

## Unidad 4

# HERRAMIENTAS DE MODELADO



**UAIOnline**  
**Ultra**»»



# Herramientas de Modelado

## Unidad 4

### ■ OBJETIVOS

- Analizar y comprender el concepto del Paradigma orientado a objetos.
- Identificar y comprender las características, propiedades y componentes del paradigma orientado a objetos.
- Estudiar y comprender el análisis orientado a objetos.
- Comprender lo que es una CLASE, cuales son sus características y la forma en que se relacionan.



# Herramientas de Modelado

## Unidad 4

### ■ RESULTADOS DE APRENDIZAJE

Comprender el paradigma de modelado Orientado a Objetos para el análisis de un sistema utilizando el estándar UML.



# Herramientas de Modelado

## Unidad 4

- **HABILIDADES Y COMPETENCIAS QUE DESARROLLA LA ASIGNATURA**
  - Construir modelos para la representación abstracta de la realidad.
  - Aplicar el análisis orientado a objetos para la representación actual de un sistema de información.
  - Aplicar el paradigma orientado a objetos para el diseño de sistemas de información.



# El Paradigma OO



**UAIOnline**  
**Ultra**»»



# INTRODUCCIÓN / I

## ¿Qué es un paradigma?

Un paradigma es una forma de entender y representar la realidad: un conjunto de teorías, estándares y métodos que, juntos, representan un modo de organizar el pensamiento

Es una forma de ver el Mundo

# INTRODUCCIÓN / 2

## El Paradigma OO

Define los programas en términos de comunidades de objetos. Los objetos con características comunes se agrupan en clases (un concepto similar al de tipo abstracto de dato (TAD))

Es una forma de ver el Mundo

# EL PARADIGMA OO SE IMPUSO POR ...

- ◆ *Conceptos comunes de modelado a lo largo de todo el ciclo de vida*
- ◆ *Reducción de la brecha entre el mundo de los problemas y el mundo de los modelos*
- ◆ *Aumento de complejidad de los sistemas*
- ◆ *Aumento de la necesidad de reutilización*
- ◆ *Uso de patrones*



# CLASES

Una **clase** especifica una estructura de datos y las operaciones permisibles que se aplican a cada uno de sus objetos.

Los objetos se vinculan mediante enlaces enviando **mensajes** a **operaciones** que activan los **métodos**

- ⑩ **Mensaje:** es una solicitud para que se lleve a cabo la operación indicada y se produzca el resultado.
- ⑩ **Operaciones:** es una función o transformación que se aplica a un objeto de una clase
- ⑩ **Métodos:** es la implementación de una operación

*"Un objeto es una instancia de una **clase**"*

# RESPONSABILIDADES DE LAS CLASES

- ◆ Una responsabilidad es un contrato u obligación de una clase
- ◆ Una clase puede tener cualquier numero de responsabilidades, pero una clase bien estructurada debería tener al menos una y como mucho unas pocas (alta cohesión)
- ◆ Las responsabilidades se describen inicialmente con texto libre
- ◆ Al ir refinando los modelos, las responsabilidades se traducirán en el conjunto de atributos y operaciones que satisfagan esas responsabilidades

(ver guía 5.1 Patrones de Diseño para Asignación de Responsabilidades)

# EJEMPLO DE CLASES

class Unidad 4.2

Automovil
- Año: int - Color: String - Marca: String - Modelo: String - Titular: String
+ Acelerar() + Frenar()

Persona
- Apellido: String - Edad: int - Nombre: String
+ Caminar() + Comer()

# OBJETOS / I

*"Un objeto es cualquier cosa real o abstracta, acerca de la cual almacenamos datos y las **operaciones** que controlan dichos datos"*

*Se opone al análisis estructurado donde los datos y el comportamiento están débilmente relacionados*

Tenemos que olvidarnos del modelo estructurado...

## OBJETOS / 2

*"Los objetos son entidades que combinan un estado (es decir, datos) y un comportamiento (esto es, procedimientos o métodos). Estos objetos se comunican entre ellos para realizar tareas "*

# PROPIEDADES DE LOS OBJETOS

*"El **estado** de un objeto abarca todas las propiedades (normalmente estáticas) del mismo, más los valores actuales (normalmente dinámicos) de cada una de esas propiedades"*

*"El **comportamiento** nos muestra como actúa y reacciona un objeto, en términos de sus cambios de estado y paso de mensajes"*

*"La **identidad** es aquella propiedad de un objeto que lo distingue de todos los demás objetos"*

# ABSTRACCIÓN / I

*La **abstracción** es la propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las restantes características (no esenciales).*

Plantea **¿Qué hace?** En lugar de **¿Cómo lo hace?**

El común denominador en la evolución de los lenguajes de programación, desde los clásicos o imperativos hasta los orientados a objetos, ha sido el nivel de abstracción del que cada uno de ellos hace uso

# ABSTRACCIÓN / 2

Desde el punto de vista **Orientado a Objetos**, la abstracción:

- Expresa las características esenciales de un objeto, la cual distingue al objeto de los demás
- Provee límites conceptuales



# ENCAPSULAMIENTO

*El **Encapsulamiento** es la propiedad que permite asegurar que el contenido de la información de un objeto está oculta al mundo exterior .*

*El encapsulamiento, al separar el comportamiento del objeto de su implantación, permite la modificación de éste sin que se tengan que modificar las aplicaciones que lo utilizan*

*Si un objeto tiene más características de las necesarias los mismos resultarán difíciles de usar, modificar, construir y comprender*

# RELACIONES DE ASOCIACIÓN

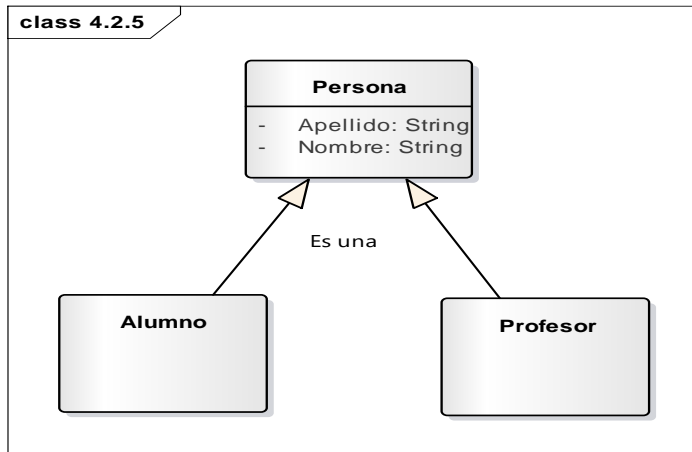
*"Se descompone (**clases**) para comprender, se une (**asociación**) para contruir"*

- ⑩ *Los enlaces entre objetos son instancias de la **asociación** entre sus clases*

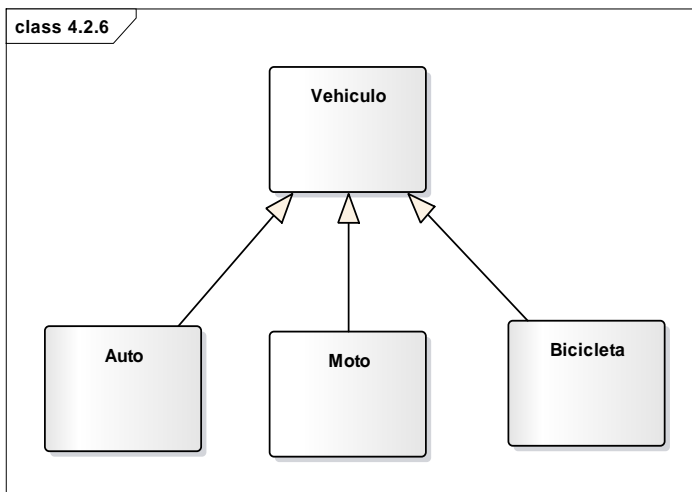
# RELACIONES DE JERARQUÍA

- ◆ *La **generalización** consiste en factorizar los elementos comunes de un conjunto de clases en una clase más general llamada superclase*
- ◆ *La **herencia** es una técnica de los lenguajes de programación para construir una clase a partir de una o varias clases, compartiendo atributos y operaciones*

# EJEMPLO DE RELACIÓN DE JERARQUÍA / I

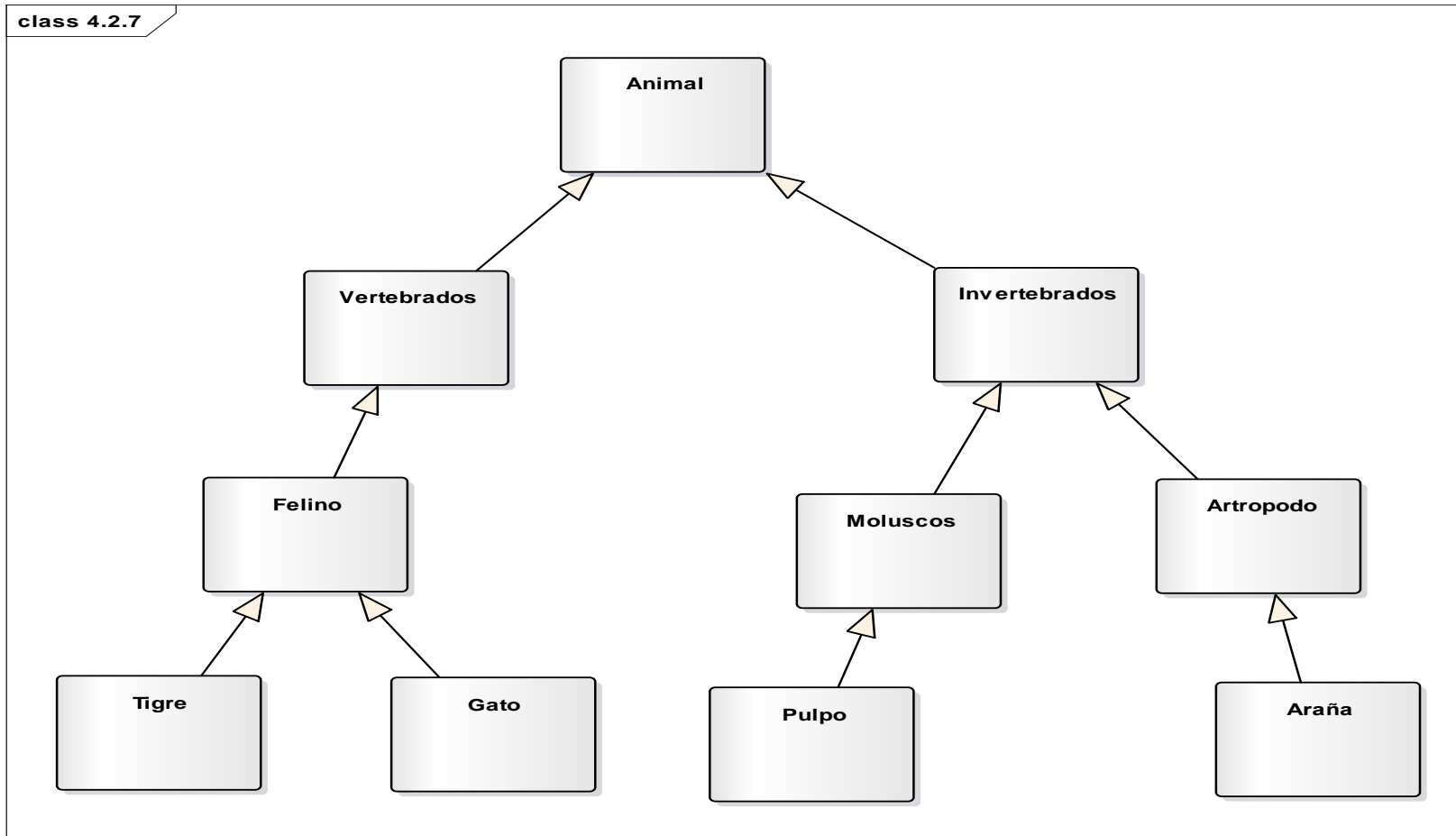


*Algunas Personas son Alumnos  
y otras Profesores  
Un Profesor es una Persona*



*Algunos Vehículos son Autos  
Una Moto es un Vehículo*

# EJEMPLO DE RELACIÓN DE JERARQUÍA / 2



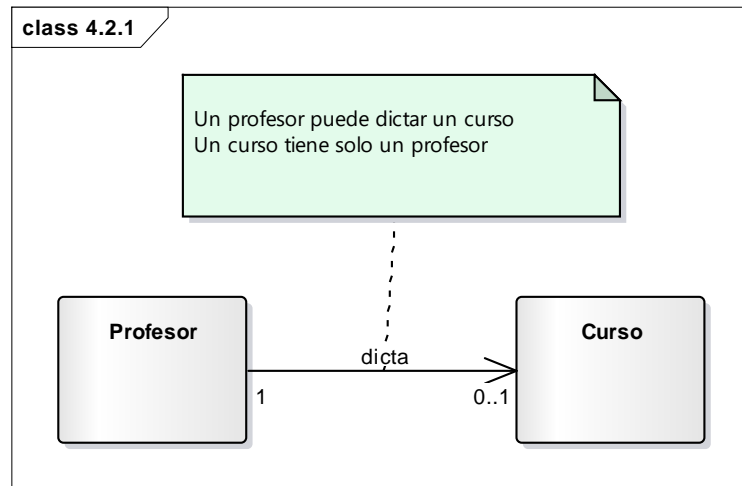
# MULTIPLICIDAD DE RELACIONES / I

- ◆ *Determina cuantos objetos de cada tipo Intervienen en una relación*
- ◆ *Cuantas instancias de UNA CLASE que se relacionan con UNA instancia de otra clase.*
- ◆ *Cada asociación tiene dos multiplicidades: una para cada extremo*
- ◆ *Cada multiplicidad tiene un valor mínimo y un valor máximo*
- ◆ *Ayuda a definir reglas de negocio o características centrales del dominio*

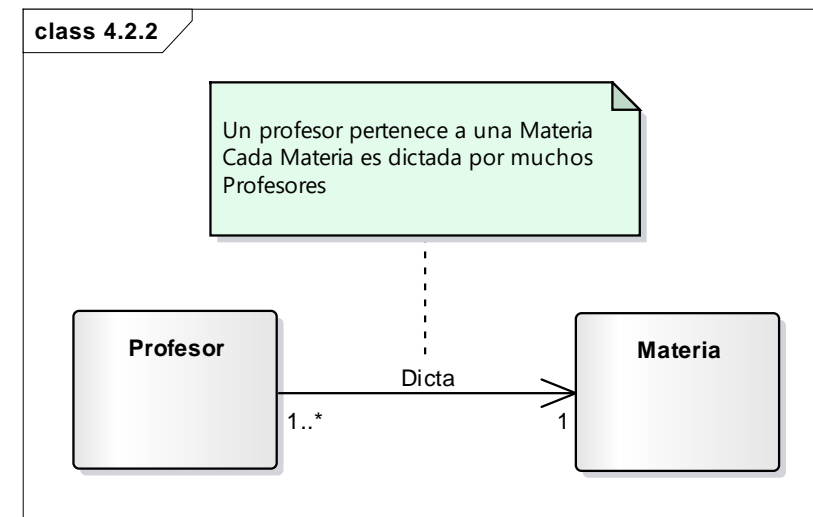
## MULTIPLICIDAD DE RELACIONES / 2

Multiplicidad	Significado
1	Uno y sólo uno
0..1	Cero o uno
N..M	Desde N hasta M
*	Cero o varios
0..*	Cero o varios
1..*	Uno o varios (al menos uno)

# MULTIPLICIDAD DE RELACIONES / 3

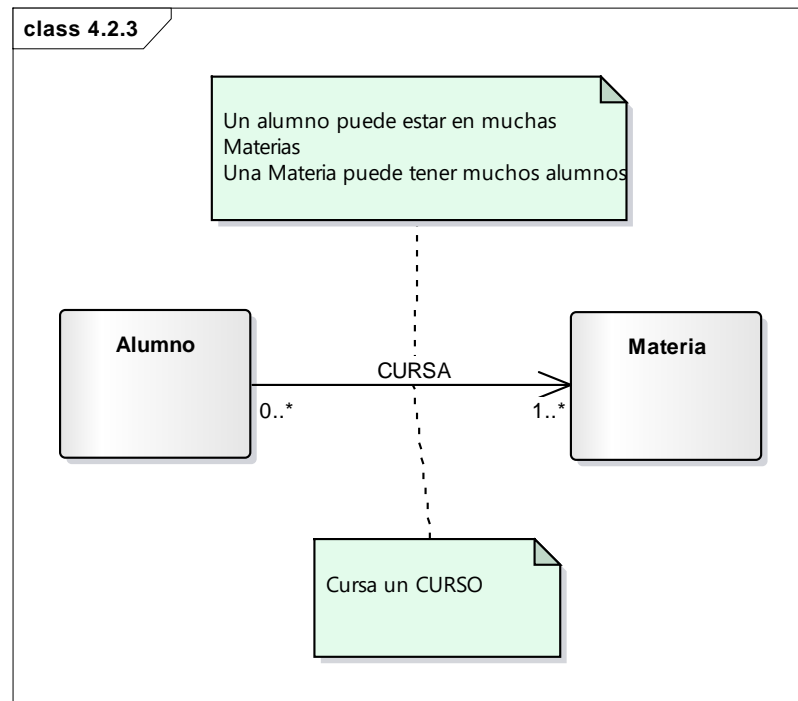


Multiplicidad	Significado
1	Uno y sólo uno
0..1	Cero o uno
N..M	Desde N hasta M
*	Cero o varios
0..*	Cero o varios
1..*	Uno o varios (al menos uno)



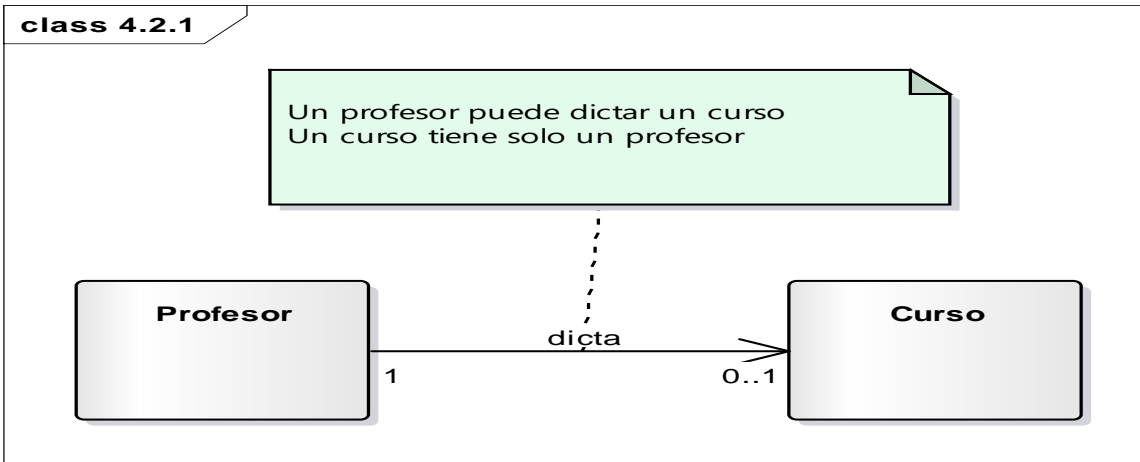


# MULTIPLICIDAD DE RELACIONES / 4



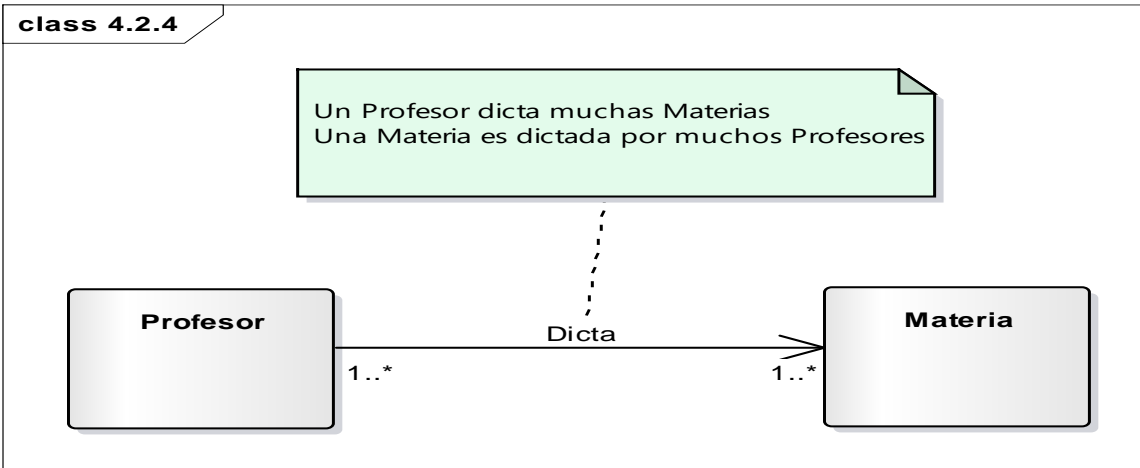
Multiplicidad	Significado
1	Uno y sólo uno
0..1	Cero o uno
N..M	Desde N hasta M
*	Cero o varios
0..*	Cero o varios
1..*	Uno o varios (al menos uno)

# MULTIPLICIDAD DE RELACIONES / 5

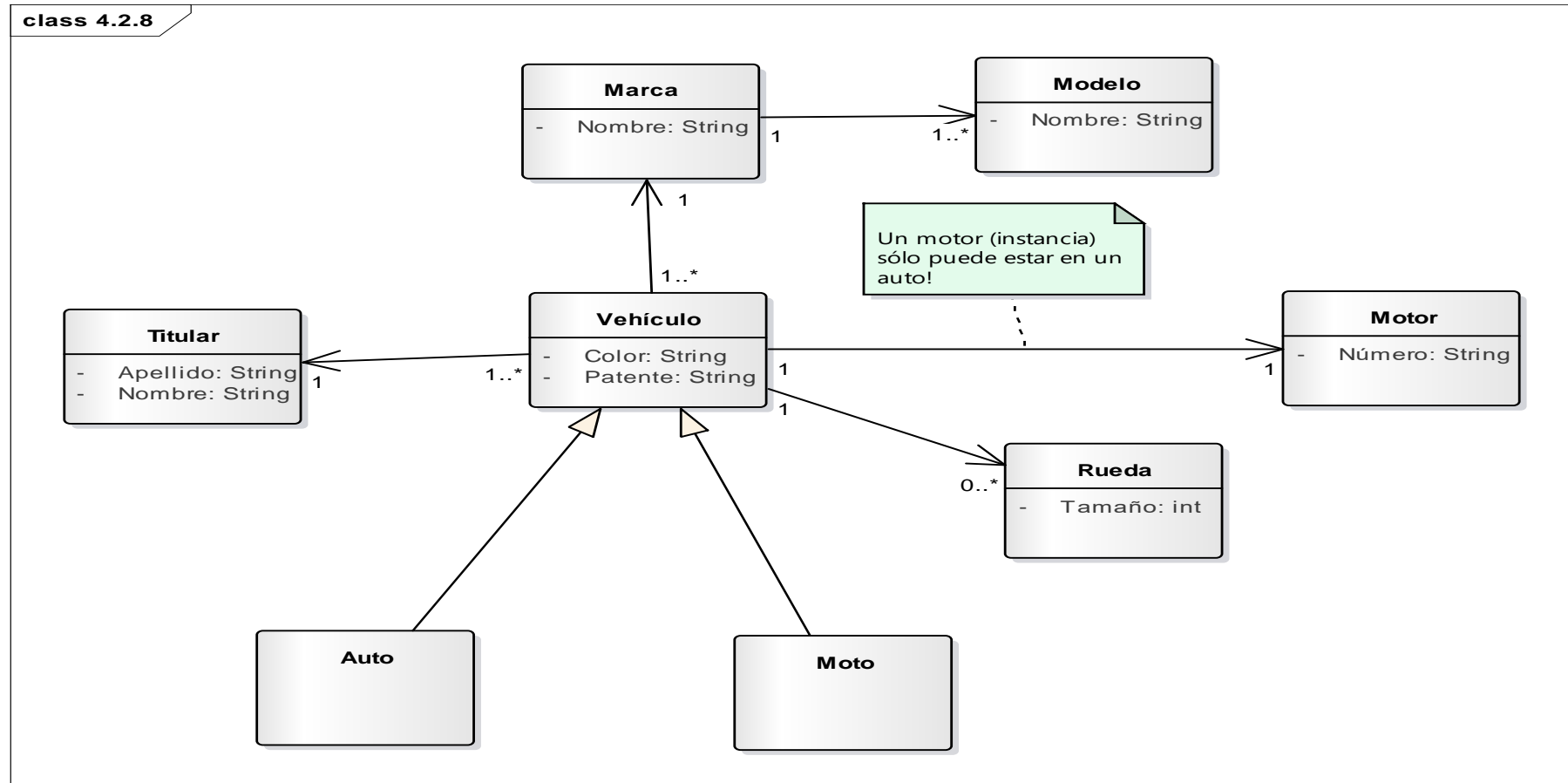


*Definir Reglas de negocio:*

*Un profesor puede dictar  
SOLO UNA MATERIA o  
puede dictar MUCHAS??*



# EJEMPLO – MODELANDO EL MUNDO REAL



# ANÁLISIS ESTRUCTURADO VS ANÁLISIS OO

*El enfoque tradicional del análisis y diseño estructurados, se descompone el problema en funciones o procesos y estructuras de datos*

*En un enfoque OO se busca descomponer el problema, no en funciones, sino en unidades más pequeñas denominadas objetos*

# BENEFICIOS DEL ENFOQUE OO

*Disminución del bache semántico entre análisis y diseño proveyendo una representación consistente en todo el ciclo de vida*

## ***Enfoque OO***

*La transición del análisis al diseño es un refinamiento*

## ***Enfoque Estructurado***

*En la transición del análisis al diseño pasamos del DFD al DER mediante un proceso heurístico no trivial*

# AUTO EVALUACIÓN / I

Comprendí los conceptos más importantes de la unidad 4.2 si puedo definir y dar ejemplos de:

- ⑩ Clase
- ⑩ Objeto
- ⑩ Estado / Comportamiento / Identidad
- ⑩ Tarjetas CRC
- ⑩ Mensaje / Operación / Método
- ⑩ Asociación / Agregación / Composición
- ⑩ Generalización / Herencia / Polimorfismo

# AUTO EVALUACIÓN / 2

Comprendí los conceptos más importantes de la unidad 4.2, si

- ⑩ Comprendo la diferencia entre el análisis y diseño Estructurado y el OO.
- ⑩ Entiendo la diferencia entre análisis, diseño e implementación y que es lo que realizo en cada una de estas actividades.
- ⑩ Vinculo la etapa de análisis con la descripción del modelo de dominio de la aplicación.
- ⑩ Comprendo cual es el uso de las tarjetas CRC.
- ⑩ Entiendo cual es el objetivo de la abstracción y el encapsulamiento.
- ⑩ Entiendo la diferencia entre mensajes, operaciones y método.
- ⑩ Comprendo la diferencia entre asociación, agregación y composición.
- ⑩ Entiendo la diferencia entre generalización y herencia.



Fin de la clase



**UAI**

**Universidad Abierta  
Interamericana**