



Asignatura: Entornos de desarrollo

UML- Unified Modeling Language



DIAGRAMAS EN UML

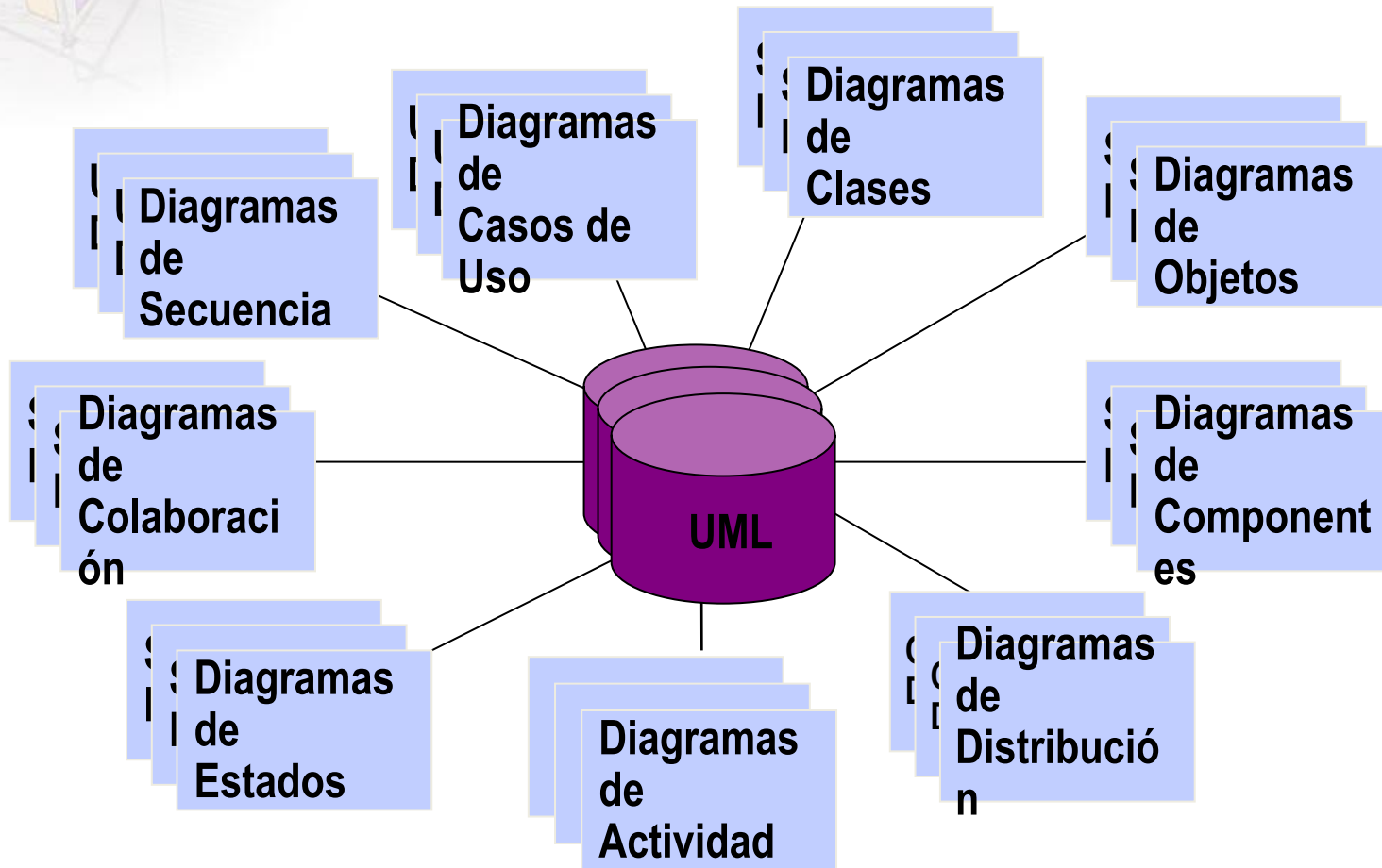




Diagrama de clases

- Es el diagrama principal para el análisis y diseño del sistema.
- Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases y las asociaciones entre ellas.
- Sirve para visualizar las relaciones entre las clases que componen el sistema.



Diagrama de clases

- Un diagrama de clases está compuesto por:
 - a) Clases: atributos, métodos y visibilidad.
 - b) Relaciones: asociación, herencia, agregación/composición, generalización/especialización.
- Cada objeto pertenece a una clase.
- Los objetos se crean por instanciación de las clases.



Diagrama de clases

CLASES: Notación gráfica

En UML, una clase se representa por un rectángulo que posee tres divisiones:

- Nombre de la clase.
- Atributos de la clase.
- Métodos(operaciones de la clase)

Motocicleta
color cilindrada velocidad máxima
arrancar() acelerar() frenar()



Diagrama de clase

En la representación de una clase los atributos y métodos pueden omitirse. Por defecto la visibilidad de los atributos debe ser *private* y la de los métodos *public*.

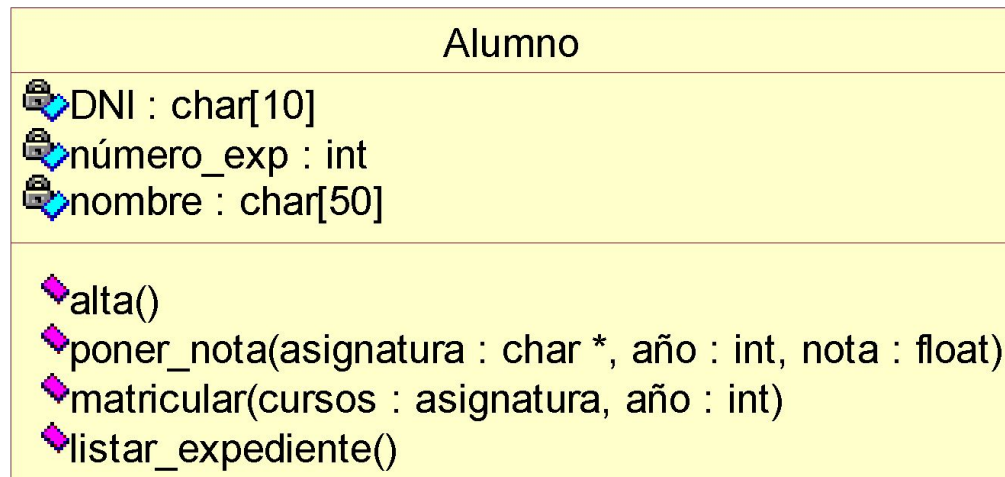




Diagrama de clases

- CLASES: Encapsulación

La encapsulación es el proceso de ocultar los atributos y métodos de un objeto a otros objetos.

La encapsulación presenta dos ventajas básicas:

- a) Se protegen los datos de accesos indebidos.
- b) Favorece la modularidad y el mantenimiento.



Diagrama de clases

- Los niveles de encapsulación son:
 - (-) **Privado** : es el más fuerte. Esta parte es totalmente invisible
 - (#) Los atributos/operaciones **protegidos** están visibles para las clases derivadas de la original
 - (+) Los atributos/operaciones **públicos** son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)



Diagrama de clases

- **ATRIBUTOS**

Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Ejemplo de atributos son: Nombre, Salario, Código, Teléfono, etc.

Al crear los atributos se indicará el tipo de dato y también su visibilidad con el entorno: público, protegido o privado.



Diagrama de clases

- MÉTODOS

Un método, también llamado operación, es la implementación de un servicio de la clase que muestra un comportamiento común a todos los objetos.

Definen la forma de cómo la clase interactúa con su entorno.

Igual que los atributos, los métodos también pueden ser: públicos, protegidos o privados.



Diagrama de clases

• Ejemplo

Reglas de visibilidad

- ◆ Atributo público : Integer
- ◆ Atributo protegido : Integer
- ◆ Atributo privado : Integer

- ◆ "Operación pública"()
- ◆ "Operación protegida"()
- ◆ "Operación privada"()



Diagrama de clases

•RELACIONES

En el mundo real muchos objetos están vinculados o relacionados entre sí, los vínculos se corresponden con asociaciones entre los objetos, por ejemplo el vínculo existente entre un alumno y el curso en el que está matriculado; o el vínculo entre un profesor y el centro en el que trabaja. En UML, estos vínculos se describen mediante asociaciones.



Diagrama de clases

- RELACIONES

Formas de relación entre clases:

- a) Asociación, composición y agregación.
- b) Herencia: generalización / especialización.



Diagrama de clases: ASOCIACIÓN

- La asociación expresa una conexión bidireccional entre objetos.





Diagrama de clases: ASOCIACIÓN

- La asociación posee una cardinalidad llamada multiplicidad que representa el número de instancias de una clase que se relacionan con las instancias de otras clases.
- En uno de los extremos de la asociación, es posible especificar la multiplicidad mínima y máxima.



Diagrama de clases: ASOCIACIÓN

- Para expresar las multiplicidades mínimas y máximas se utiliza la siguiente notación:

1	Uno y sólo uno
0..1	Cero o uno
M..N	Desde M hasta N (enteros naturales)
*	Cero o muchos
0..*	Cero o muchos
1..*	Uno o muchos (al menos uno)

La multiplicidad mínima ≥ 1 establece una restricción de existencia.

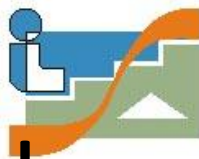


Diagrama de clases: ASOCIACIÓN

- Ejemplo

En el primer ejemplo se muestra la asociación suministra entre Proveedor y Artículo, un proveedor suministra 0 o muchos artículos. Un artículo es suministrado por un proveedor.

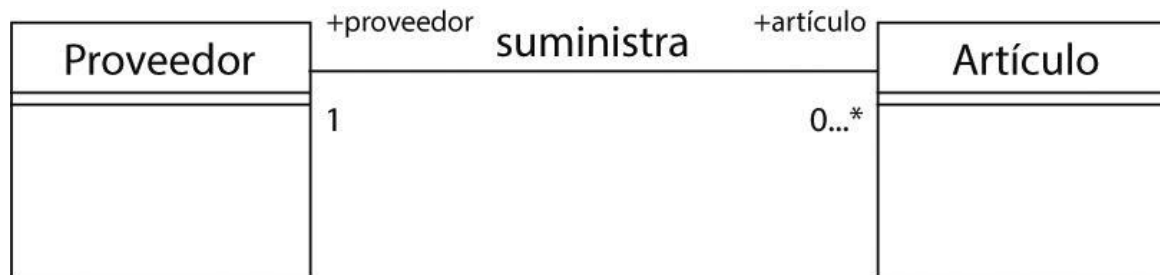




Diagrama de clases: ASOCIACIÓN

- Ejemplo

En el segundo se muestra la asociación tutoriza entre Tutor y Curso, un tutor tutoriza un curso, y un curso es tutorizado por un tutor, es una asociación 1 a 1.





Diagrama de clases: ASOCIACIÓN

- Una clase puede asociarse consigo misma creando una asociación reflexiva. Estas asociaciones unen entre sí instancias de una misma clase.

En la siguiente figura se muestran dos asociaciones reflexivas, un alumno (delegado) es delegado de muchos alumnos y un empleado (jefe) es jefe de muchos empleados.

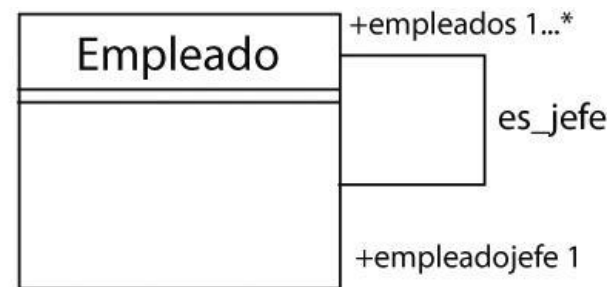
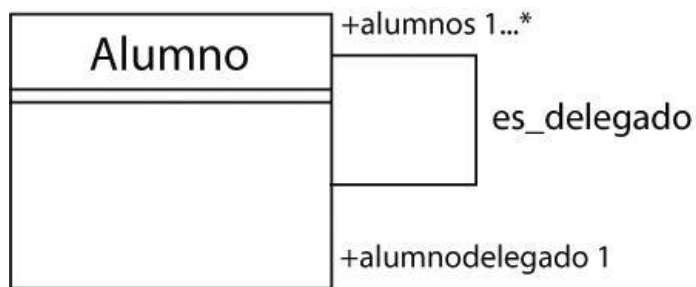


Diagrama de clases: CLASE ASOCIACIÓN



- Una asociación entre dos clases puede llevar información necesaria para esa asociación, a esto se le llama clase asociación.
- Al igual que el resto de las clases, éstas pueden estar dotadas de atributos y operaciones y estar vinculadas a otras clases a través de asociaciones.

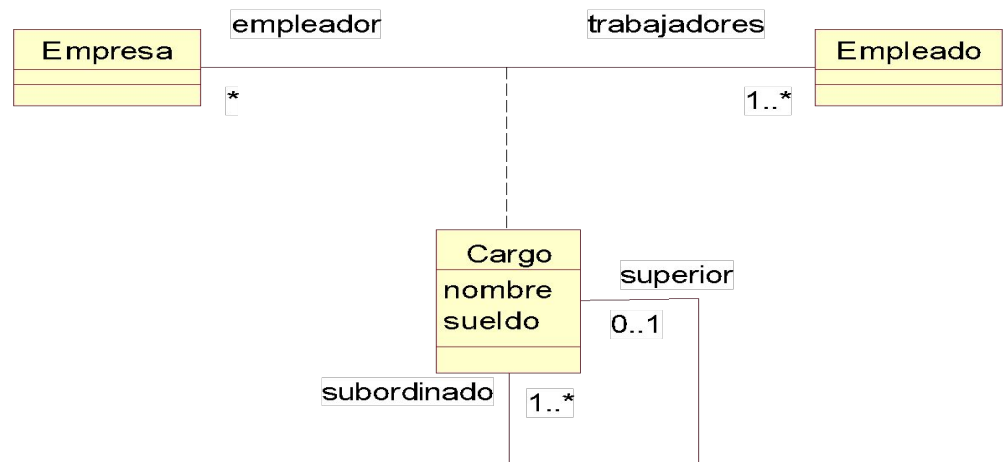




Diagrama de clases: COMPOSICIÓN

- Un objeto puede estar compuesto por otros objetos, en estos casos nos encontramos ante una asociación entre objetos llamada ***Asociación de Composición***. Ésta asocia un objeto complejo con los objetos que lo constituyen, es decir, sus componentes.
- Existen dos formas de composición, fuerte o débil. La fuerte se la conoce como ***composición*** y la débil se la conoce como agregación.



Diagrama de clases: COMPOSICIÓN

- En la composición fuerte los componentes constituyen una parte del objeto compuesto, y éstos no pueden ser compartidos por varios objetos compuestos. La supresión del objeto compuesto comporta la supresión de los componentes.
- Se representa por una línea con un rombo relleno:

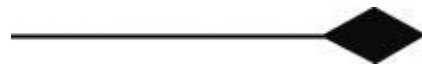




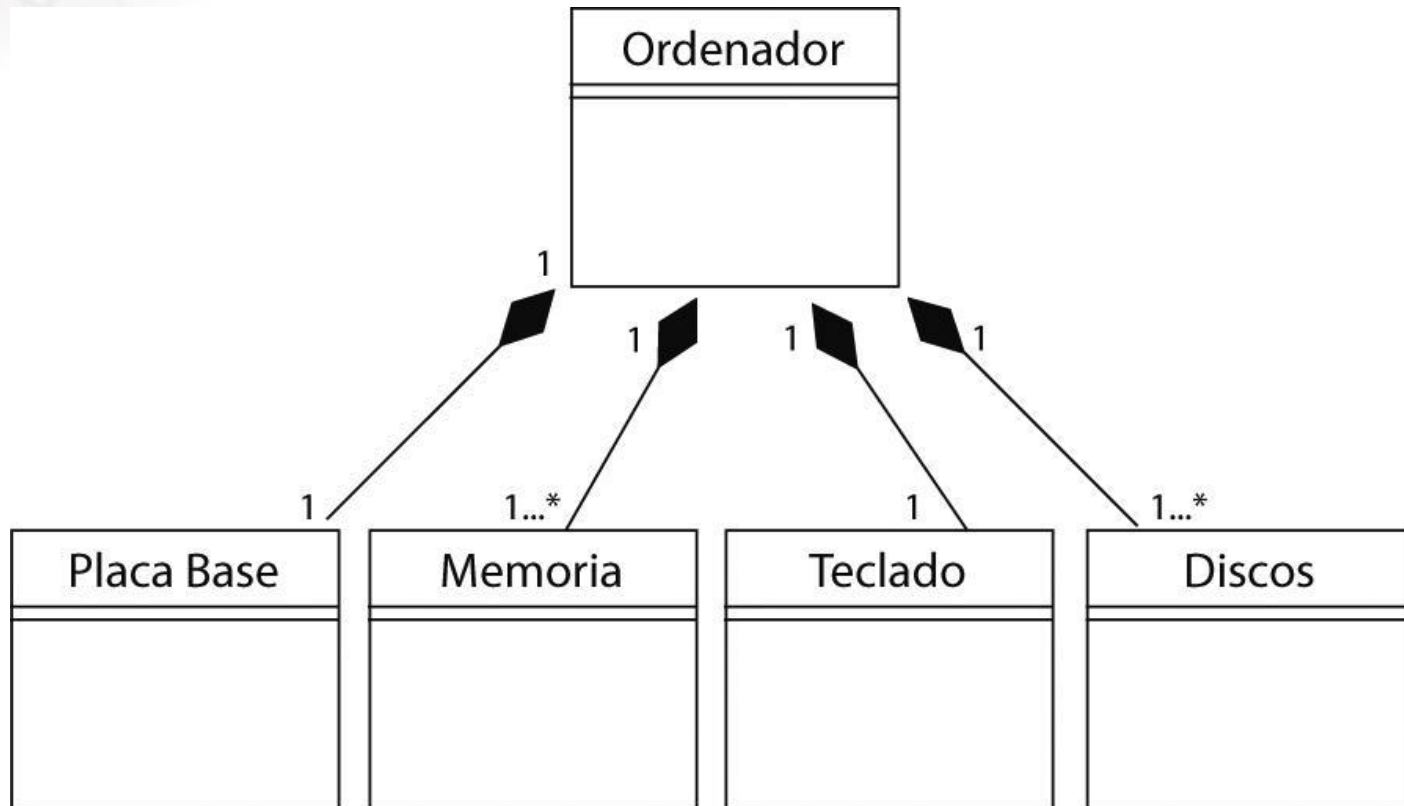
Diagrama de clases: COMPOSICIÓN



En la siguiente figura se muestra la asociación de composición entre un ordenador y sus partes. Se considera que el ordenador se compone de una placa base, una o varias memorias, un teclado y uno o varios discos.



Diagrama de clases: COMPOSICIÓN



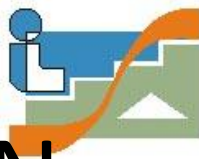


Diagrama de clases: AGREGACIÓN

- La agregación es la composición débil, en este caso los componentes pueden ser compartidos por varios compuestos y la destrucción del compuesto no implica la destrucción de los componentes. La agregación se da con mayor frecuencia que la composición.
- Se representa por un línea con un rombo vacío:



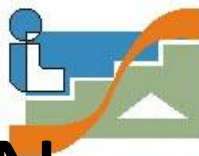


Diagrama de clases: AGREGACIÓN

En la siguiente figura se muestra la asociación de agregación entre una clase *Equipo*, y la clase *Jugador*. Un equipo está compuesto por jugadores, sin embargo, el jugador puede jugar también en otros equipos. Si desaparece el equipo, el jugador no desaparece.





Diagrama de clases: Generalización y Especialización

- Las clases con atributos y operaciones comunes se pueden organizar de forma jerárquica, mediante la herencia.
- La generalización define una relación entre una clase más generalizada, y una o más versiones refinadas de ella.



Diagrama de clases: Generalización y Especialización

- La generalización indica que una clase (clase secundaria o subclase) hereda los atributos y métodos de otra (clase principal o superclase). La superclase generaliza a sus subclases, y las subclases especializan a la superclase.
- Nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada



Diagrama de clases: Generalización y Especialización.

Para representar esta asociación se utiliza una flecha: —————>

Vehículo

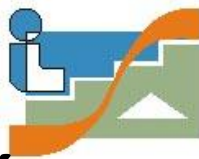


Diagrama de clases: Generalización y Especialización

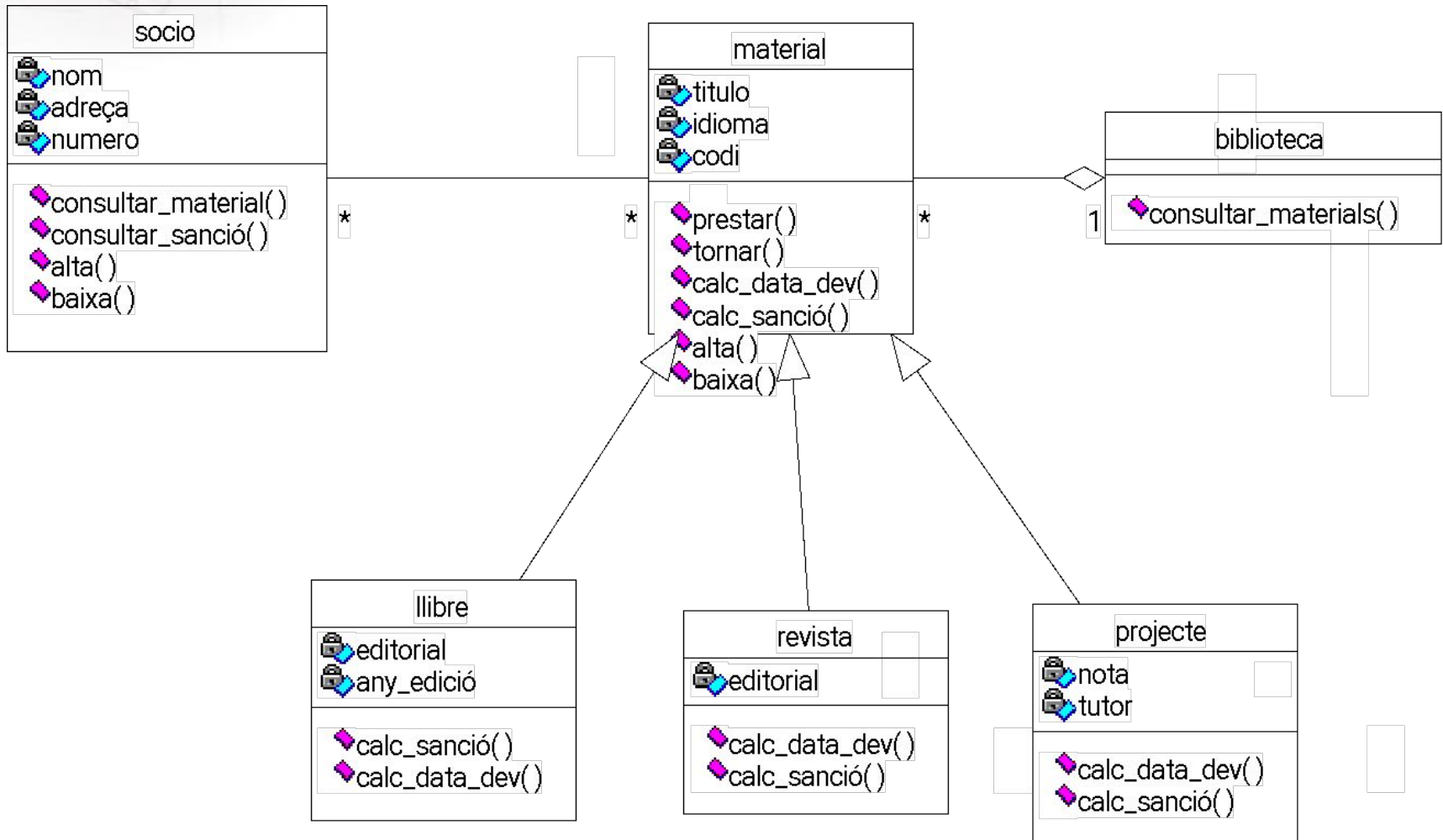




Diagrama de clases: Generalización y Especialización

UML ofrece cuatro especificaciones sobre la herencia entre una superclase y sus subclasses:

- Total** {complete}: significa que el conjunto de las subclasses está completo y cubre la superclase.
- Parcial** {incomplete}: significa que el conjunto de subclasses está incompleto y no cubre la superclase.
- Disjunta** {disjoint}: significa que las subclasses no tienen ninguna instancia en común.
- No disjunta** {overlapping}: significa que las subclasses pueden tener una o varias instancias en común.



Diagrama de clases: Generalización y Especialización

La figura ilustra una relación de herencia entre la superclase Équido y dos subclases: los caballos y los burros. Estas dos subclases no cubren la clase de los équidos (existen otras subclases, como las cebras). Además, también están las mulas, que derivan de un cruce y son a la vez caballos y burros. Por ese motivo, la figura presenta las especificaciones {incomplete} y {overlapping}.

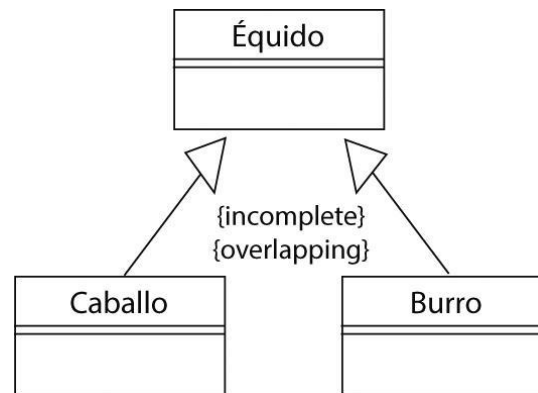




Diagrama de clases: Generalización y Especialización

La figura muestra otra relación de herencia entre la superclase Caballo y dos subclases: CaballoMacho y CaballoHembra. Estas dos subclases cubren la clase de los caballos. No existe ningún caballo que sea a la vez macho y hembra. Por ello, la figura presenta las especificaciones {complete} y {disjoint}.

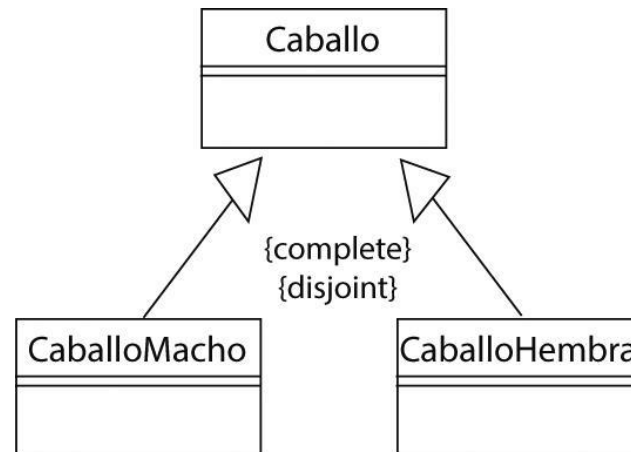




Diagrama de clases:

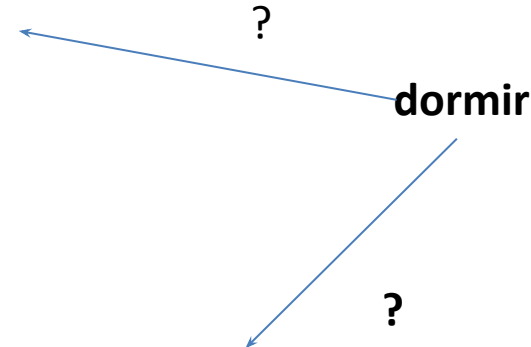
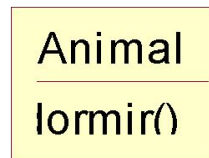
POLIMORFISMO

- El término polimorfismo se refiere a que una característica de una clase puede tomar varias formas en las subclases.
- El polimorfismo representa en nuestro caso la posibilidad de desencadenar operaciones distintas en respuesta a un mismo mensaje.
- Cada subclase hereda las operaciones pero tiene la posibilidad de modificar localmente el comportamiento de estas operaciones.



Diagrama de clases: POLIMORFISMO

Ejemplo: todo animal duerme, pero cada clase lo hace de forma distinta.



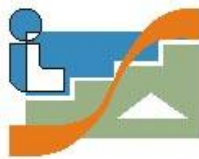


Diagrama de clases: POLIMORFISMO

