



# Certified Tech Developer

The Ultimate Degree

## Modelado con UML (relaciones y herencia)

### Objetivo

Identificar y modelar las clases involucradas en el enunciado especificando sus atributos, responsabilidades y relaciones entre las mismas.

### Caso de estudio

La sede de licencias de conducir de una jurisdicción nos pide un sistema para gestionar los exámenes tomados. Todo examen consta de una fecha (String) y la persona que lo hizo (DNI, nombre y apellido). Como parte del examen, se debe realizar un circuito de prueba. En caso de concursar para licencia de auto se debe realizar 1 circuito y en el caso de concursar para licencia de moto son muchos circuitos.

Los exámenes de moto, además de los circuitos hechos, tienen información sobre la moto con la que se hicieron las pruebas (patente, marca, cilindrada y si llevaba todos los espejos). Los exámenes de auto guardan, además del circuito, información del auto con el que se hizo (patente, marca, cilindrada y cantidad de puertas). De los circuitos (para cualquier tipo de examen) se sabe el tiempo en segundos que tomó y el número de faltas cometidas. La lógica de aprobación para un examen de moto y de auto son diferentes.



### Microdesafíos - Paso I

Entonces, ¿cuáles son las Clases dentro del problema?

1. Leé detenidamente el enunciado buscando las posibles clases de objetos (recordá que los objetos generalmente se nombran con un sustantivo).
2. De cada clase de objeto vamos a pensar cuáles son las características, o sea, los atributos que la representan.
3. Luego de cada clase identificaremos los comportamientos (generalmente es un verbo).



## **Microdesafíos - Paso II**

Ahora debemos representar nuestras clases en un diagrama que nos sirva para poder programar a futuro el sistema necesario:

1. Con la herramienta de diagramación ( [app.diagrams.net](http://app.diagrams.net) ) generá un nuevo diagrama.
2. Agregá por cada clase de objeto un elemento de clase.
3. Para cada clase agregá todos sus atributos y métodos (responsabilidades).



## **Microdesafíos - Paso III**

Ahora debemos representar las relaciones entre las clases

4. Simplemente modelá las asociaciones. De acá en adelante no te vamos a pedir discriminar entre agregación y composición, ya que su implementación en Java no varía.

5. Una vez plasmada la relación vamos a especificar su multiplicidad. Para esto parate primero sobre una de las clases y preguntate ¿por cada instancia de esa clase cuántas instancias de la otra clase podría tener la relación? Recordá que si pueden ser muchos lo especificamos con un "\*" y este resultado lo ponemos en el extremo de la relación con la otra clase donde estamos parados. Luego hacemos lo mismo parándonos con la otra clase de la relación.



## **Microdesafío - Paso IV**

Identificar posibles relaciones de herencia.

6. Recordá que existen dos procesos mentales para identificar posibles relaciones de herencia, estas son la generalización y la especialización.
7. Si hay clases que identificaste que tienen los mismos atributos a través de la generalización, podes armar una relación de herencia.

**¡Hasta la próxima!**