

Ejemplo Unidad 1

Problema

Un sistema de tareas en línea, tendrá dos tipos de usuarios: el profesor y el estudiante, la tarea va a tener descripción, fecha de entrega y puntaje. Un estudiante puede realizar 1 entrega por tarea, mientras que el profesor puede revisar todas las entregas que han sido realizadas por los alumnos.

Requerimientos funcionales

- En línea: este requerimiento nos dice que va a ser un sistema que utilice la red, es decir no es un sistema aislado. Desde ya tenemos que pensar en usar una tecnología que soporte la comunicación web. Otro requerimiento que podemos obtener de esto es que puede ser: una página web o una aplicación para dispositivos móviles o para computadoras, cualquiera de las dos puede cumplir el requerimiento de que sea en línea. Este es un punto que se tiene que discutir y aclarar con los clientes que saben lo que quieren realizar con la aplicación.
- Dos tipos de usuarios: como nos dicen que tienen que haber profesores y estudiantes eso nos dice que habrán dos roles, es decir dos tipos de usuarios que tendrán acceso a características diferentes de la aplicación que estamos desarrollando.
- Las tareas tienen una fecha de entrega: de las 3 características que tienen las tareas esta es importante, que la tarea tenga una fecha de entrega nos puede servir para agregar comportamientos al sistema de entregas, por ejemplo, luego de la fecha ya no se puede entregar la tarea.
- Un estudiante puede entregar 1 respuesta por tarea: esto nos dice que no se pueden entregar varias respuestas a la misma tarea, y al tratarse de un sistema en línea también nos dice que hay que planear una forma de que se realicen las entregas, por ejemplo, que podamos colocar enlaces a sitios de almacenamiento como Dropbox o Google Drive como respuesta.
- Un profesor puede ver todas las respuestas de los estudiantes: este requerimiento puede parecer obvio, sin embargo es de vital importancia. Las tareas que asigna un profesor van a ser respondidas por un estudiante, y el profesor va a necesitar poder acceder a dichas respuestas, entonces toda respuesta que se envíe a una tarea asignada por un profesor podrá ser revisada por dicho profesor.

Clases que vamos a necesitar

Ahora vamos a listar las clases más importantes, pueden haber más clases dependiendo de su propio análisis, pero estas son las que considero esenciales

Profesor: hay dos tipos de usuarios, este es el primero, en este análisis simplemente vamos a decir que será el que asigne y revise las tareas.

Estudiante: el segundo tipo de usuario, lo que hará en estos momentos será revisar las tareas asignadas y responder a las mismas.

Tarea: la asignación que realiza el profesor. Una tarea tiene tres atributos, la descripción, la fecha de entrega y un puntaje, ya hablamos anteriormente de la importancia de la fecha de entrega.

Entrega: Una entrega será la respuesta de un estudiante a una tarea determinada. Tendrá 2 atributos: La fecha en la que fue entregada y el contenido de la entrega, este puede ser: un archivo de texto como un documento de Word o un PDF, un enlace a un sitio de almacenamiento como DropBox o Google Drive o un archivo comprimido.

Ya con los requerimientos y las clases podemos realizar los diagramas de clases. Vamos a definir acá también las relaciones entre las clases:

Relaciones entre las clases

- Profesor

- 1 a muchas con Tarea: un profesor puede asignar tareas, será el profesor el que decida la descripción, la fecha de entrega y el puntaje de las mismas.
- 1 a muchas con Entrega: las entregas son las respuestas de un alumno a una tarea, el profesor va a revisar las tareas esto solo si ya hay respuestas, por lo que puede no estar relacionado con una entrega en un momento dado.

- Estudiante

- De 1 a varios estudiantes se les van a asignar muchas tareas. Esta relación nos va a decir que pueden existir muchas tareas asignadas por un profesor, y cada tarea se le va a asignar a todos los estudiantes. Si por ejemplo tuviéramos 10 estudiantes y 3 tareas, a cada uno de los 10 estudiantes se le

asignan las mismas 3 tareas. Veamos brevemente una comparación entre las multiplicidades comparadas con las de profesor:

profesor con tarea 1 profesor asigna * tarea
estudiante con tarea 1..* estudiante tiene asignado * tarea

¿Qué nos dice aquí la multiplicidad?

Nos dice que un profesor puede asignar 0 o más tareas y al mismo tiempo todas las tareas que hayan sido asignadas por un profesor estarán asociadas solamente con 1 profesor, aquel el que las asignó.

Con un estudiante, la relación tiene un rango, lo que nos dice que varias tareas pueden estar relacionadas con 1 o más estudiantes. Es decir, la misma tarea la van a tener asignada varios estudiantes.

Cada tarea estará relacionada solamente con un profesor, pero puede estar relacionada con más de 1 alumno. Por esta razón es importante definir correctamente las distintas multiplicidades de las relaciones, si la multiplicidad de estudiante también fuera 1 a * significa que 1 solo estudiante tiene asignadas muchas tareas, el problema es que esas tareas ya no se le podrían asignar a otros estudiantes. Los rangos pueden ser cualquier rango de M a N donde M y N son números naturales (incluyendo el 0).

Ahora volvamos a ver el resto de las relaciones.

- De 1 a muchos con entrega: Esto nos dice que en un momento dado un alumno puede tener 1 entrega o puede no haber realizado la entrega. Es de 1 a muchos porque un alumno puede tener varias entregas o puede no tener ninguna, no es de 1 a 1 porque el alumno no va a realizar solo 1 entrega, sino que se va a realizar 1 entrega por tarea.

Esto nos presenta una dificultad en el diagrama: en nuestros

requerimientos nos indica que solo se puede realizar 1 entrega por tarea ¿como mostramos eso en el diagrama? No podemos colocar la relación de 1 a 1 ya que eso significa que 1 alumno solo tiene 1 entrega asociada, sin

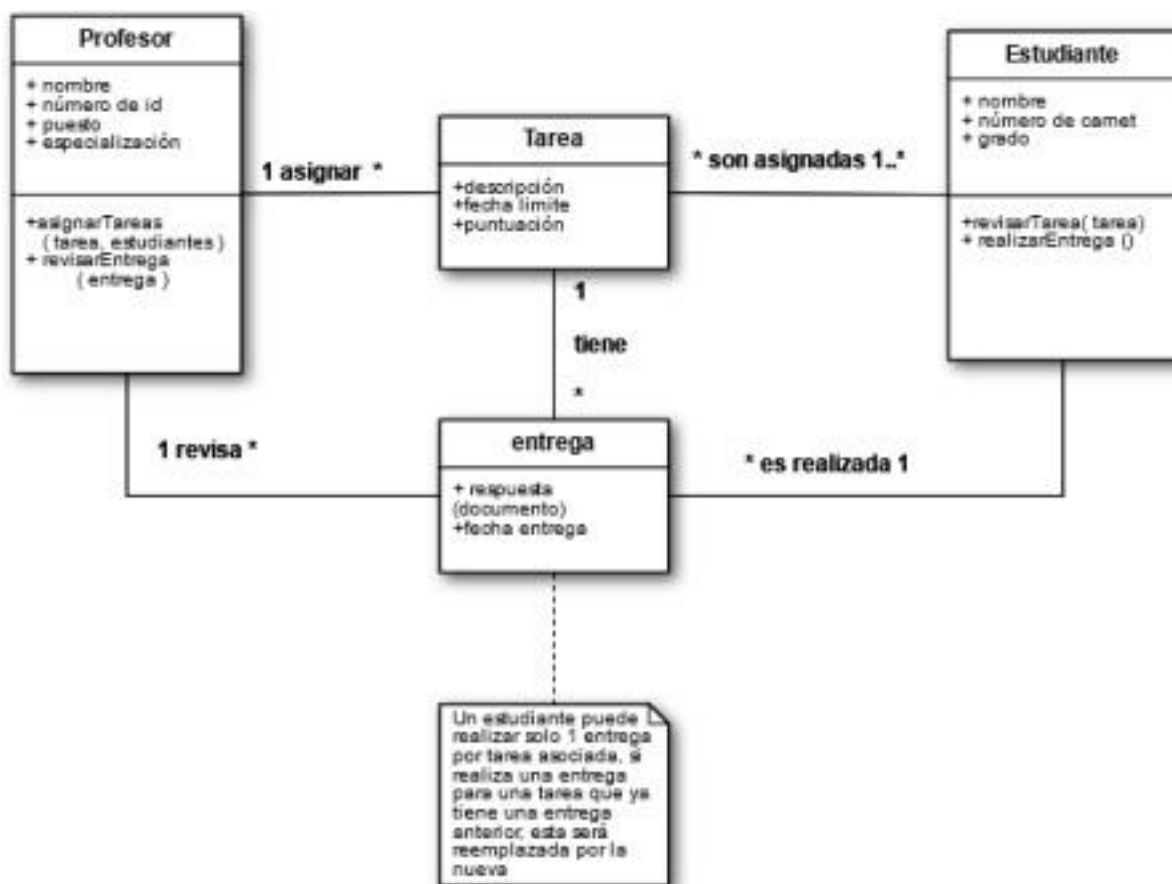
importar la cantidad de tareas que pueda tener, va a tener solamente 1 entrega. Este problema no se puede mostrar claramente en el diagrama de clases ya que es un comportamiento, el diagrama de clases no puede mostrar eso, en cambio tenemos que indicar en nuestra documentación detalla el comportamiento esperado de esta relación para que se pueda resolver en la implementación del sistema.

- Tarea:

- 1 a muchos con Entrega: una tarea es solamente la descripción de lo que se ha de realizar, una entrega es la respuesta de la misma y como dijimos anteriormente, hay varios estudiantes que van a enviar una respuesta a una tarea de ahí la relación de 1 a muchos

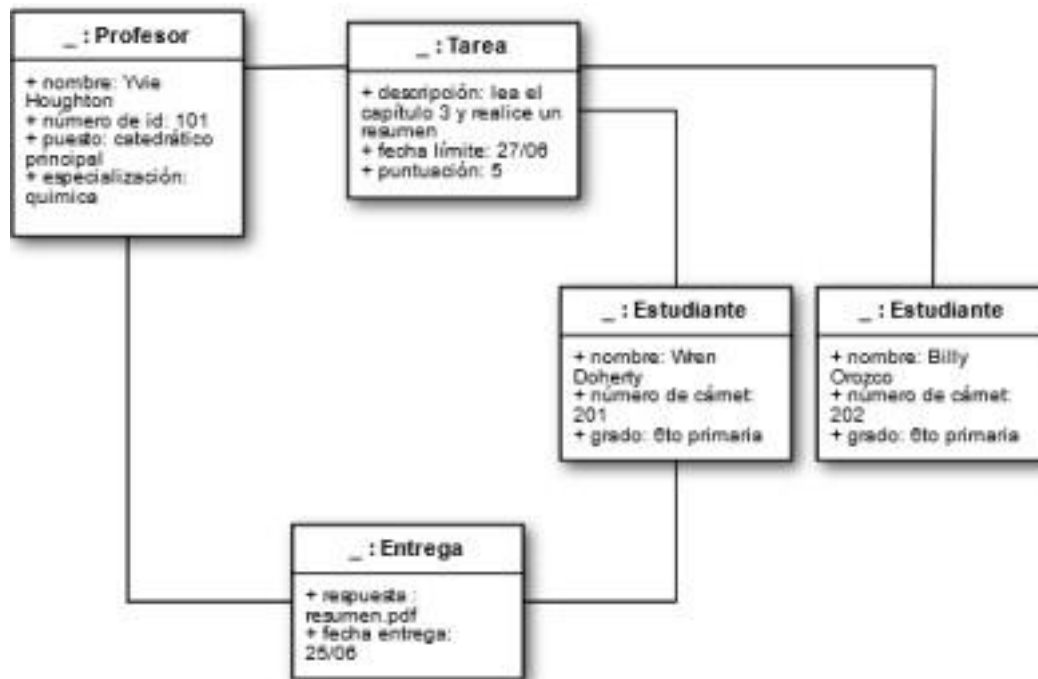
Estas serán las clases y las relaciones de las mismas que se usarán para realizar los diagramas.

Diagrama de Clases



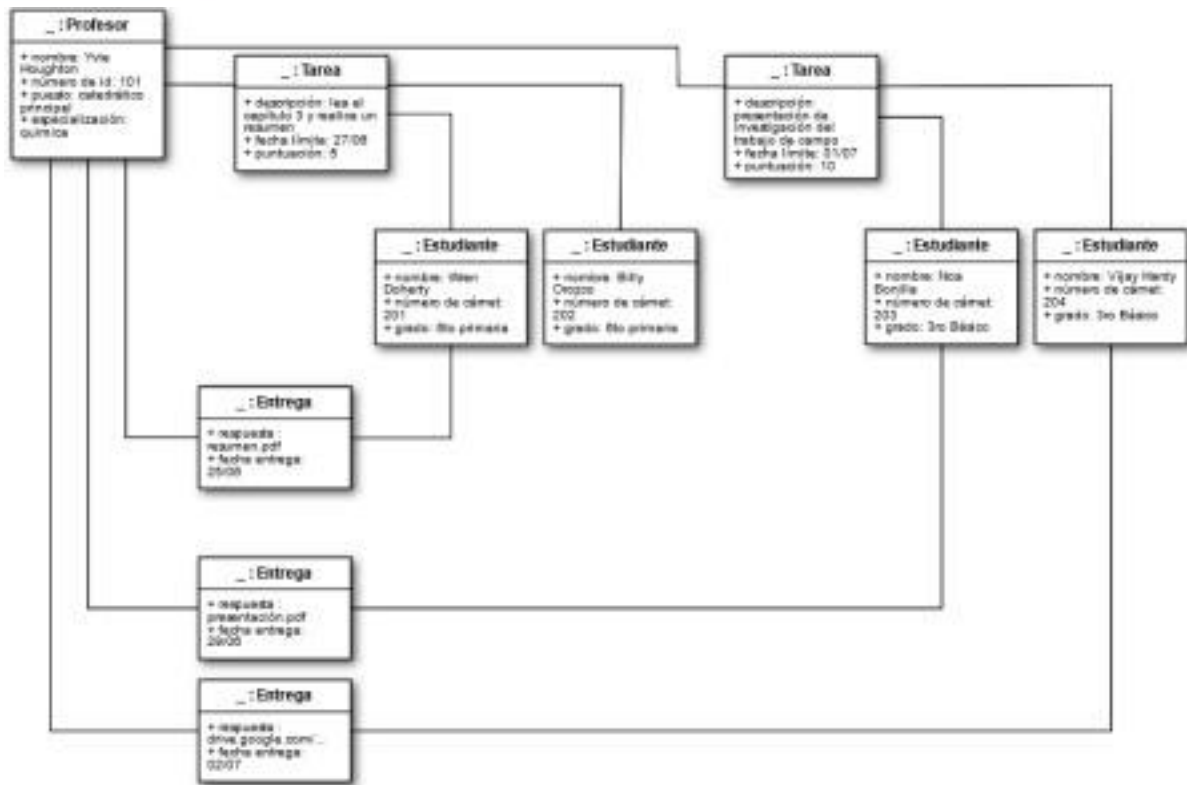
Aquí colocamos una nota, las notas son elementos de UML que se pueden usar para dar una mejor explicación o algo más de contexto a algún elemento, lo usamos generalmente para colocar restricciones, condiciones o consideraciones que sean difíciles de mostrar usando solamente los símbolos.

Diagrama de Objetos

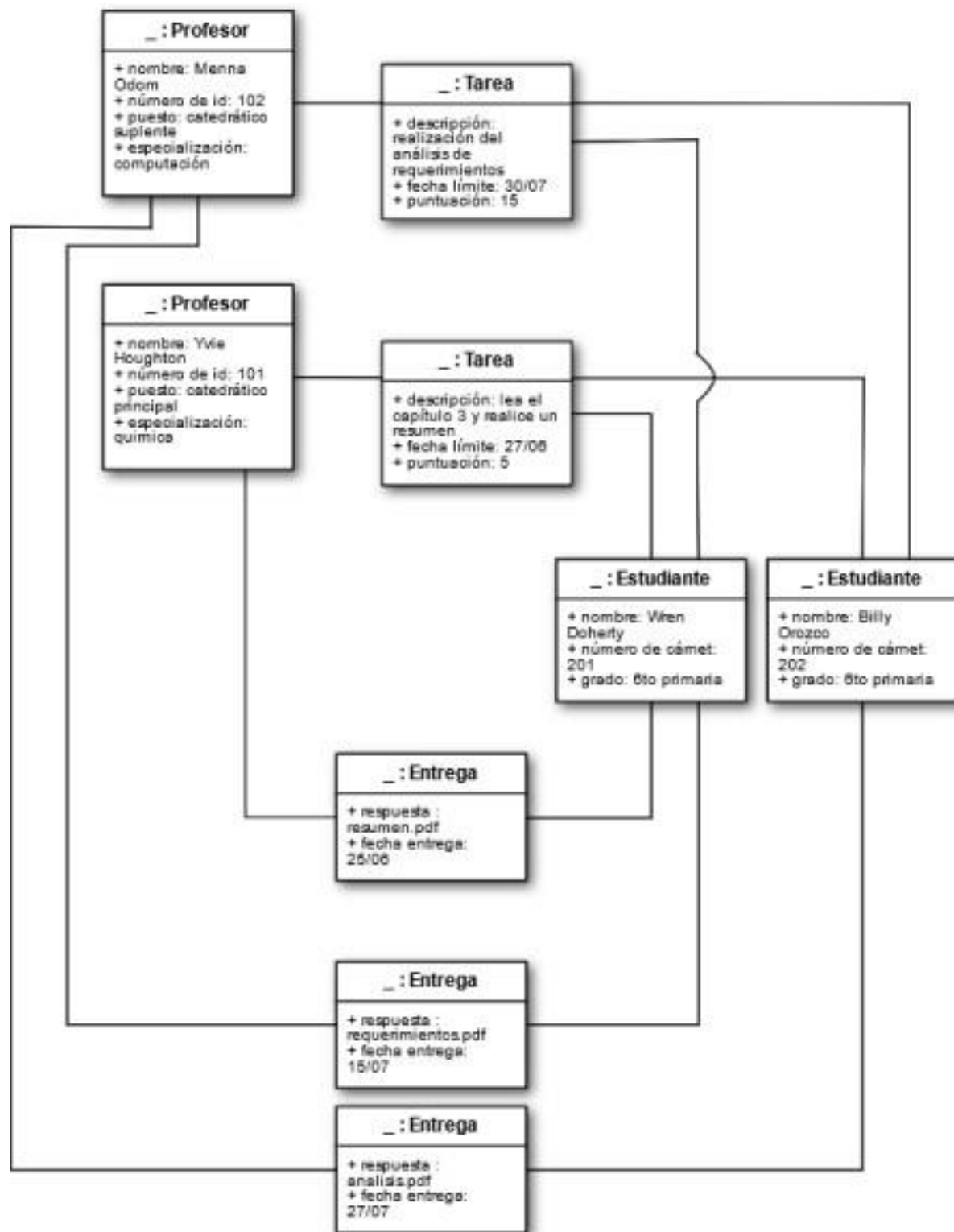


Un

diagrama de objetos simple es aquel que cumple con las relaciones mínimas que plantean las multiplicidades, es decir si dice de 1..4 con que aparezca 1 es suficiente. Hay que tener en cuenta que si el límite inferior es 1, no puede dejar de aparecer el objeto en el diagrama, si hay una multiplicidad de 1 y no se coloca el objeto en el diagrama de objetos, este es incorrecto, ya que no representa el mismo diagrama de clases.



Este diagrama de objetos también es válido, representa un sistema donde hay varias tareas asignadas a múltiples estudiantes, y cada uno tiene su entrega particular. Las tareas se pueden asignar a muchos estudiantes, pero las entregas solo son realizadas por 1 estudiante, es decir, no puede haber una entrega realizada por 2 o más estudiantes.



En este último caso vemos que hay varios profesores asignando tareas a los mismos estudiantes, se puede entender como que los estudiantes llevan varios cursos durante un periodo de tiempo y los profesores de los mismos asignan tareas de sus cursos.