

# Diagramas de clases (parte 1)

Sitio: Agencia de Habilidades para el Futuro  
Curso: Modelado y Diseño de Software 1º D  
Libro: Diagramas de clases (parte 1)

Imprimido por: Eduardo Moreno  
Día: domingo, 23 de marzo de 2025, 10:33

# Tabla de contenidos

- 1. Introducción
- 2. Objetos y clases
- 3. Diagramas de clases
  - 3.1. Definición
- 4. Ejemplo de aplicación



## Introducción

En este libro vas a recorrer contenidos sobre **qué es una clase y cómo se diagrama** .

Pero antes de avanzar, en la semana anterior recordamos y usamos el modelo E\_R de Base de Datos como práctica, también aprendiste qué es modelar cuando vas a desarrollar un producto software. En esta semana, vas a conocer las **primeras herramientas de UML**.

UML permite representar en diagramas y para eso es necesario conocer las clases (que, como viste en el capítulo anterior, clasifican a los objetos). En el **índice** de la derecha podrás recorrer los contenidos revisando los contenidos todas las veces que lo necesites.

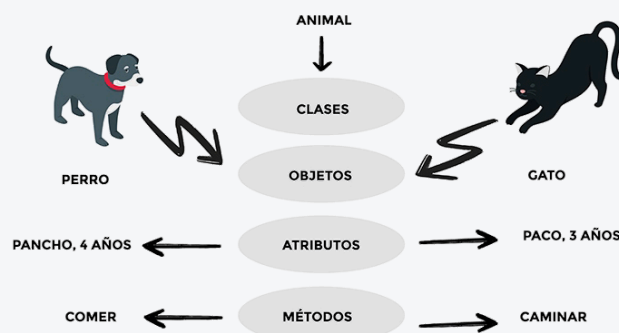


## Objetos y clase: modos de reconocerlos

Como te contamos anteriormente, un objeto de software es una representación de un objeto del mundo real, compuesto de una **serie de características y un comportamiento específico**.



A continuación te invitamos a observar la siguiente imagen:



Como se ve en la imagen **los objetos** tienen características tales como:

- **Identidad o nombre:** en este caso son Gato – Perro
- **Atributos:** en este ejemplo es la edad y el nombre. En este caso es un atributo común a ambos objetos pero pueden tener atributos diferentes.

¡Te desafiamos a que encuentres al menos 1 atributo que los diferencie!!

- Comportamiento o acción: Comer(), caminar()

¿Qué otra acción puedes encontrar en este ejemplo de la imagen?



¿Notaste que ambos pertenecen a la clase “Animal”?

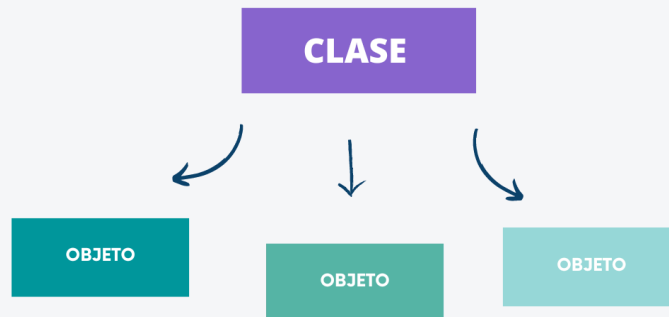




## ¿Cómo se componen los diagramas de clase?

Como te contamos, los objetos pertenecen a clases.

El **objeto auto** pertenece a la **clase vehículo** como la moto, el colectivo, el tren, la bicicleta.



En forma genérica podemos representar la relación entre clases y objetos con la imagen anterior.

Ahora, ¿para qué necesito saber esto?

No solo para **modelar** sino también para **programar** en la PC con el **paradigma de orientación a objetos**, más conocido como **POO** que vas a realizar durante la cursada del módulo de Desarrollo de Sistemas Orientados a Objetos.

Sigamos con el diagrama de clases.



## ¿Cómo definir clase y diagrama de clase?

La **clase** es una descripción de un conjunto de objetos que comparten una estructura y un comportamiento común. Y a partir de la clase, se crean tantas "copias" o "instancias" como necesitemos. Esas copias son los objetos de la clase.

Las clases constan de **datos y métodos** que resumen las características comunes de un conjunto de objetos. Un programa informático está compuesto por un conjunto de clases, a partir de las cuales se crean objetos que interactúan entre sí.

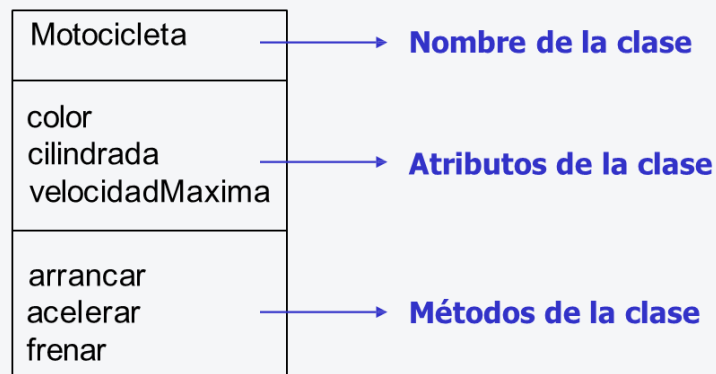
En otras palabras, **una clase es una plantilla o prototipo** donde se especifican:

- Los **atributos** comunes a todos los objetos de la clase.
- Los **métodos** que pueden utilizarse para manejar esos objetos.

El **diagrama de clases**, como su nombre lo indica es un diagrama (recordá que un diagrama es un dibujo) donde se muestran las clases y sus relaciones.

En **UML** las clases se representan por rectángulos compartimentados

- El primer compartimento contiene el nombre de la clase, que debe ser único en el paquete que la contiene.
- El segundo contiene los atributos.
- El último los métodos.

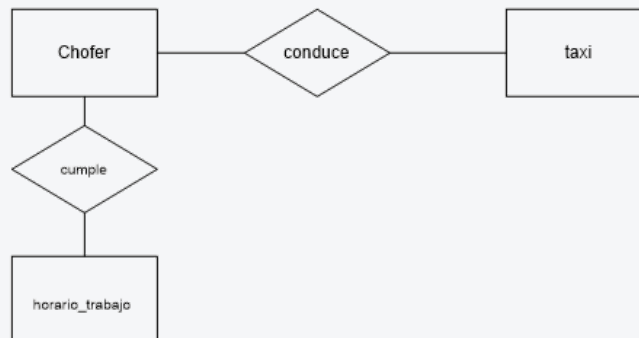




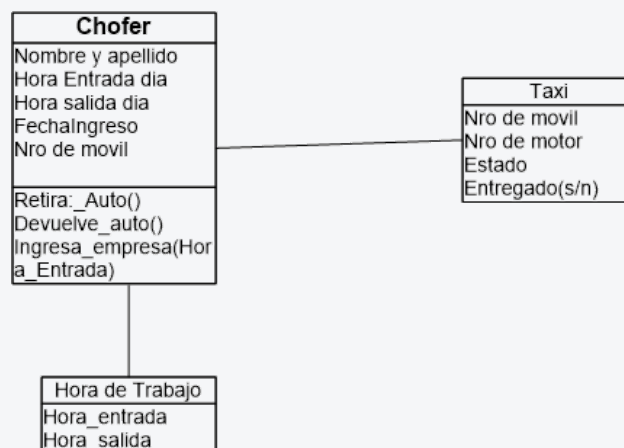
## Ejemplo de aplicación: La empresa de alquiler de taxis

La semana anterior vimos un ejemplo donde, según requerimientos del cliente, se nos pedía administrar y controlar los taxímetros y los choferes al comienzo del turno de trabajo (recordá el caso haciendo clic [aquí](#)). El sistema debía gestionar la entrega de taxis a los choferes de acuerdo al horario de trabajo de los mismos y la devolución de los coches por el chofer.

A partir de este requerimiento, hicimos el modelo E-R de 2 de los requerimientos.



Ahora, luego de conocer las nociones de clase y diagrama, vamos a transformar esos requerimientos en clases:



Si observas el ejemplo, tenemos las mismas entidades que en el modelo E-R solo que ahora se transforman en clases, con atributos y métodos. No todas las clases deben tener métodos, pero eso queda para la semana próxima.