

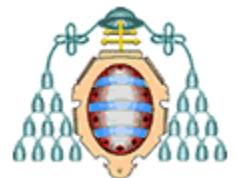
Técnica: Casos de Uso

Ingeniería del Software

José García Fanjul (<http://www.di.uniovi.es/~fanjul/>)
Grupo de Investigación en Ingeniería del Software
<http://giis.uniovi.es/>



Curso 2022-2023





Contenido

- Introducción
- Casos de Uso
- Actores
- Relaciones
- Conclusiones
- Anexos

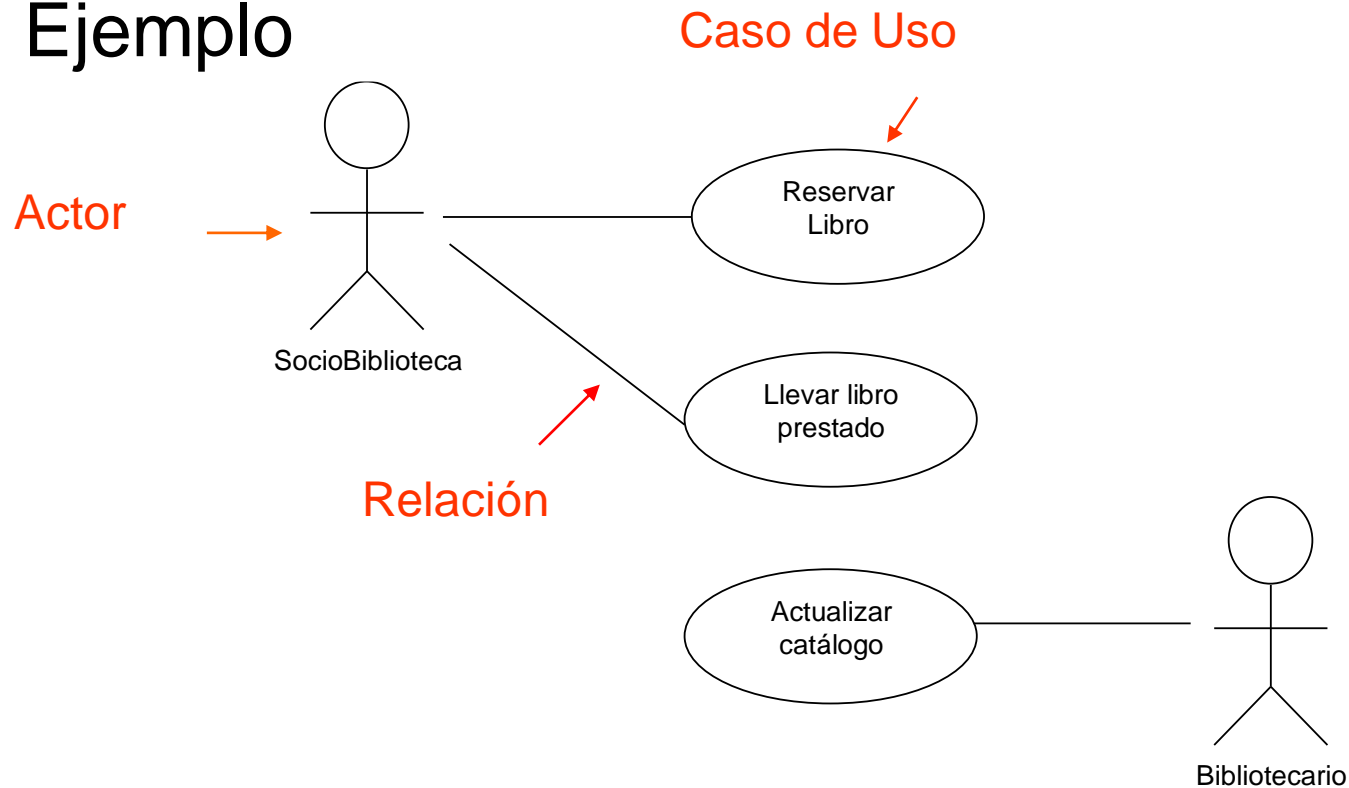
Introducción

- Técnica gráfica para modelar el comportamiento de un sistema desde el **punto de vista del usuario**
- Determinan los requisitos funcionales del sistema
 - representan las funciones que un sistema puede ejecutar
- Importante:
 - Reflejan lo *esencial* del sistema
- Se pueden usar durante diferentes fases del desarrollo:
 - Captura de Requisitos
 - Especificación Funcional del Sistema
 - Planificación de iteraciones de desarrollo
 - Validación del sistema



Introducción

■ Ejemplo



Casos de Uso



- **Secuencia de acciones** que describen una tarea o proceso realizado por un actor con el sistema
- Importante
 - Es un modelo de un proceso
 - En general, no incluir como casos de usos procesos relativos a:
 - Aspectos no funcionales
 - Mantenimiento de maestros
 - Consultas/Informes no relevantes
 - ...
- Cada uno de los casos de uso debe especificarse textualmente mediante uno o varios flujos.
 - Flujo principal
 - Flujos alternativos

Casos de Uso: flujos

Caso de Uso “Llevar prestado un libro”

Objetivo: Un socio de la biblioteca desea llevarse un libro prestado.

Flujo (un primer borrador):

1. El socio de la biblioteca solicita llevarse un ejemplar de un libro.
2. El sistema comprueba que los libros que ya tiene en préstamo no sobrepasan el número máximo de libros que se pueden tener.
3. Si la comprobación anterior es correcta, el sistema comprueba si el ejemplar que se desea llevar está reservado por otra persona.
 1. Si el ejemplar está reservado, entonces no se permite el préstamo.
 2. Si no lo está, el sistema registra el préstamo y se emite un recordatorio en papel para el cliente con la fecha máxima de devolución del ejemplar.

Excepciones: Si quien quiere realizar el préstamo no es un socio de la biblioteca sino parte del personal de la misma, el número máximo de libros en préstamo aumenta (hasta 12)

Casos de Uso: flujos

Caso de Uso “Llevar prestado un libro”

Objetivo: Un socio de la biblioteca desea llevarse un libro prestado.

Flujo (un primer borrador):

1. El socio de la biblioteca solicita llevarse un ejemplar de un libro.
2. El sistema comprueba que los libros que ya tiene en préstamo no sobrepasan el número máximo de libros que se pueden tener.
3. Si la comprobación anterior es correcta, el sistema comprueba si el ejemplar que se desea llevar está reservado por otra persona.
 1. Si el ejemplar está reservado, entonces no se permite el préstamo.
 2. Si no lo está, el sistema registra el préstamo y se emite un recordatorio en papel para el cliente con la fecha máxima de devolución del ejemplar.

Excepciones: Si quien quiere realizar el préstamo no es un socio de la biblioteca sino parte del personal de la misma, el número máximo de libros en préstamo aumenta (hasta 12)

Este flujo está redactado sin la participación del bibliotecario. Haz el ejercicio de redactarlo con su participación.

¿Cómo se identifica el socio ante el sistema? ¿cómo sabe el sistema qué ejemplar desea llevarse el socio?

¿Cómo sabe el sistema cuál es el número máximo de libros? ¿Parámetros del sistema!... y por tanto funcionalidad para configurarlos.

¿Qué datos se registran sobre el préstamo? ¿sólo la fecha? ¿Cómo se calcula la fecha máxima? ¿otro parámetro? ¿qué pasa si la fecha máxima cae en festivo? ¿cómo puede saber el sistema si una fecha es festivo o no?

Casos de uso: flujos



- Las acciones de los flujos deben ser sencillas.
 - Utilizar un enfoque jerárquico, explicar acciones complejas utilizando acciones más sencillas.
- Las acciones representan:
 - Algo que hace un actor
 - Algo que hace el sistema
- Analizar el orden en que se ejecutarán las acciones, numerarlas.
- Los flujos deben ser completos: indicar todas las acciones necesarias para llevar a cabo el caso de uso.
- Centrarse en lo esencial, evitando referencias a diseño detallado de pantallas (hacer click en un botón).

Casos de uso: flujos



En lugar de especificar condicionales complejos es preferible usar flujos alternativos.

- Flujo principal y flujos alternativos:
 - Especificar en el flujo principal lo que sucederá normalmente al ejecutar el caso de uso.
 - En flujos alternativos: variantes o tratamientos excepcionales.
- Especificar claramente los flujos:
 - Al finalizar un flujo alternativo, el caso de uso puede volver al flujo principal o puede terminar, indicarlo.
 - Especificar en qué momentos el usuario puede cancelar un caso de uso. Algunas cancelaciones requerirán la ejecución de acciones compensatorias.
- Si es necesario:
 - Indicar precondiciones y postcondiciones (el estado en que debe encontrarse el sistema antes/después de ejecutar un caso de uso).

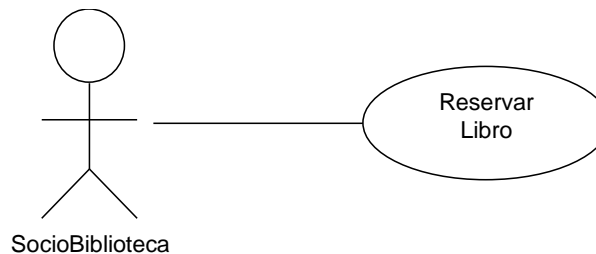
Actores

- Representan a las partes que interactúan con el sistema o un tipo de usuario del sistema
- No siempre usuarios “humanos”
 - Unidades organizativas
 - Otros sistemas
- Una clasificación (hay más)
 - Iniciador: inicia un caso de uso
 - Participante: involucrado en el caso de uso, pero no lo inicia
- La misma persona física puede interpretar varios papeles (roles) como actores distintos
 - El nombre del actor describe el papel desempeñado, no la persona física

Relaciones

■ Asociación

- Relaciones entre casos de uso y actores
- El actor interactúa (o participa) con el sistema para llevar a cabo el caso de uso
- Se representan por una línea no dirigida y sin etiqueta



- No entender como obligatoriedad
 - El actor “puede” interactuar con el sistema a través del caso de uso

Conclusiones



- **Técnica para modelar funcionalmente el sistema**
- Modelar lo esencial, independientemente de la implementación.
 - Casos de Uso
 - Procesos desde el punto de vista del usuario
 - No especificar procesos “triviales”
 - Mantenimiento de maestros
 - Consultas no relevantes
- **Lo fundamental: las descripciones de los casos de uso.**
- **Completar con prototipos, revisar con stakeholders y contrastar con requisitos de usuario.**

Bibliografía

- Cockburn A (2001). Writing Effective Use Cases. Addison-Wesley 2001.
- Stevens P, Using UML: Software Engineering with Objects and Components (2nd edition)", Addison-Wesley, 2006.

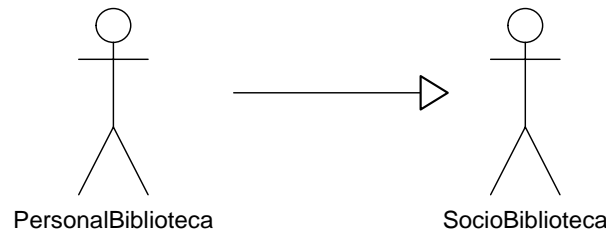
Anexo: Proceso (Cockburn)

1. **Actors & Goals.** List what actors and which of their goals the system will support. Review this list, for accuracy and completeness. Prioritize and assign to teams and releases.
2. **Main Success Scenarios.** For the use cases you have selected to pursue, write the stakeholders, trigger and main success scenario. These you can review in draft form to make sure that the system really is delivering the interests of the stakeholders you care about.
3. **Failure conditions.** Brainstorm all the failures that could occur. Draft this list completely before working out how the system must handle them all. Filling in the next step, failure handling, will take very much more energy than simply listing the failures. People who start writing the failure handling immediately often run out of the energy needed to finish listing all the failure conditions.
4. **Failure handling.** Finally, write how the system is supposed to respond to each failure. This is often tricky, tiring and surprising work. It is surprising because, quite often, a question about an obscure business rule will surface during this writing. Or the failure handling will suddenly reveal a new actor or a new goal that needs to be supported.

Anexo: Más sobre relaciones

■ Generalización

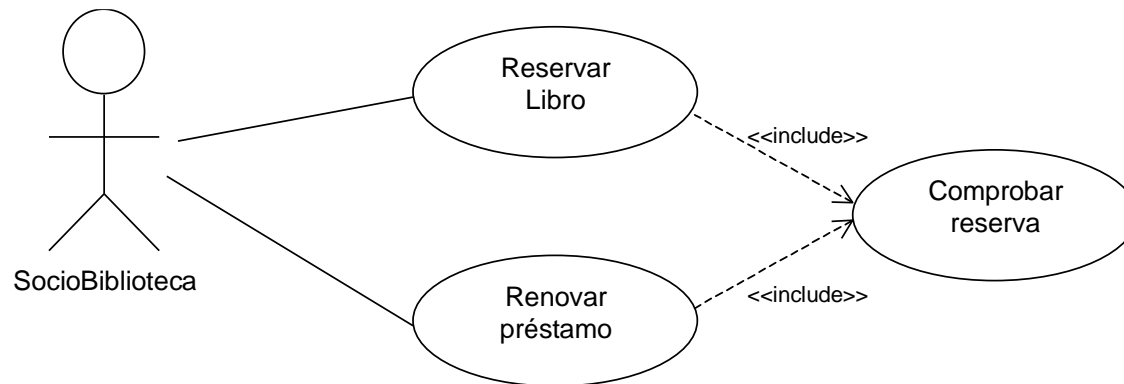
- ☐ Relación entre casos de uso o actores
- ☐ Entre casos de uso: poco útil en situaciones normales, revisar el estándar si se identifica un posible uso.
- ☐ Entre actores



Anexo: Más sobre relaciones

■ Inclusión

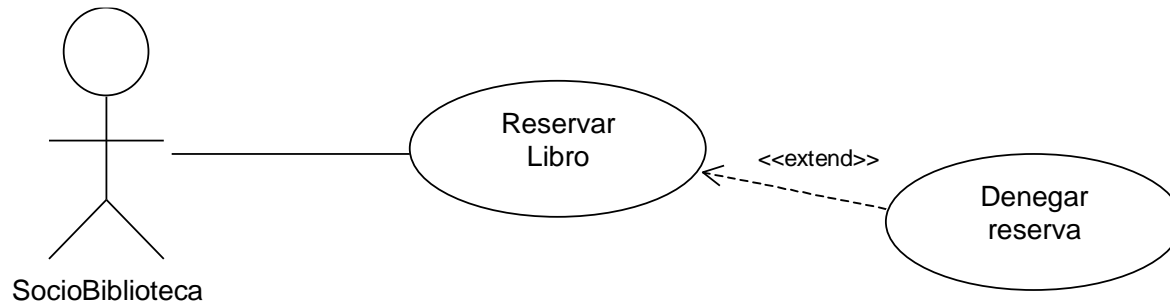
- Relación entre casos de usos
- El caso de uso incluye siempre el comportamiento del caso de uso incluido



Anexo: Más sobre relaciones

■ Extensión

- Relación entre casos de usos
- Bajo ciertas circunstancias, el comportamiento del caso de uso se extiende con otro caso de uso



- Opcionalmente se pueden especificar en el caso de uso extendido puntos de extensión
 - Indican explícitamente el instante en el que el caso de uso se extiende