

Diagrama de Clases de UML

Corrección del diagrama de clases de UML

Curso 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

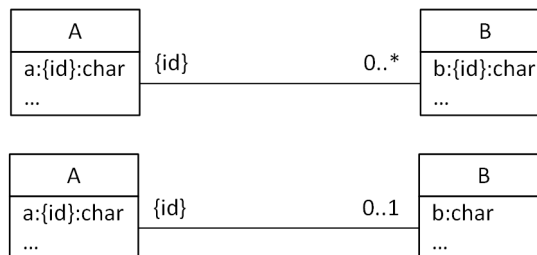
1 INTRODUCCIÓN

En este documento os listo una serie de reglas que debe cumplir un diagrama de clases para ser sintácticamente correcto cuando es utilizado en el proceso de diseño de una base de datos relacional para representar el esquema conceptual del sistema de información es su vertiente estática, para que os sea útil, debéis tener claro todos los elementos del diagrama de clases ya que aquí no se va a incluir la explicación de éstos. Sería bueno que reflexionaraís sobre el significado de cada regla.

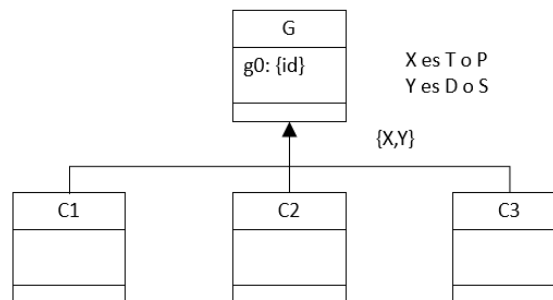
Obviamente, el no tener errores sintácticos no asegura la corrección semántica de diagrama. Antes de presentar las reglas os incluyo unos comentarios que me parecen importantes.

En un diagrama de clases UML, una clase puede ser de uno de los tipos siguientes:

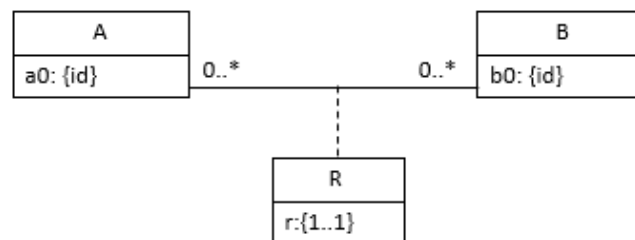
- **Débil:** en este caso alguna de las asociaciones en las que participa la clase tiene la etiqueta {id} en la otra clase. En la figura, B es una clase débil en los dos diagramas.



- **Especializada:** en este caso la clase está conectada mediante una flecha a otra clase. En la figura, C1, C2 y C3 son clases especializadas. Se dice que G es una superclase de C1, C2 y C3.

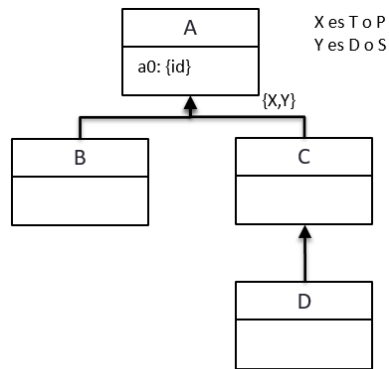


- **Clase-asociación:** en este caso la clase está conectada a una asociación mediante una línea discontinua. En la imagen, R es una clase asociación.



- **Fuerte:** clase que no está en ninguno de los tres casos anteriores.

En el caso de una especialización a varios niveles:



diremos que:

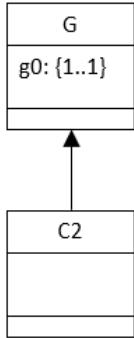


- A es una superclase (o antepasado) de B y de C.
- C es una superclase (o antepasado) de D y por tanto también A es una superclase (o antepasado) de D.
- A es la clase raíz de B, C y D.

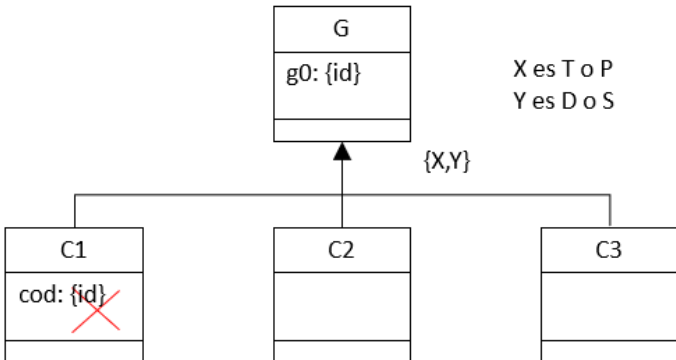
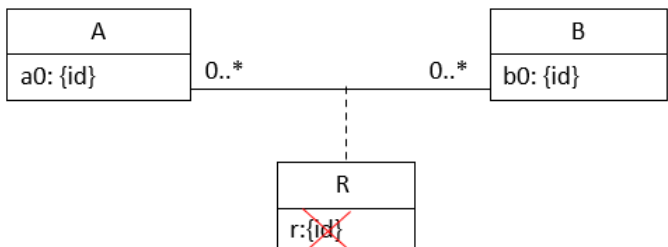
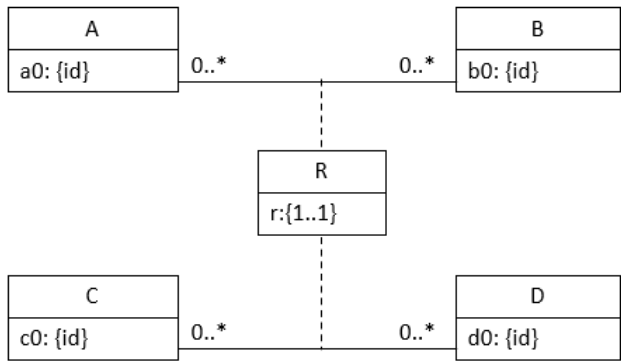
2 REGLAS

Reglas generales		
Nombre regla		Propiedad que debe cumplir el diagrama
G	01	No puede haber dos nombres iguales entre las clases, las asociaciones ni los roles.

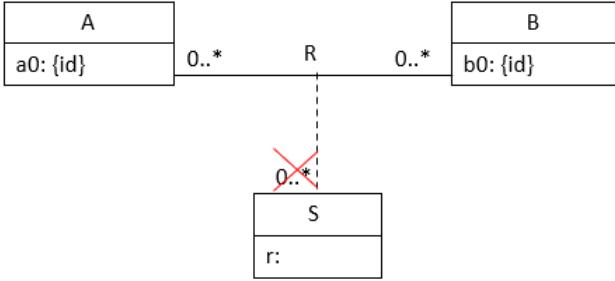
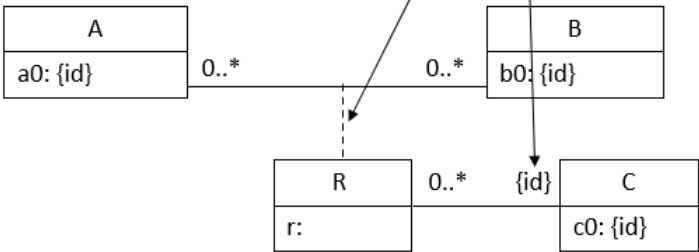
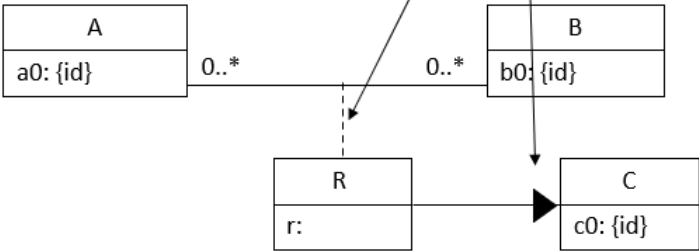
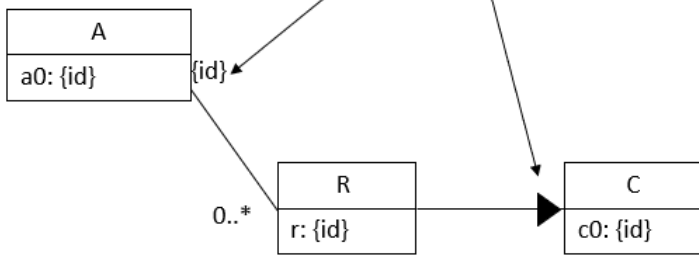
Reglas sobre atributos		
Nombre regla		Propiedad que debe cumplir el diagrama
A	01 ¹	Todos los atributos deben tener un tipo de datos asociado.
A	02	En una clase no puede haber dos atributos con el mismo nombre.
A	03	En una clase no puede haber un atributo con el mismo nombre que un atributo de una de sus superclases.
A	04	Todo atributo tiene que tener cardinalidad excepto en el caso de que ésta sea 0..1 (aun así se recomienda hacerlo explícito en todos los casos).

¹ En los diagramas que voy a incluir como ejemplo en este documento, no cumplo esta regla, pero antes de dar por definitivo el diagrama de clases y pasar a realizar el diseño lógico habría que incluir los tipos de datos.

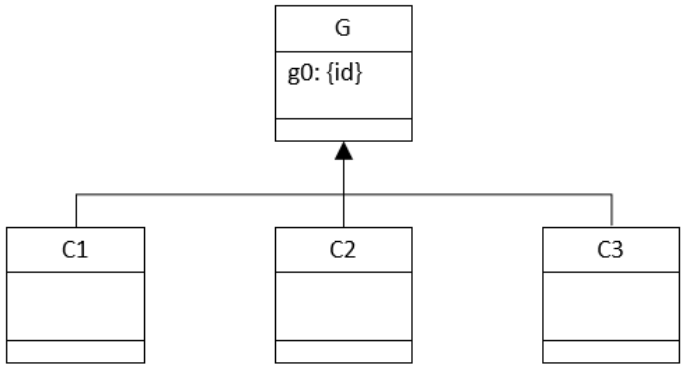
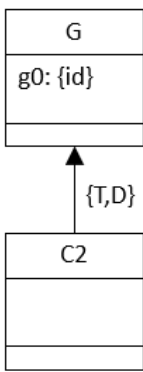
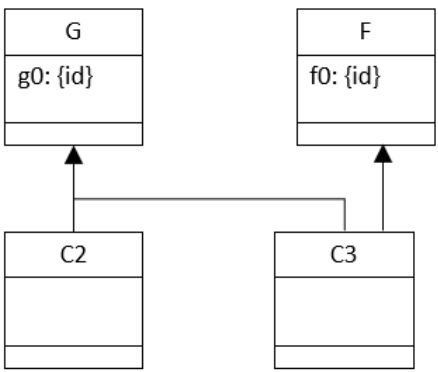
Reglas sobre clases		
Nombre regla		Propiedad que debe cumplir el diagrama
C	01	<p>Si una clase no es débil, ni especializada ni clase-asociación entonces es una clase fuerte y debe tener identificador (conjunto de atributos con la etiqueta {id}). Con esta regla aseguramos que toda clase es de uno de los cuatro tipos enumerados (débil, especializada, clase-asociación o fuerte).</p> <p>Esquema incorrecto: la clase G no es ni débil, ni especializada, ni clase-asociación y no tiene identificador.</p> 
C	02	<p>Las clases débiles no pueden tener identificador (llamado semiidentificador) cuando alguna de las clases con las que se conecta con la etiqueta <i>id</i> tiene cardinalidad máxima a 1.</p> <p>Esquema incorrecto: B es débil y la cardinalidad máxima de A es 1</p> 
C	03	<p>Las clases débiles que se conectan a una sola clase con la etiqueta <i>id</i> deben tener identificador (llamado semiidentificador) cuando esa otra clase tiene cardinalidad máxima a *.</p> <p>Esquema incorrecto: B es débil, la cardinalidad máxima de A es * y B no tiene semiidentificador</p> 


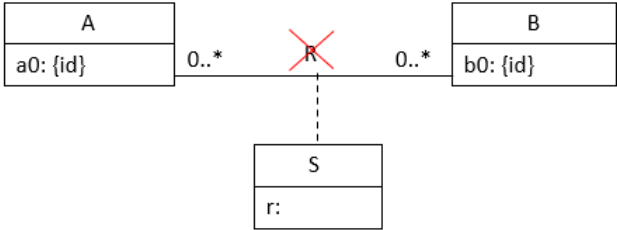
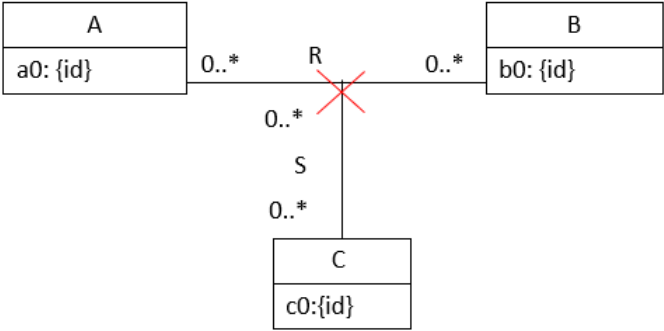
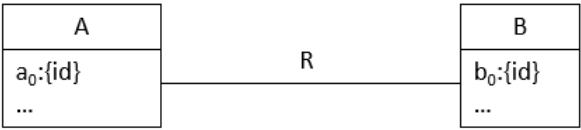
Reglas sobre clases		
Nombre regla		Propiedad que debe cumplir el diagrama
C	04	<p>Las clases especializadas nunca tienen identificador.</p> <p>Esquema incorrecto: C1 es una clase especializada y tiene identificador</p>  <pre> classDiagram class G { g0: {id} } class C1 { cod: {id} } class C2 class C3 G < -- C1 G < -- C2 G < -- C3 </pre> <p>X es T o P Y es D o S</p>
C	06	<p>Las clases-asociación nunca tienen identificador</p> <p>Esquema incorrecto: R es una clase-asociación y tiene identificador</p>  <pre> classDiagram class A { a0: {id} } class B { b0: {id} } class R { r: {id} } A "0..*" -- "0..*" B A -.- R B -.- R </pre>
C	07	<p>Una clase asociación sólo puede conectarse con una asociación mediante una línea discontinua como mucho.</p> <p>Esquema incorrecto: La clase R tiene dos conexiones con líneas discontinuas.</p>  <pre> classDiagram class A { a0: {id} } class B { b0: {id} } class C { c0: {id} } class D { d0: {id} } class R { r: {1..1} } A "0..*" -- "0..*" B C "0..*" -- "0..*" D A -.- R B -.- R C -.- R D -.- R </pre>

Reglas sobre clases

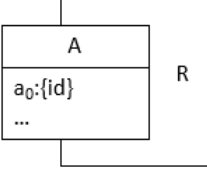
Nombre regla	Propiedad que debe cumplir el diagrama
C 08	<p>La línea discontinua de una clase asociación no tiene cardinalidades.</p> <p>Esquema incorrecto: La línea discontinua no es una asociación.</p> 
C 09	<p>Una clase no puede ser a la vez clase-asociación y débil, ni clase-asociación y especializada, ni débil y especializada</p> <p>Esquema incorrecto: R es una clase-asociación y débil</p>  <p>Esquema incorrecto: R es una clase-asociación y especializada</p>  <p>Esquema incorrecto: R es una débil y especializada</p> 

Reglas sobre clases									
Nombre regla		Propiedad que debe cumplir el diagrama							
C	10	<p>En una clase no pueden aparecer nunca atributos que hagan referencia a otra clase.</p> <p>Si en el diagrama, con el atributo b_0 de A se pretende representar una relación con B, esto es incorrecto. Ese atributo debe eliminarse de A y definirse una asociación con B con la cardinalidad adecuada en cada caso.</p> <div><table><tr><th>A</th></tr><tr><td>$a_0:\{id\}$</td></tr><tr><td>...</td></tr><tr><td>$b_0:\{1..1\}: tipo_b_0$</td></tr></table><table><tr><th>B</th></tr><tr><td>$b_0:\{id\}: tipo_b_0$</td></tr><tr><td>...</td></tr></table></div>	A	$a_0:\{id\}$...	$b_0:\{1..1\}: tipo_b_0$	B	$b_0:\{id\}: tipo_b_0$...
A									
$a_0:\{id\}$									
...									
$b_0:\{1..1\}: tipo_b_0$									
B									
$b_0:\{id\}: tipo_b_0$									
...									

Reglas sobre la especialización		
Nombre regla		Propiedad que debe cumplir el diagrama
E	01	<p>Toda especialización debe llevar las propiedades Total o Parcial y Disjunta o Solapada excepto en el caso de que sólo haya una clase especializada en cuyo caso no es necesario ponerlo.</p> <p>Esquema incorrecto: faltan las propiedades de la especialización T o P y D o S</p>  <p>Esquema incorrecto: cuando sólo hay una especialización, las propiedades son P y S</p> 
	02	<p>Una clase especializada sólo puede tener una clase raíz.</p> <p>Esquema incorrecto: la clase C3 tiene dos clases raíz</p> 

Reglas sobre las asociaciones		
Nombre regla	Propiedad que debe cumplir el diagrama	
X	01 ²	<p>Una asociación siempre tiene un nombre excepto si de ella cuelga una clase asociación, en cuyo caso no debe tener nombre.</p> <p>Esquema incorrecto: La asociación entre A y B no tiene nombre.</p>  <p>Esquema incorrecto: La asociación entre A y B tiene nombre y de ella cuelga una clase asociación.</p> 
X	02	<p>Una asociación siempre conecta dos clases.</p> <p>Esquema incorrecto: R no es una clase. S es incorrecta</p> 
X	03	<p>Toda asociación debe llevar las cardinalidades especificadas, teniendo en cuenta las siguientes simplificaciones (que aconsejo evitar)</p> <ul style="list-style-type: none"> • $* \cong 0..*$ • $1 \cong 1..1$ <p>Esquema incorrecto: faltan las cardinalidades de A y B en R</p> 

² Esta regla no aparece en el documento UD4.2, pero sí en el estándar UML.

Reglas sobre las asociaciones		
Nombre regla		Propiedad que debe cumplir el diagrama
X	04 ³	<p>Toda asociación reflexiva debe tener nombres de roles en ambos sentidos.</p> <p>Esquema incorrecto: R es una asociación reflexiva y no tiene nombres de roles</p> 

³ Esta regla no aparece en el documento UD4.2, pero sí en el estándar UML.