

PROGRAMACIÓN EN JAVA
PROF. ING. GUIDO ACOSTA

Diagrama de Clases

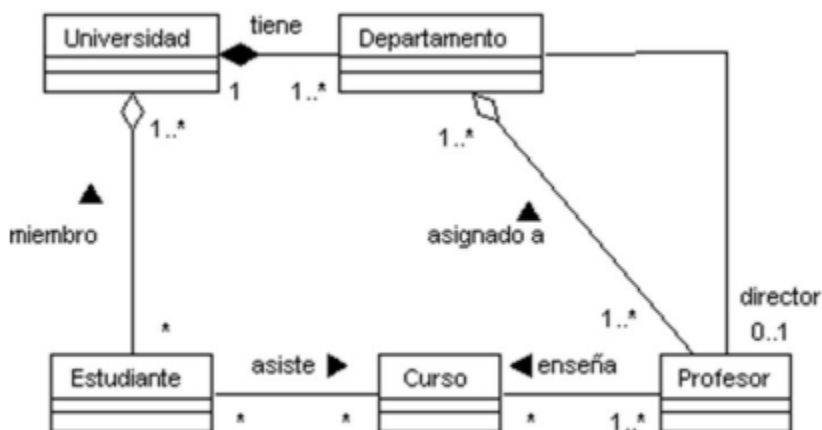
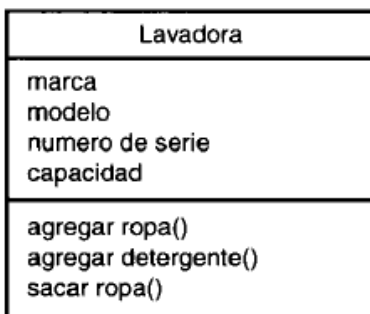
¿Qué es un diagrama de clases?

En ingeniería de software, un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

¿Qué es UML?

El UML (Lenguaje Unificado de Modelado) es una herramienta que permite a los desarrolladores de sistemas generar diseños que capturen sus ideas en forma convencional y fácil de comprender para comunicarlas con otras personas.

Ejemplo de diagrama de clases



Herramienta para diseñar diagramas de clases

www.lucidchart.com

StarUML

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

Diseñando el diagrama de clases en lucidchart

1. Panel de control del Lucidchart

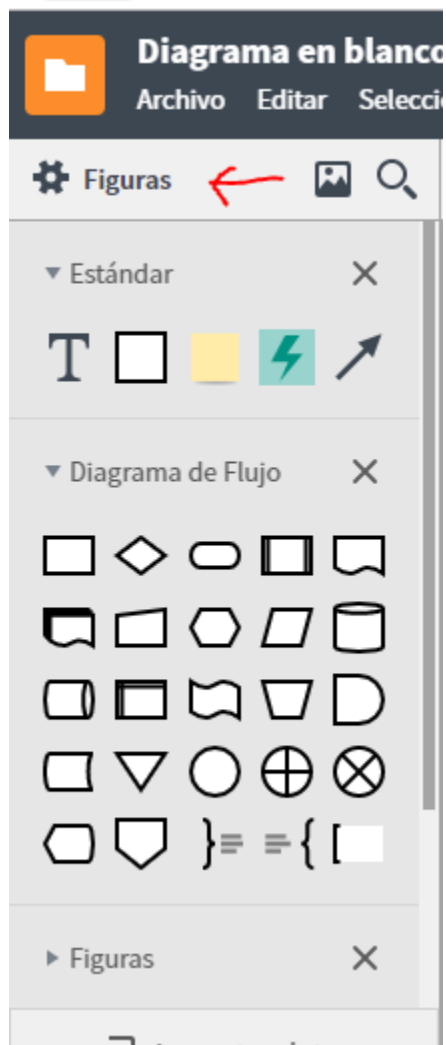


2. Generar un nuevo documento

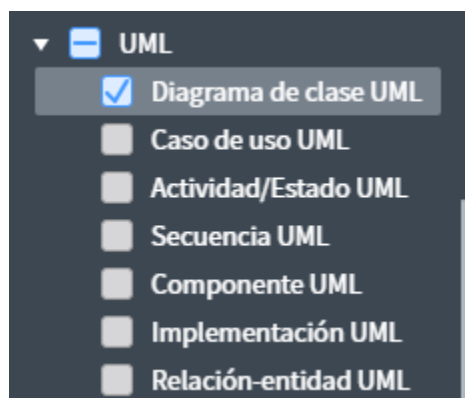


3. Seleccionar las figuras que queremos utilizar

PROGRAMACIÓN EN JAVA
PROF. ING. GUIDO ACOSTA



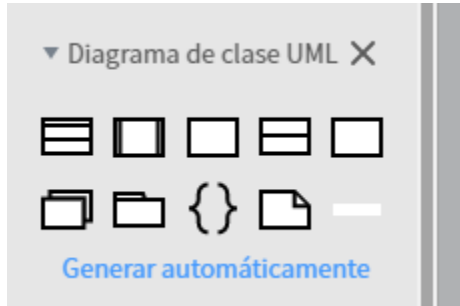
4. Seleccionar los gráficos de UML



5. A partir del grupo de **Diagrama de Clases UML** diseñar el diagrama:

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA



Analizando el ejercitario de Gestión Académica – Parte 1

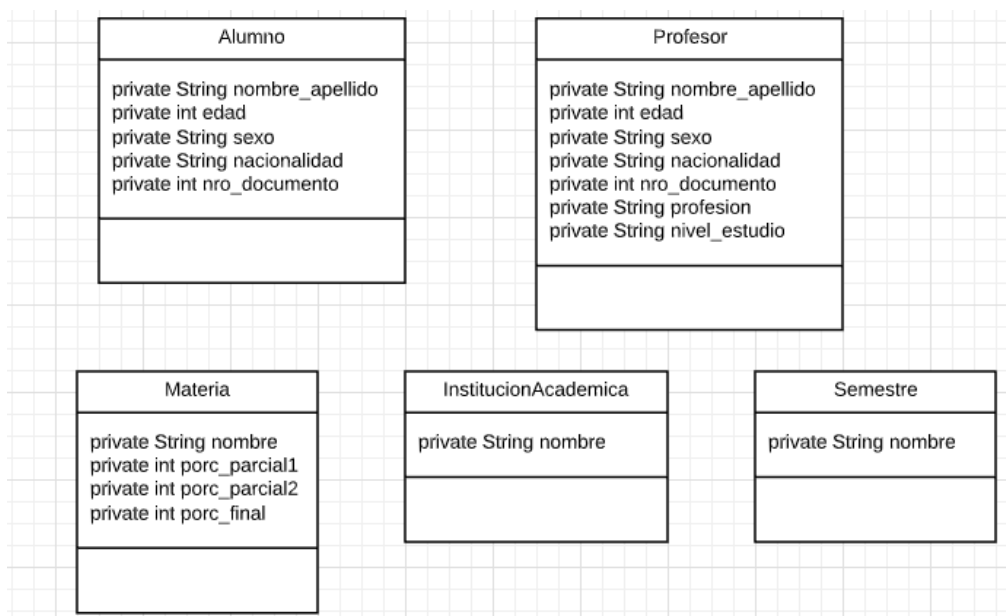
1. Identificando las clases [sustantivos] (las posibles clases se marcan en verde)

Escribir un programa que gestione el rendimiento de alumnos y profesores de una institución académica. Los alumnos tienen la posibilidad de matricularse en varias materias en cada semestre. Por cada materia, los alumnos tienen dos exámenes parciales y un examen final. El alumno tiene derecho a rendir examen final si logra un promedio mayor a 50% en los exámenes parciales. La nota final del alumno se obtiene aplicando la siguiente fórmula:

$$\text{Nota final} = 0.5 * \text{promedio_parciales} + 0.5 * \text{porcentaje_final}$$

Todos los exámenes se basan en una escala de 100.

Clases identificadas hasta el momento



PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

Además, para calificar al profesor, el programa deberá ofrecer una funcionalidad para el cálculo del promedio de la materia. El rendimiento del profesor se obtiene a partir del promedio de la materia.

Cada materia tiene una cantidad límite de alumnos que debe ser validada al momento de que el administrador matricule a los alumnos.

¿Administrador es una clase?

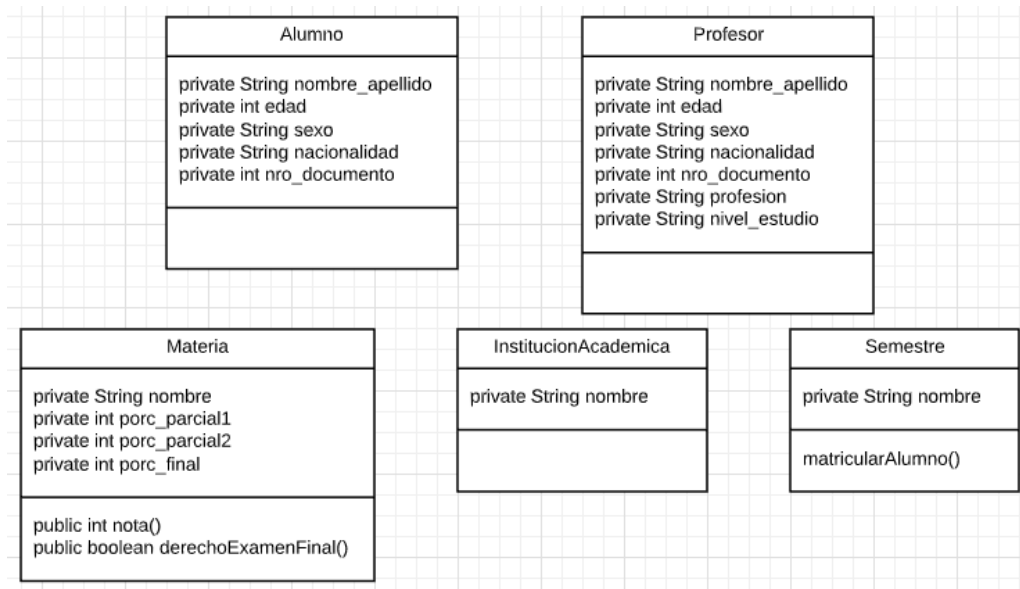
2. Identificando los métodos [verbos] (los posibles métodos están marcados en naranja)

Escribir un programa que gestione el rendimiento de alumnos y profesores de una institución académica. Los alumnos tienen la posibilidad de matricularse en varias materias en cada semestre. Por cada materia, los alumnos tienen dos exámenes parciales y un examen final. El alumno tiene derecho a rendir examen final si logra un promedio mayor a 50% en los exámenes parciales. La nota final del alumno se obtiene aplicando la siguiente fórmula:

$$\text{Nota final} = 0.5 * \text{promedio_parciales} + 0.5 * \text{porcentaje_final}$$

Todos los exámenes se basan en una escala de 100.

Métodos identificados hasta el momento



Además, para calificar al profesor, el programa deberá ofrecer una funcionalidad para el cálculo del promedio de la materia. El

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

rendimiento del profesor se obtiene a partir del promedio de la materia.

Cada materia tiene una cantidad límite de alumnos que debe ser validada al momento de que el administrador matricule a los alumnos.

Tareas

1. Definir las clases necesarias

2. Crear un menú con las operaciones disponibles:

1. Matricular alumno

2. Cargar datos del profesor

3. Cargar notas de parciales: para la carga de las notas parciales el programa mostrar el nombre del alumno y solicitar las notas de los dos parciales.

4. Cargar nota de examen final: para la carga de la nota del examen final, el programa deberá mostrar el nombre del alumno y solicitar el porcentaje obtenido en el examen final. El programa solo solicitará la nota del alumno que tuvo derecho a rendir examen final.

5. Mostrar resultados

1. Calificación del profesor de una materia dada.

2. Calificación de un alumno en particular en una materia dada

3. Nombre de alumnos aplazados de una materia dada

4. Calificación de un alumno en todas las materias en las que está matriculado

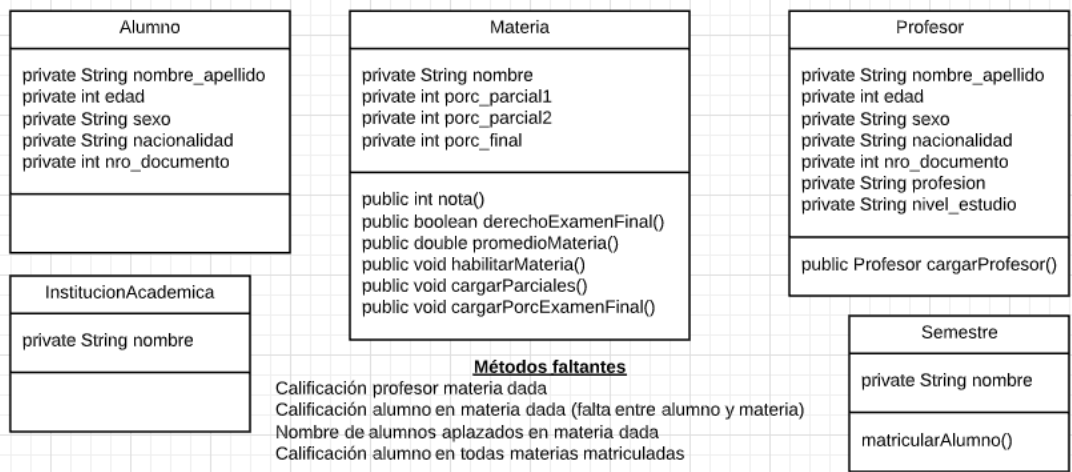
¿Los métodos de los reportes en qué clase van?

3. Implementar métodos para validaciones

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

Métodos identificados hasta el momento



3. Identificando las relaciones [acciones] (las posibles relaciones están marcados en azul)

Observación: las relaciones indican que desde una clase se necesita acceder a los miembros de otra clase.

Escribir un programa que gestione el rendimiento de alumnos y profesores de una institución académica. Los alumnos tienen la posibilidad de matricularse en varias materias en cada semestre. Por cada materia, los alumnos tienen dos exámenes parciales y un examen final. El alumno tiene derecho a rendir examen final si logra un promedio mayor a 50% en los exámenes parciales. La nota final del alumno se obtiene aplicando la siguiente fórmula:

$$\text{Nota final} = 0.5 * \text{promedio_parciales} + 0.5 * \text{porcentaje_final}$$

Todos los exámenes se basan en una escala de 100.

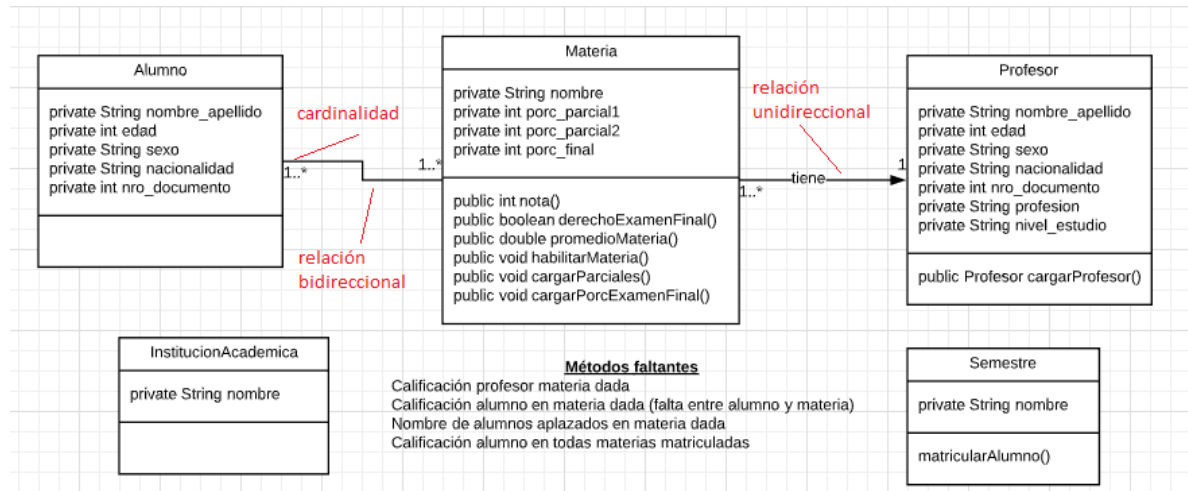
Además, para calificar al profesor, el programa deberá ofrecer una funcionalidad para el cálculo del promedio de la materia. El rendimiento del profesor se obtiene a partir del promedio de la materia.

Cada materia tiene una cantidad límite de alumnos que debe ser validada al momento de que el administrador matricule a los alumnos.

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

Relaciones identificadas hasta el momento



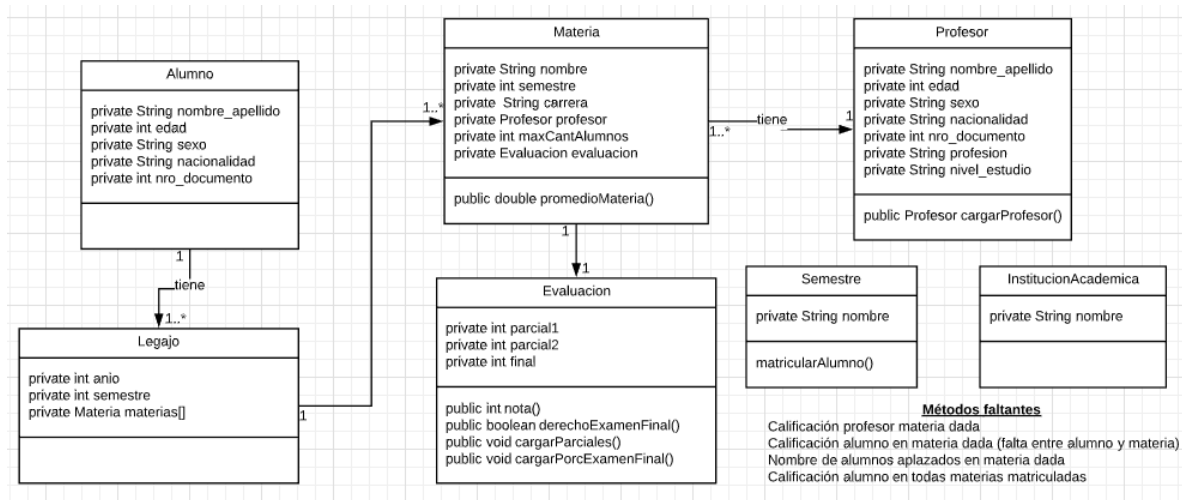
- a) Un alumno puede cursar materias. Una materia puede ser cursada por varios alumnos.
 - a. ¿Un vector de alumnos se tiene en la clase Materia o un vector de materias se tiene en la clase Alumno?
- b) Las materias tienen dos exámenes parciales y un final. El alumno rinde los exámenes
 - a. ¿Examen es una clase?
- c) El promedio de notas de la materia representa la calificación del profesor.
 - a. Desde profesor se necesita acceder a materia para obtener el promedio. Se requiere las notas de todos los alumnos para dicha materia.
- d) Cada materia tiene un profesor.
 - a. ¿El profesor forma parte de la institución académica?
 - b. ¿Cómo obtengo la calificación del profesor en todas las materias que enseña?
 - c. ¿Cómo obtengo el listado de todos los profesores?
 - d. ¿Cómo obtengo el profesor con mejor calificación?
- e) Cada materia tiene una cantidad máxima de alumnos.
- f) Acomodar los métodos faltantes
 - a. Calificación profesor materia dada
 - i. El método puede ir en Materia. El promedio de la materia será la calificación del profesor de la materia.
 - b. Calificación alumno en materia dada
 - i. El método debe ir en Alumno. Cada alumno tiene sus materias.
 - c. Nombre de alumnos aplazados en materia dada.
 - i. El método debe ir en Materia. Se recorrerá el vector de alumnos de la materia.
 - d. Calificación alumno en todas la materias matriculadas
 - i. El método debe ir en Alumno. Se recorre el listado de materias y se obtiene la calificación en cada una de ellas.

PROGRAMACIÓN EN JAVA

PROF. ING. GUIDO ACOSTA

¿Cómo Materia tendrá los atributos parcial1, parcial 2 y final y a la vez tiene el vector de alumnos? Eso significa que si Materia está en alumnos, el vector de alumnos también estará en todos los alumnos? ¿Qué pasa si el alumno repite la materia? ¿Cómo se representa esto?

Relaciones identificadas hasta el momento



¿El método **promedioMateria** debe estar en la clase Materia?

¿Podemos crear una clase Seccion?

¿Existe alguna relación de herencia?

4. Escribir las relaciones en Java