc.

Table of Contents

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, acronyms, and abbreviations
   - 1.4 References
   - 1.5 Overview
2. Overall description
   - 2.1 Product perspective
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 Constraints
   - 2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

Appendixes

Index

1 INTRODUCTION
- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms and abbreviations
- 1.4 References
- 1.5 Overview

2 GENERAL DESCRIPTION
- 2.1 Relation to current projects
- 2.2 Relation to predecessor and successor projects
- 2.3 Function and purpose
- 2.4 Environmental considerations
- 2.5 Relation to other systems
- 2.6 General constraints
- 2.7 Model description

3 SPECIFIC REQUIREMENTS
(The subsections may be regrouped around high-level functions)
- 3.1 Functional requirements
- 3.2 Performance requirements
- 3.3 Interface requirements
- 3.4 Operational requirements
- 3.5 Resource requirements
- 3.6 Verification requirements
- 3.7 Acceptance testing requirements
- 3.8 Documentation requirements
- 3.9 Security requirements
- 3.10 Portability requirements
- 3.11 Quality requirements
- 3.12 Reliability requirements
- 3.13 Maintainability requirements
- 3.14 Safety requirements

4 REQUIREMENTS TRACEABILITY MATRIX

Tipos de Especificación (Semi-formal)

- Lenguajes y técnicas de especificación semi-formales son usados para *sustituir* o *complementar* el lenguaje natural
- Habitualmente son modelos/diagramas o notaciones específicas:
  - CU, Diagramas de Actividad, Workflows, URN (User Requirements Language), MSC (Message Sequence Chart – graphical/textual),…
- *Semi-formal* significa:
  - elementos en consideración y sus relaciones son declarados formalmente
  - sentencias que describen o prescriben sus propiedades son especificados en lenguaje natural
- El aspecto *formal* de las técnicas semi-formales se refiere a:
  - "procesable" por el ordenador
  - *sintaxis* y *semántica* bien definida
    - *sintaxis concreta* habitualmente *gráfica* para facilitar la comunicación

Tipos de Especificación (Formal)

- Una especificación ***semi-formal*** declara algunas partes de la ERS formalmente pero deja sentencias descriptivas y prescriptivas en LN
- En una **especificación formal**, todas las sentencias de la ERS corresponden a un lenguaje formal
  - alto grado de precisión en la formulación de sentencias
  - reglas precisas de interpretación e inferencia
  - mecanismos avanzados de verificación y validación
    - model checking, animación, demostración de propiedades, etc.
- Especialmente importantes en **sistemas altamente críticos** para ayudar a asegurar especificaciones libres de errores
- Lenguajes de especificación formal:
  - Z, VDM, Prolog, OBJ, CSP, CCS, Maude, …

Contenido

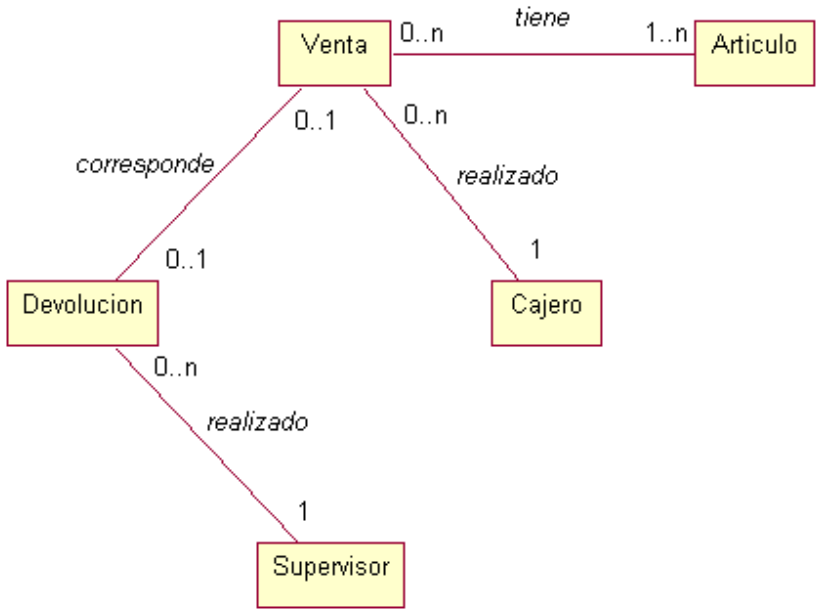
- Introducción
- Tipos de Especificación de Requisitos
  - informal
  - semi-formal
  - formal
- Modelos de Análisis
  - Modelo del Dominio
  - Modelo del Negocio
- Casos de Uso
- Detalle de Casos de Uso
  - Guiones
  - Interacciones

Modelos de Análisis

- Modelo de Dominio
  - Identificar y nombrar *conceptos* importantes: *entidades* que se deben manipular o sobre los que se registra información y *eventos* que ocurren en el contexto del sistema (ej. artículo, venta, devolución)
  - Identificar y nombrar *relaciones* entre conceptos
    - Relaciones (simples) y multiplicidades
  - Notación: Diagrama de Clases (UML)
  - Origen: Glosario, Requisitos de Información, Stakeholders, Experto,…
- Modelo del Negocio (estático) Modelo del Proceso del Negocio (dinámico)
  - Procesos / Comportamiento (*workflows*)
  - Trabajadores, responsabilidades y operaciones (*tareas*)
  - Objetivos, metas, … (*i**)
- Decidir cuándo/cómo construir un modelo del negocio / modelo del proceso del negocio, o un modelo de dominio.

Modelos de Análisis

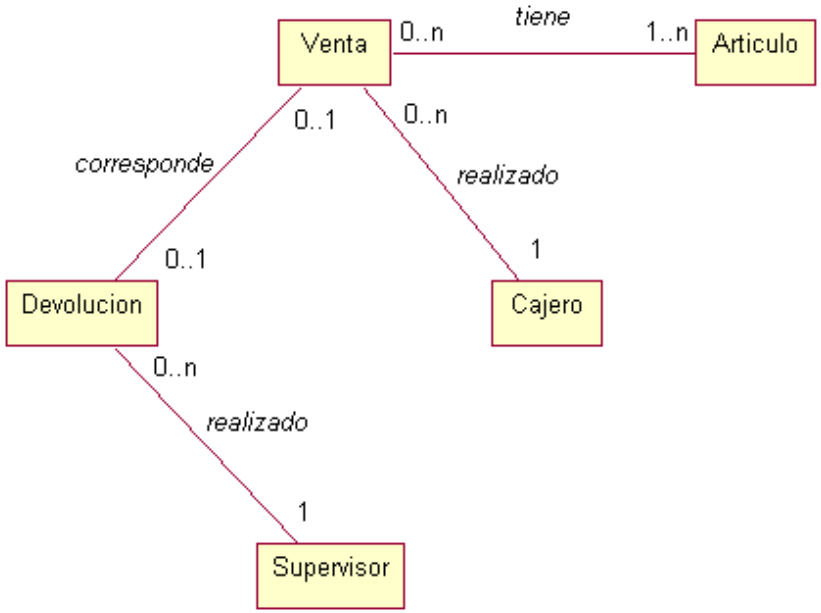
- Modelo de Dominio
  - Capturados como clases en un Diagrama de Clases
  - Establecen un *lenguaje común* entre clientes y desarrolladores
  - Se centran en el modelado del dominio
  - Posteriormente, en sucesivas iteraciones, puede diferir con elementos del modelado del sistema a nivel de análisis y diseño o implementación: factorización, especialización/generalización, …
  - Ej. *Terminal de Punto de Venta:*
    - **Artículo**
    - **Venta**
    - **Cajero**
    - **Devolución**
    - **Supervisor**



Modelos de Análisis

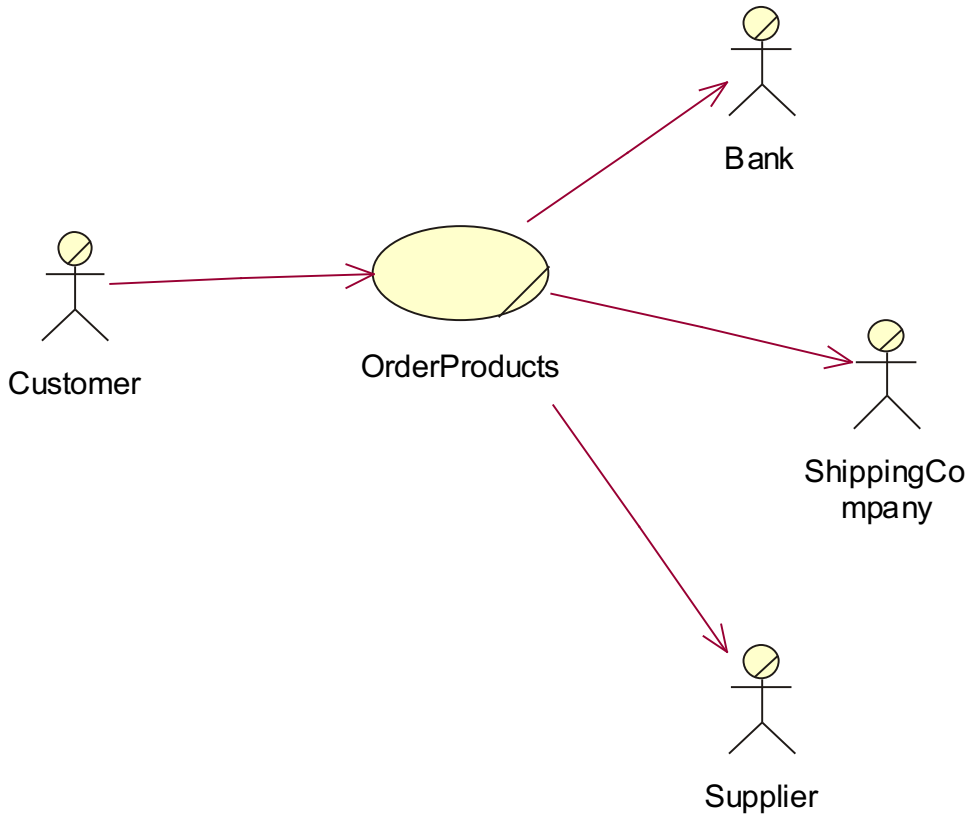
- Del Modelo de Dominio al Modelo de Análisis
  - Habitualmente, a medida que aumenta la información del dominio del problema, el modelo de dominio *evoluciona* a un modelo de análisis/diseño:
    - **Factorización**
    - **Generalización**
    - **Especialización**
    - **Asociación/Agregación/Composición**
    - **Atributos**
    - **Servicios**
    - **Paquetes y Dependencias**
    - **…**

Ejemplo: *Terminal de Punto de Venta*



Modelo del Negocio. Casos de Uso de Negocio

- **Modelo de CU del negocio**: describe el contexto de operación del negocio
- **Actor del negocio**: entidades externas a la organización (o área) estudiada. Entidades internas (usuarios) no se identifican como actores
- **Caso de uso del negocio**: describe procesos del negocio (tareas realizadas en la organización) que proveen valor a un actor de negocio

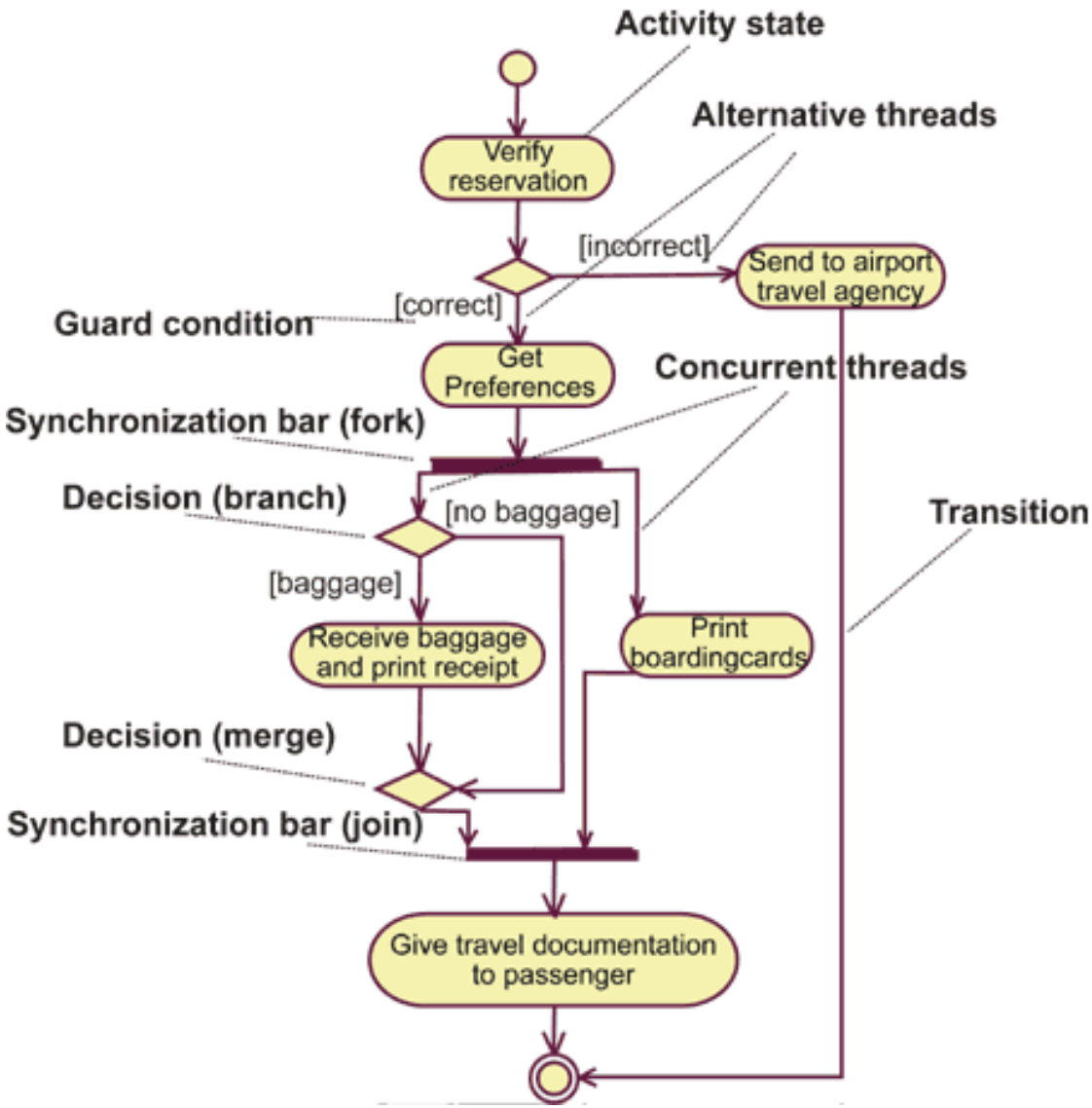


Modelo del Negocio. Workflows

- Describe ***QUÉ*** hace la organización para proveer un servicio a un actor y no ***CÓMO*** se va a solucionar una necesidad

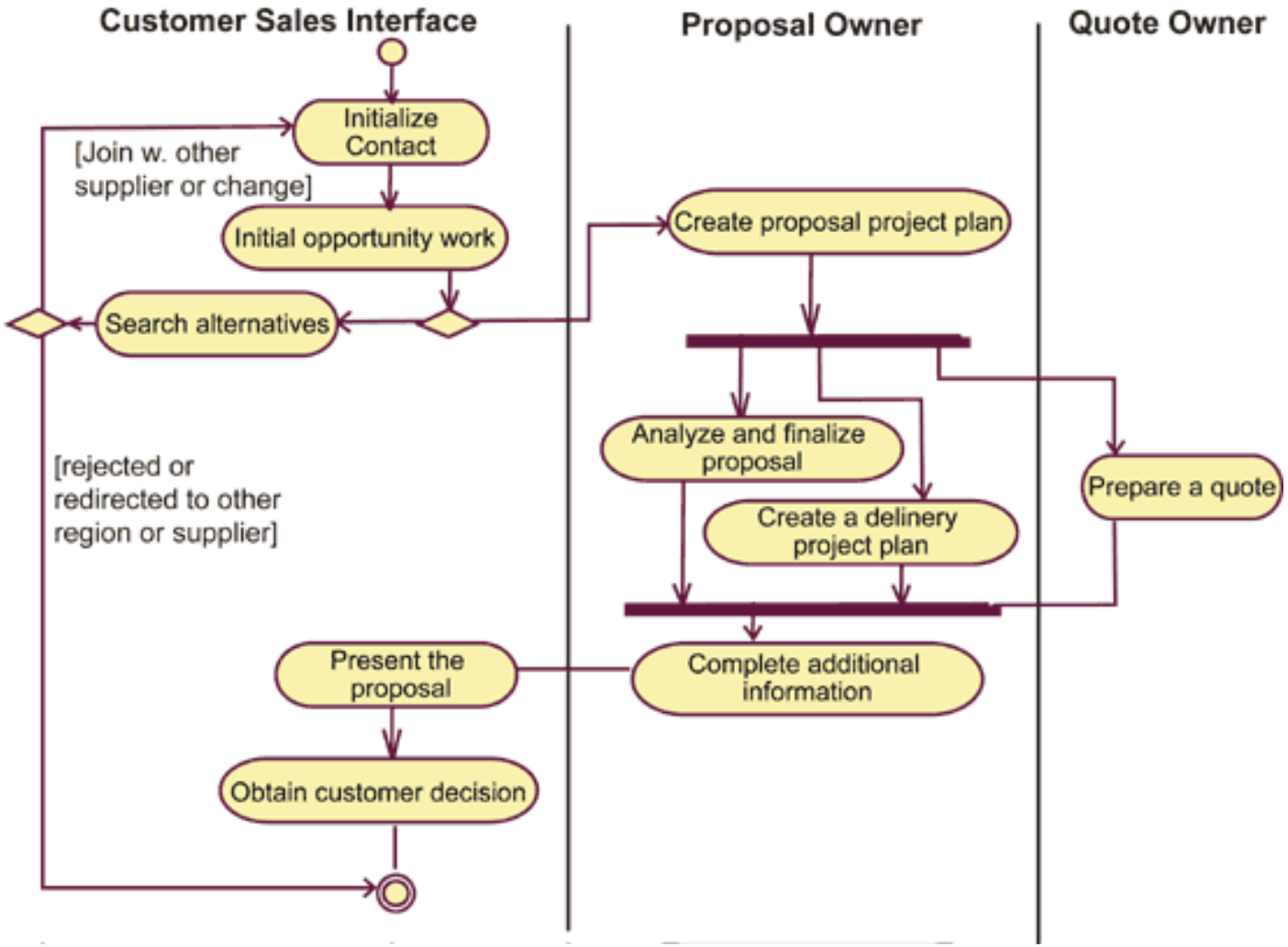
- **Ejemplo**:

  Workflow para el Caso de Uso de negocio: *Emitir billete de avión (Check-in)*



Modelo del Negocio. Workflows

- Otro ejemplo de workflow con *calles* (swimlines) para indicar los actores del dominio responsables para realizar las actividades



Contenido

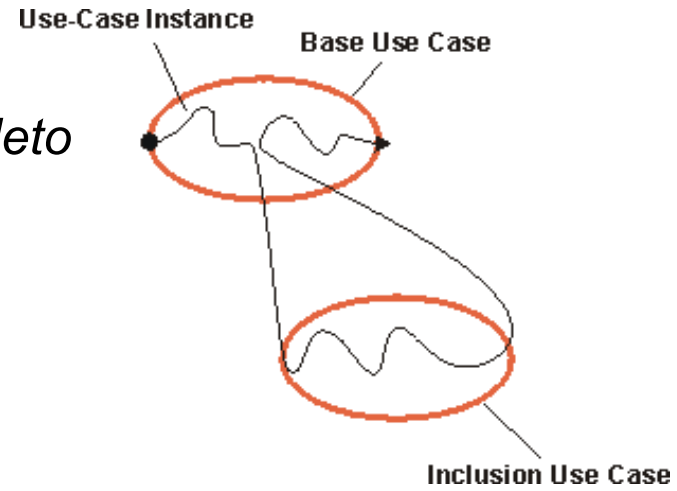
- Introducción
- Tipos de Especificación de Requisitos
  - informal
  - semi-formal
  - formal
- Modelos de Análisis
  - Modelo del Dominio
  - Modelo del Negocio
- Casos de Uso
- Detalle de Casos de Uso
  - Guiones
  - Interacciones

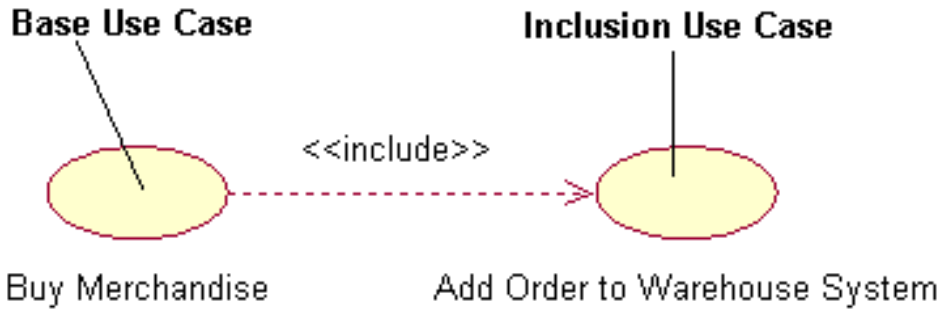
Conceptos de Casos de Uso

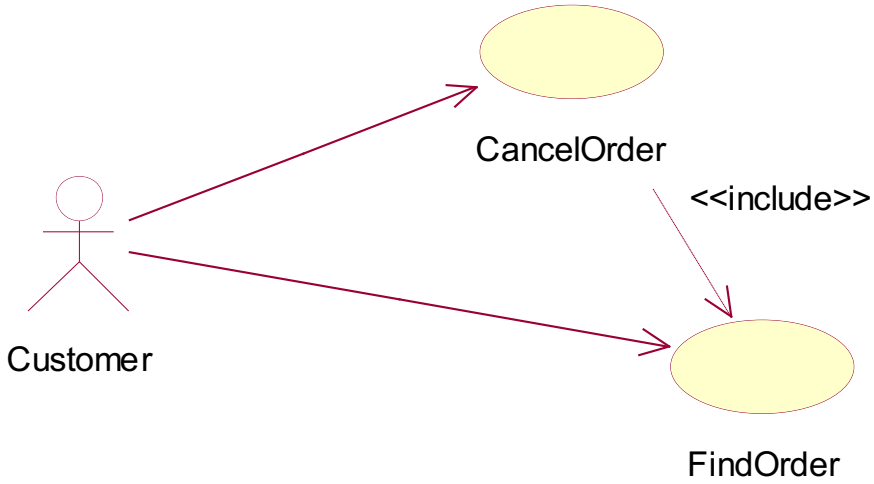
- Actor
- Relaciones entre actores
- Casos de Uso
- Comunicación Actor – Caso de Uso
- Relaciones entre Casos de Uso
  - Include, Extends, Inheritance

Include

- La relación ***include*** se utiliza para “extraer” comportamiento común de un CU base
- El CU incluido **habitualmente no es** un CU *completo* e *independiente* (no necesariamente! ver ejemplo)
- El CU incluido **no sabe cuándo** es incluido
- Un CU puede incluir varios otros CU y se pueden incluir varios niveles de *include*

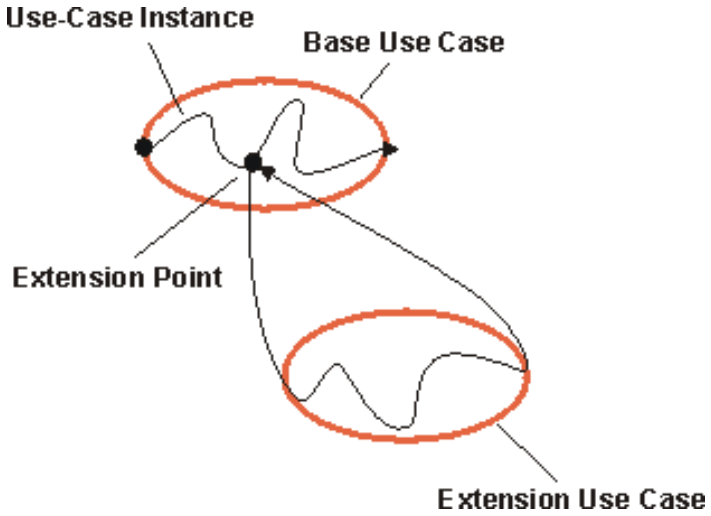


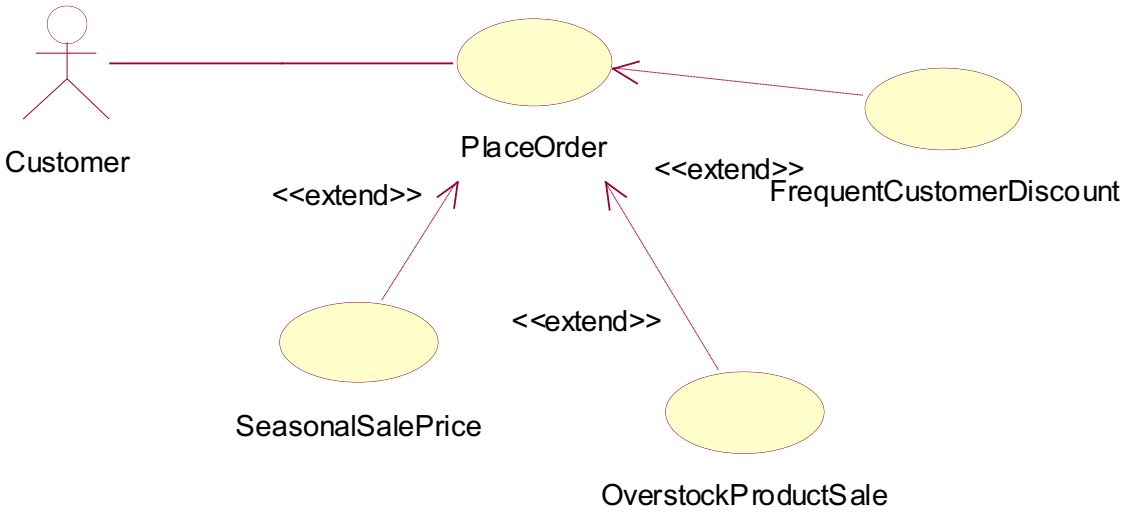




Extend

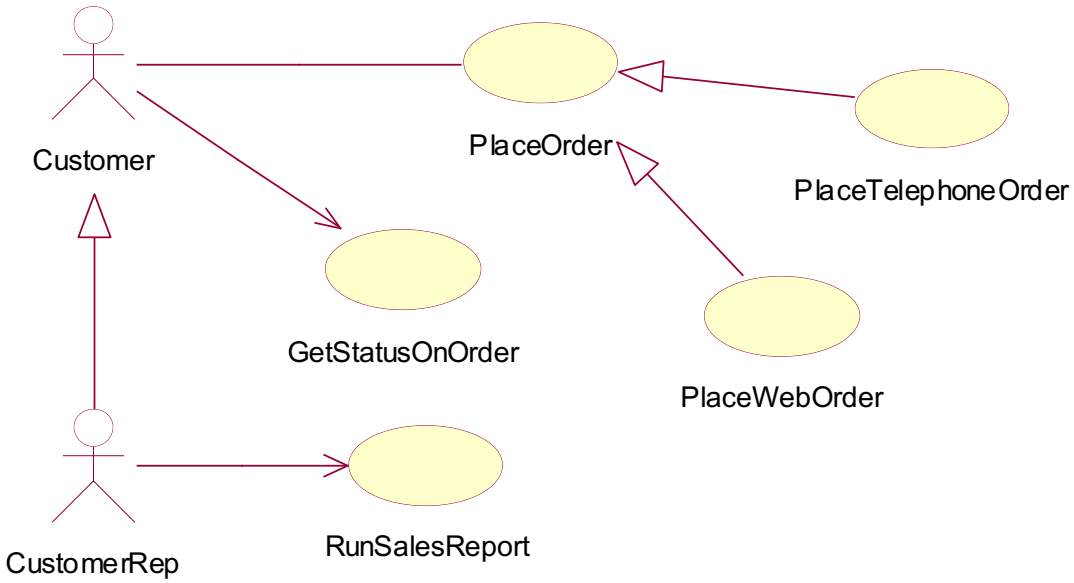
- La relación ***extend*** es usada para añadir comportamiento condicional/opcional a un CU base
- La relación ***extend*** incluye una **expresión condicional** (que pertenece a la relación y no al CU base ni al CU extensión)
  - Cuando el **punto de extensión** es alcanzado, si la condición es cierta, el CU extensión es ejecutado.

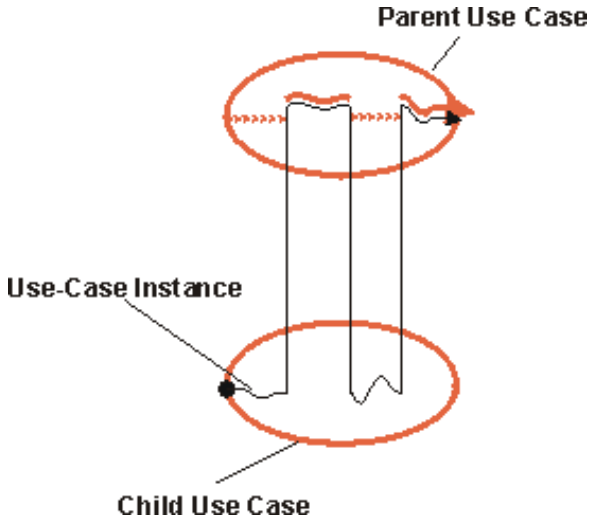




Inheritance (Herencia)

- Herencia entre actores significa que el **actor hijo** hereda todos los roles del **actor padre**
- Herencia entre CU significa que un **CU hijo** es una versión especializada del **CU padre**
  - alternativamente puede representarse con relaciones *extend*
  - pueden haber CU abstractos





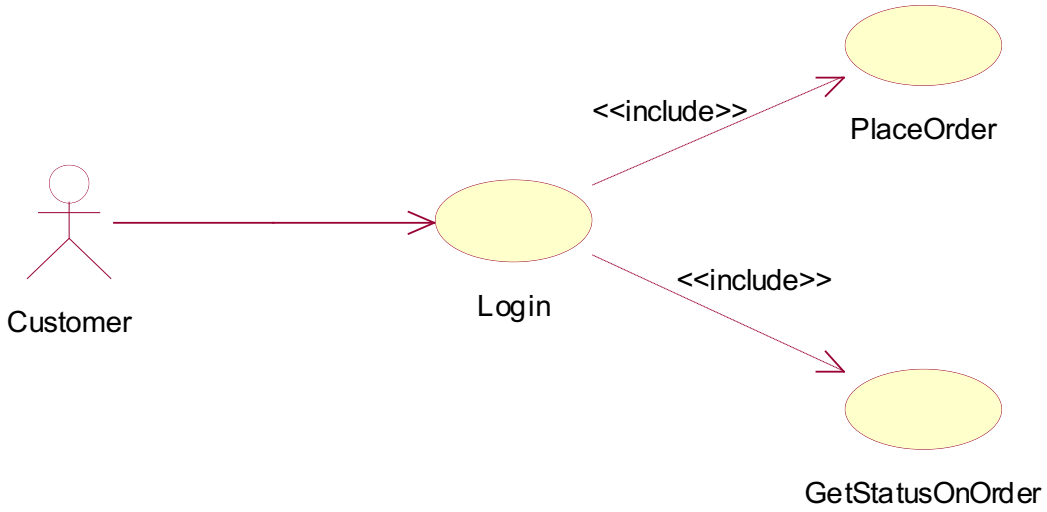
Una instancia de un CU hijo sigue el flujo de eventos del padre *insertando* o *modificando* partes de este flujo

Ejemplo: Login

- (Al menos) 4 formas de especificar el CU login
  - CU login incluyendo los otros CU
  - otros CU incluyendo el CU login
  - otros CU extendiendo el CU login
  - CU login independiente

Login Incluye al resto de Casos de Uso

- **Ventaja**: el diagrama de casos de uso tiene el aspecto esperado, desde el punto de vista del uso.
- **Desventaja**: no es una buena opción de cara al mantenimiento posterior del sistema. La significado del CU login no es apropiado.



Login Incluye al resto de Casos de Uso (cont.)

Caso de Uso Log In

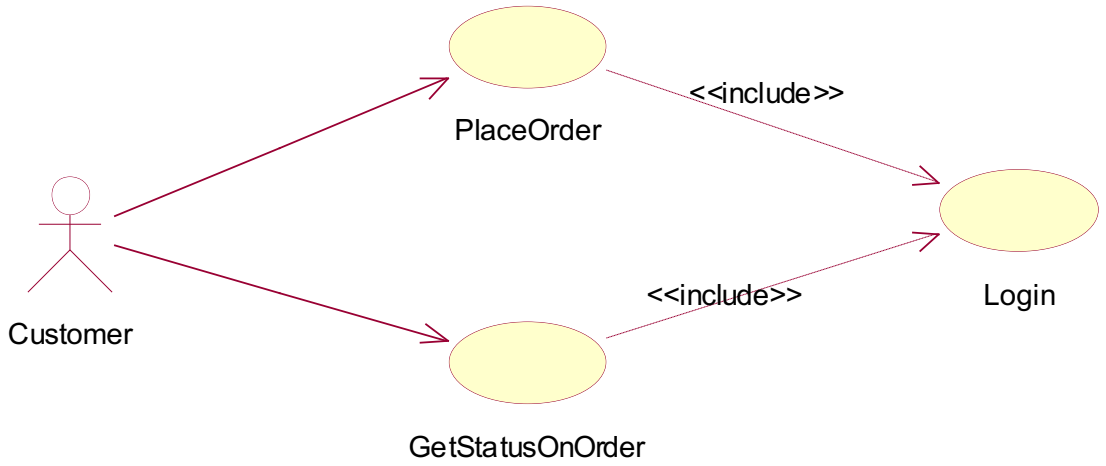
- El caso de uso comienza cuando el usuario inicia la aplicación
- El sistema solicita al cliente que introduzca su nombre de usuario y contraseña
- El cliente introduce su nombre de usuario y contraseña
- El sistema verifica que es un usuario válido
- ***Mientras*** el cliente no seleccione salir, realizar los siguientes pasos en cualquier orden.
  - El cliente puede elegir localizar una orden (include Place Order).
  - El cliente puede elegir recuperar la situación de una orden (include Get Status on Order).

Fin bucle.

8. El caso de uso termina

Otros Casos de Uso incluyen a Login

- **Ventaja**: el caso de uso Login describe la autenticación y nada más.
- **Desventaja**: el cliente tiene que autenticarse cada vez que quiere hacer algo diferente.



Otros Casos de Uso incluyen a Login (cont.)

Caso de Uso Log In

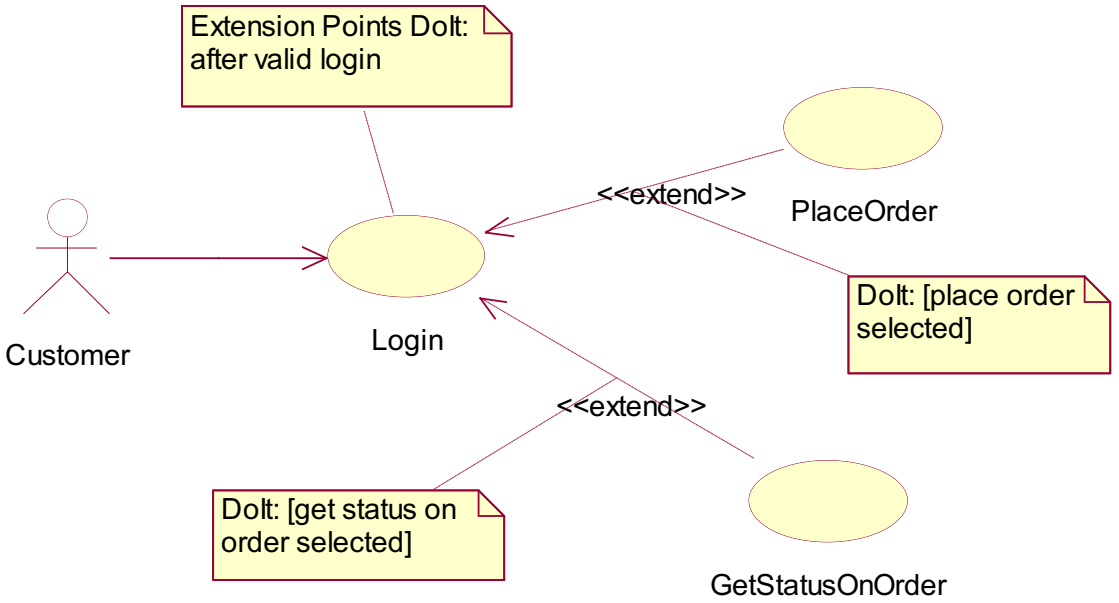
- El sistema solicita al cliente que introduzca su nombre de usuario y clave
- El cliente introduce su nombre de usuario y clave
- El sistema verifica que el cliente es un usuario válido.
- El caso de uso finaliza.

Caso de Uso Parcial Place Order

- El caso de uso comienza cuando el cliente se autentica en el sistema (include Login).
- …

Otros Casos de Uso extienden a Login

- **Ventaja**: el caso de uso Login no necesita ser modificado cada vez que se añadan nuevos casos de uso.
- **Desventaja**: puede ser una relación difícil de explicar, especialmente a personas que no desarrollen (programen). Significado de los CU no son apropiados.



Otros Casos de Uso extienden a Login (cont.)

Caso de Uso Log In

- El caso de uso comienza cuando el cliente inicia la aplicación.
- El sistema solicita al cliente que introduzca su nombre de usuario y contraseña.
- El cliente introduce su nombre de usuario y contraseña.
- El sistema verifica que el cliente es un usuario válido.
- **Mientras** el cliente no seleccione salir
  - Punto de extensión: realizarlo
- **Fin bucle**.

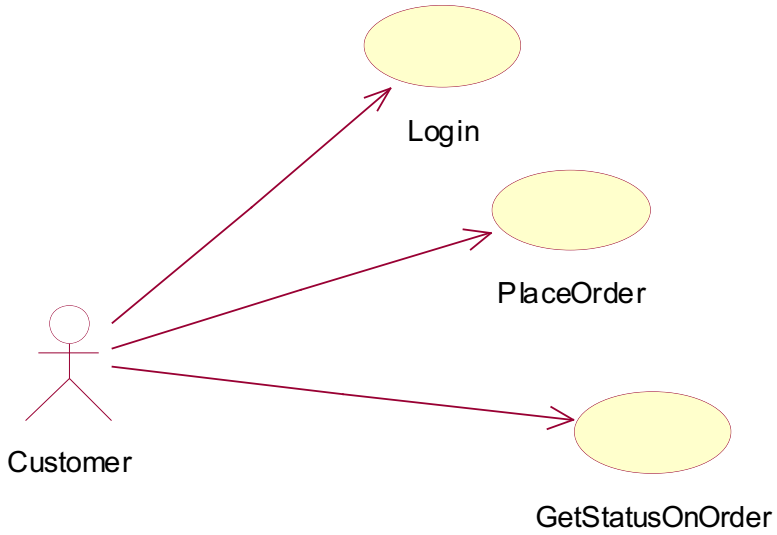
7. El caso de uso termina

Caso de Uso Parcial Place Order

Precondición: El cliente ha seleccionado Place Order.

Login como un Caso de Uso independiente

- **Ventaja**: Sólo describe la autenticación y nada más. El diagrama y el texto son claros y fáciles de entender y tenemos una mayor flexibilidad en el sistema.
  - Ahora *PlaceOrder* y *GetStatusOnOrder* no requieren que se ejecute login, pero como precondición tendrán que el cliente debe ser un usuario válido.
- **Desventaja**: gráficamente no se ve la relación entre los CU.



Login como un Caso de Uso independiente (cont.)

Caso de Uso Log In

- El caso de uso comienza cuando el usuario inicia la aplicación.
- El sistema solicita al usuario que introduzca el nombre de usuario y la contraseña.
- El cliente introduce un nombre de usuario y una contraseña.
- El sistema verifica que se trata de un usuario válido. user.
- El cado de uso finaliza.

Caso de Uso Place Order

Precondición: Un usuario válido se ha autenticado en el sistema.

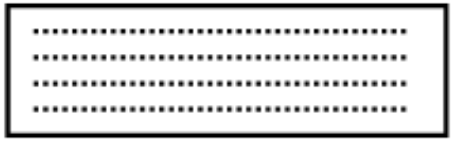
Contenido

- Introducción
- Tipos de Especificación de Requisitos
  - informal
  - semi-formal
  - formal
- Modelos de Análisis
  - Modelo del Dominio
  - Modelo del Negocio
- Casos de Uso
- Detalle de Casos de Uso
  - Guiones
  - Interacciones

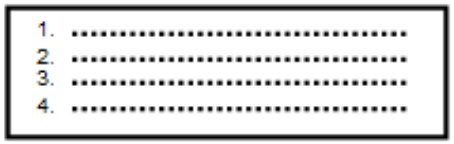
Detalle de Casos de Uso: Guiones e Interacciones

Hay 2 formas principales y complementarias de especificar CU:

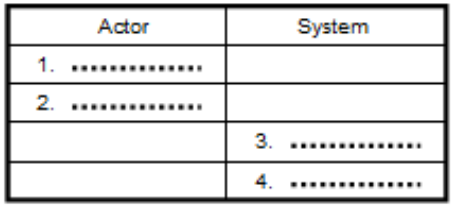
- Guión del caso de uso:
  - *narrativa*



  - *escenario*



  - *diálogo*



- Interacciones del caso de uso (realización del caso de uso):
  - *diagramas de interacción (sequencia/colaboración)*
  - *diagramas de actividad*
  - *MSC (Message Sequence Chart – graphical/textual)*
  - *Etc.*

Detalle de Casos de Uso: guiones

- Guión (narrativo):

Caso de Uso: Compra de artículo
Descripción: Este caso de uso permite comprar artículos por el cliente. El cajero utiliza el sistema TPV para registrar la venta, realizar el cobro correspondiente al cliente y finalmente emitir el recibo.

- Guión (escenario):
  1. El actor envía al sistema una petición y los datos necesarios para llevarla a cabo
  2. El sistema valida la petición y los datos
  3. El sistema altera su estado interno
  4. El sistema devuelve el resultado al actor

- Guión (diálogo):

Actor	Sistema
1. Se envía al sistema una petición y los datos necesarios para llevarla a cabo	
	2. Se valida la petición y los datos
	3. Se altera su estado interno
4. Se muestra el resultado	

Detalle de Casos de Uso: guiones

- Ejemplo de plantilla de especificación de escenario con alternativas

Caso de uso: Nombre del caso de uso
Descripción: El caso de uso descrito con una frase corta
Actores: lista de actores
Precondición: Condición
Escenario Principal:

1. El actor envía al sistema una petición y los datos necesarios para llevarla a cabo
2. El sistema valida la petición y los datos
3. El sistema altera su estado interno
4. El sistema devuelve el resultado al actor

Alternativas:

2: error en la validación
2.1 El sistema debe volver a solicitar los datos

Detalle de Casos de Uso: guiones

- **Ejemplo 1 de escenario**: CU "*registro de un siniestro*"

1. El contable introduce los datos de la reclamación (datos del cliente, fecha del siniestro, etc.)
2. El sistema extrae la información de la póliza y de la reclamación
3. El contable introduce los costos
4. El sistema confirma que los costos son conformes a la póliza y asigna un número de reclamación
5. El contable selecciona y asigna un tasador
6. El contable confirma los datos
7. El sistema guarda la información y anota el trabajo en la lista del tasador

Detalle de Casos de Uso: guiones

- Guías para escribir pasos en un escenario:
  - Usar construcciones gramaticales simples
    - Sujeto...verbo...objetos
  - Mostrar claramente quién actúa
  - Tomar un punto de vista elevado
  - Mostrar cómo se progresa hacia el objetivo
  - Incluir un conjunto “razonable” de acciones
  - Validar “razonablemente”

  - Pasos de control: “Llamar CU”, “Incluir CU”, …
  - Pasos de control: “Repetir *pasos x-y* hasta <*condición*>”
    “Si <*condición*> entonces *pasos x-y* Fin Si”

Detalle de Casos de Uso: guiones

Ejemplo 2 de escenario: CU “login”

Este caso de uso describe el proceso de conexión al sistema de gestión de averías. Además establece los permisos para las diferentes categorías de actores.

Actores: Maestro, Jefe

Precondición: el usuario no está conectado

Escenario Principal:

1. El caso de uso se inicia cuando el usuario inicia la aplicación
2. El sistema presentará la pantalla de login
3. El usuario introduce su nombre y password
4. El sistema verificará la información
5. El sistema establecerá los permisos de acceso
6. El sistema mostrará la información de confirmación

Detalle de Casos de Uso: Interacciones

- Hay 2 tipos principales de diagramas para especificar interacciones:
  - Diagramas de secuencia
  - Diagramas de comunicación (anteriormente **diagrama de colaboración**)

- Un diagrama de interacción básicamente consiste en:
  - Un conjunto de objetos / clases
  - Los mensajes que se pueden enviar entre ellos

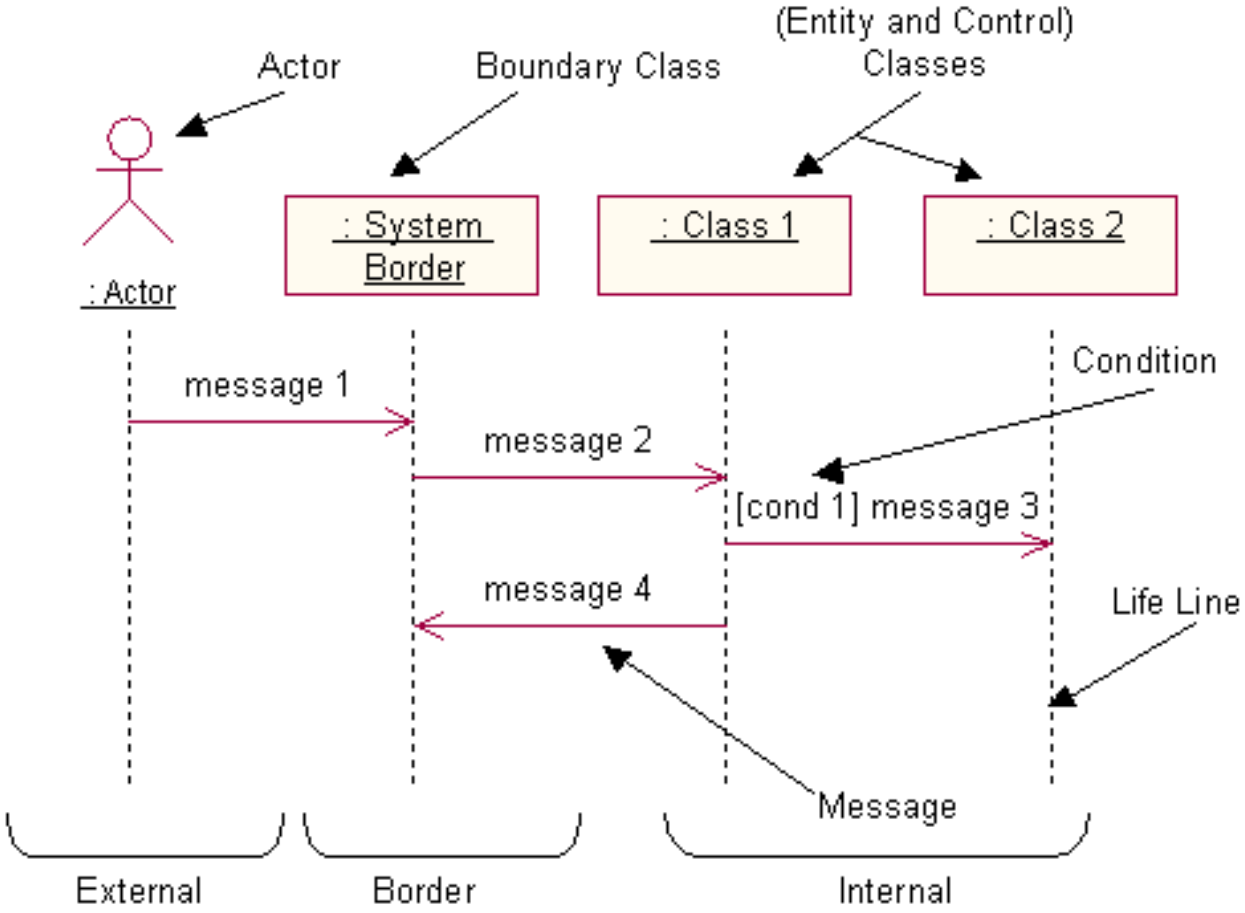
- ***Alternativamente***, también pueden utilizarse otros tipos de diagramas que describan el comportamiento del CU:
  - diagramas de actividad, diagramas de estado, etc

Detalle de Casos de Uso: Interacciones

- Un diagrama de interacción (sequencia/colaboración) tiene dos aspectos:
  - Estructural: especifica las clases, interfaces o componentes
    - Se organizan utilizando las relaciones normales de UML (asociaciones, generalizaciones y dependencias)
    - Se trata de una *vista conceptual* de la arquitectura de una parte del sistema
  - Comportamiento: dinámica de cómo interactúan estos elementos
    - Se representa habitualmente mediante diagramas de interacción
    - Debe ser consistente con la visión estructural
    - Puede haber más de un diagrama de interacción por colaboración, mostrando distintos aspectos o escenarios de una colaboración

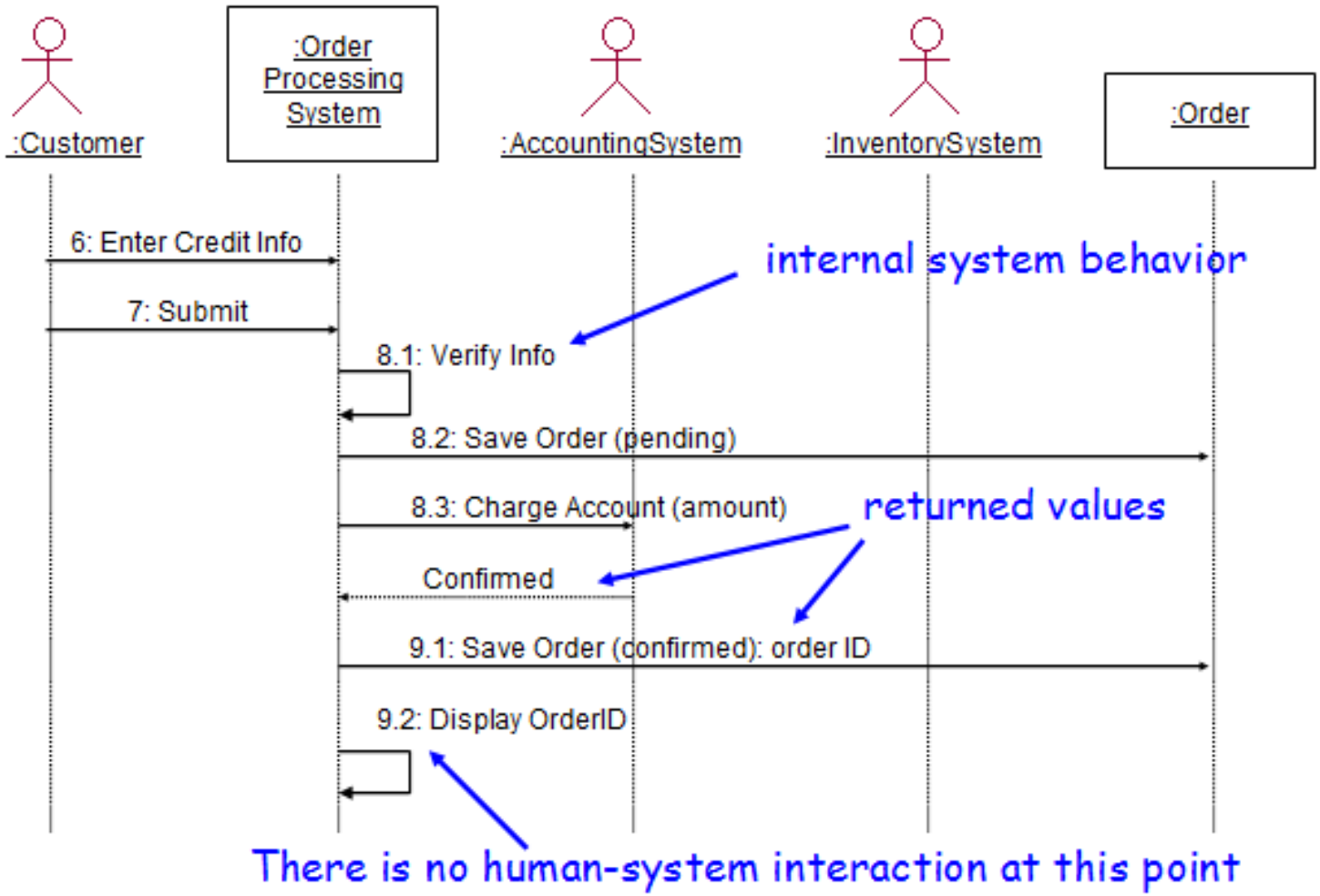
Detalle de Casos de Uso: Diagramas de Secuencia

- Los *diagramas de secuencia* se suelen asociar a los casos de uso, mostrando cómo se realizan los CU a través de interacciones de objetos



Detalle de Casos de Uso: Diagramas de Secuencia

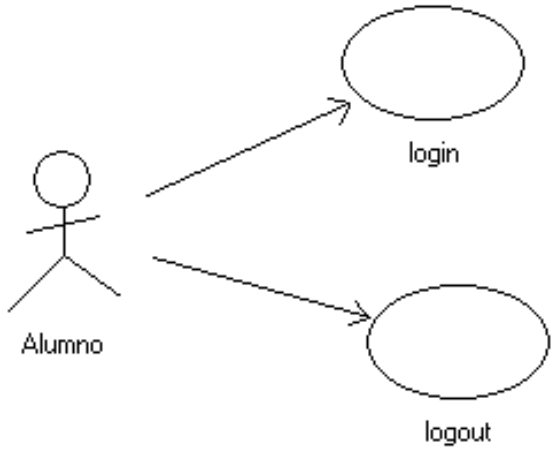
- Ejemplo:

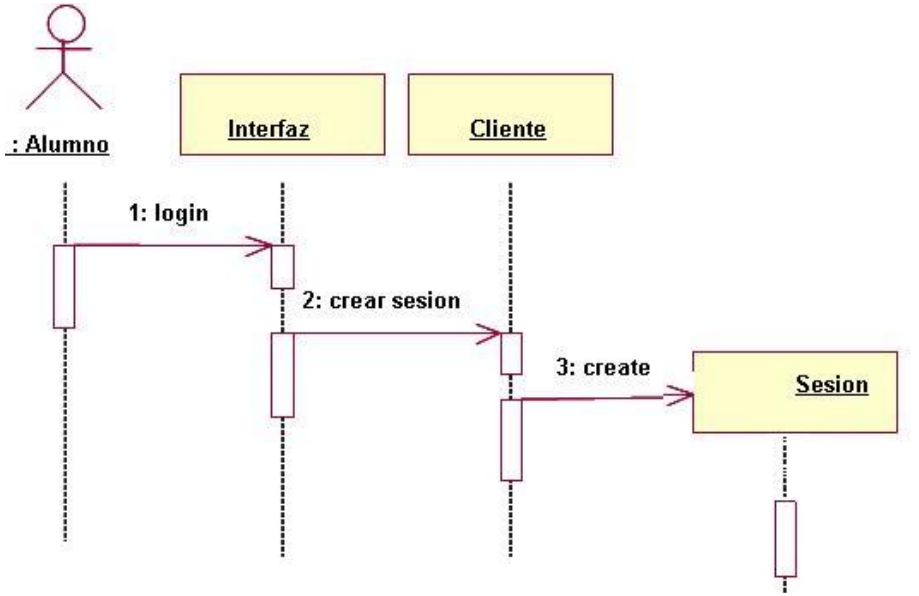


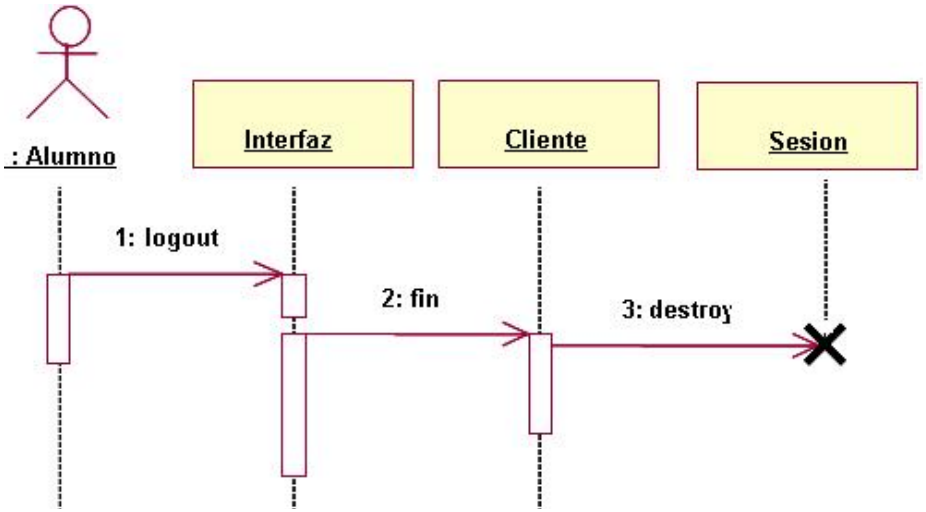
Detalle de Casos de Uso: Diagramas de Secuencia

- Los mensajes se corresponden generalmente a métodos de los objetos

- Los objetos pueden ser creados/destruidos durante la ejecución



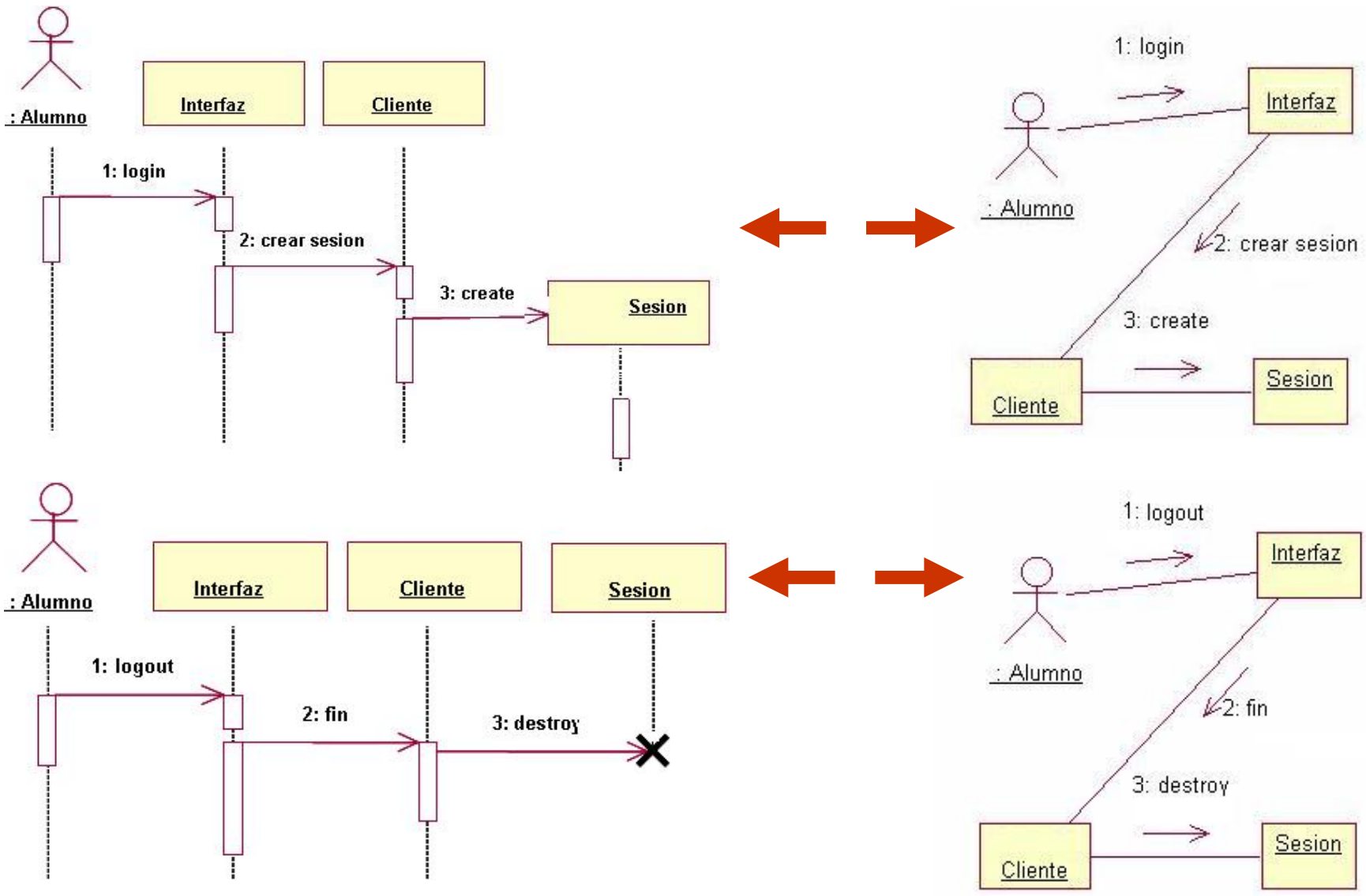




Detalle de Casos de Uso: Diagramas de Comunicación

- Destacan la organización de los objetos que participan en una interacción
- Es un grafo en el que los nodos son los objetos y los arcos los enlaces
- Los arcos se etiquetan con los mensajes que envían y reciben los objetos
- Dan una visión del flujo de control en el contexto de la organización de los objetos que colaboran

Detalle de Casos de Uso: Diagramas de Comunicación



Detalle de Casos de Uso: Diagramas de Interacción
Consideraciones finales

- No tienen por que utilizarse sólo para los CU
- Son útiles para modelar aspectos dinámicos de cualquier interacción entre cualquier instancia en cualquier vista del sistema (clases, interfaces, componentes,...)
- Pueden ser utilizados a nivel de Requisitos, Análisis o Diseño.