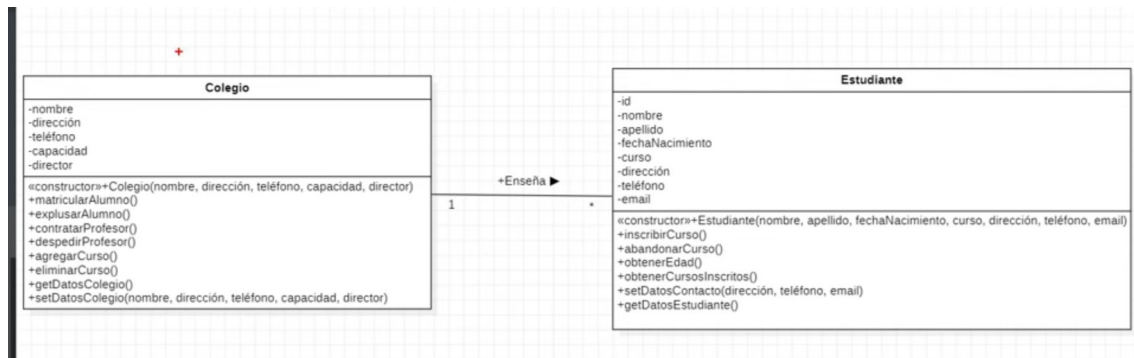


## 1. Realiza el siguiente diagrama de clases en StarUML:



## 2. Descripción del problema

El Ministerio de Educación ha encargado a BK Programación que desarrolle una plataforma de aprendizaje electrónico para que los alumnos de ciclos formativos a distancia tengan acceso a los materiales y puedan comunicarse con sus profesores. Para que los chicos puedan empezar a crear los primeros diagramas de la aplicación Ada les pasa la siguiente descripción del ámbito del problema:

“Los alumnos y alumnas de Ciclos Formativos a Distancia se matriculan de varios módulos formativos al año. Los módulos formativos son impartidos por profesores y profesoras que pondrán los contenidos del módulo a disposición de los alumnos y alumnas. Para superar un módulo hay que hacer una tarea y un examen que se calificarán de uno a diez, y sacar en ambos casos una puntuación superior a cinco. Los exámenes se componen de 30 preguntas que se eligen y ordenan al azar. Las preguntas tienen un enunciado y cuatro posibles respuestas, sólo una de ellas válida. Un ciclo formativo se compone de una serie de competencias profesionales, que tienen una descripción y que, a su vez, están formadas por uno o varios módulos, que tienen un nombre, y un número de horas. Cuando un alumno o alumna supera los módulos correspondientes a una capacidad se le certifica esa capacidad. Cuando se han superado todos los módulos (y por tanto se han adquirido todas las competencias profesionales) se aprueba el ciclo. Cuando un alumno o alumna finaliza el ciclo se emite un certificado de competencias a su nombre donde aparece la descripción de las competencias que forman el ciclo y la nota media obtenida. Si un alumno o alumna no termina de cursar el ciclo completo puede pedir un certificado que acredite las competencias que sí tenga adquiridas. El alumnado y el

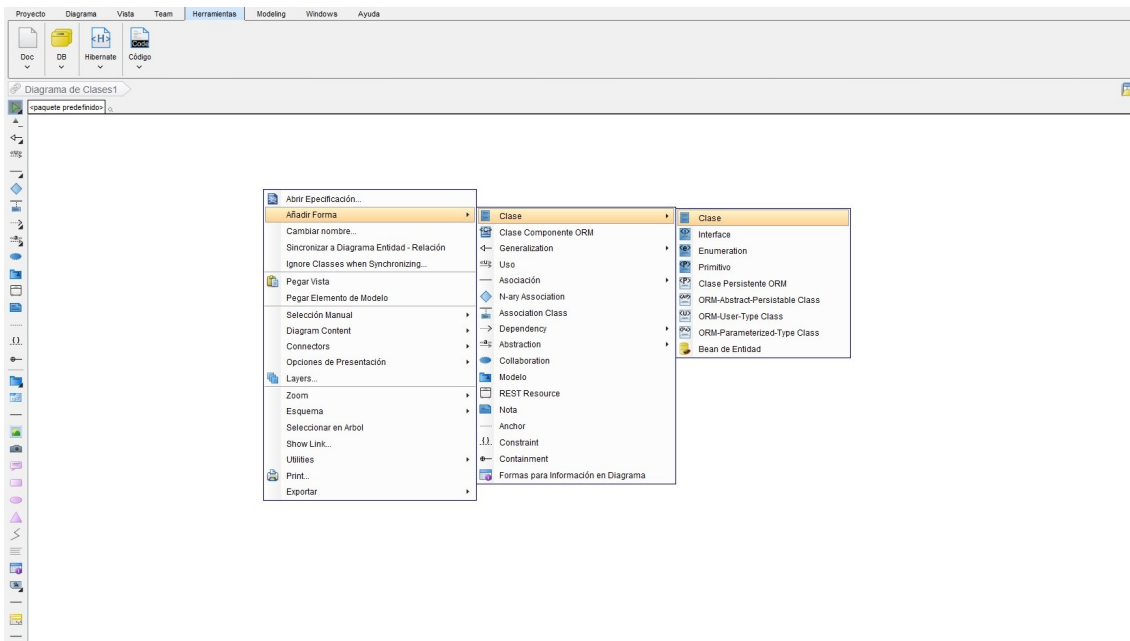
profesorado se identifican con un alias en el sistema y se comunican a través de correo electrónico. Por motivos administrativos es necesario conocer el nombre y apellidos, dirección completa y teléfono de todas las personas que participan en el sistema, sea como profesores o como alumnos. Para el profesorado, además, se debe conocer su número de registro personal (NRP)”.

**Crear un diagrama de clases nuevo en Visual Paradigm UML que incluya su nombre y su descripción.**

Para crear un diagrama de clases en VP-UML seleccionamos **Archivo >> Nuevo Diagrama** y seleccionamos Diagrama de clases. También podemos acceder al Navegador de diagramas, que se encuentra en el panel de la izquierda y en diagramas de clases hacer clic con el botón secundario y Seleccionar Nuevo diagrama de clases. Cuando generamos un diagrama nuevo tenemos que indicar su nombre y una descripción. Esto es importante para la generación de la documentación posterior.

Cuando creamos un diagrama nuevo aparece en blanco en el panel central de la aplicación. Si es necesario cambiar sus propiedades podemos hacerlo seleccionándolo en el Navegador de diagramas, a través de la opción "Abrir Specification" del menú contextual. También podemos abrir el menú contextual, haciendo clic con el botón derecho del ratón sobre el panel central de la aplicación.

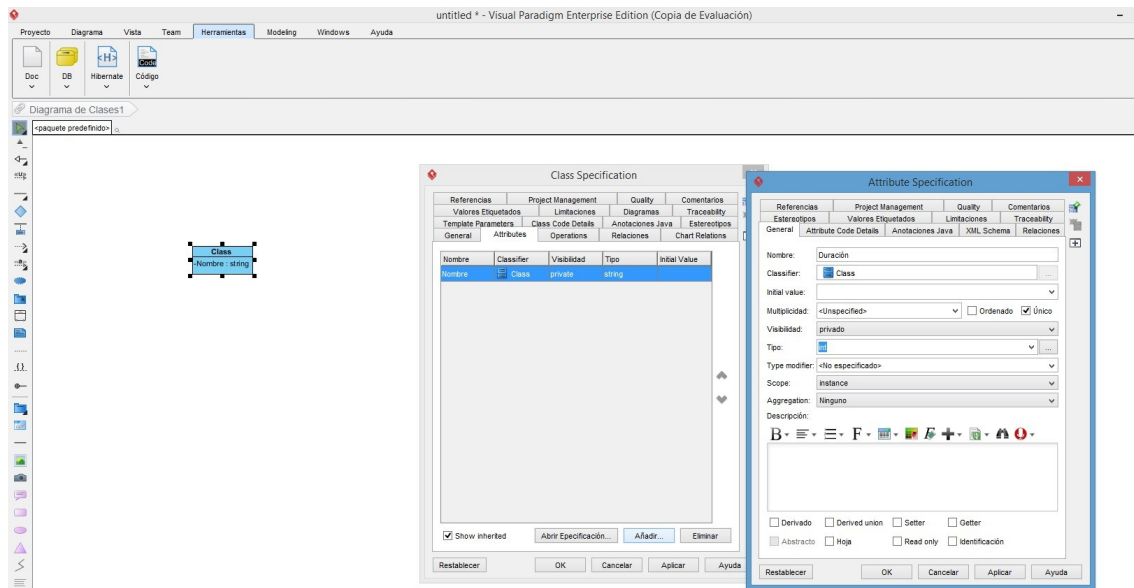
Crear una clase nueva en el diagrama de clases del punto anterior.



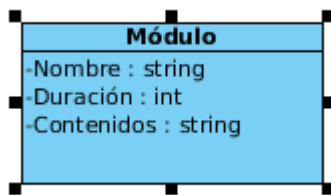
**Crear una clase de nombre "Módulo" y que tenga tres atributos:**

- **Nombre, de tipo string.**
- **Duración de tipo Int.**
- **Contenidos de tipo string.**

Crear la clase como hemos visto en el punto anterior y modificar su nombre a "Módulo". Para añadir un atributo a una clase basta con seleccionar **Añadir atributo** del menú contextual y escribir su nombre. Si queremos añadir más información podemos hacerlo desde la especificación de la clase en la pestaña Atributos, en la imagen vemos la especificación de una clase llamada Módulo, y de su atributo Contenidos para el que se ha establecido su tipo (string) y su descripción. Por defecto la visibilidad de los atributos es privado y no se cambia a menos que sea necesario.



Así queda la representación de la clase, los guiones al lado del atributo significan visibilidad privada.



Tenemos la posibilidad de añadir, desde el menú contextual de la clase, con el atributo seleccionado dos métodos llamados **getter** y **setter** que se utilizan para leer y establecer el valor del atributo cuando el atributo no es calculado, con la creación de estos métodos se contribuye al encapsulamiento y la ocultación de los atributos.

#### Añadir a la clase creada anteriormente los métodos:

- **matricular(alumno : Alumno) : void**
- **asignarDuración(duracion : int) : void**

El método más directo para crear un método es en el menú contextual seleccionar "Añadir operación" y escribir la signatura del método:

+nombre() : tipo\_devuelto

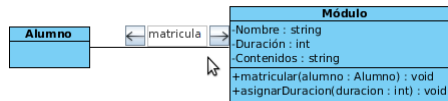
También se puede añadir desde la especificación de la clase en la pestaña *Operations*.

En la imagen vemos la especificación de la clase y del método "asignarDuración" que asigna el número de horas del módulo.

El signo + en la signatura del método indica que es público. Así queda la clase en el diagrama:



Crea una clase nueva llamada Alumno y establece una relación de asociación con el nombre "matrícula" entre ésta y la clase Módulo.



Creamos la clase como hemos visto en puntos anteriores. Para crear una relación utilizamos el elemento asociación de la paleta o bien el icono **Association >> Class** del menú contextual de la clase. Otra forma consiste en hacer clic sobre la clase **Alumno**, seleccionar **Association >> Class** y estirar la línea hasta la clase **Módulo**, aparecerá un recuadro para nombrar la relación.

**Establece la cardinalidad de la relación que has creado en el punto anterior para indicar que un alumno debe estar matriculado en al menos un módulo, o varios y que para cada módulo se puede tener ningún alumno, uno o varios.**

Si queremos establecer la **cardinalidad** abrimos la especificación de la relación y establecemos el apartado Multiplicidad a alguno de los valores que indica, si necesitamos utilizar algún valor concreto también podemos escribirlo nosotros mismos. En el caso que nos ocupa seleccionaremos la cardinalidad 0..\* para los alumnos y 1..\* para los módulos.

Association Specification

Traceability

Referencias

Project Management

Quality

Comentarios

General

Estereotipos

Valores Etiquetados

Limitaciones

Diagramas

Nombre:

matricula

Visibilidad:

<No especificado>

Association End From

Role:

Element:

Alumno

Multiplicidad:

0..\*

Navegable:

True

Association End To

Role:

Element:

Modulos

Multiplicidad:

1..\*

Navegable:

True

Descripción:

B

F

☐ Abstracto

☐ Hoja

☐ Derivado

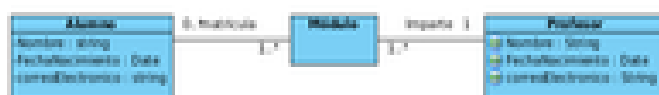
Restablecer

OK

Cancelar

Aplicar

Ayuda



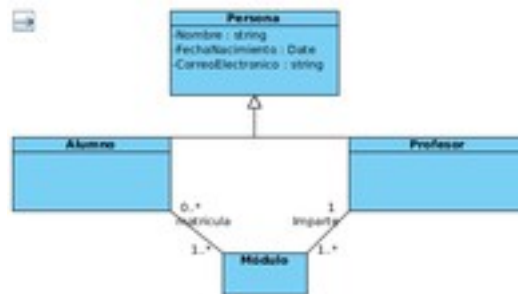
En nuestro diagrama tenemos Alumnos y Profesores. Aún no hemos hablado de su definición y estructura, pero en nuestro sistema tanto un alumno como un profesor tienen unas

características comunes como el nombre, la fecha de nacimiento o el correo electrónico por el hecho de ser personas:

Transforma este diagrama para hacer uso de la herencia añadiendo una clase "Persona".

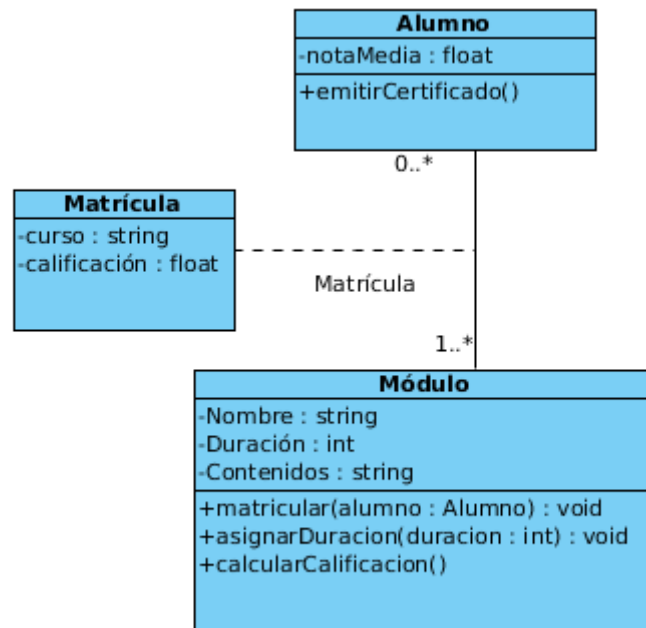
Podemos utilizar la relación de herencia para crear una clase nueva que se llame Persona y que recoja las características comunes de profesor y alumno. Persona será la **clase base** y Profesor y Alumno las **clases derivadas**.

Como los atributos Nombre, FechaNacimiento y correoElectronico se heredan de la clase base no hace falta que aparezcan en las clases derivadas, por lo que las hemos eliminado. Después podemos añadir atributos o métodos propios a las clases derivadas. La relación se añade de igual manera que una relación de asociación, pero seleccionando la opción Generalization.



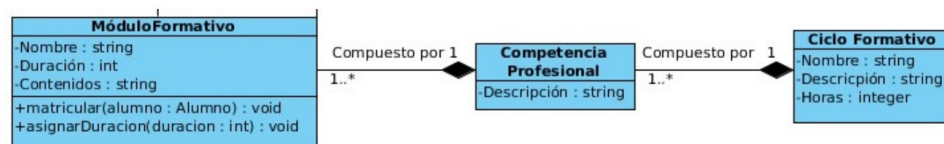
**Cuando un alumno se matricula de un módulo es preciso especificar el curso al que pertenece la matrícula, las notas obtenidas en el examen y la tarea y la calificación final obtenida. Estas características no pertenecen totalmente al alumno ni al módulo sino a la relación específica que se crea entre ellos, que además será diferente si cambia el alumno o el módulo. Añade estos atributos al enlace entre Alumno y Módulo.**

Para modelar esto en Visual Paradigm creamos una clase nueva (Matrícula) junto a Alumno y Módulo, y la unimos a la relación utilizando el icono que rodea a la propia clase "Association class", el diagrama queda así:



Con las clases ya extraídas y parcialmente definidas (aún faltan por añadir métodos y atributos inferidos de posteriores refinamientos y de nuestro conocimiento) podemos empezar a construir relaciones entre ellas.

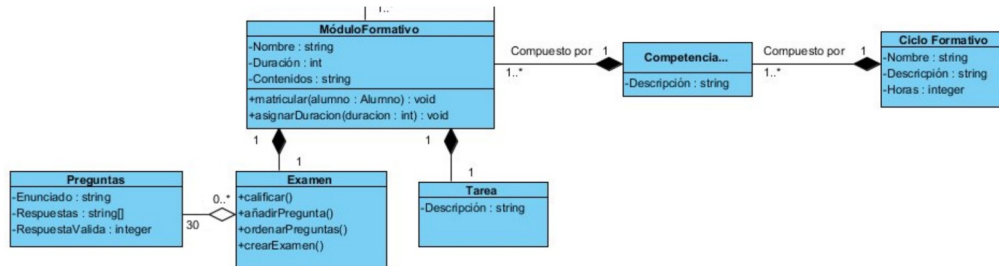
Comenzaremos por las clases que hacen referencia a la estructura de los Ciclos, cada Ciclo se compone de una o más competencias profesionales, que no tienen la capacidad de existir por sí mismas, es decir, la competencia no tiene sentido sin su ciclo, por lo que vamos a crear una relación entre ambas clases de composición. De igual manera una competencia profesional se compone de un conjunto de módulos formativos (1 o más) por lo que relacionaremos ambas, también mediante composición.



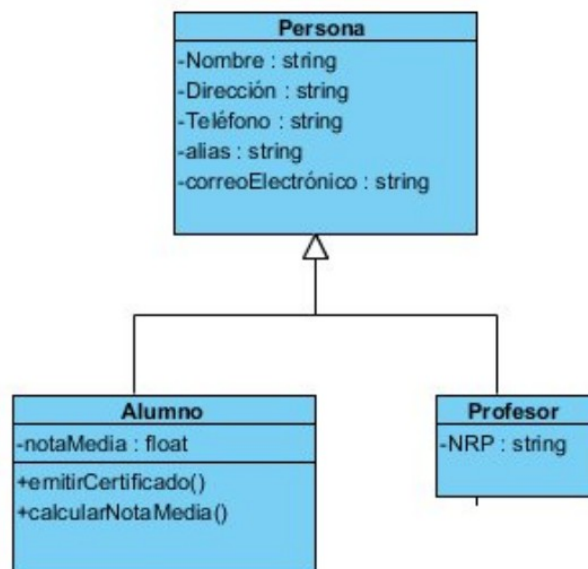
Un módulo formativo a su vez, contiene un examen y una tarea, que tampoco tienen sentido por sí mismos, de modo que también los vamos a relacionarlos mediante composición. El examen por su parte se compone de 30 preguntas, pero éstas pueden tener sentido por sí mismas, y pertenecer a diferentes exámenes, además, el hecho de eliminar un examen no va



a dar lugar a que las preguntas que lo forman se borren necesariamente, si leemos con atención el enunciado, podemos deducir que las preguntas se seleccionan de un repositorio del que pueden seguir formando parte [... [Los exámenes se componen de 30 preguntas que se eligen y ordenan al azar...], así que en este caso usaremos la relación de agregación para unirlos.



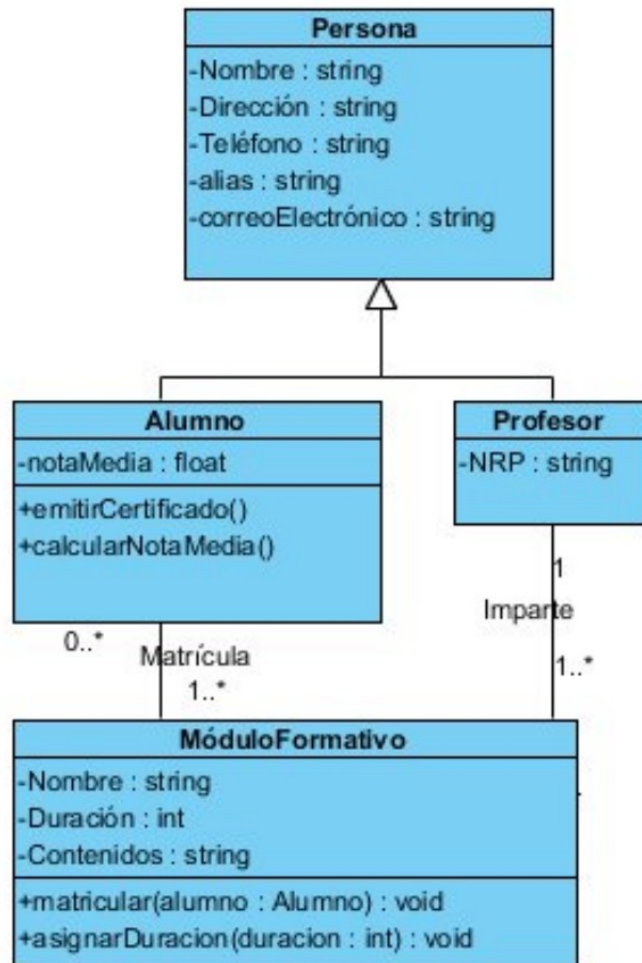
Por otra parte alumnos y profesores comparten ciertas características, por necesidad del sistema, como son los datos personales, o el correo electrónico, esto induce a pensar que podemos crear una abstracción con los datos comunes, que de hecho, ya hemos obtenido del enunciado en la clase persona, que recoge las coincidencias entre alumnos y profesores y añadir una relación de herencia de la siguiente manera:



Por último queda relacionar a alumnos y profesores con los módulos formativos. Un alumno se matricula de un conjunto de módulos formativos, y un profesor puede impartir uno o varios módulos formativos.

Más concretamente, de cara a la cardinalidad, un alumno puede estar matriculado en uno o varios módulos, mientras que un módulo puede tener, uno o varios alumnos matriculados. Por su parte un profesor puede impartir uno o varios módulos, aunque un módulo es impartido por un profesor.

Éste análisis da como resultado lo siguiente:



### Añadir Getters, Setters y constructores

Por último añadimos los métodos que permiten crear los objetos de las clases (constructores) así como los que permiten establecer los valores de los atributos no calculados y leerlos (getters y setters), recuerda que para tener éstos métodos completos es necesario que el atributo tenga establecido su tipo, para que sea tenido en cuenta.

Para añadir los getters y setters en Visual Paradigm, basta con desplegar el menú contextual del atributo y seleccionar la opción Create Getter and Setter.