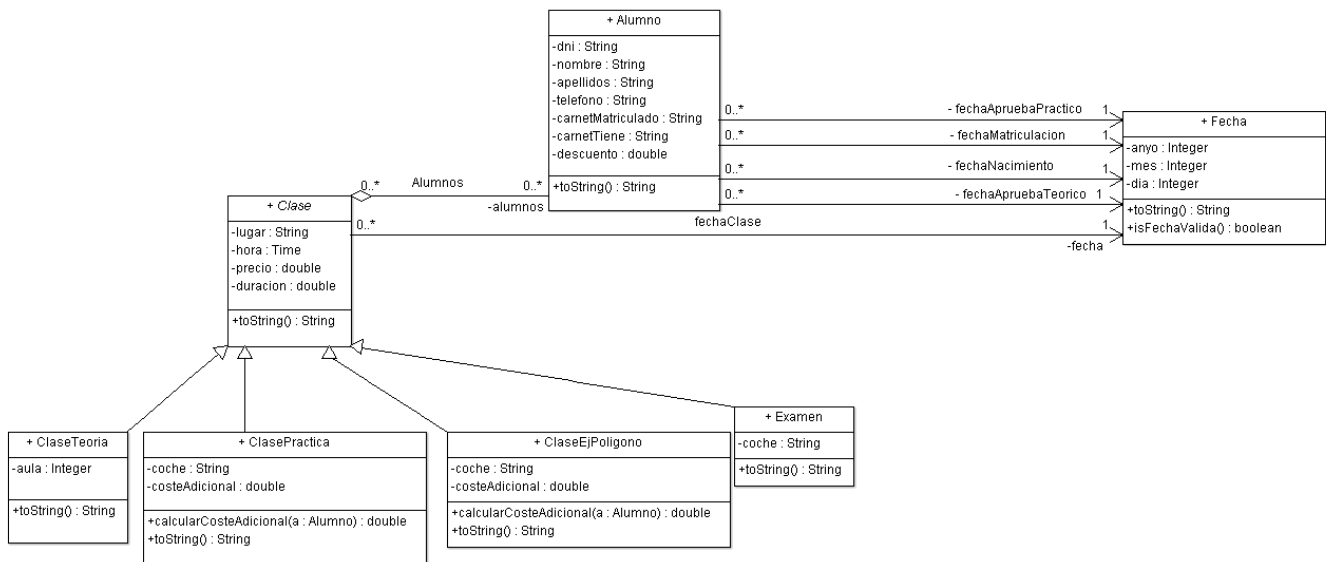


## MEMORIA PRACTICA 2

### Ejercicio 1

Se adjunta el diagrama de clases por separado también para que su visualización sea más sencilla.



En este caso el diseño del digrama es bastante simple, y no requiere de ninguna explicación especial. Como mucho aclarar que como es obvio, la Clase es una asociación de Alumno para que, en caso de borrarse la clase (ej, cancelarse), los alumnos no desaparezcan y puedan ir a otra.

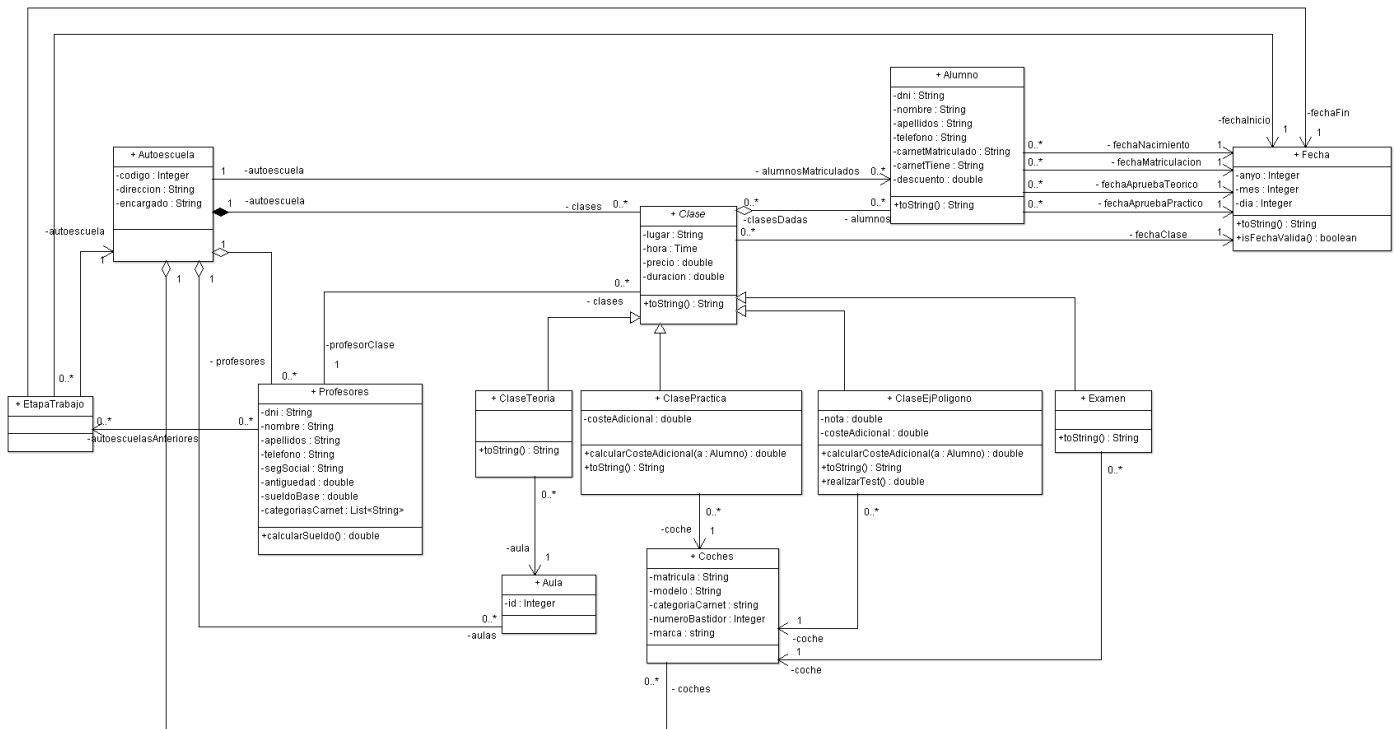
Se pide entonces una implementación de las clases Fecha y Alumno de dicho diagrama.

En el caso de la clase Fecha, no hay mayor complicación, simplemente implementar el método *isFechaValida* que nos dice si una fecha existe o no, para ello decidimos en la implementación tener en cuenta la existencia de meses de 31, 30, 28 y 29 días (en caso de ser año bisiesto). Implementamos también un método *toString* que crea una string con la fecha separada por guiones '- '.

Tras esto creamos Alumno con dni, nombre, apellidos, teléfono, fecha de nacimiento, fecha de matriculación, fecha de aprobado del teórico, fecha de aprobado del práctico, una lista con los carnets que ya tiene y el carnet para el que esta matriculado. Sin embargo, para el tester que se nos proporcionaba solo necesitabamos los campos dni, nombre, apellidos, carnet para el que está matriculado y fecha de matriculación. Es por esto último que en Alumno.java se pueden ver dos constructores, uno completo, y otro con tan solo los parámetros empleado por el TesterAutoescuela.java que se nos daba, de modo que este funcionase correctamente.

## Ejercicio 2

Se adjunta el diagrama de clases por separado también para que su visualización sea más sencilla.



Al aumentar el diagrama de clases para cumplir los requisitos del segundo ejercicio decidimos que, como el sueldo de los profesores es variable, tan solo guardaríamos su sueldo base y que el sueldo total se calcularía a través de un método a partir del base y los extras que obtenga por las clases prácticas que haya realizado, que leemos del array de clases prácticas que hay en la clase profesor.

Como no sabemos que define al encargado de una Autoescuela (nombre, DNI, un número propio de la autoescuela, etc), lo guardamos como un simple String.

La clase profesores y la clase Alumno podrían heredar de una misma clase que fuese, por ejemplo, Persona, y que tuviese DNI, nombre, apellidos y teléfono, sin embargo, a la hora de representarlo en el diagrama quedaba extremadamente confuso, con lo que decidimos dejar el diagrama así para facilitar su lectura, teniendo esto en cuenta.

Decidimos también que la antigüedad de un profesor estaría guardada como double y no como int puesto que puede ser de 2,5 años, por ejemplo.

Se pide además poder tener un historial de todas las autoescuelas donde ha trabajado un profesor. Esto lo hemos implementado creando la clase EtapaTrabajo, que tiene la fecha de inicio, fin y la autoescuela, de esta forma cada profesor tiene un número de estas etapas para poder saber sus antiguos lugares de trabajo.

Javier Delgado del Cerro y Javier López Cano

Los coches y los profesores no están unidos a la autoescuela como contenido puesto que, si la autoescuela desaparece, estos no desaparecerán, sino que pueden ir a otra autoescuela de la misma cadena.

Además, el hecho de que una clase práctica tenga solo dos alumnos lo implementamos en el código, como una comprobación, y no en el diagrama de clases, para simplificarlo.

En este ejercicio, las únicas decisiones que hemos tenido que tomar han sido respecto al taller en sí. Hemos decidido entonces que la cadena de autoescuelas está asociada únicamente con una cadena de talleres. Además, toda la información relativa al cobro de las reparaciones y revisiones, como el precio, el IVA o el descuento por ejemplo, se hace mediante código.

## **Ejercicio opcional**

Hemos programado el ejercicio opcional implementando únicamente las clases y los métodos estrictamente necesarios para su funcionamiento de acuerdo a lo pedido en el enunciado del ejercicio (Profesor, Autoescuela y EtapaTrabajo), además de haber generado toda la documentación mediante Javadoc.

Así, para probarlo basta con compilarlo y ejecutar Autoescuela, donde se encuentra el main del programa.