

## **Práctica 2**

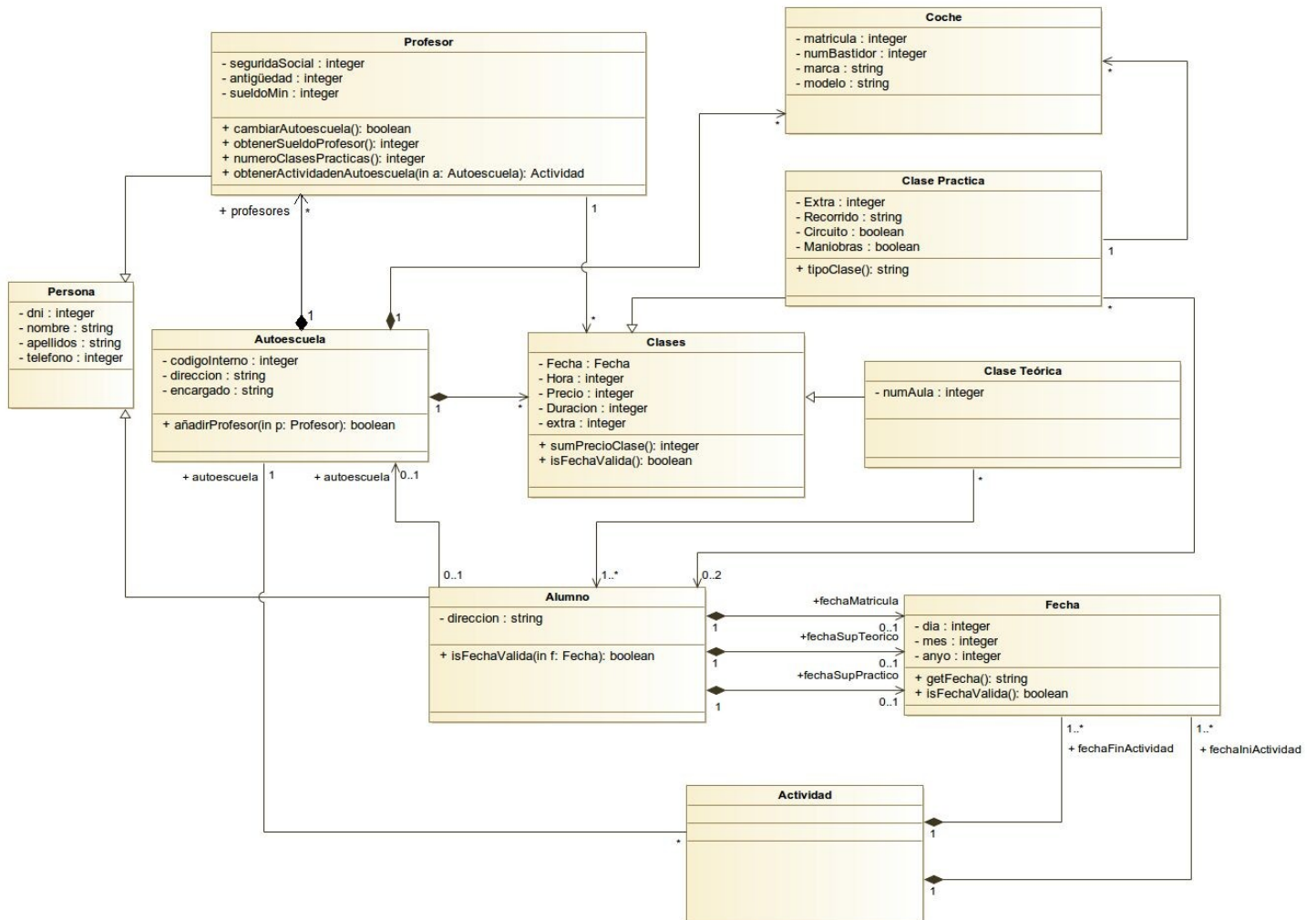
Miguel García Moya y Lucía Colmenarejo Pérez

### **Apartado 2:**

En este apartado lo que teníamos que hacer era desarrollar un diagrama de clases destinado a la gestión de una cadena de autoescuelas. Para la realización de este diagrama nos hemos servido del diagrama adjuntado en el ejercicio anterior que estaba compuesto de dos clases: Alumno y Fecha. Sin embargo en Alumno hemos introducido ciertas modificaciones. Una de ellas es que hemos creado una clase abstracta llamada Persona que va a englobar ciertos atributos que son comunes a las clases de Alumno y Profesor y que, por tanto, éstas van a heredar. Un profesor es una clase que va a tener los siguientes métodos: cambiarAutoescuela (nos va a devolver un boolean notificándonos de que el cambio de dicho profesor se ha hecho correctamente o no), obtenerSueldoProfesor(devuelve un entero con la cantidad de dinero que cobra el profesor), numeroClasesPracticas (va a devolver el número de clases prácticas que ha impartido, dato necesario para la obtención de su sueldo) y obtenerActividadenAutoescuela ( a la cual le pasamos un dato tipo Autoescuela para que nos devuelva la actividad del profesor en dicha autoescuela). En cambio en Alumno no tenemos ningún método adicional que no sea isFechaValida, para determinar si las tres fechas de las que está compuesta alumno (fecha de matriculación, fecha de superación del examen teórico y fecha de superación del examen práctico).

Otra de las clases más importantes es Autoescuela, la cual tiene los atributos que nos indicaron en el enunciado y únicamente tiene un método: añadirProfesor el cual te devuelve un boolean si has añadido correctamente al profesor en la autoescuela). Cabe destacar de que en Autoescuela hemos introducido como atributo encargado como un string ya que nos era más fácil a la hora de representarlo. Autoescuela y Profesor están unidos mediante una asociación que nos dice que una autoescuela tiene n profesores. En cambio en Alumno la cardinalidad de la asociación varía dado que un alumno como mucho tiene una autoescuela. Por otra, una autoescuela está formada por n clases. La clase Clase es una clase abstracta dado que después distinguiremos entre clases teóricas y clases prácticas. De esta forma nos ahorramos tener que escribir en dos clases los mismos atributos y luego también será mas fácil de visualizar para el usuario a la hora de relacionarla con otras clases. En esta clase sólo tendremos dos métodos, los cuales son: sumPrecioClase(devolverá un entero que será el precio de cada clase, que luego se especializará dependiendo del tipo de clase que sea) e isFechaValida. Las dos clases “hijas” de Clase , es decir, Clase Teórica y Clase Práctica tienen sus distintos tipos de atributos adecuándose a los datos proporcionados en el enunciado. La peculiaridad es el método de la Clase Práctica que se llama tipoClase el cual nos devuelve un string con el tipo de clase práctica que se imparte, dado que dependiendo de qué tipo sea el profesor cobrará un sueldo distinto. Una Clase Práctica tiene n Coches, y cada coche tiene sus atributos correspondientes. También se puede observar como una autoescuela está formada por n Coches.

Una medida que hemos optado es la creación de una clase adicional llamada Actividad que va a estar formada por fechaFinActividad y fechaIniActividad que nos permite obtener las fechas de la actividad del profesor en esa autoescuela(que pueden ser más de una, de ahí la cardinalidad). Para obtener dichas fechas utilizaremos los métodos que nos dieron en el apartado 1.



### Apartado 3:

Este apartado está implementado sobre el diagrama del apartado anterior, pero añadiendo las clases de Taller, Reparación, Revisión, Trabajo Mayor y Pieza. Un taller está formado por un número n de reparaciones y de revisiones. Estas dos últimas clases están relacionadas con Coche ya que un coche puede tener varias revisiones y reparaciones. Cada una de estas tiene un método para calcular su precio ya que cada una se calcula de una forma distinta y no tenía sentido realizar una clase para un método que prácticamente no tiene nada que ver.

Hemos implementado una clase Trabajo Mayor que está relacionado con Pieza, ya que para distintos trabajos es necesario un cierto número de piezas para su realización. En Pieza podemos ver que aparecen los dos atributos que nos han informado en el enunciado. En Trabajo Mayor hay un método que se llama IsTrabajoEnAcuerdo, que devuelve un true o un false (boolean) para ver si está de acuerdo o no con la compañía.

