Modelado Conceptual de Bases de Datos

Universidad Autónoma
de Entre Ríos

Características – Modelo Conceptual

- El diseño de una base de datos, implica una serie de pasos, se va avanzando de un nivel de **abstracción** menor a uno mayor, utilizando para ello distintos modelos.
- La arquitectura ANSI/SPARC tiene distintas formas de representación abstracta de la información
- Debemos centrarnos en el aspecto **lógico** de la información, del aspecto **físico** se encargará el DBMS
- El modelo conceptual se define en forma exterior del DBMS

UML: Diagramas de Clases

- Diseño Conceptual de Bases de Datos
- UML
- Diagrama de Clases
- Componentes de un Diagrama de Clases
- Propiedades Extendidas
- Uso y Ejemplos

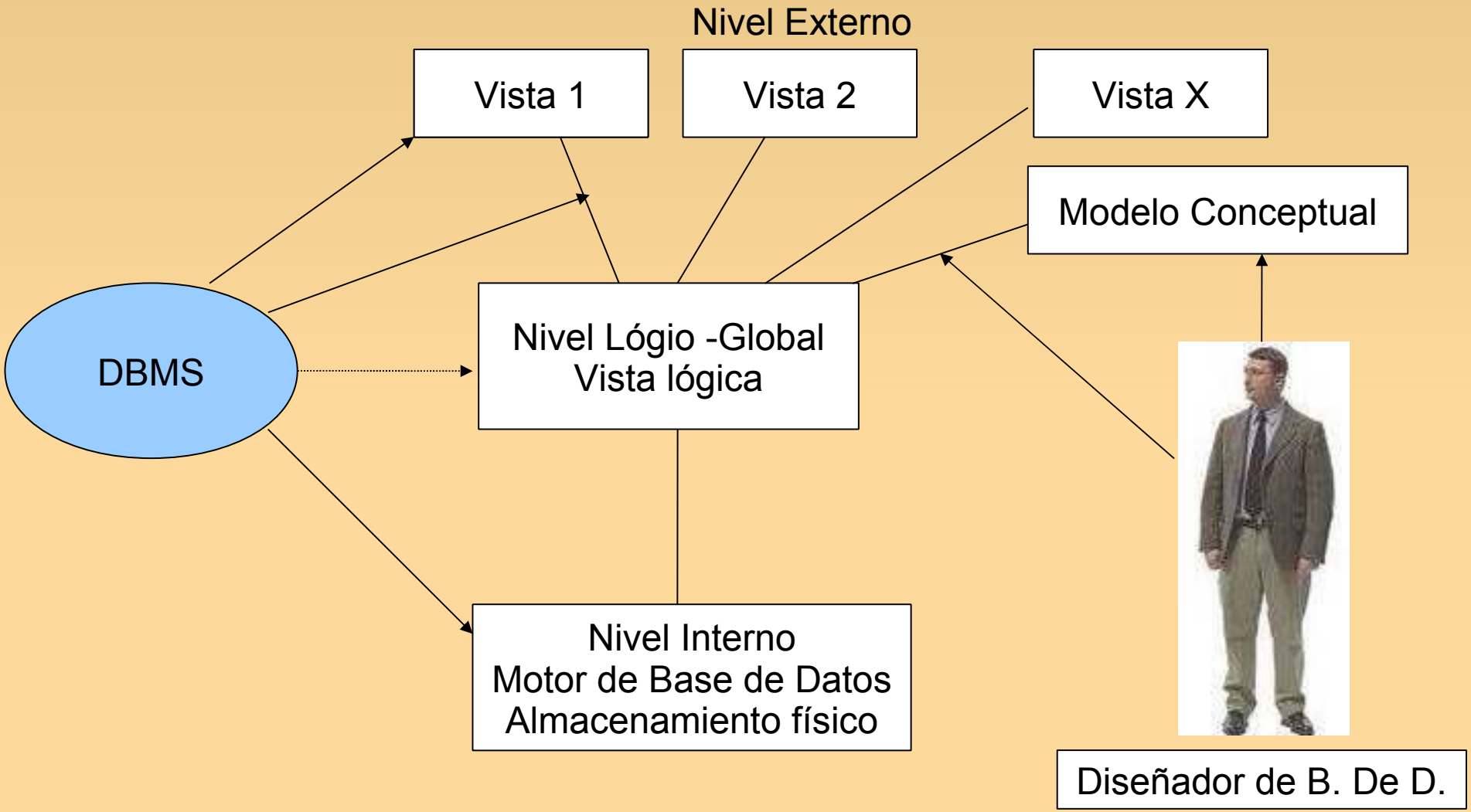
Objetivos

- Conocer la aplicación del Diagrama de Clases en el contexto del Diseño Conceptual de Bases de Datos Relacionales
- Introducir a los alumnos en el uso práctico del Diagrama de Clases para modelar esquemas de bases de datos
- Conocer una alternativa del Diagrama E / R

Diseño Conceptual de Bases de Datos

- Sirve como auxiliar en el Diseño de Base de Datos
- Es un Lenguaje de Alto Nivel
- Permite Identificar Conceptos Semánticos Importantes
- Independiente del DBMS

Vista global



¿Para qué Modelar?

- Permite comunicar conceptos que de otra forma sería dificil
- Es una abstracción de la realidad
- Permite entender lo que estamos construyendo y guiar el proceso
- Documenta las decisiones tomadas

UML

- Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema software.
- Brinda una forma standard de escribir conceptos, procesos de negocios, funciones del sistema, hasta cosas concretas como clases escritas en un lenguaje específico, esquemas de bases de datos y partes reutilizables de software.
- Es extensible, pudiéndose adaptar a necesidades específicas de una aplicación u organización.

Diagrama de Clases

- El diagrama de clases es usado para hacer modelado estructural, pero el enfoque particular que vamos a darle es el Diseño Conceptual de Bases de Datos
- Un diagrama de clases es una vista estática del diseño de un sistema
- Es el diagrama mas comúnmente encontrado en cualquier diseño OO
- Muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones

¿Y el Diagrama E / R no sirve para hacer Diseño Conceptual?

- El Diagrama E / R es el mas ampliamente difundido
- Lo entienden la mayoría de los diseñadores de Bases de Datos
- Existen diversas herramientas CASE que permiten exportar e importar esquemas practicamente sin intervención del usuario

¿Entonces que ventajas existen al utilizar el Diagrama de Clases?

- El Diagrama E / R solo se enfoca en los datos
- El Diagrama de Clases es un Superconjunto del Diagrama E / R
- Permite modelar comportamiento
- Permite utilizar el mismo lenguaje (UML) para modelar el sistema completo

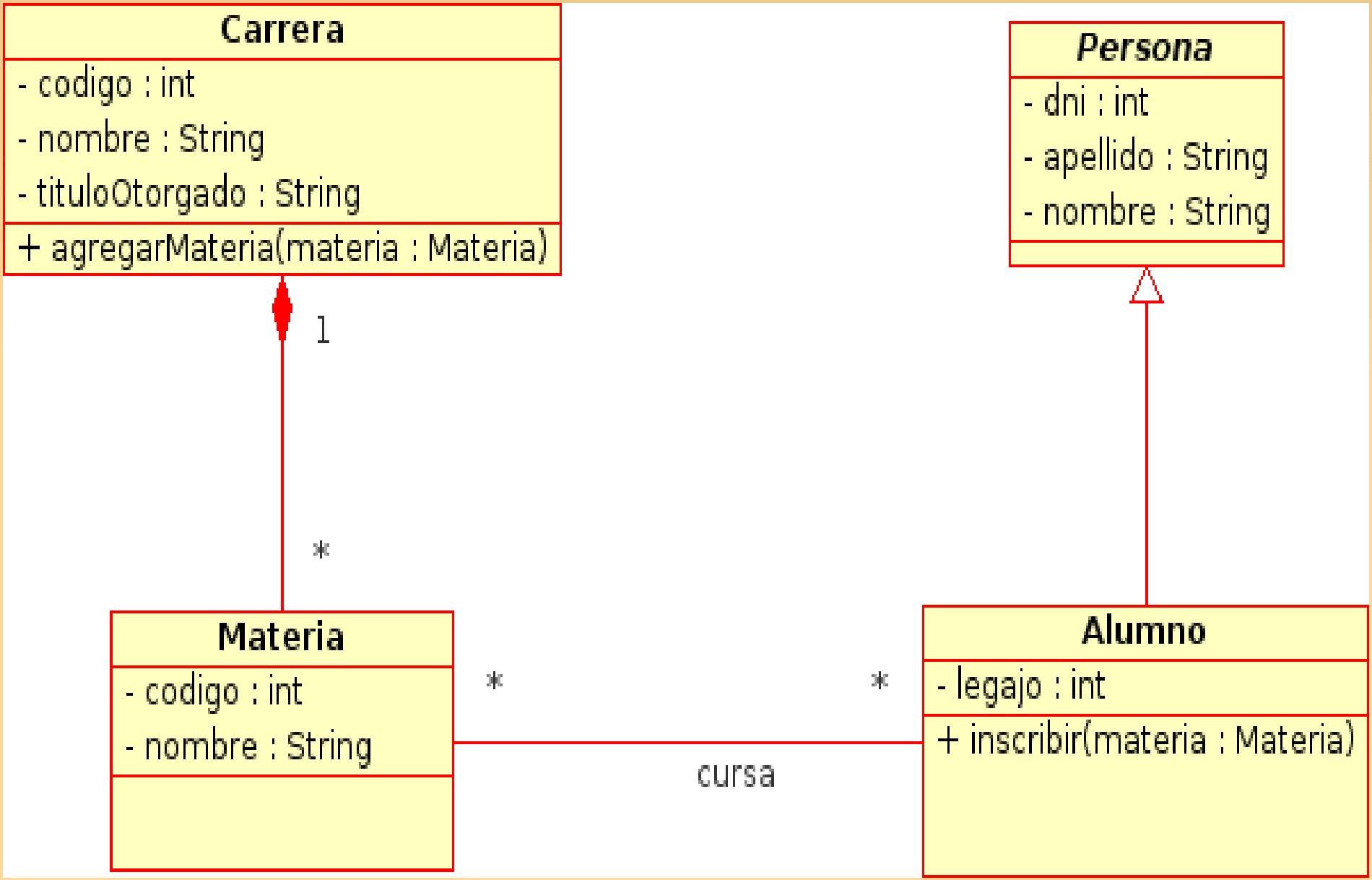
También como consecuencia de esto

- Facilita la comunicación entre analistas, programadores y diseñadores de bases de datos.
- Permite utilizar las mismas herramientas CASE
- Permite utilizar un standard actual de documentación y diseño
- Permite relacionar mejor el sistema en su conjunto y no tenerlo en diseños separados

Visualización de un Diagrama de Clases

- Al ver un diagrama de clases generalmente se pueden apreciar a simple vista un conjunto de cajas unidas por líneas de diversas formas.
- Las cajas son representaciones de Clases o de Interfaces
- Las lineas representan las Relaciones entre las clases, estas relaciones pueden ser de Asociación o Generalización.

Ejemplo de un Diagrama de Clases



Estructura de una Clase

- Se representa mediante una caja con tres divisiones
- En la primera división se indica el nombre de la clase
- La segunda división se especifican los atributos
- La tercera división se especifican las operaciones, las cuales están fuera del alcance del contenido de la materia

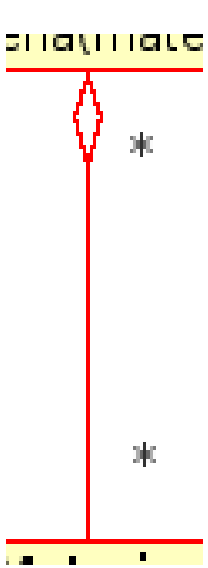
Otras propiedades visibles en una Clase

- Los nombres de clases con estilo de fuente en itálica son clases abstractas. Lo que significa en que no puden instanciarse directamente.
- Los prefijos en los atributos u operaciones + - # indican la visibilidad de los mismos.
    - - privado
    - + público
    - # protegido
- Luego del nombre del atributo o de la operación aparece el nombre de un tipo de dato. En el contexto de las bases de datos relacionales, estos tipos deberían ser tipos simples como int, String, float, boolean.

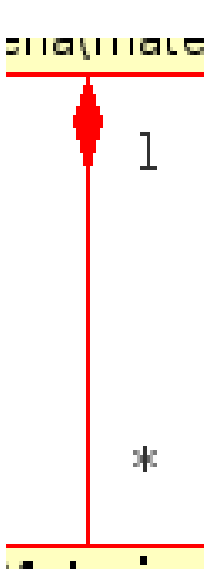
Relaciones

- Se representan mediante líneas de diversos formas y estilos
- Agregan semántica al diagrama
- Pueden ser relaciones de Asociación o de Generalización
- Dentro de la asociación podemos encontrar la asociación propiamente dicha, la agregación, que a su vez puede ser agregación compartida o bien composición

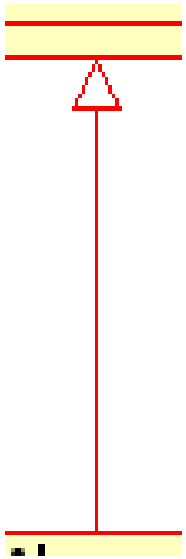
Ejemplos de Relaciones



Agregación



Composición



Generalización

Asociación

Otras propiedades visibles en Relaciones

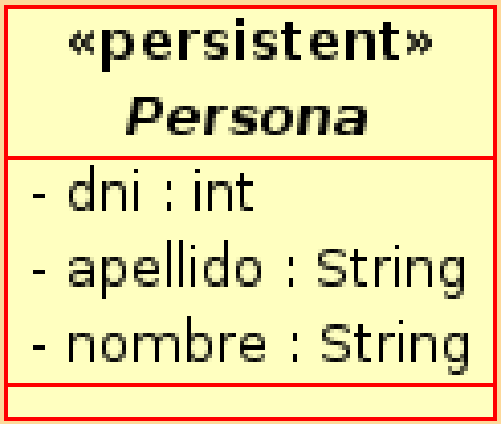
- En los extremos de las relaciones figuran símbolos como 0..1, 1, 1..*, *. Estos indican la multiplicidad, o sea con la cantidad de instancias del objeto que pueden relacionarse
- La multiplicidad es un concepto importante al momento de transformar el diagrama al esquema de la base de datos. Esto define las claves foráneas.
- En la linea puede aparecer un comentario que sirve para darle significado a la relación.

¿Es suficiente con esto?

- Un principio fundamental del Modelo Relacional es que toda relación posee una clave primaria. ¿Como representamos esto?
- ¿Como determina una persona que no participó del diseño del diagrama si se trata de un diagrama de clases de una vista estática de diseño o de un esquema de base de datos?
- ¿Como se puede hacer para documentar otros aspectos a tener en cuenta en la implementación?

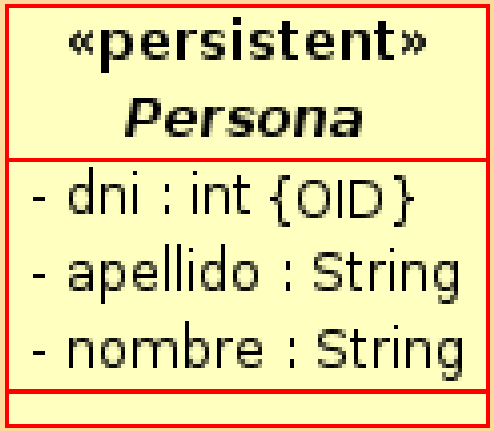
Propiedades Extendidas

- UML es extensible y nos proporciona determinadas técnicas para documentar o extender el lenguaje
- Stereotipos: mediante el uso de una simple anotación podemos documentar otras cosas. Generalmente el stereotipo se lo representa por un texto encerrado entre << >> El stereotipo que indica que una clase representa una estructura en una base de datos es Persistent. Si el stereotipo se encuentra en una relación se lo puede representar entre llaves, por ejemplo {ordered}



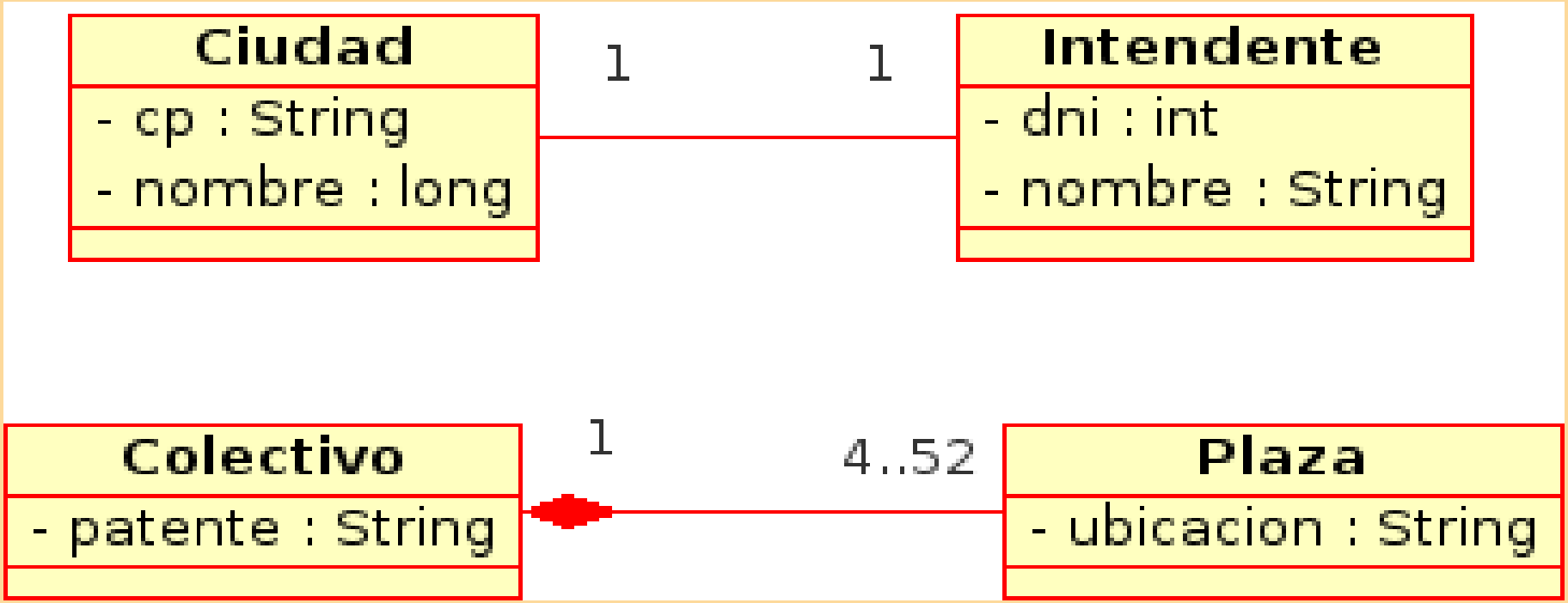
Mas Propiedades Extendidas

- Propiedades: permiten indicar propiedades adicionales a los atributos. Las propiedades se especifican entre llaves y se ponen al final de los atributos.
- El uso mas común que podemos darle en el contexto del diseño conceptual de bases de datos es definir la clave primaria. Por convención se utiliza {OID}
- Tambien se podríamos llegar a indicar si un atributo puede tomar valores nulos {nullable}, o la clave alternativa {alternate OID}



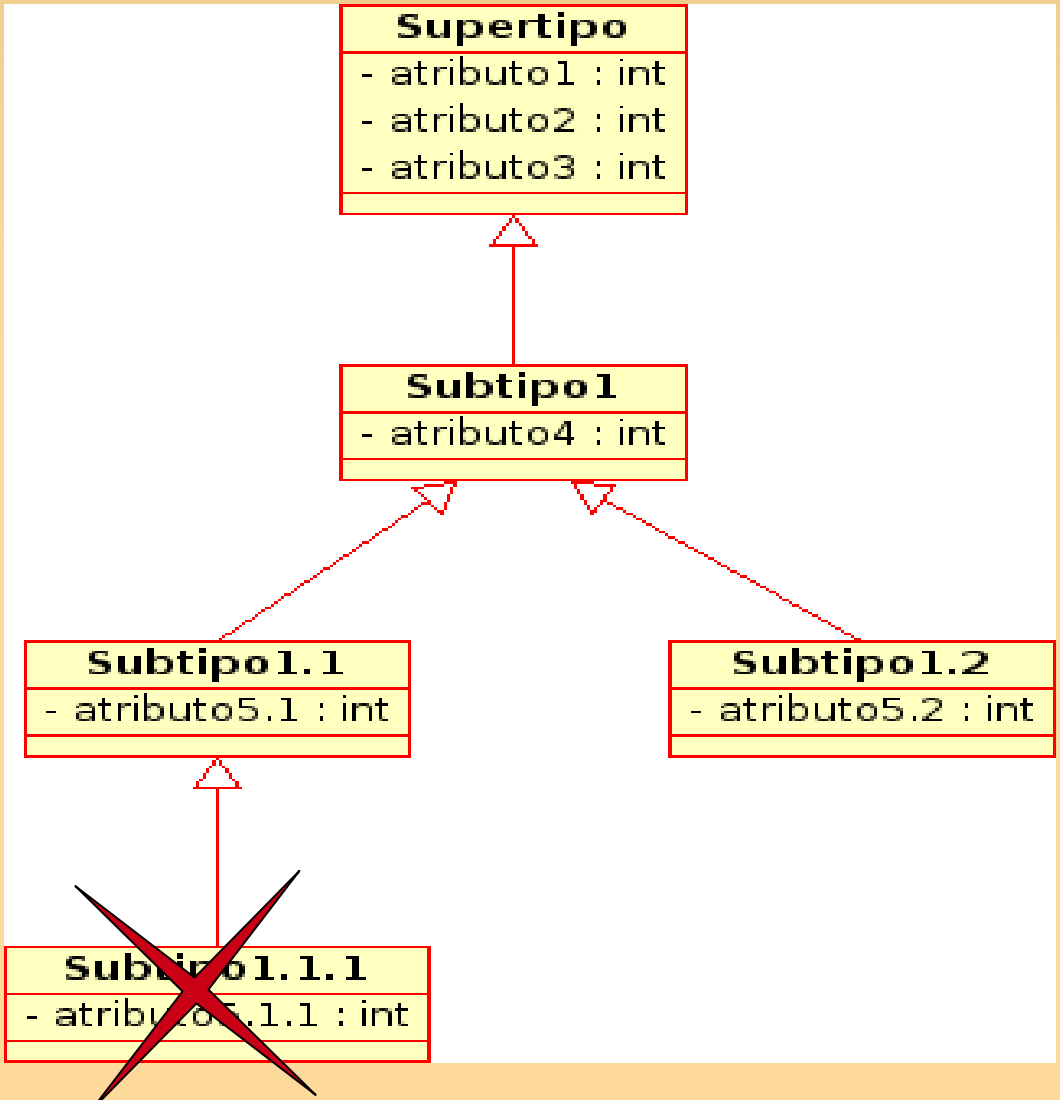
Algunas Consideraciones

- Nombrar las clases en singular
- Nombrar todas las asociaciones n a n. Facilita la transformación
- Evitar las relaciones 1 a 1 y las n-arias



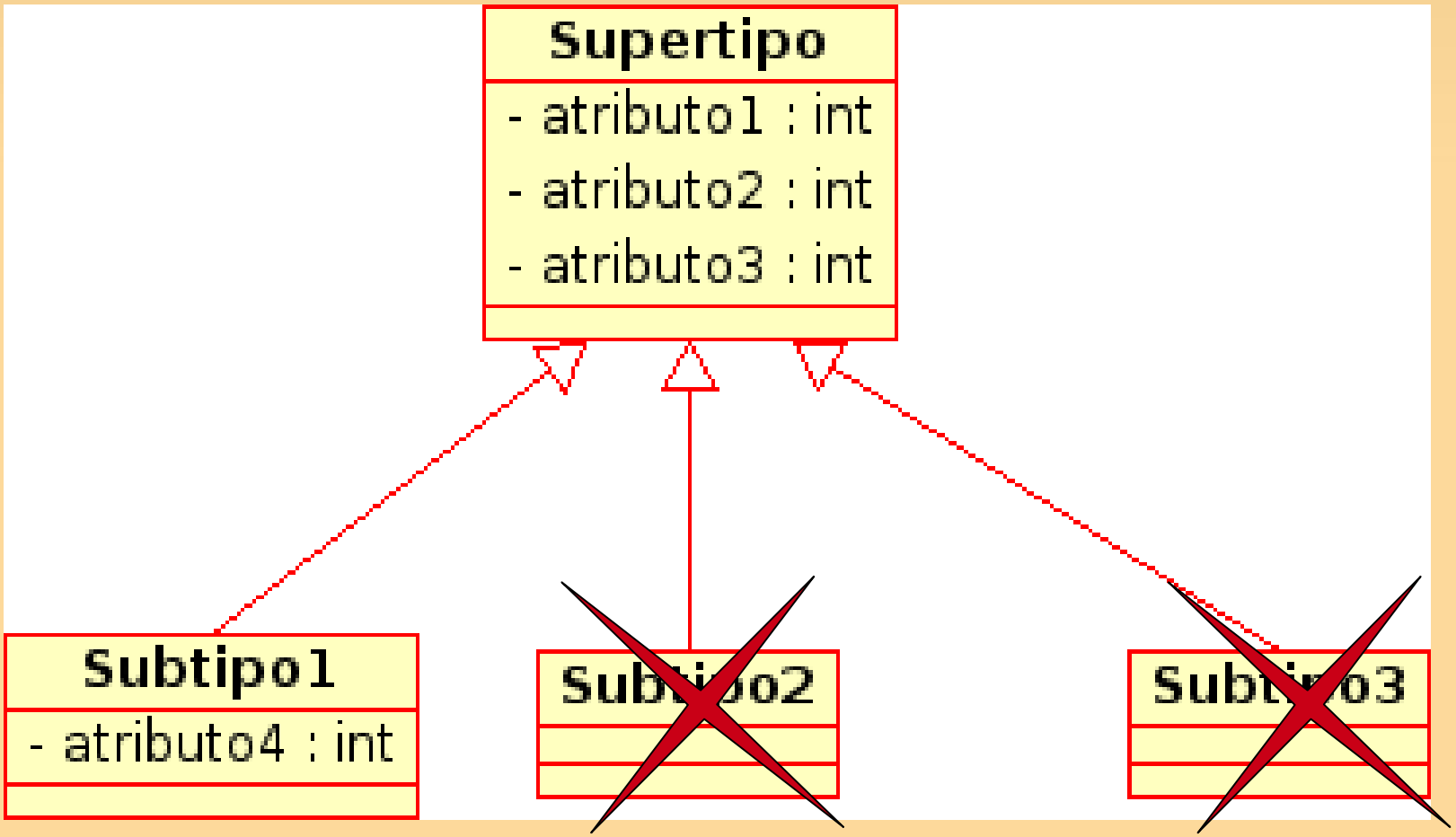
Algunas Consideraciones

- Evitar tener grandes jerarquías de clases, es complicada la implementación



Algunas Consideraciones

- Las clases especializadas sin atributos tienen solamente significado semántico. Al hacer la transformación no se obtiene una relación como resultado



Ejemplo

Se desea almacenar notas sobre el cursado de materias de alumnos de una universidad. De los alumnos nos interesa guardar el numero de legajo, el nombre y la carrera que cursa.

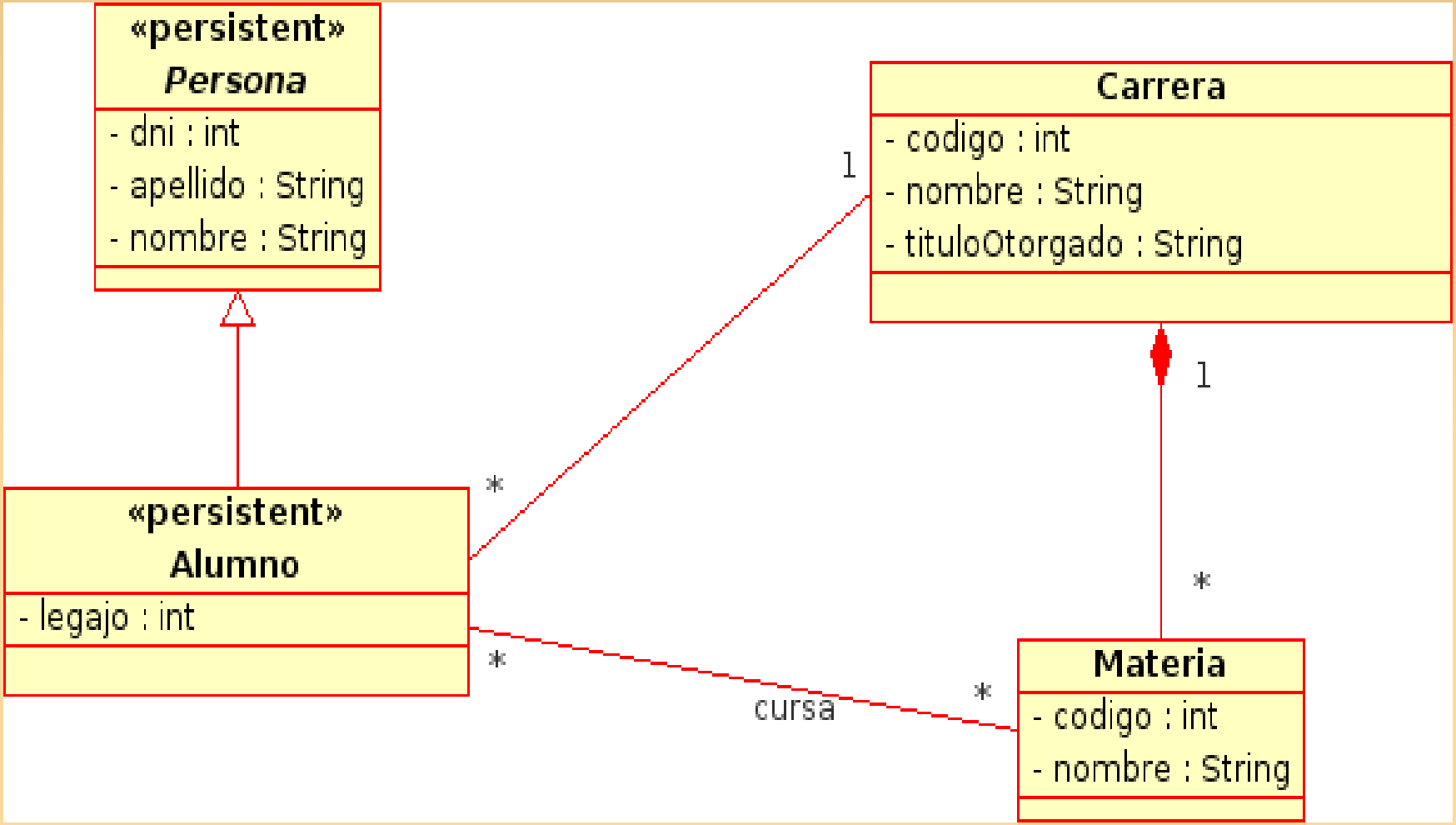
Todos los años se inscribe en un conjunto de materias, y al finalizar de cursarlas obtiene una calificación.

Las materias se dictan en una sola carrera.

El primer paso es identificar las clases, que son cosas de existencia real (una persona) o cosas de existencia conceptual (una materia).

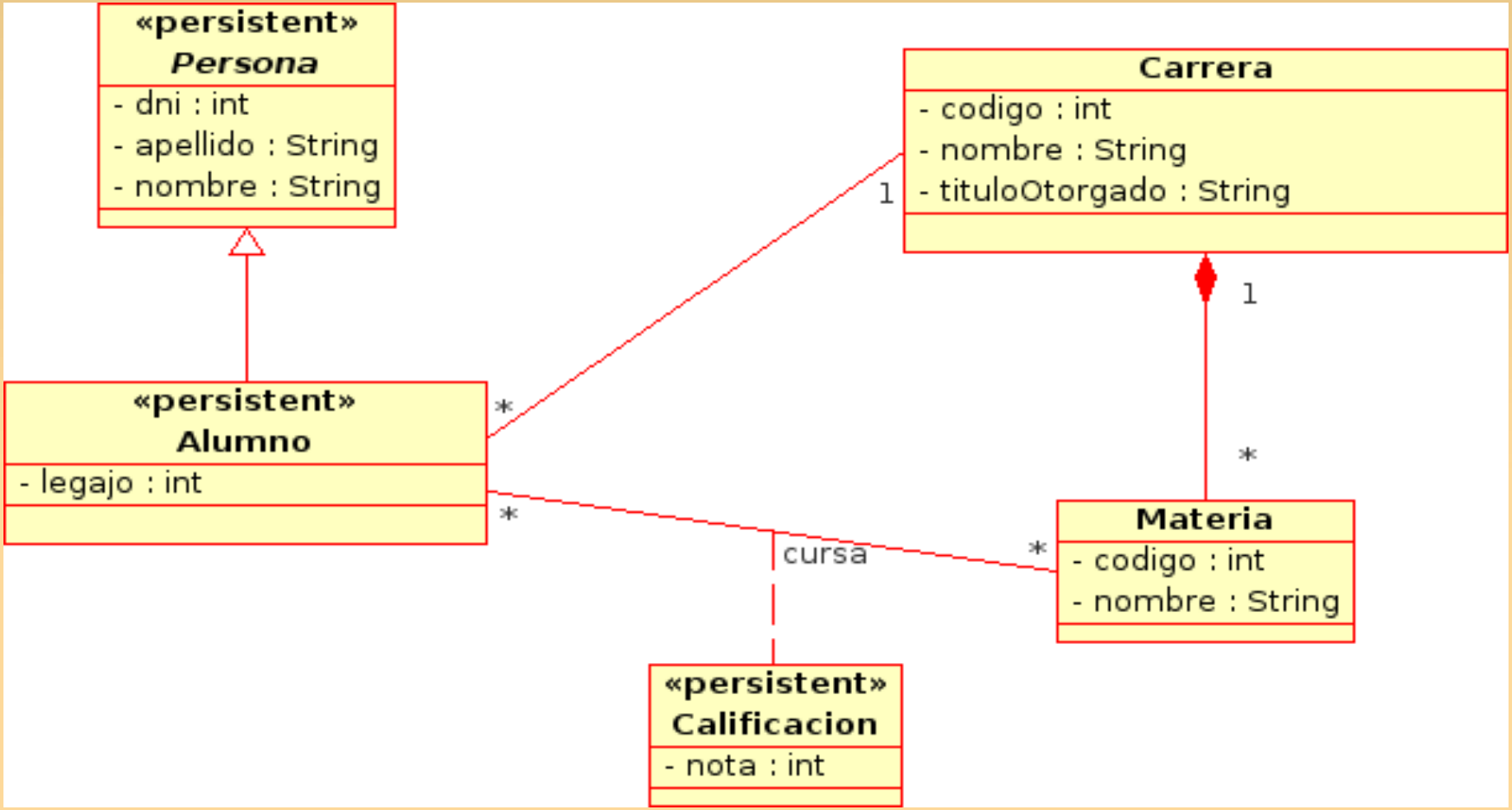
El segundo paso es identificar los atributos y las relaciones entre las clases.

Diagrama de Ejemplo



- ¿Es suficiente con esta representación?
- ¿No está faltando la nota del alumno?

Diagrama de Ejemplo



- Para poder llevar la calificación de un alumno para una materia agregamos una Clase de Asociación
- Esta Clase nos permite agregar atributos a la relación

Limitaciones Diagrama Anterior

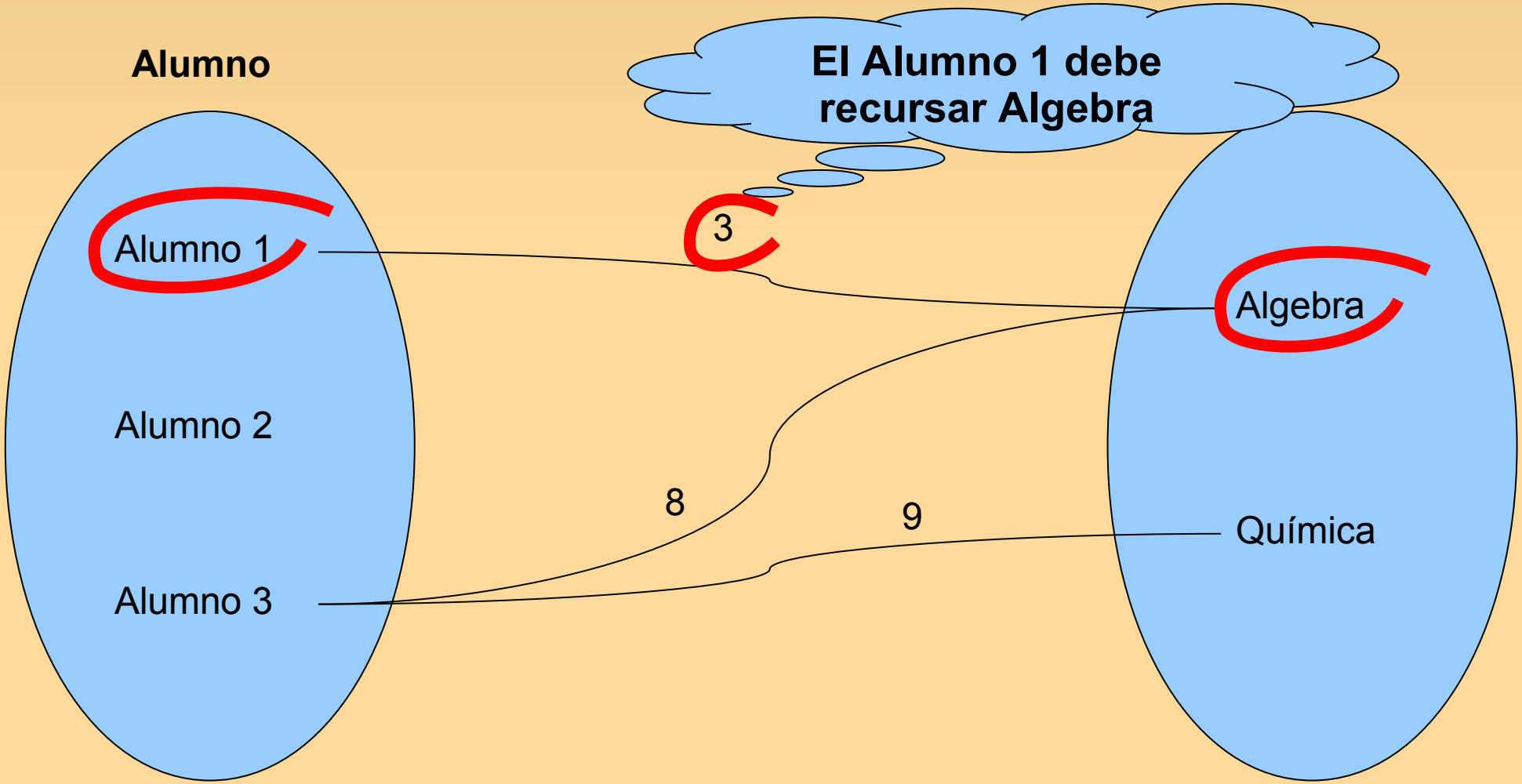
- Cuando se diseña, hay que tener en cuenta ciertas consideraciones que posiblemente no surjan del relevamiento.
- Estas consideraciones generalmente se encuentran implícitas en el problema y juegan un rol fundamental.
- ¿Que sucede si un alumno debe recursar una materia?
- Las relaciones muchos a muchos se pueden dar como máximo entre 2 instancias 1 sola vez

¿Qué significa esto?

- Cuando representamos una Clase, estamos representando una estructura donde se almacenarán instancias de objetos del mismo tipo
- Veamoslo gráficamente
  - Cada uno de los globos representa la colección de objetos de cada tipo
  - Los arcos representan la ocurrencia de una relación entre instancias de diferentes clases
  - El valor asignado al arco representa la nota que obtuvo el alumno en la materia

¿Qué significa esto?

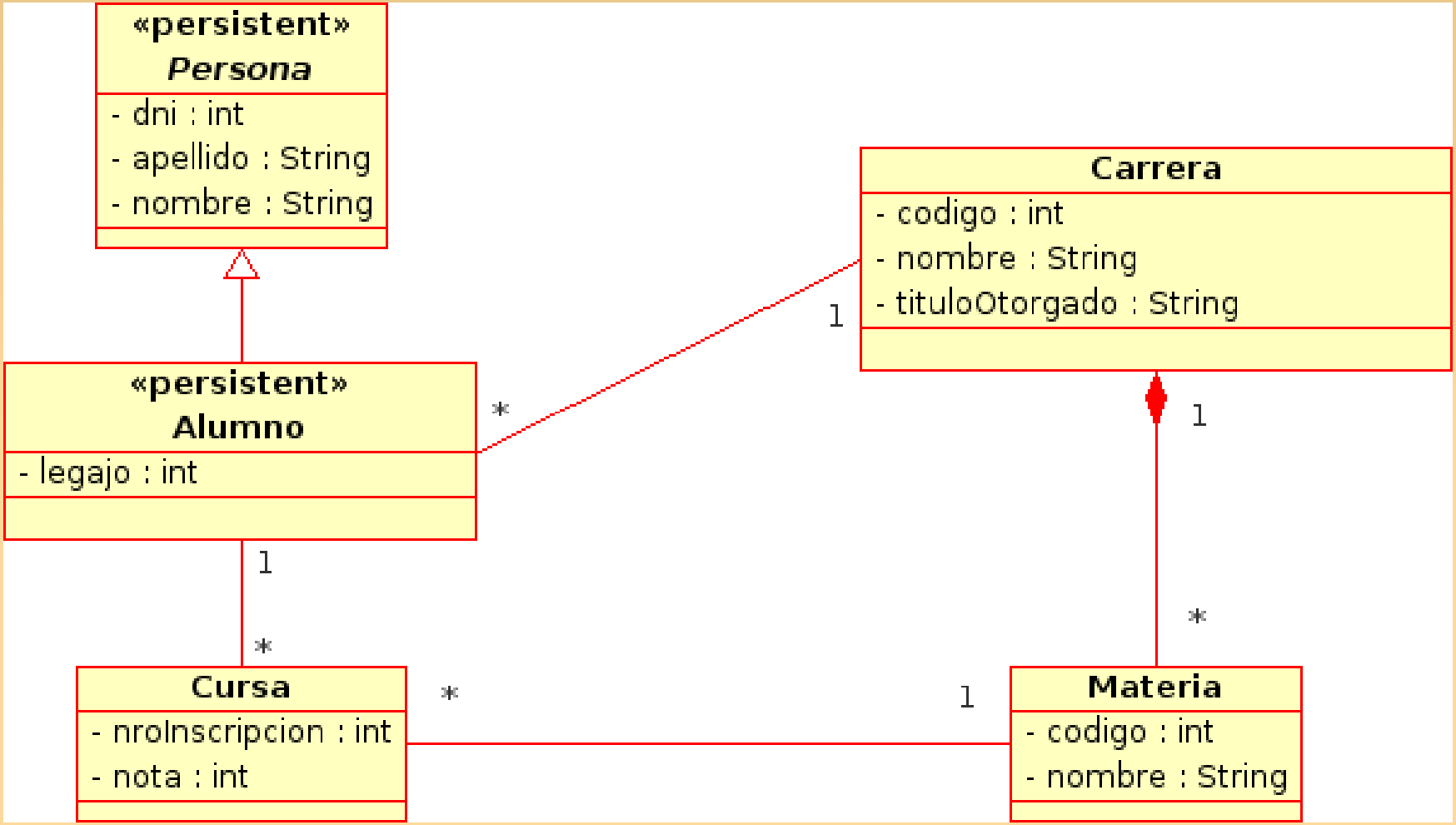
- Cuando representamos una Clase, estamos representando una estructura donde se almacenarán instancias de objetos del mismo tipo



Promover una Clase de Asociación a una Clase Completa

- Una vez que la relación se ha dado, no hay forma de que se repita.
- Esto se traduce en que tengamos que sobreescribir la historia. En nuestro caso perderemos la nota anterior.
- Y en el peor de los casos que se impida cambiar. En nuestro caso no se permitiría recursar una materia.
- Este caso se resuelve promoviendo la Clase de Asociación a una Clase Completa.
- Quiere decir que la nueva Clase va a tener existencia por derecho propio.

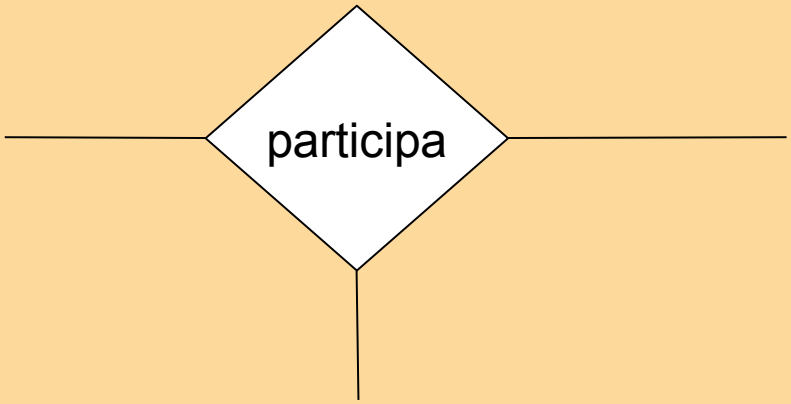
Diagrama de Ejemplo Arreglado



- Nuestra Clase de Asociación Calificación pasó a ser la Clase Cursa.

Relaciones n-arias

- Existen otro tipo de relaciones mas complejas donde la aridad de la relación es mayor a 2
- Es bastante frecuente encontrar relaciones ternarias
- Las relaciones de aridad mayores a 3 son muy poco frecuentes
- Las relaciones n-arias resuelven problemas que no se pueden resolver con relaciones binarias
- La forma de representar una relación n-aria es con un rombo



Piense en el siguiente enunciado

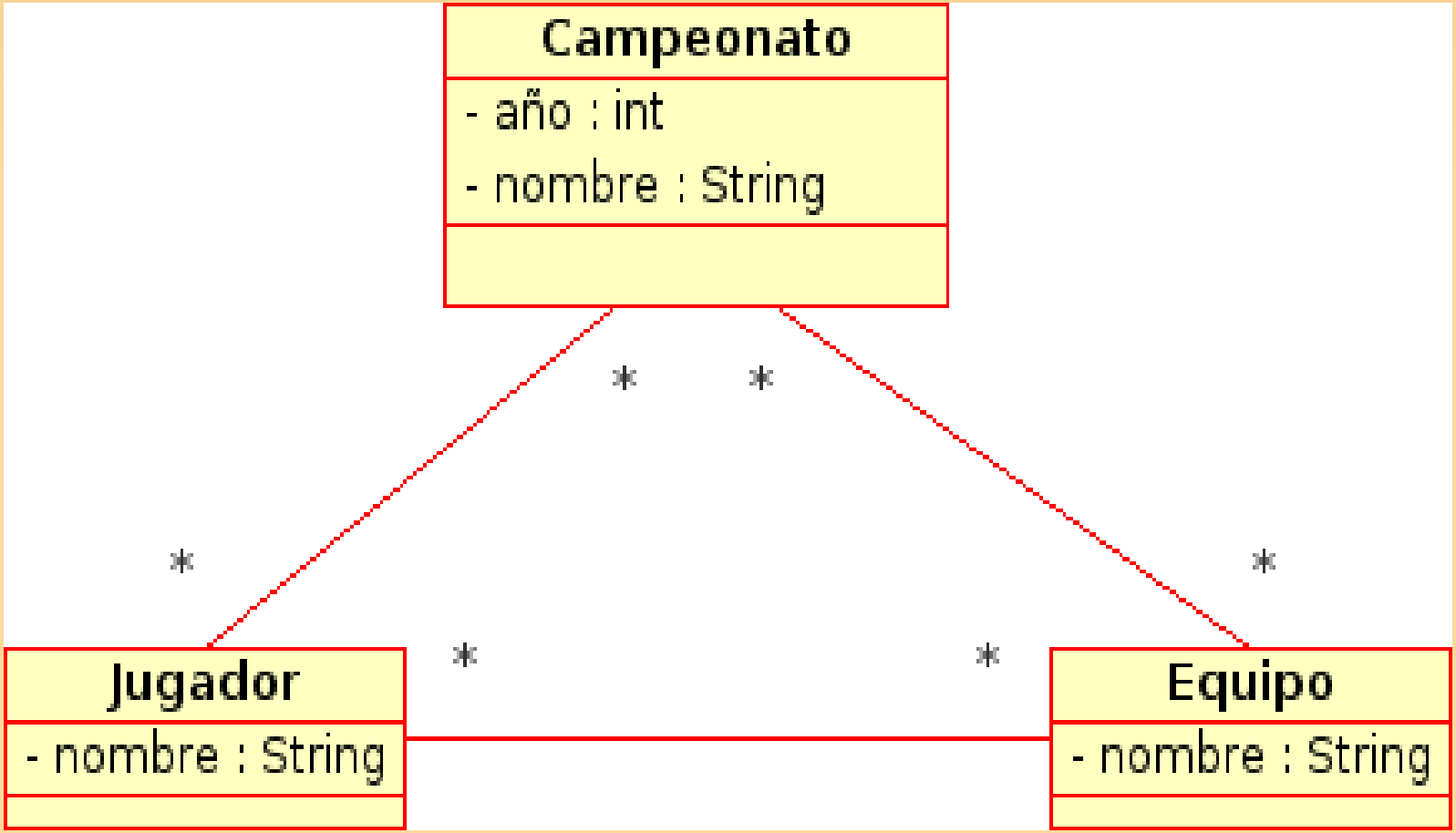
Suponga que se desea almacenar información acerca de campeonatos de fútbol.

Cada campeonato se identifica por su nombre y el año en que se juega.

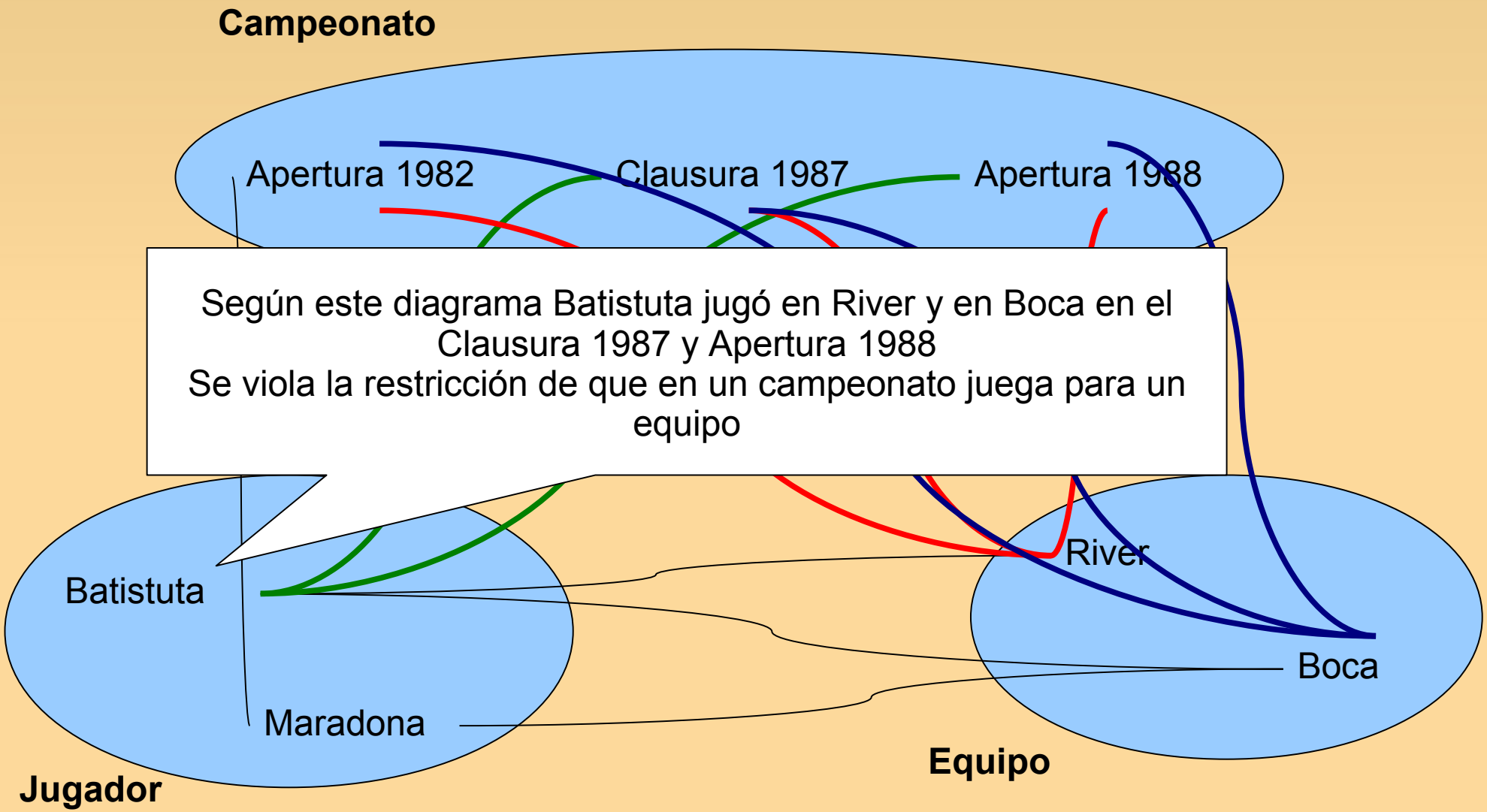
- Del campeonato participan varios equipos
- A cada equipo lo integran varios jugadores
- Los jugadores pueden cambiar de equipo al finalizar un campeonato, antes no

Resolución Incorrecta

- La forma mas sencilla de verla es la siguiente
- ¿Pero si nos preguntamos en qué equipo jugó Batistuta en el campeonato apertura de 1988?

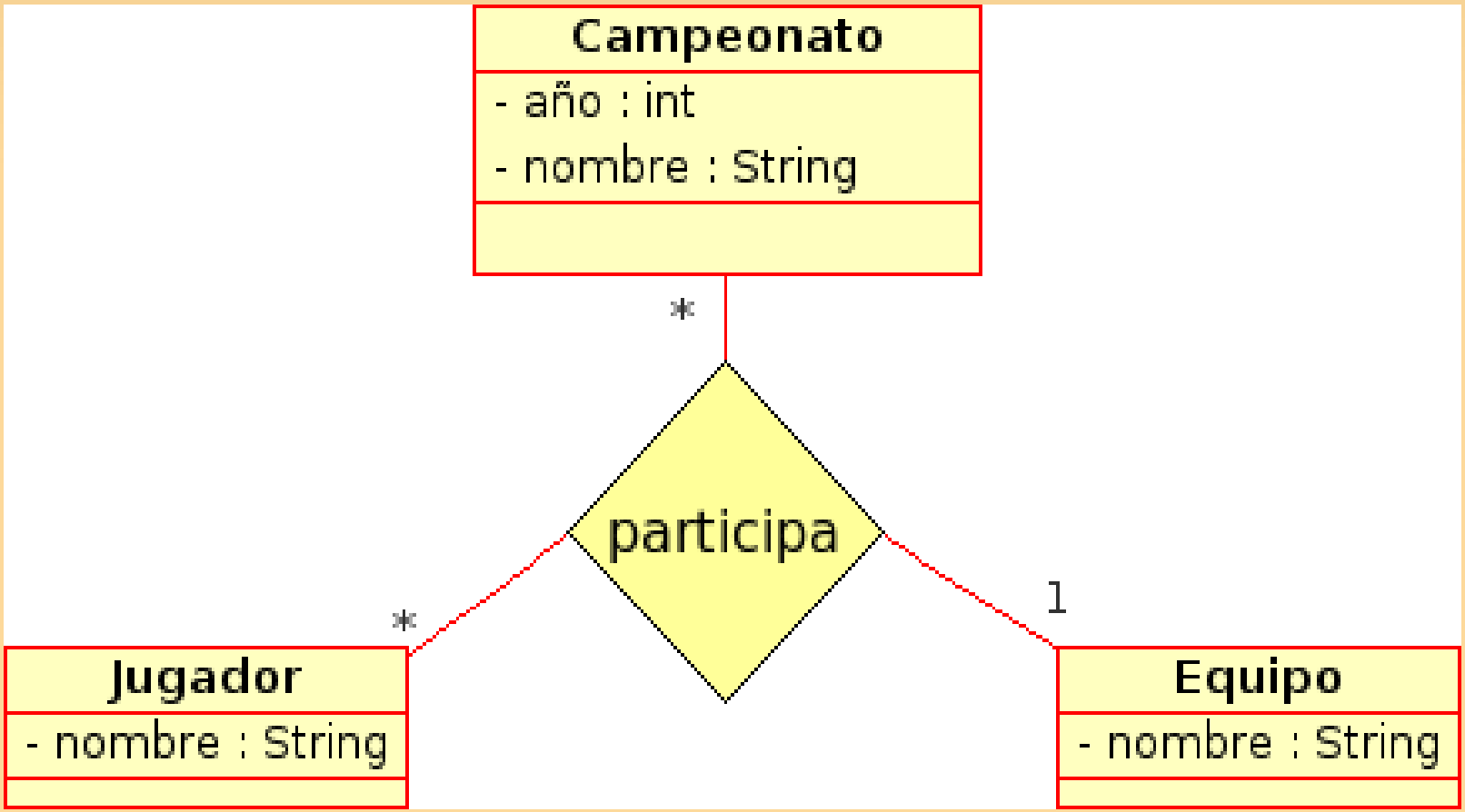


Por transitividad podemos sacar la información adicional



Resolución Correcta

- Con una relación ternaria se soluciona el problema
- Fijense que también se agregó la restricción mencionada anteriormente



CONCLUSIONES

- Conocimos una forma alternativa al Diagrama E / R para modelar conceptualmente una base de datos
- Vimos algunas propiedades a tener en cuenta para identificar que el Diagrama de Clases representa un modelo conceptual de datos
- También vale mencionar que lo visto es solo una parte de las posibilidades del Diagrama de Clases. Eso fue simplificado lo mas posible para facilitar el aprendizaje y su aplicación
- Vimos algunos ejemplos de resolución de casos problemáticos puntuales.

PREGUNTAS