

Programación Orientada a Objetos

Tarea 3.1. Relaciones entre clases

José Fidel Argudo Argudo Francisco Palomo Lozano
Inmaculada Medina Bulo Gerardo Aburrizaga García
Pedro Delgado Pérez



Versión 2.0



Tarea 3.1. Cuestiones

Ejercicio 1

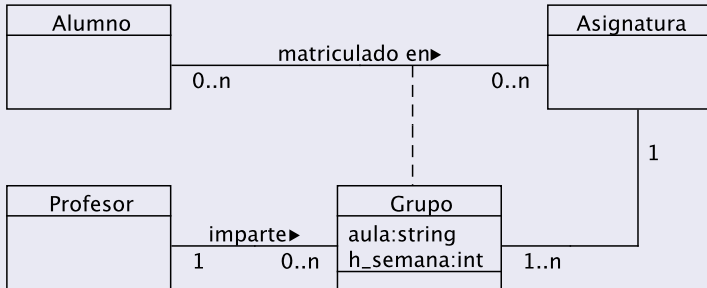
Suponga que hay que desarrollar la interfaz de usuario de una aplicación. Dicha interfaz estará formada por **menús**, **opciones** y **formularios**. Hay que tener en cuenta que:

- Desde cada menú se puede ejecutar una serie de opciones.
- La selección de una opción desencadena la activación de un formulario.
- Una opción solo puede aparecer en un menú, pero un mismo formulario puede ser compartido por varias opciones.

Describa e implemente las relaciones que se establecerán entre estas clases.

Tarea 3.1. Cuestiones

Sea el diagrama de clases siguiente:



Ejercicio 2

Implemente las clases del diagrama, declarando exclusivamente los miembros imprescindibles para implementar las relaciones.

Tarea 3.1. Cuestiones

Ejercicio 3

Defina los dos métodos siguientes:

- `Alumno::matriculado_en()` para matricular a un alumno en una asignatura asignándole un grupo.
- `Profesor::imparte()` para vincular un profesor a un grupo.

Tarea 3.1. Cuestiones

Ejercicio 4

Declare una clase de asociación `Alumno_Asignatura` para la relación `matriculado_en`. Para ello declare los atributos que considere necesarios y dos métodos `matriculado_en()` y `matriculados()`. El primero registra a un alumno en una asignatura asignándole el grupo y el segundo devuelve todas las asignaturas (y los correspondientes grupos) en que se encuentre matriculado un alumno. Declare sobrecargas de estos dos métodos para el otro sentido de la asociación.

Tarea 3.1. Cuestiones

Ejercicio 5

Declare una clase de asociación **Profesor_Grupo** para la relación **imparte**. Incluya en ella los atributos oportunos y dos métodos `imparte()` e `impartidos()`. El primero enlaza un profesor con un grupo y el segundo devuelve todos los grupos que imparte un profesor. Sobrecargue ambas funciones miembro para el sentido inverso de la relación.

Tarea 3.1. Cuestiones

Ejercicio 6

Defina el método `Alumno_Asignatura::matriculado_en()` (y su sobrecarga) para matricular a un alumno en una asignatura asignándole un grupo. Esta función también permitirá cambiar el grupo al que pertenece un alumno ya matriculado en la asignatura.

Tarea 3.1. Cuestiones

Ejercicio 7

Escriba la definición de `Profesor_Grupo::imparte()`. Si el grupo ya tiene un profesor asociado, se deberá desvincular del mismo y enlazarlo con el nuevo.

Tarea 3.1. Cuestiones

Ejercicio 8

Defina `Profesor_Grupo::impartidos()`.