Tema 3. Diagramas de Clases y Objetos

Diagramas UML

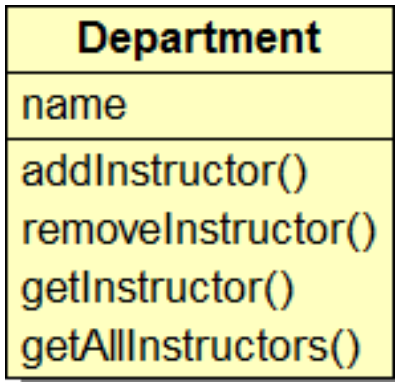
- Diagramas Estructurales
  - Diagrama de Casos de Uso
  - **Diagrama de Clases**
  - **Diagrama de Objetos**
- Diagramas de Comportamiento
  - Diagrama de Estados
  - Diagrama de Actividad
- Diagramas de Interacción
  - Diagrama de Secuencia
  - Diagrama de Colaboración
- Diagramas de Implementación
  - Diagrama de Componentes
  - Diagrama de Despliegue/Distribución

Diagrama de Clases

- Describe la definición de cada uno de los posibles objetos pertenecientes al sistema
- Muestra las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos
- Diagrama cercano a la implementación. Construido y refinado a través del desarrollo
- Desarrollado por analistas, diseñadores y desarrolladores
- **Utilidad**
  - Organizar el sistema, describiendo sus diferentes entidades, así como sus características y relaciones entre ellas
  - Ayuda en la implementación del sistema
  - Permite ver los esquemas lógicos de las estructuras de datos

Diagrama de Clases

- Cada clase se representa por un rectángulo con tres compartimentos
  - Nombre de la clase
  - Atributos de la clase
  - Operaciones de la clase



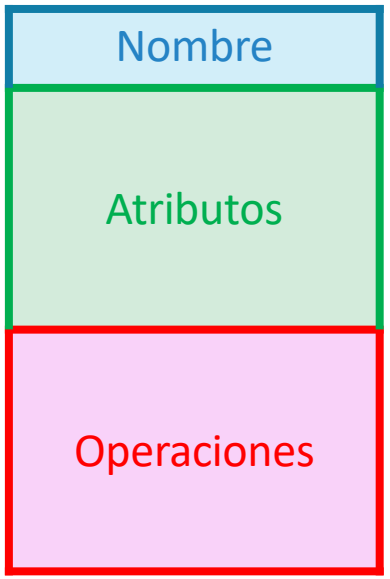


Diagrama de Clases. Encapsulación

- **PREGUNTA: ¿Qué es la encapsulación?**
- Ocultamiento de los datos de un objeto de tal forma que solo sean accesibles mediante operaciones definidas por el propio objeto

- La encapsulación presenta una serie de ventajas
  - Se protegen los datos privados del objeto de lecturas y escrituras no permitidas
  - Permite una mejor estructuración y manipulación de los datos

- Los atributos de un objeto no deberían ser manipulables directamente por el resto de los objetos. En caso de querer hacerlos manipulables, se debe implementar los procedimientos Set y Get

Diagrama de Clases. Encapsulación

- En UML, los niveles de encapsulación vienen heredados de C++
  - **- Privado:** Atributo o proceso totalmente invisible
  - **# Protegido:** Visibles para las clases amigas (*friends)* o para clases derivadas de la original
  - **+ Públicos:** Visibles a otras clases

Diagrama de Clases. Asociación

- La asociación expresa una conexión entre elementos, esto es, que existe algún tipo de relación entre ambos

- Se representa mediante una línea que une ambas clases. Se puede indicar el tipo de asociación y el sentido de la misma

- Se indica la multiplicidad de cada clase, que representa con cuantos objetos de la clase unida por la asociación se puede relacionar un objeto determinado
  - 1
  - 0..1
  - M..N
  - *
  - 0..*
  - 1..*

- La multiplicidad >= 1 establece una restricción de existencia

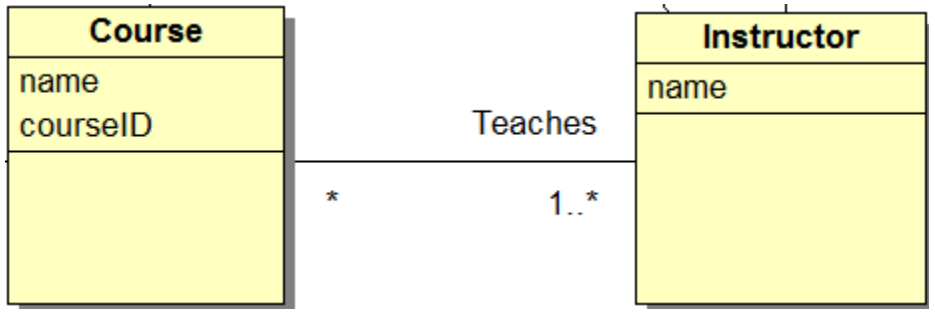
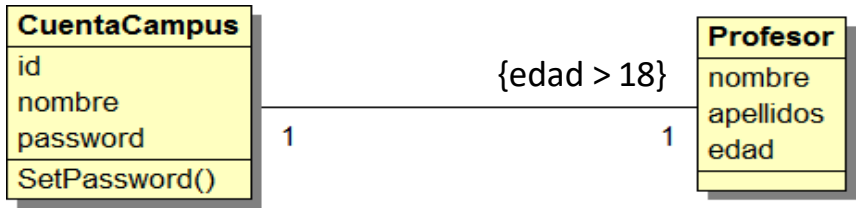


Diagrama de Clases. Asociaciones

- **Asociación con restricciones.** Indica que sólo se realiza la asociación si se cumple una determinada condición



- **Asociación excluyente.** Indica que 2 posibles asociaciones no se pueden realizar a la vez

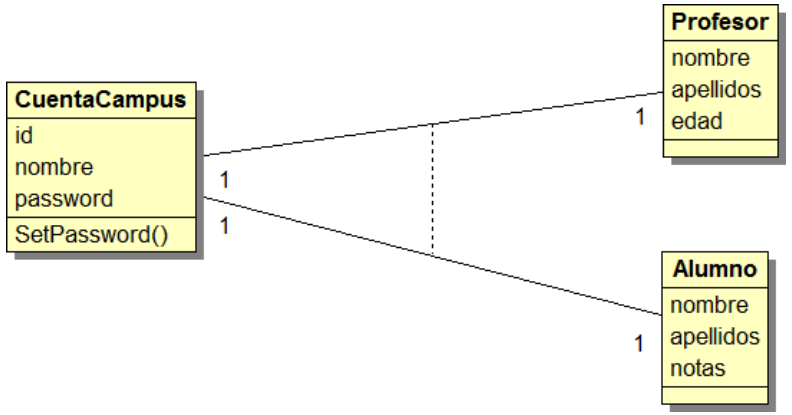


Diagrama de Clases. Agregación

- Una clase puede ser puede estar relacionada por un conjunto de clases que la representen y, sin las cuales, no tenga sentido.

- A esta relación se le llama **Agregación** o **Composición débil**, y se representa mediante un rombo blanco

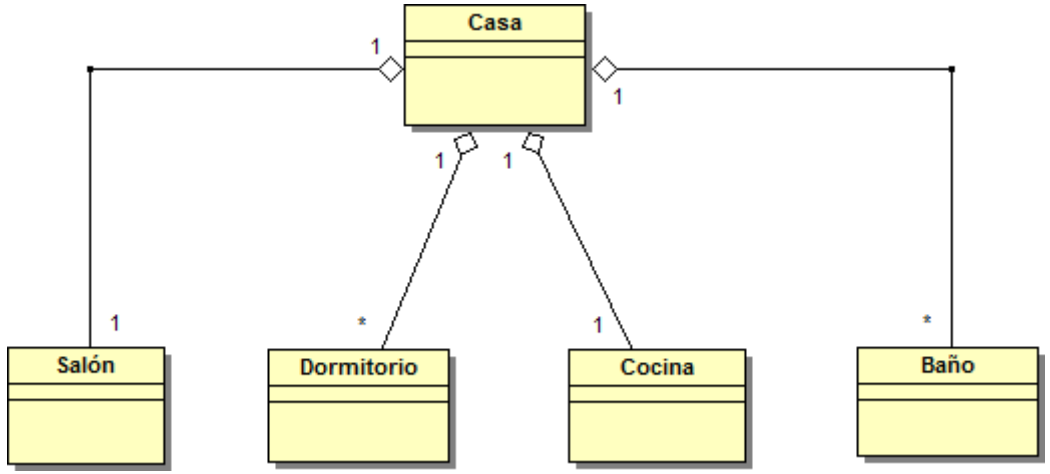


Diagrama de Clases. Agregación. Ejemplo

- Un ordenador posee, como mínimo, los siguientes elementos:
  - 1 Torre
  - 1 Teclado
  - 1 Monitor
  - 1 Ratón
- Una torre se compone por:
  - 1 o varias unidades de disco duro
  - Varios módulos de memoria RAM
  - 1 unidad óptica
  - 1 tarjeta gráfica

Diagrama de Clases. Agregación. Ejemplo

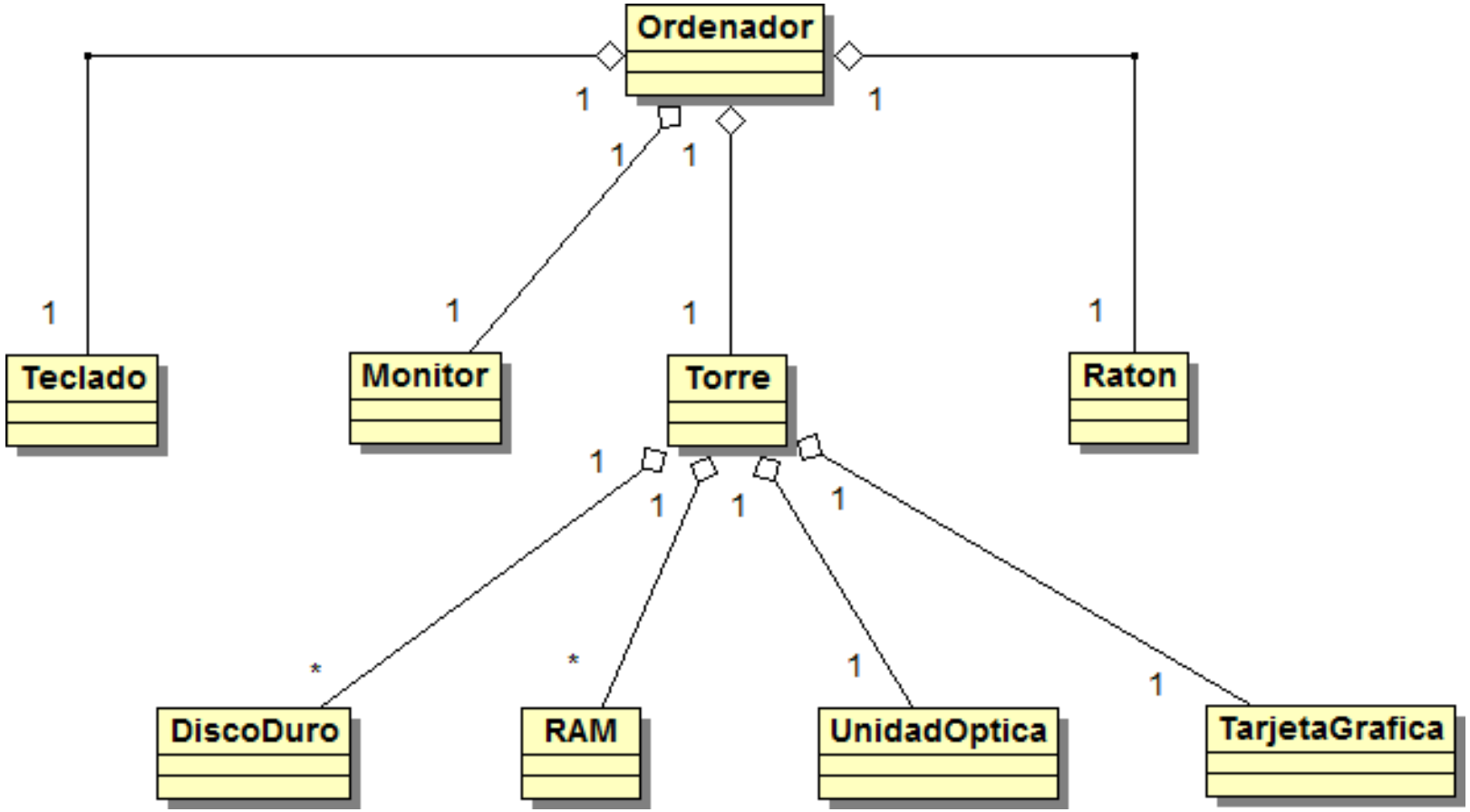


Diagrama de Clases. Composición

- La **Composición** o **Composición fuerte** es una relación entre clases similar a la agregación, pero en la que las clases que componen a la principal no tienen sentido sin dicha clase principal

- Se representa mediante un rectángulo de color negro

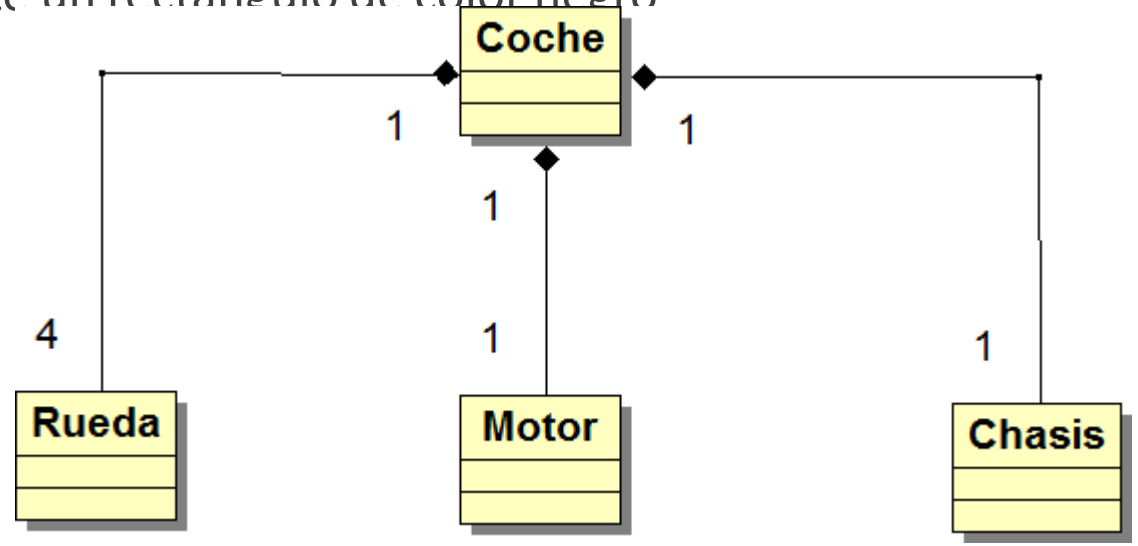


Diagrama de Clases. Generalización

- Consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. **Herencia**

- Las subclases heredan propiedades de sus clases padre, esto es, los atributos, operaciones y asociaciones de la clase padre están disponible en sus clases hijas

- Existen 2 conceptos complementarios: Generalización y Especialización

- En la fase de análisis nos movemos desde la generalización hacia la especialización

Diagrama de Clases. Generalización

- La generalización se expresa mediante una flecha hueca

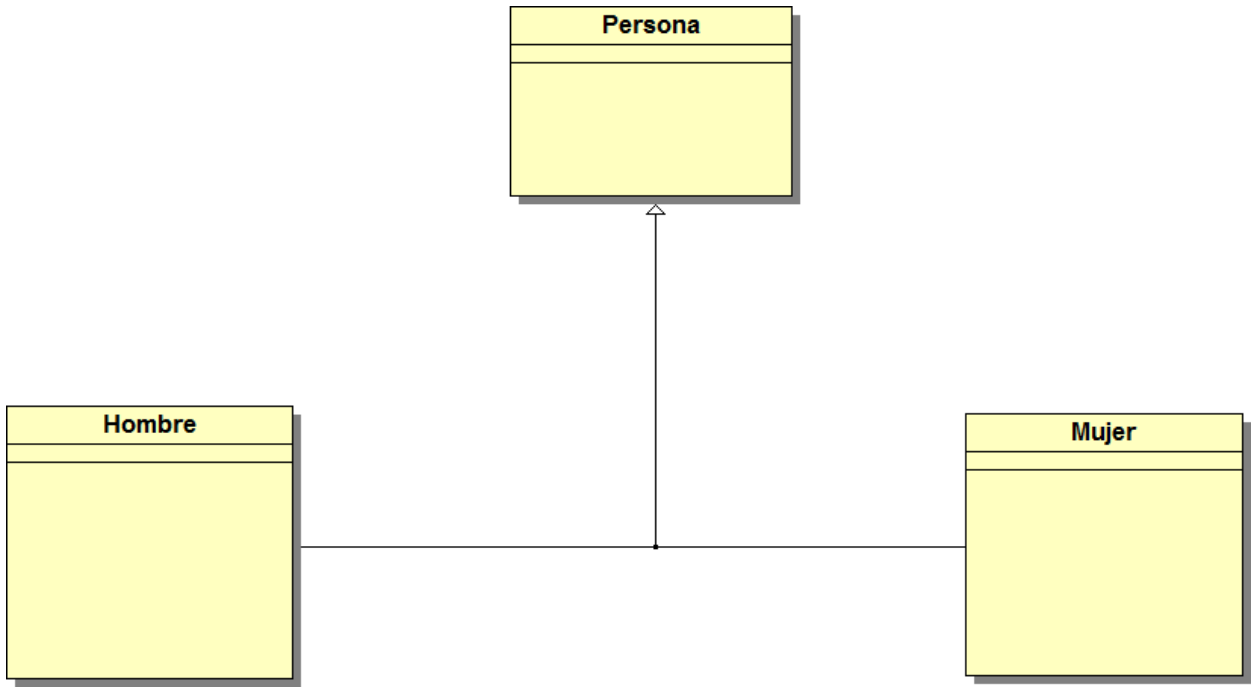


Diagrama de Clases. Ejemplos generalización

1. Representar mediante un diagrama de clases la clasificación de los seres vivos:
   - Los animales se deben organizar por su alimentación, esto es: Carnívoros, Herbívoros y Omnívoros
   - Las plantas se deben clasificar por su tipo: Árboles, Arbustos y Hierbas
   - Para los animales, se ponen 3 ejemplos: Conejo, León y Vaca

Diagrama de Clases. Ejemplos generalización

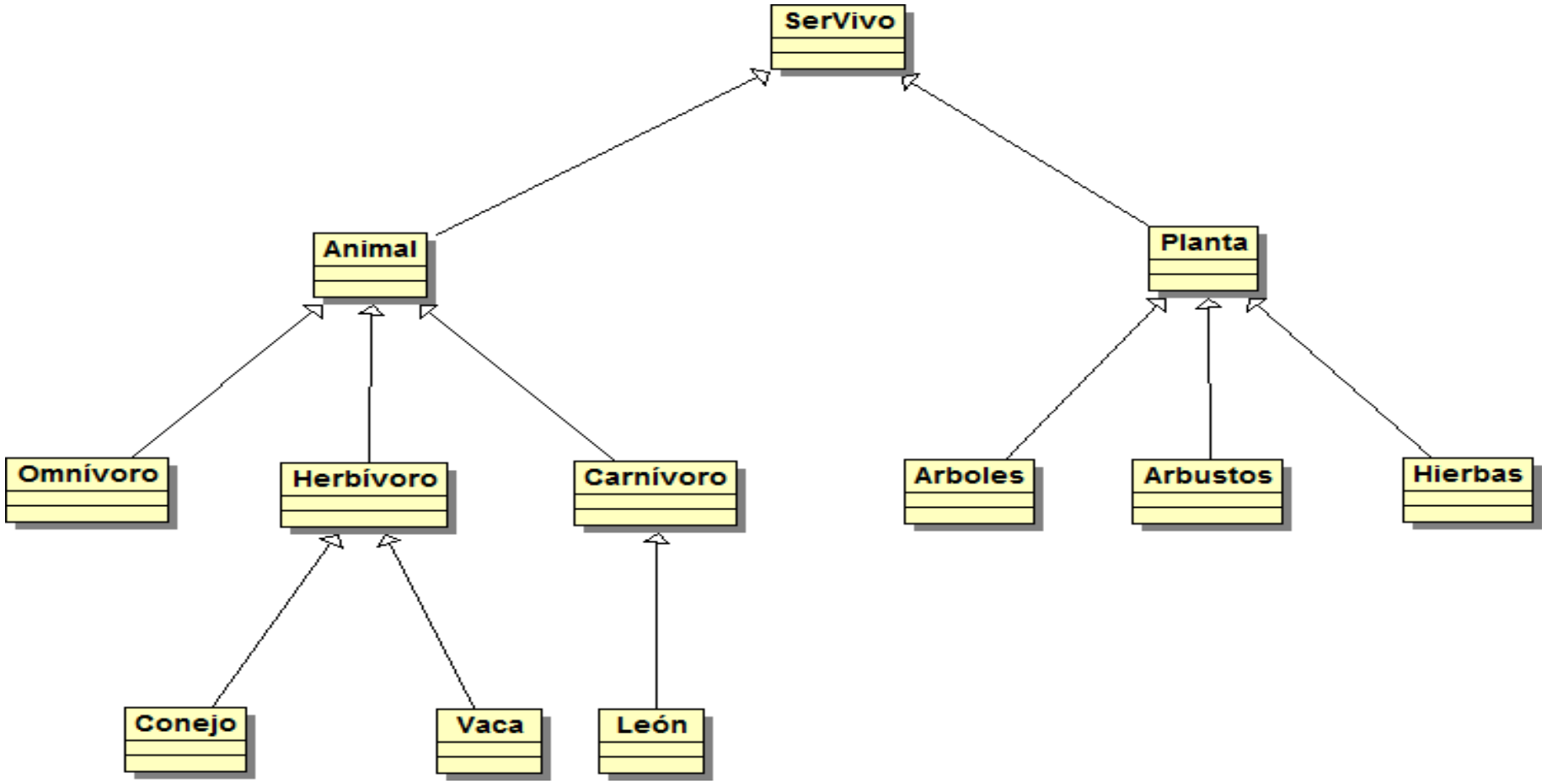
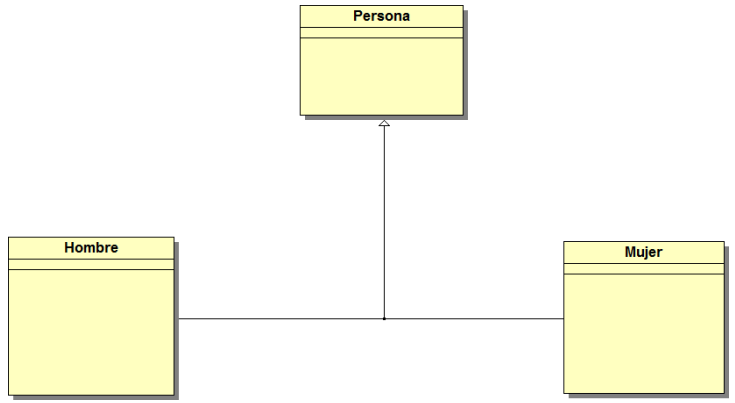


Diagrama de Clases. Generalización y conjuntos

- Se puede equiparar el concepto de **clase** al de **conjunto**, de forma que ambos sirven para clasificar distintos elementos

- Generalización y especialización expresan relaciones entre conjuntos

Clases:

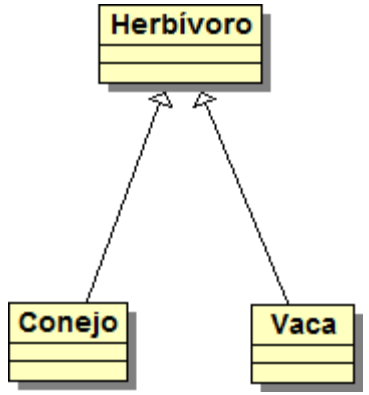


Conjuntos:

Hombres U Mujeres = Personas

Diagrama de Clases. Clasificación

- Se puede hacer uso de la herencia para clasificar los distintos elementos
- Si la clasificación se hace por cualidades del objeto, se dice que es **Clasificación Estática**



- Si la clasificación se hace por el estado del objeto, se dice que es **Clasificación Dinámica**

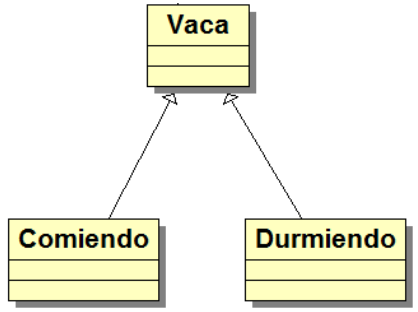


Diagrama de Clases. Herencia múltiple

- Se produce cuando una subclase tiene más de una superclase
- No se suele recomendar, ya que puede dar conflictos de nombre y procedencia
- Algunos lenguajes de programación como Java o Ada95 no permiten herencia múltiple

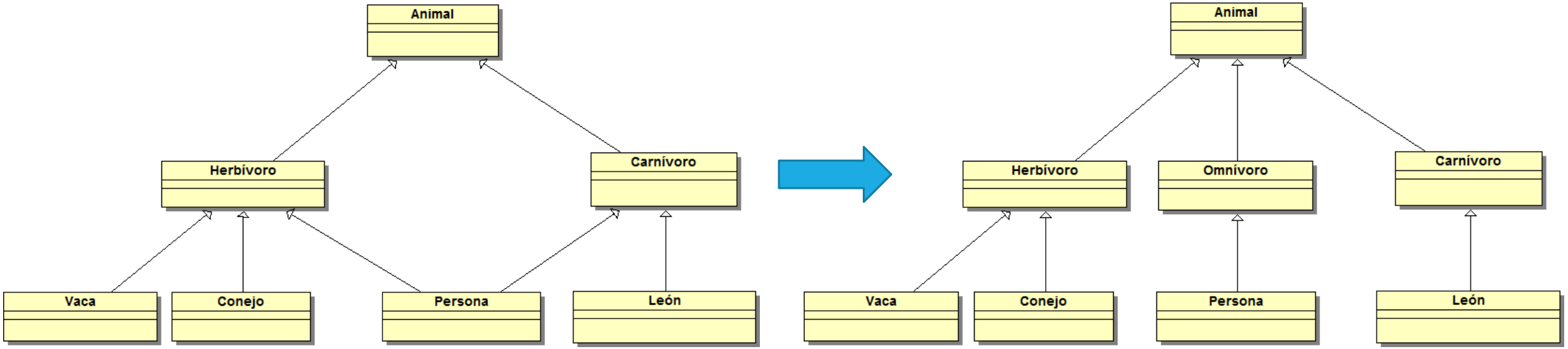


Diagrama de Clases. Principio de Sustitución

- El principio de sustitución (Liskow 1987) dice:

Debe ser posible utilizar cualquier objeto instancia de una subclase en el lugar de cualquier objeto instancia de su superclase sin que la semántica del programa escrito en los términos de la superclase se vea afectado.

- Este principio debe seguirse siempre que se utilice el **Polimorfismo**

Diagrama de Clases. Polimorfismo

- El término polimorfismo se encuentra ligado al concepto de herencia e indica que una característica de una clase padre puede tomar diferentes formas dependiendo de la clase hija que la ejecute

- Permite que, ante un mismo estímulo, se desencadene una respuesta distinta dependiendo de la clase que la ejecute

- Este concepto permite dotar de flexibilidad al conjunto de clases implementado, siendo uno de los mecanismos mas potentes que posee el uso de herencia

Diagrama de Clases. Ejemplo de polimorfismo

- **Ejemplo:** ¿Qué comen los animales?

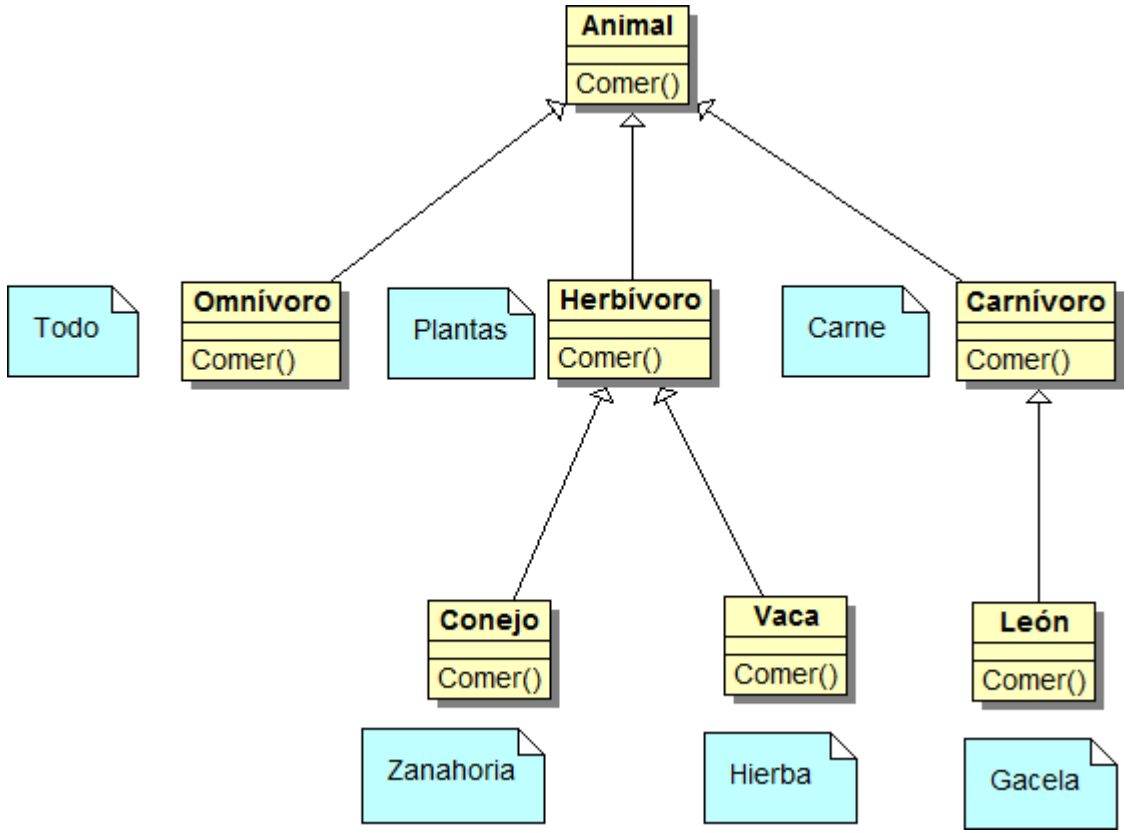


Diagrama de Clases. Ejemplo

El dueño de un hotel te pide a desarrollar un programa para consultar sobre las habitaciones disponibles y reservar habitaciones de su hotel

El hotel posee tres tipos de habitaciones: simple, doble y matrimonial, y dos tipos de clientes: habituales y esporádicos. Una reserva almacena datos del cliente, de la habitación reservada, la fecha de comienzo y el número de días que será ocupada la habitación

El recepcionista del hotel debe poder hacer la siguientes operaciones:

- Obtener un listado de las habitaciones disponible de acuerdo a su tipo
- Preguntar por el precio de una habitación de acuerdo a su tipo
- Preguntar por el descuento ofrecido a los clientes habituales
- Preguntar por el precio total para un cliente dado, especificando su numero de DNI, tipo de habitación y número de noches.
- Dibujar en pantalla la foto de un habitación de acuerdo a su tipo
- Reservar una habitación especificando el número de la habitación, DNI y nombre del cliente.
- Eliminar una reserva especificando el número de la habitación

El administrador puede usar el programa para:

- Cambiar el precio de una habitación de acuerdo a su tipo
- Cambiar el valor del descuento ofrecido a los clientes habituales
- Calcular las ganancias que tendrán en un mes especificado (considere que todos los meses tienen treinta días).

El hotel posee información sobre que clientes son habituales. El diseño a desarrollar debe facilitar la extensibilidad de nuevos tipos de habitación o clientes y a su vez permitir agregar nuevas consultas

Diagrama de Clases. Ejemplo

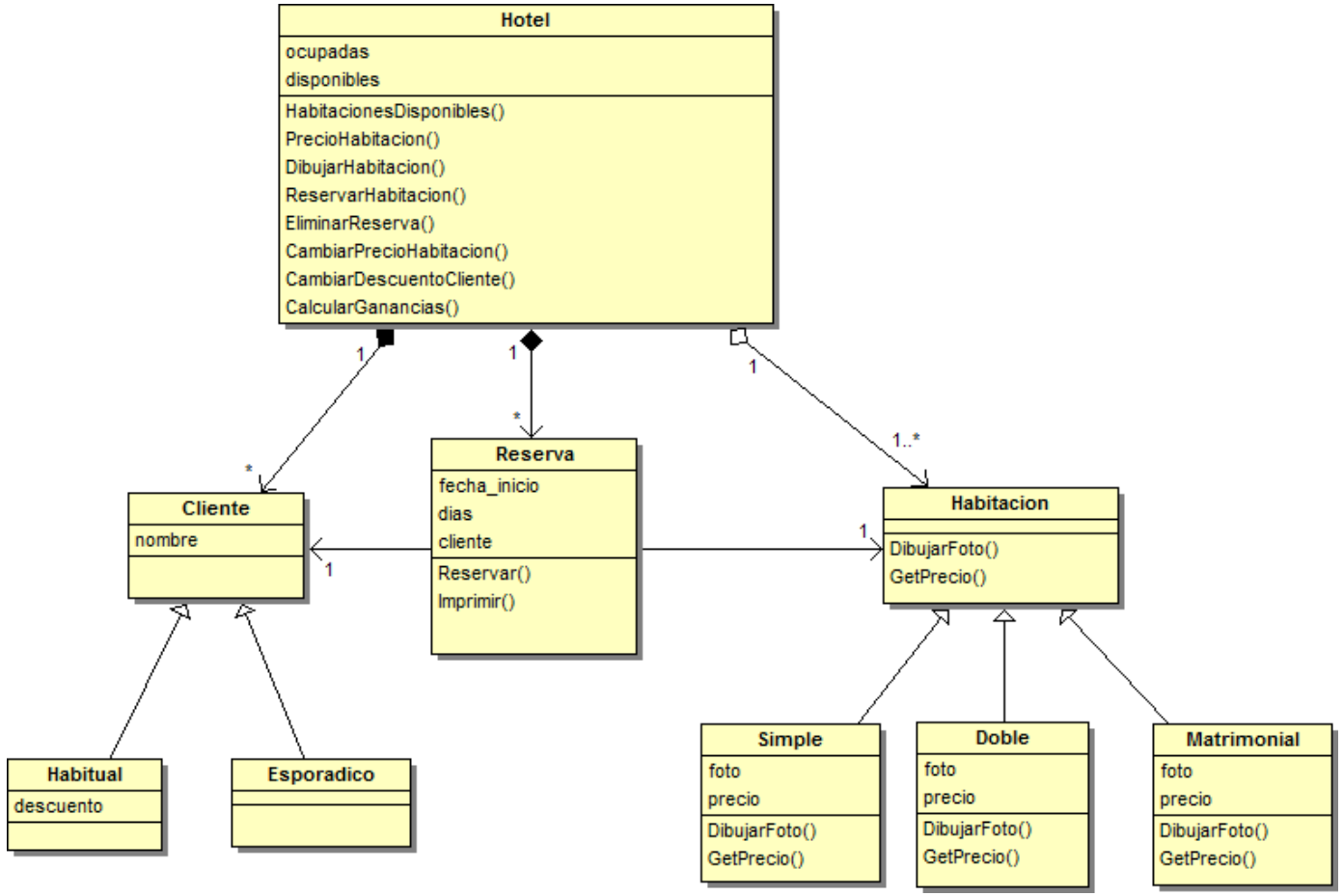


Diagrama de Objetos

- Muestra una vista completa o parcial de los objetos de un sistema **en un instante de ejecución determinado**

- Comparte la misma notación que los diagramas de clases. El nombre del objeto se representa subrayado, a diferencia del nombre de las clases

- **Utilidad.**
  - Ilustrar las estructuras de datos/objetos del sistema
  - Especificar detalles del modelo
  - Obtener una “foto” del sistema en un determinado punto

Diagrama de Objetos. Ejemplo práctico