

2. MODELADO DE CASOS DE USO

Diseño de Software

Grado en Ingeniería Informática

Curso 2024/2025

José Varela Pet - Departamento de Electrónica y Computación

Bibliografía

- **Fowler.** *UML distilled*. 3ª ed. Addison-Wesley, 2004
 - Capítulo 9 (pp. 99-105)
- **Booch, Rumbaugh, Jacobson.** *El Lenguaje Unificado de Modelado*. 2ª ed. Addison-Wesley, 2006
 - Capítulo 17 (pp. 243-257)
 - Capítulo 18 (pp. 259-269)

Casos de uso (Jacobson, 1992)

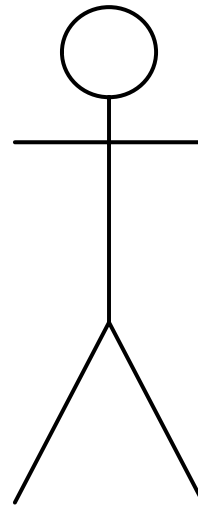
- Cualquier sistema *interesante* **interactúa** con sus usuarios y con otros sistemas
- Los casos de uso capturan el **comportamiento** deseado del sistema sin especificar cómo implementarlo ⇒ **requisitos funcionales**
- Permiten **comunicación** entre expertos del dominio, desarrolladores y usuarios

Casos de uso

- **Caso de uso:** conjunto de *secuencias* de acciones que ejecuta un sistema para producir un resultado observable de valor para un *actor*
- **Secuencia:** interacción del sistema con agentes externos
- **Actor:** rol que adopta un usuario o un sistema externo al interaccionar con el sistema representado

Representación gráfica

Caso de Uso



Actor1

«actor»
Actor2

Escenarios

- **Escenario:** secuencia específica de acciones que describe un comportamiento
- **Caso de uso:** conjunto de escenarios ligados por un **objetivo** de usuario común
- Los escenarios son **instancias** de los casos de uso

Especificación de casos de uso

- Se describen textualmente como un flujo de eventos, incluyendo un escenario **principal** y sus **alternativas**
- El grado de detalle necesario depende del **riesgo** asociado al caso de uso
- Representan **vista externa** del sistema (*caja negra*) \Rightarrow no cabe esperar correlación con sus clases

Ejemplo [Fowler, 2004]

- **Escenario:** “El cliente navega por el catálogo y añade artículos a su cesta. Para pagar, introduce su dirección y los datos de su tarjeta, y confirma la compra. El sistema comprueba la validez de la tarjeta de crédito y confirma la transacción en pantalla y mediante correo electrónico”

Escenarios alternativos

- Un caso de uso presenta un escenario **principal** y, habitualmente, varios escenarios **alternativos** en los que:
 - Algún paso falla
 - Se sigue un camino alternativo
- **No hay una regla general** para considerar una secuencia como caso de uso separado o como escenario dentro de un caso de uso existente

Ejemplo [Fowler, 2004]

Comprar un producto

1. El cliente selecciona artículos del catálogo
2. El cliente va a la caja
3. El cliente introduce la información de reparto
4. El sistema muestra el precio final
5. El cliente introduce los datos de su tarjeta
6. El sistema autoriza la venta
7. El sistema confirma la transacción

Alternativa nº 1: Fallo en autorización (paso 6)

Se permite reintroducir los datos de la tarjeta (paso 5)

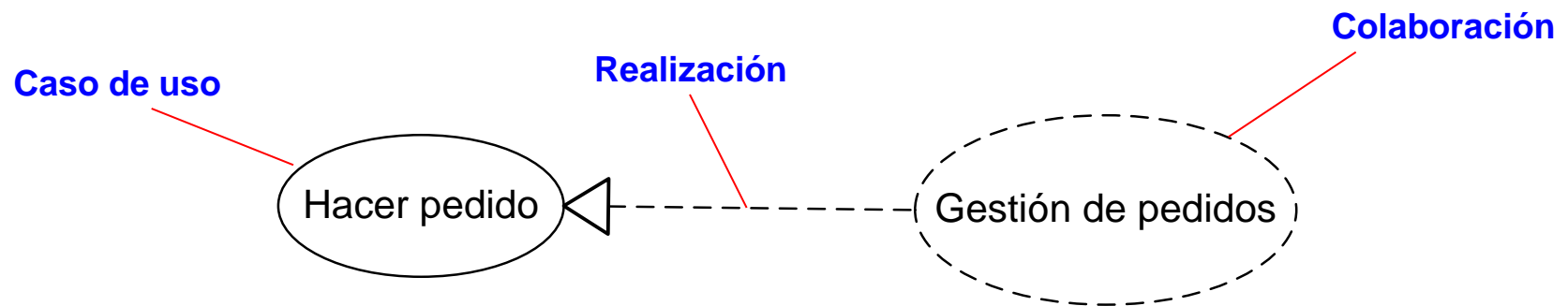
Alternativa nº 2: Cliente habitual

Se muestran valores por defecto de reparto y pago

El cliente acepta o modifica datos y salta al paso 6

Colaboraciones

- Los casos de uso capturan el comportamiento sin especificar cómo se implementa \Rightarrow análisis no influenciado por implementación
- Los casos de uso se **realizan** por medio de **colaboraciones** entre clases y otros elementos



Actores

- No son necesariamente humanos
- En general, **cualquier agente externo** que intercambia información con el sistema modelado
- El mismo usuario puede adoptar varios *roles* diferentes
- Cada actor puede participar en varios casos de uso y viceversa

Caso de uso bien estructurado

- Denota comportamiento simple e identificable
- Identifica **actores** que interactúan con él
- Incorpora comportamiento común **incluyendo** otros casos de uso y coloca variantes en casos de uso que lo **extienden**
- Describe flujo de eventos con los actores por medio de un conjunto mínimo de **escenarios** que muestran semántica normal y variantes
- Especifica **pre** y **post condiciones**

Diagramas de casos de uso

- Introducidos por Jacobson en 1994
- Visualizan el **comportamiento** de un sistema a alto nivel para poder usarlo e implementarlo
- Se emplean para:
 - Modelar el **contexto** de un sistema
 - Modelar sus **requisitos** funcionales

Modelado del contexto del sistema

- ✓ Modelar el **contexto** es establecer lo que está dentro o fuera del sistema
- ✓ Elementos del sistema son responsables del **comportamiento** esperado por el entorno
- ✓ **Contexto**: formado por elementos externos (actores) que interactúan con el sistema
- ✓ Se puede rodear el sistema por un rectángulo para enfatizar lo que queda fuera

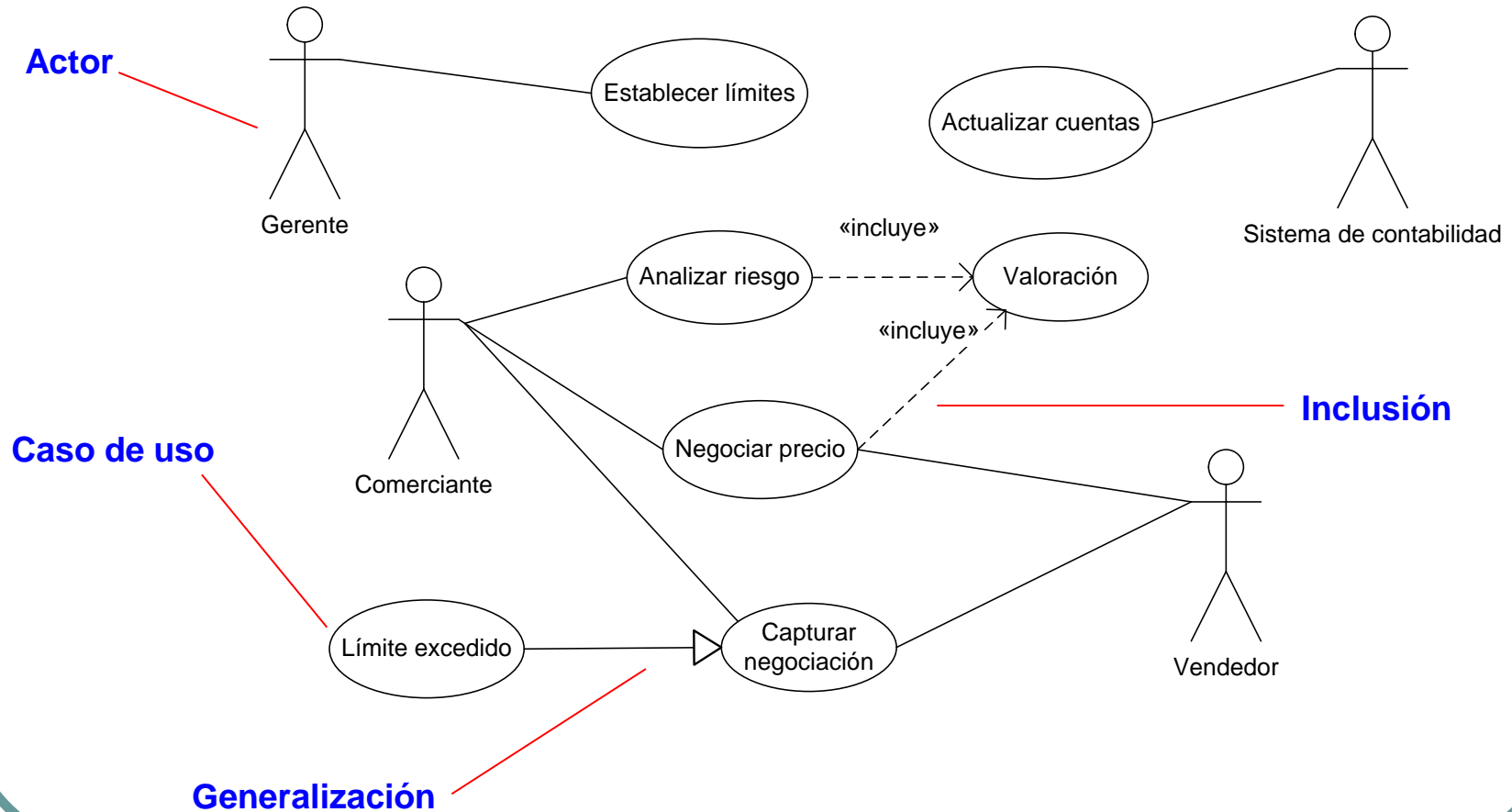
Modelado de requisitos funcionales

- ✓ Modelar **requisitos** implica especificar qué debería hacer el sistema
- ✓ **Requisito**: característica de diseño, propiedad o comportamiento del sistema
- ✓ Un requisito es un **contrato** entre el sistema y los elementos externos
- ✓ Un sistema debe satisfacer sus requisitos de manera predecible y fiable

Bloques en un diagrama de casos de uso

- **Actores:** roles que juegan los elementos que forman el entorno del sistema
- **Casos de uso**
- **Asociaciones** entre actores y los casos de uso en que están implicados
- **Relaciones**
 - Entre casos de uso: generalización, inclusión, extensión
 - Entre actores: generalización

Ejemplo



Actores y casos de uso

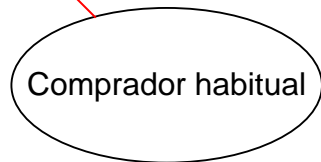
- Primero se debe identificar actores y extraer después los casos de uso en que **intervienen**
- Fowler sólo considera actores que obtienen un **beneficio** del caso de uso
- Algunos casos de uso pueden no estar **ligados** a ningún actor
- La mayor dificultad estriba en decidir el **tamaño** adecuado para los casos de uso

Relaciones entre casos de uso

- **Generalización:** el caso de uso hijo hereda el comportamiento del padre y puede añadir o redefinir comportamiento
- **Inclusión:** un caso de uso incorpora el comportamiento de otro
- **Extensión:** un caso de uso amplía el comportamiento de otro sólo en ciertos **puntos de extensión**

Relación de extensión

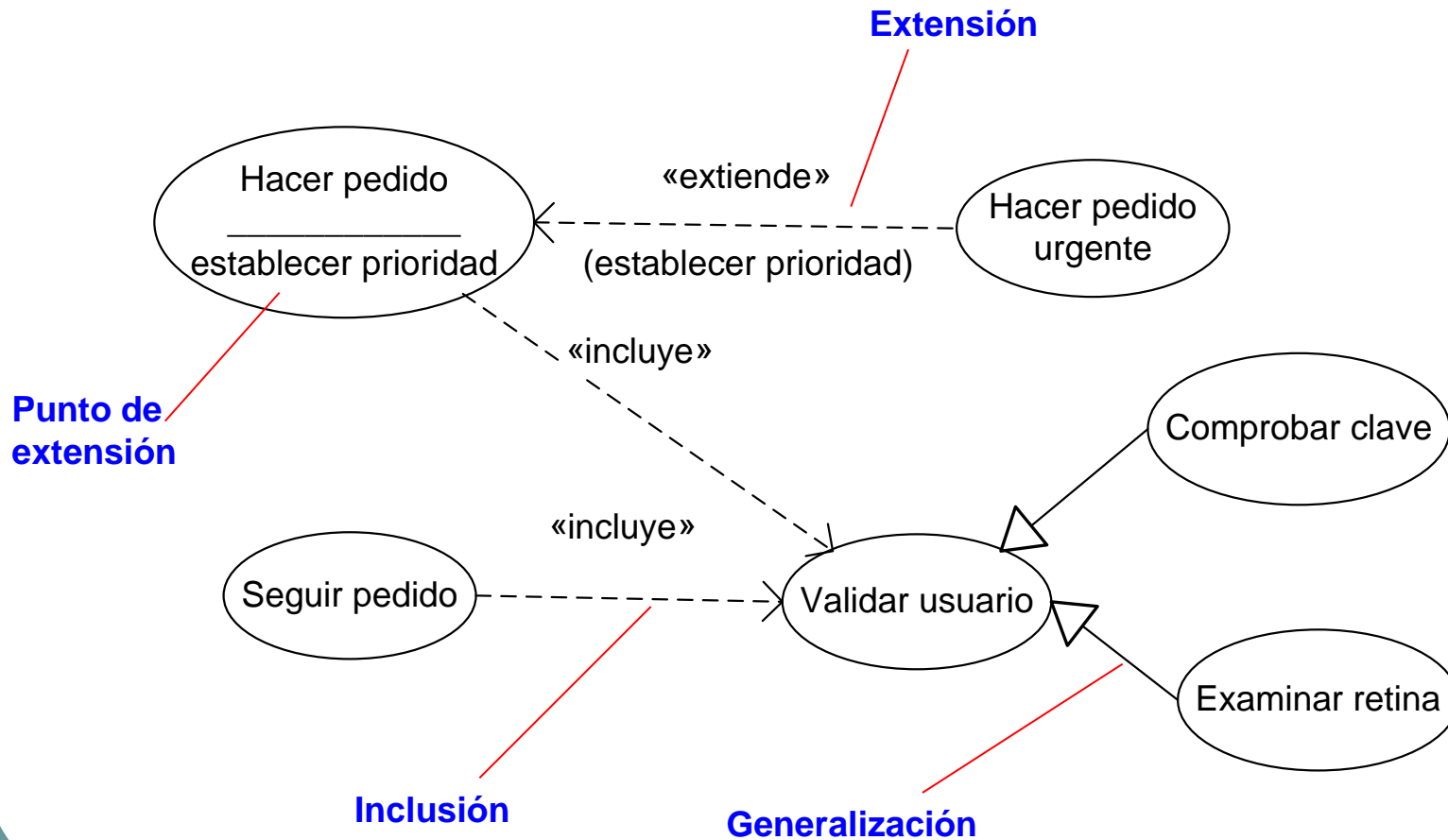
Caso de uso extendido



Caso de uso base

Puntos de extensión

Ejemplo



Modelado del contexto

- ✓ Decidir qué se debe incluir y qué se debe omitir como actor
- ✓ Identificar y clasificar **actores** en torno al sistema
- ✓ Organizar actores en **jerarquías** de generalización/especialización
- ✓ Especificar vías de **comunicación** de cada actor con los casos de uso

Modelado de requisitos

- ✓ Considerar **comportamiento** que cada actor espera del sistema e introducirlo como **caso de uso**
- ✓ Factorizar comportamiento común por medio de **inclusiones**
- ✓ Expresar comportamiento variante a través de **extensiones**
- ✓ Enunciar requisitos no funcionales dentro de **notas**

Para qué NO usar un diagrama de casos de uso

- No debe mostrar **estructura** del sistema (¡visión de *caja negra*!)
 - En los nombres de los casos de uso y en los escenarios aparecen referencias a conceptos del dominio de aplicación
- No debe representar **flujo**
 - Los escenarios sí que lo recogen
- No debe ilustrar **despliegue** del sistema
 - Diagramas de componentes para ensamblado
 - Diagramas de despliegue para topología