

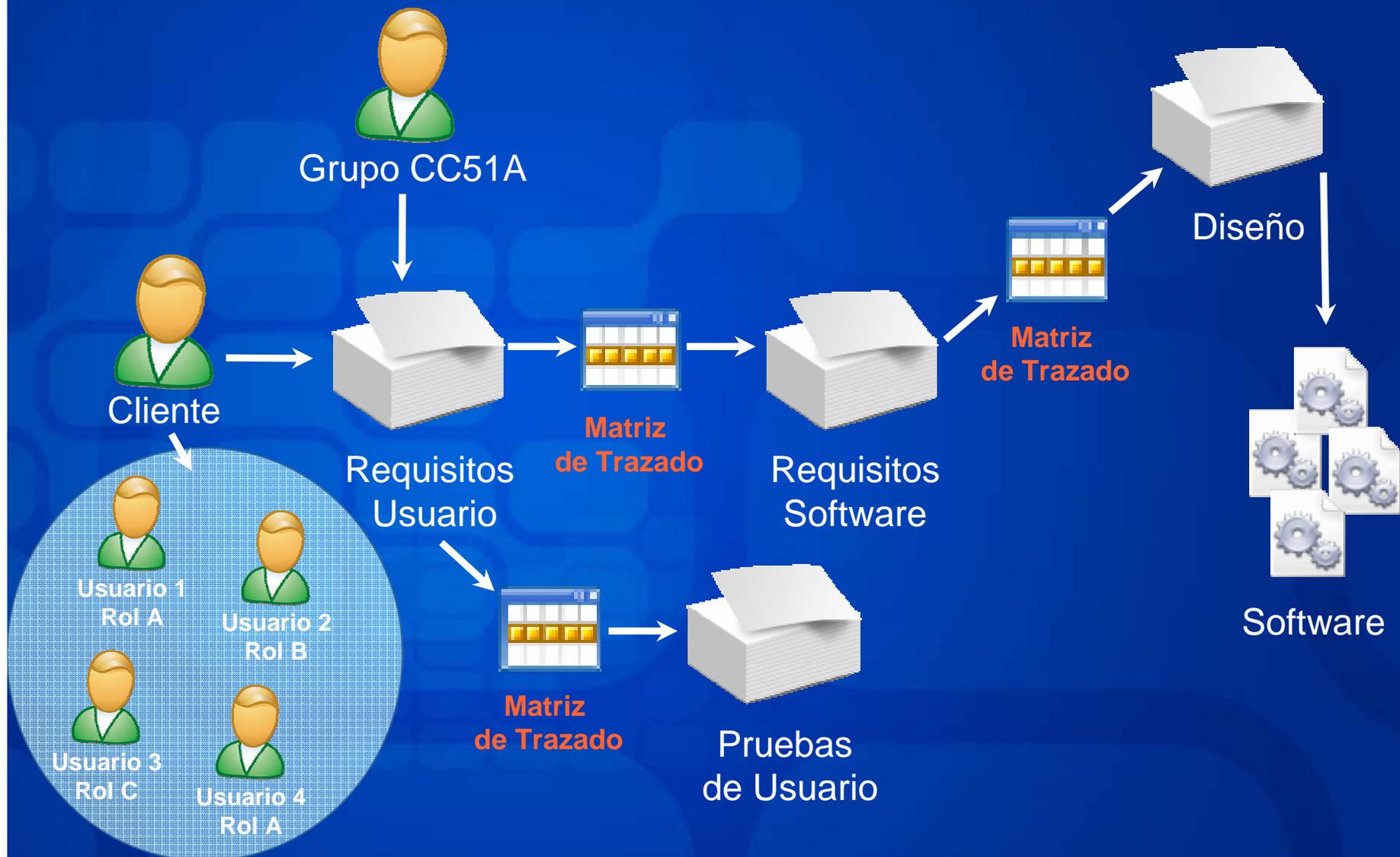
Ingeniería de Software – CC51A

Clase Auxiliar

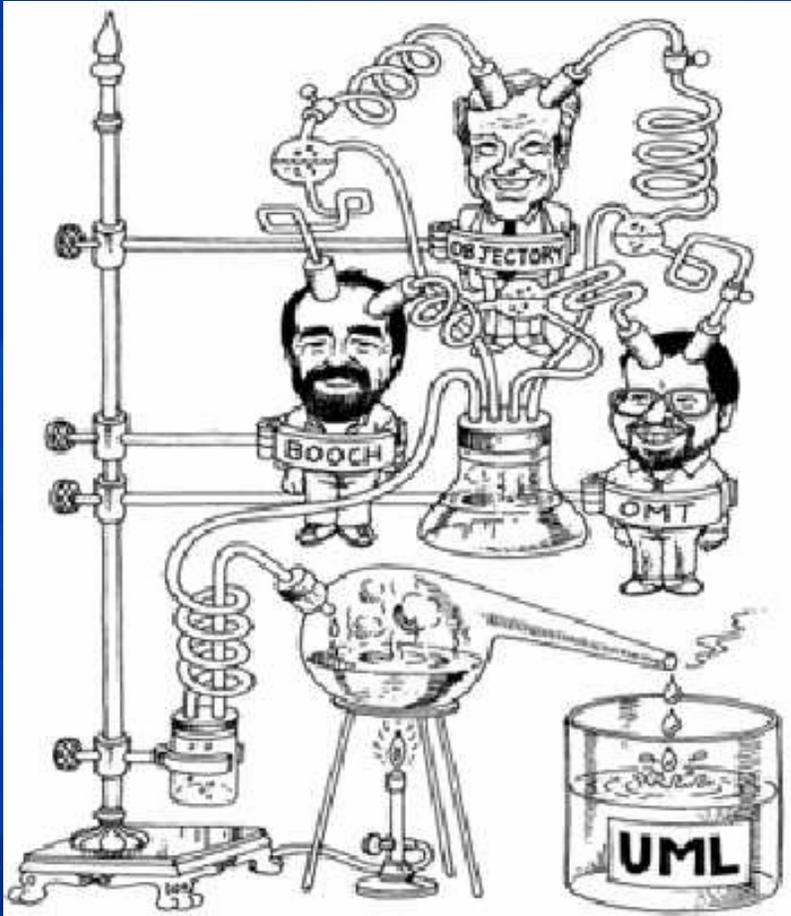
Auxiliar: Andrés Neyem
Oficina 418 de Doctorado
aneyem@dcc.uchile.cl

6 de Agosto de 2007

Aspectos Generales



Que es UML?



Historia de los lenguajes de modelamiento OO

- 1994 Rumbaugh se une a Booch (vers. 0.8 de Unified Method) en Rational.
- 1995 presentan el Unified Method en OOPSLA.
- Jacobson se une a Rational (versión 0.9 de UML, junio de 1996).
- Versión 1.0 enero 1997, aprobación de OMG, Object Management Group.
- Versión 1.1 en noviembre de 1997 (con nuevas recomendaciones).
- Versión 1.2 en junio de 1998.
- Versión 1.3 a fines de 1998.
- Versión 1.4 en septiembre de 2001.
- Versión 1.5 en marzo de 2003 (versión oficial de OMG).
- Versión 2.0 en octubre de 2004.

Que es UML?

- **Lenguaje Unificado de Modelado**
- **Unified (UNIFICADO):**
 - El aporte de muchos métodos y notaciones
 - El concepto de Ciclo de Vida de Desarrollo (completo)
 - Para un amplio conjunto de dominios de aplicación
 - Más allá de implementaciones, plataformas y lenguajes
 - Para todo tipo de proceso de desarrollo
 - Internamente autodefinido como un metamodelo
- **Modeling (MODELADO):**
 - Los modelos son utilizados en todas las ingenierías
- **Language (LENGUAJE):**
 - Si hay gente, requieren comunicarse, si se tienen que comunicar se tienen que entender, necesitan un lenguaje.

Objetivos del UML

- Establecer un lenguaje visual de modelado, expresivo y sencillo en su uso
- Mantener una independencia de los procesos de modelado y de los lenguajes de programación
- Establecer bases formales
- Integrar las mejores prácticas
- Imponer un estándar mundial

Herramientas del UML

- Modelado de requerimientos
 - Diagrama de casos de uso
- Modelado de la estructura
 - Diagrama de clases
 - Diagrama de objetos
- Modelado de la interacción
 - Diagrama de secuencias
 - Diagrama de colaboraciones
- Modelado del comportamiento
 - Diagrama de estados
 - Diagrama de actividades
- Herramientas de diseño
 - Diagrama de componentes
 - Diagrama de despliegue
- Organización del modelo
 - Diagrama de paquetes

Diagrama de Casos de Uso



- **Introducido formalmente por Ivar Jacobson**
- **Aceptado por la comunidad usuaria de T00 y por muchos metodologistas**
- **De empleo en la etapa de relevamiento para captar los requerimientos de los usuarios**
- **De fácil comprensión por parte de los usuarios de los sistemas**

Diagrama de Casos de Uso

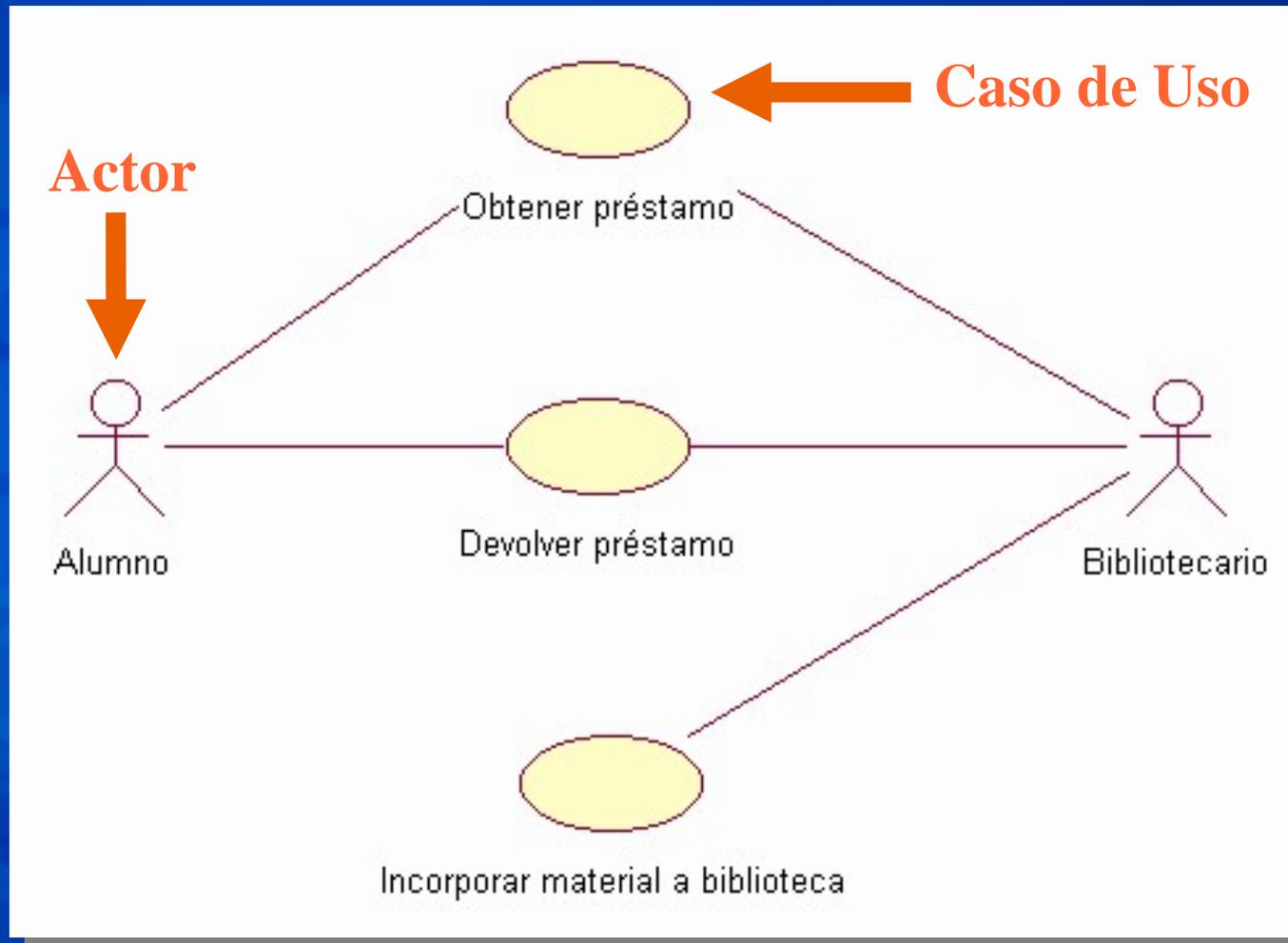


Diagrama de Casos de Uso

- **Actor:**

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso juegan al interactuar con el sistema. Normalmente, un actor representa un rol que es jugado por una persona, un dispositivo de hardware, una base de datos o incluso otro sistema que interactúe.

- **Caso de Uso:**

Un caso de uso es la representación abstracta de una funcionalidad del sistema que provee un resultado de valor desde el punto de vista de sus actores. Además de poseer una representación gráfica como todos los elementos del UML, se describe por medio de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable, de valor para un actor como el cálculo de un resultado, la generación de un nuevo objeto o el cambio de estado de un objeto.

Diagrama de Casos de Uso

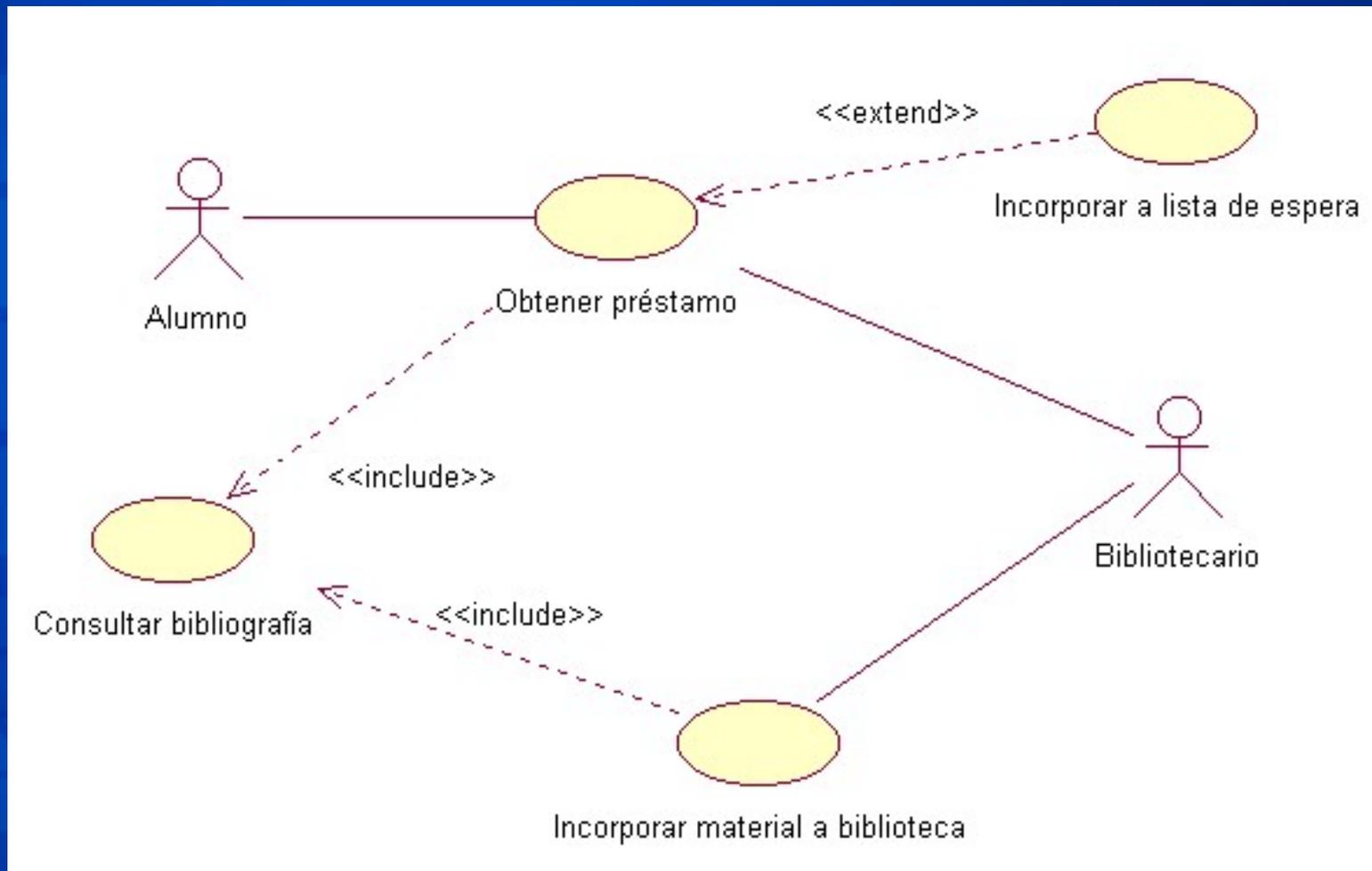


Diagrama de Casos de Uso

- «extender» (extend) se emplea para establecer una situación excepcional o de error, que no corresponde al camino normal del caso de uso básico.
- «incluir» (include) se utiliza para extraer las parte comunes de los casos de uso; son casos de uso abstractos

Diagrama de Casos de Uso

Descripción Inicial del Caso de Uso

Nombre:
Código:
Actores:
Descripción:

Más que suficiente para este Curso!!!

Descripción Base del Caso de Uso

Nombre:
Código:
Actores:
Descripción:
Curso Eventos:
Precondiciones:
Poscondiciones:
Suposiciones:

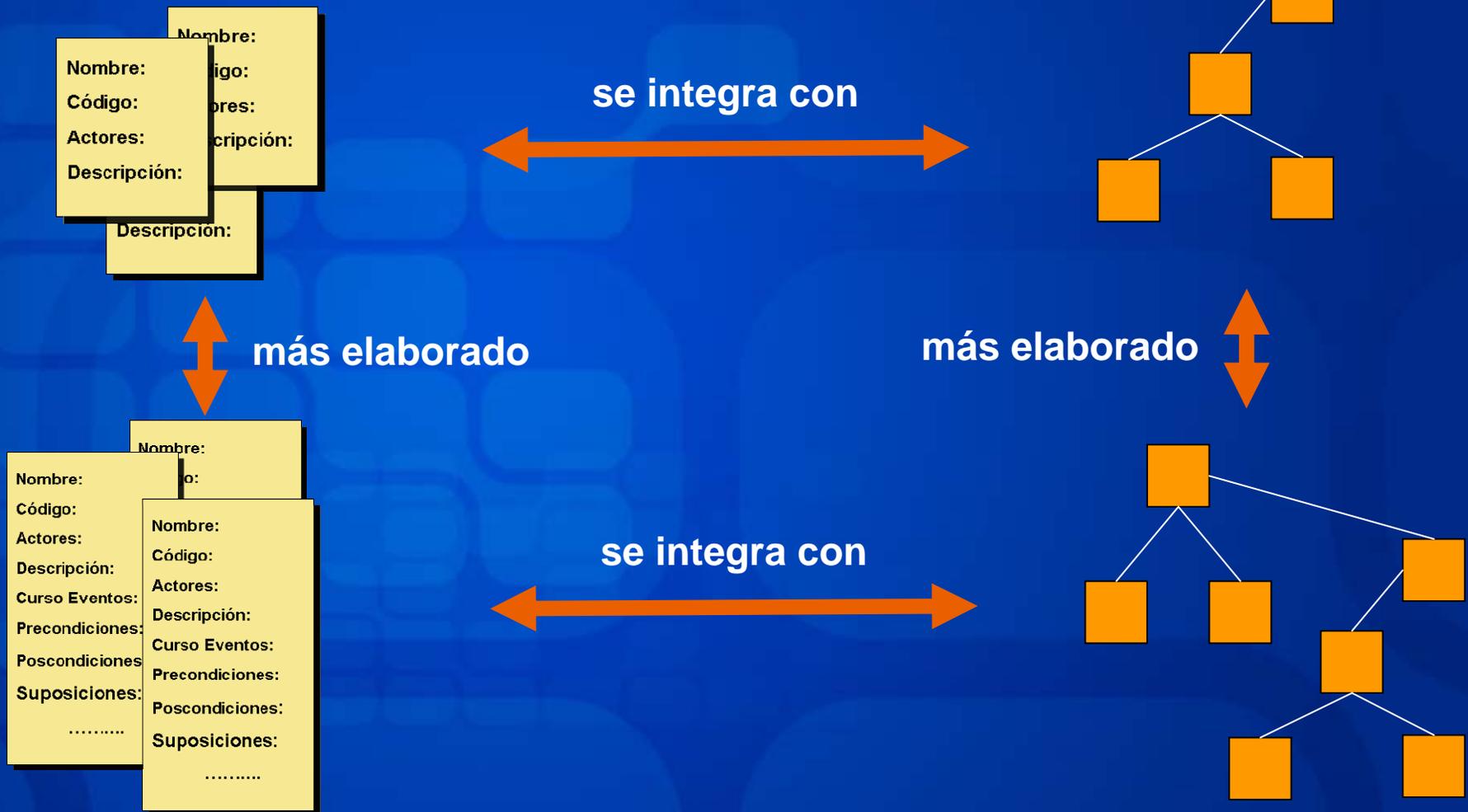
Descripción Elaborada del Caso de Uso

Nombre:
Código:
Actores:
Descripción:
Curso Eventos:
Precondiciones:
Poscondiciones:
Suposiciones:

.....

Diagrama de Casos de Uso

Descripción Inicial del Caso de Uso



Construcción de los diagramas

Pasos recomendados:

- elaborar una lista de actores y definir sus roles
- elegir el actor más representativo del sistema para comenzar el diagrama
- agotar todas las necesidades funcionales del actor incorporando los casos de uso de la funcionalidad base
- para cada caso de uso, buscar los actores que deban colaborar con él
- repetir los dos pasos anteriores para cada actor
- incorporar la funcionalidad necesaria para excepciones y errores
- factorizar los casos de uso
- obtener los actores abstractos mediante generalización
- describir cada caso de uso a medida que se incluye en el modelo
- validar y verificar el modelo junto con los usuarios

Ejercitación de Casos de Uso

Problema:

- Plantear como dominio del problema el sistema de la biblioteca de la Universidad
- Considerar que los usuarios de la biblioteca deberían poder realizar consultas acerca del diverso material existente
- Los usuarios deberían poder obtener material en calidad de préstamo o para uso en sala de lectura
- Los usuarios deberían poder devolver el material en su poder
- El personal de biblioteca debería gestionar la compra del material que se le solicita o que detecta que hace falta ante el área de la Universidad encargada de compras
- Deberían considerarse sanciones para quienes no cumplan con las reglamentaciones de la biblioteca por diversas causas

Ejercicio:

- Realizar el modelo de casos de uso que contemple los requerimientos del problema.
- Considere soluciones alternativas para atender todos los errores o excepciones que detecte.