BASES DE DATOS Y SISTEMAS DE INFORMACIÓN

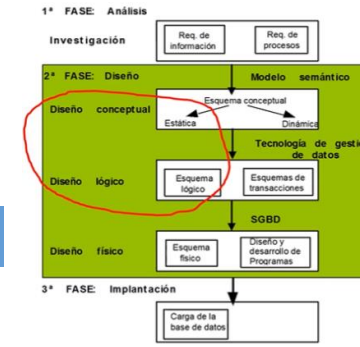
Tema 4.2

Miquel Gómez Corral Ing. Informática ETSINF, UPV

Tema 4.2

Diagrama de Clases de UML

Es un poco lo de ISW y IPC. Pero solo nos vamos a fijar en el diseño conceptual y el diseño lógico estáticos. Diferenciamos entre estática y dinámica.



- **Estática**: Dice cómo se van a almacenar los datos.
- **Dinámica**: Como se van a usar los datos.

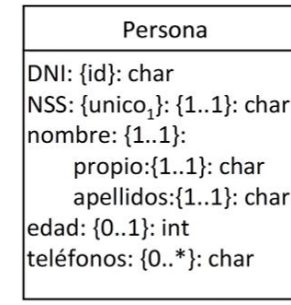
DISEÑO CONCEPTUAL

Fase del diseño de una BD cuyo objetivo es "obtener una representación de la realidad que captura las propiedades estáticas y dinámicas de la misma que son necesarias para satisfacer los requisitos: Esta representación debe suponer una imagen fiel del comportamiento real"

Para la representación se usará UML, representación gráfica. LO DE ISW

Clases

Son los objetos de los que se quiere almacenar información. Si son del mismo tipo están en la = clase: Ej persona. Las clases tienen atributos y acciones. *SOLO NOS FIJAREMOS EN LOS ATRIBUTOS.*



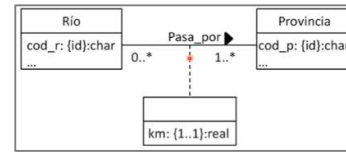
Atributos: Tendremos que indicar el **tipo** y las **restricciones** (*unicidad*, *cardinalidad*, *identificación*). Puede ser "**derivado**", si **NO se almacena** sino se calcula en base a algo.

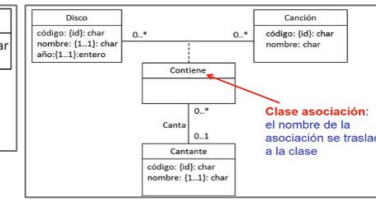
- **Unicidad** que solo haya 1 con ese valor. **Cardinalidad** 1..*, 0..1…. **Identificación** que cual es clave primaria.
- **Formado por**: Un atributo puede estar formado por otros atributos. Por ejemplo nombre, está formado por un "propio" y un "apellidos".

Asociaciones

Representan **relaciones existentes en el sistema de información** "SI". Un objeto está relacionado con otro. A las asociaciones se les puede poner: **Sentido ->**, **nombre R**, **Rol** de cada tabla y **cardinalidad**. *Lo mismo que ISW.*

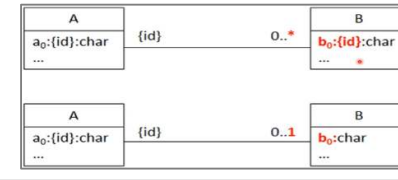
Enlace: Cuando queremos **guardar información sobre una relación** añadimos a la línea de la relación una caja con los atributos que queramos guardar. Esta nueva relación puede tener nombre y conectar con otra tabla si es que la relación contiene información de esta





Tipos de Clases

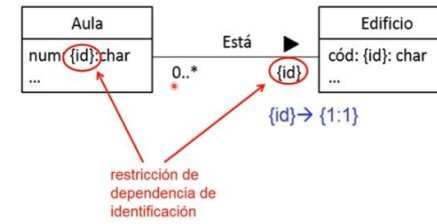
Débil: Sufre "***restricción de dependencia de identificación***". Esto pasa cuando no se puede identificar una instancia de dicha clase solo con sus atributos. Le hace falta atributos de otra clase (como claves ajenas). *Hace falta ver sus atributos y con quien se relaciona para poder identificarla*.



Si es de uno a muchos pones "id", sino no hace falta xq lo identifica la otra clase

Ejemplo

Los números de las aulas no son suficientes para identificar una aula xq hay una aula 0.1 aquí y otra en el edificio de al lado. Entonces hace falta relacionar las aulas con los edificios y decir que la ID de este edificio es necesaria para identificar el aula: ***La clase aula es Débil***.

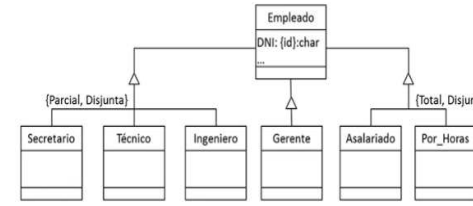


Especializada/Generalización *Herencia*

Se produce cuando hay una clase que incluye a otra → **una es subclase de la otra / varias clases se generalizan en una superclase**. *Las subclases tienen los atributos de las superclase + de los suyos propios.* Solo la super tiene ID.

Definición total y parcial: **Total** es como **abstract**, todas las instancias de la superclase se han de especificar como una de las subclases. **Parcial** puede haber ocurrencias de las superclase y de las subclases.

Disjunta y solapada: Una ocurrencia de la super se especialice en varias subclases. Solapada lo contrario.



En el ejemplo se hace ver que empleado puede especializarse en varias cosas: **Según el tipo** (*al ser parcial puede ser que un empleado sea de alguno de esos tipo, pero que NO lo seas de varios a la vez*), según si es **asalariado** o por **hora** (*al ser total sí o sí se ha de especificar*) y opcionalmente si es **gerente**.

Restricciones de integridad o condiciones extra

Si hay algunas propiedades o cosas que no se pueden representar con lo de antes, tenemos que usar unas notitas para indicar que cierto atributo ha de cumple x, o cierta clase ha de cumplir y.