

ERRORES FRECUENTES DIAGRAMAS DE CLASES

1. Nomenclatura¹:
 - a) Nombres de clase en plural
 - b) No comienza por mayúscula
 - c) Nombres de clase con caracteres ilegales: espacios en blanco, acentos, eñes, comienza por cifra o contiene sólo cifras.
 - d) No separa correctamente palabras²
 - e) Cometer errores similares en los nombres de atributo y método, que deben además comenzar por minúscula.
2. No elegir correctamente las clases, poner atributos como clases o no poner todas las clases.³
3. Elegir mal los símbolos de relación:
 - a) Poner agregación o composición (rombos blanco y negro) cuando se trata de una generalización ("es un"). ⁴
 - b) No poner nombre a una relación genérica (flecha o línea continua normal, sin rombo, sin triángulo). ⁵
 - c) Poner la relación en sentido opuesto al correcto, es decir, poner el rombo o el triángulo o la flecha en el extremo contrario.
 - d) No usar correctamente la relación de dependencia⁶
4. No poner las multiplicidades en las relaciones o ponerlas en un sólo extremo o ponerlas en StarUML en un apartado que no es "Multiplicity" dentro de "Properties".
5. Poner nombre a las relaciones de Generalización, Agregación, Composición o Dependencia.⁷
6. Poner nombres de atributos con verbos, en lugar de con sustantivos
7. Poner nombres de métodos con sustantivos en lugar de con verbos.
8. Poner métodos en la casilla de atributos y viceversa.
9. No poner los tipos (Integer, Boolean, ...) de los atributos.
10. Poner varios atributos en lugar de un array.⁸
11. No poner tipo de retorno de un método que lo requiere.⁹
12. No poner argumentos a un método.¹⁰

¹ La nomenclatura es importante porque del diagrama de clases, StarUML y el resto de herramientas UML generan automáticamente un esquema del programa JAVA con la definición de las clases. Si se ponen nombres ilegales, el código generado dará errores de compilación.

2 Cada palabra en un nombre de clase debe comenzar por mayúscula: EJ: **AnimalDeTiro**

3 En el enunciado los sustantivos serán nombres de clase o nombres de atributo. Será una clase cuando podemos deducir para ella unos atributos y métodos y además se ve que la clase va a tener varias instancias. La clase **Perro**, tendrá varias instancias, cada una con valores de atributo diferentes: Perro.nombre puede valer: "Toby", "Cuqui", "Chispa", para cada instancia de **Perro**. Los sustantivos serán atributos cuando sólo puede atribuírseles un valor: nombre de **Perro**, por ejemplo, sólo puede tener valores atómicos como los anteriores.

4 Un **Presidente** de una Comunidad de vecinos "es un" **Cargo** o "es una" **Persona**. Es una relación de herencia o generalización. Si se pone rombo, se estaría diciendo (mal) que las partes físicas en que se descompone la clase "**Cargo**" son las clases "**Presidente**", "**Vocal**" y "**Administrador**", cuando, en todo caso, las partes de un **Cargo** son **Cabeza**, **Tronco** y **Extremidades** ¿no? Estas últimas sí llevarían rombo negro (composición).

5 Si la relación no tiene un símbolo concreto, se debe poner una relación genérica **con su nombre**.

6 La dependencia se pone cuando una clase genera o crea una instancia de la clase dependiente. Un Administrador convoca reuniones, luego habrá un método en **Administrador** que será convocarReunión(fecha:String) que, cuando se ejecute, generará una nueva instancia de la clase **Reunion**.

7 Estas relaciones no tienen nombre. Sus nombres son siempre implícitamente: "es un", "se compone de" o "Genera una instancia de", el nombre de estas relaciones se presupone siempre el mismo.

8 color1: String; color2: String; color3:String debe ponerse colorSemaforo:String[3];

9 EstaAbierto() debe retornar True o False, luego debe definirse como estaAbierto(): Boolean

10 cambiaColorSemaforo(color:String) es lo correcto. Incorrecto definir tres métodos: cambiaColorVerde(); cambiaColorRojo(); cambiaColorAmarillo();