



Lenguaje Unificado de Modelado - UML

Autor(es):

Gustavo Eduardo Gil Prado
Docente TC

Resumen:

El objetivo de esta presentación es presentar los conceptos del Lenguaje Unificado de Modelado – UML, como herramienta necesaria en el curso de Programación Orientada a Objetos

FACULTAD DE INGENIERÍA

Ingeniería Informática
Institución Universitaria Colegio Mayor del Cauca



Guía de Curso – 30/05/2022

Descripción:

Esta presentación tienen la intención de servir de apoyo al desarrollo del curso de Programación Orientada a Objetos

Palabras clave:

Algoritmo, Clase, Instancia, Aplicación, Atributo, Método

Referencie este documento así: Gil, G.E. (2023). Lenguaje Unificado de Modelado - UML [Guía de Curso]. Institución Universitaria Colegio Mayor del Cauca.

Programación Orientada a Objetos

Lenguaje Unificado de Modelado - UML

Docente: Gustavo Eduardo Gil Prado

Introducción

<https://www.uml.org>



[Esta foto de Autor desconocido está bajo licencia CC BY-NC-ND](#)

Tipos de diagramas en UML

Dentro de una posible clasificación, encontramos dos (2) tipos principales de diagramas, hasta la versión 2.5 de UML.

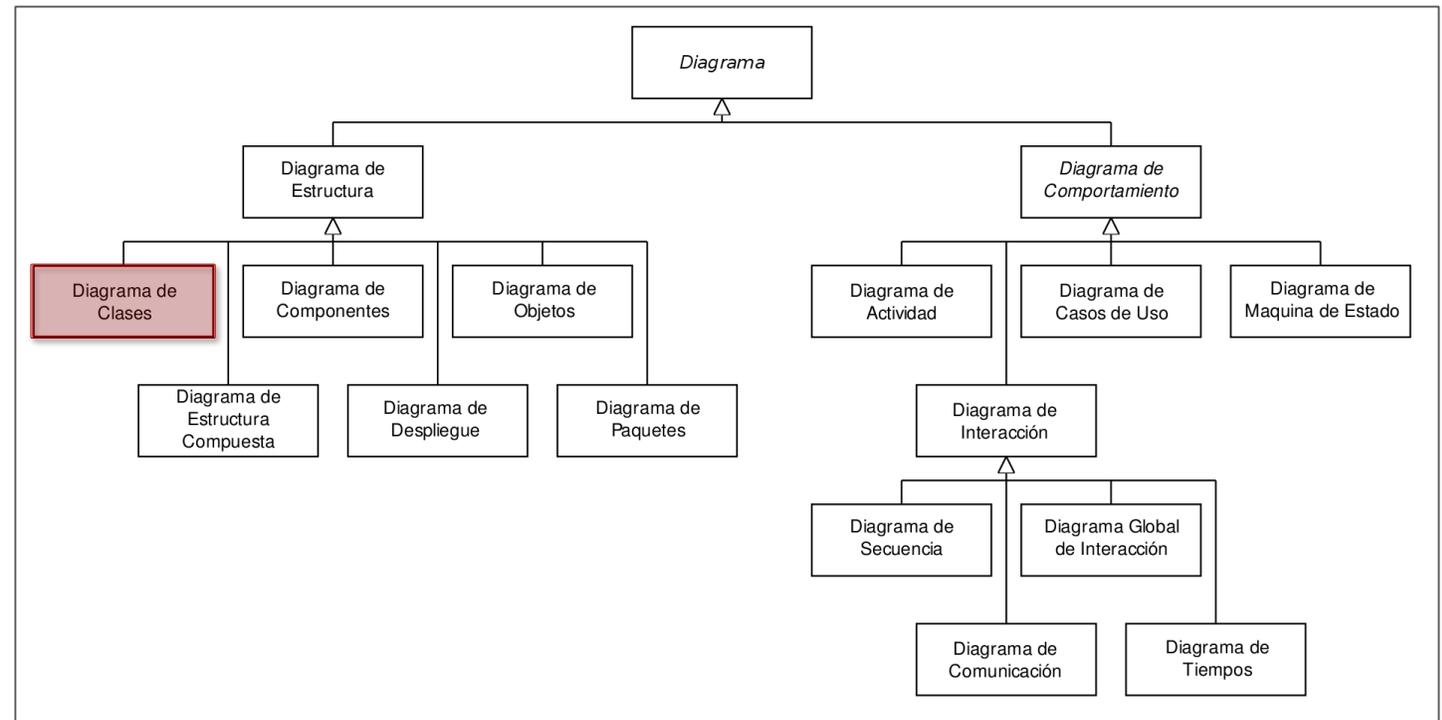
Diagramas Estructurales:

Estos representan la estructura estática del sistema y sus partes, según el nivel de abstracción

Diagramas de comportamiento:

Mediante los que se busca representar el comportamiento del sistema y sus componentes (objetos)

- Los diagramas de interacción, según UML son todos aquellos derivados del diagrama de comportamiento más general, los que se pueden considerar una subcategoría.



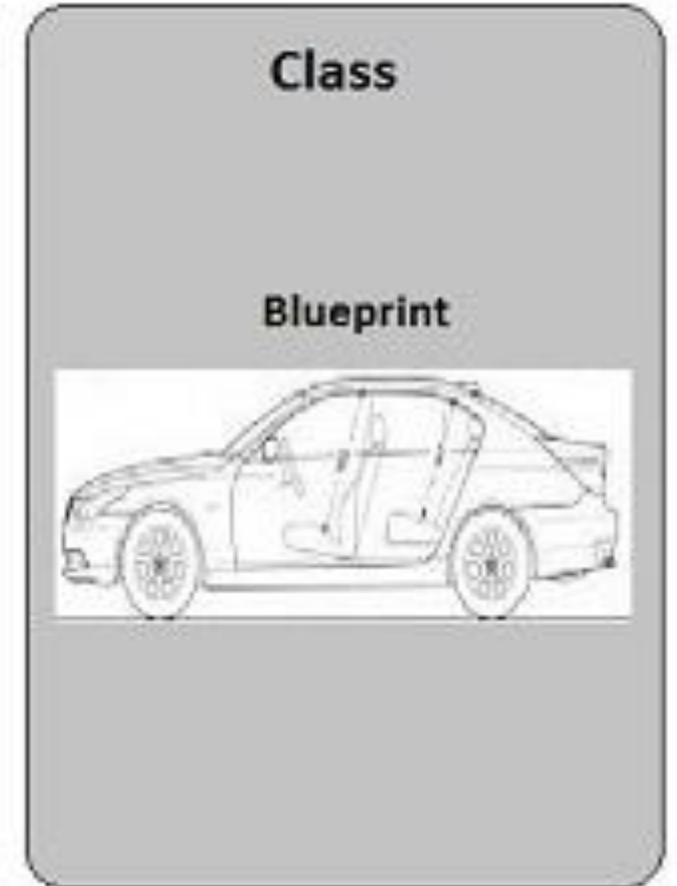
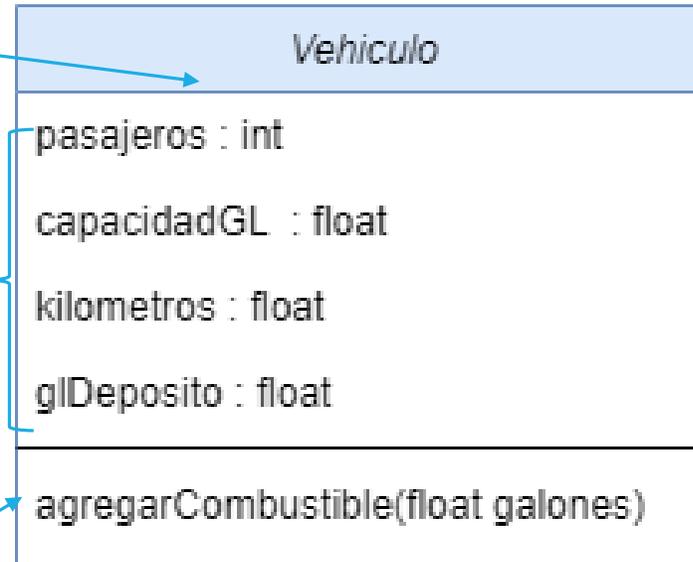
De Uml_diagram.svg: Dave A Ryanderivative work: Wazaraki (talk) - Uml_diagram.svg, Dominio público, <https://commons.wikimedia.org/w/index.php?curid=6777523>

Clase

Nombre de la clase. Esta sección siempre es necesaria

Sección para los atributos de la clase, estos son los descriptores de las instancias (Objetos) de la clase

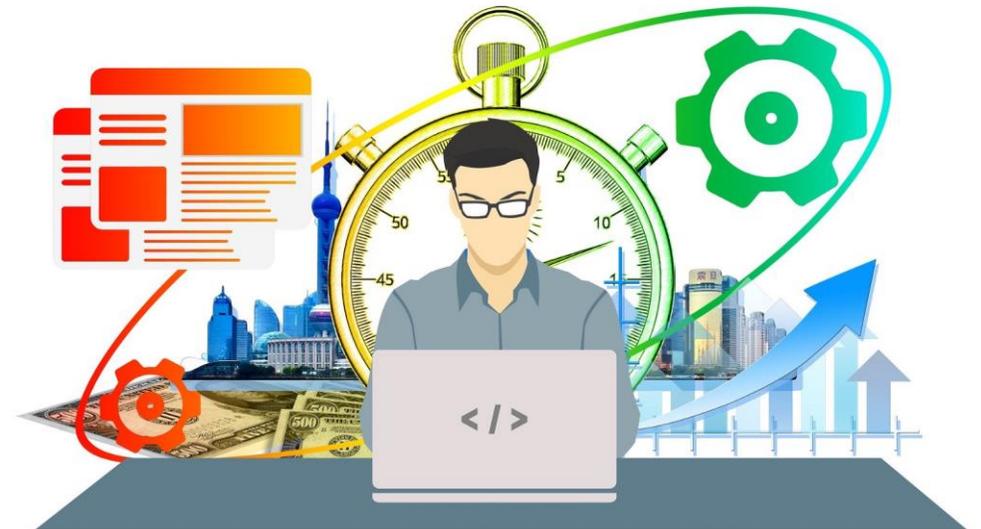
Métodos de la clase.
Se deben listar cada operación por línea.
Las operaciones describen cómo una clase puede interactuar con los datos.



Modificadores de acceso de los miembros

Para especificar la visibilidad de un miembro de la clase se coloca uno de los siguientes signos delante de ese miembro:

- Público (+)
- Privado (-)
- Protegido (#)
- Paquete (~)
- Derivado (/):
Este se puede combinar con otros.
- Estático (subrayado)



Esta foto de Autor desconocido está bajo licencia [CC BY](#)

Ejemplo modificadores de acceso de los miembros

Público (+)

Privado (-)

Protegido (#)

Derivado (/)

<i>Persona</i>
- identificacion
+ nombre
+ telefono
- fecha_nacimiento
/ edad
correo_electronico
+ comprar_tiket_parqueadero ()
+ setIdentificacion ()
+ setEdad ()

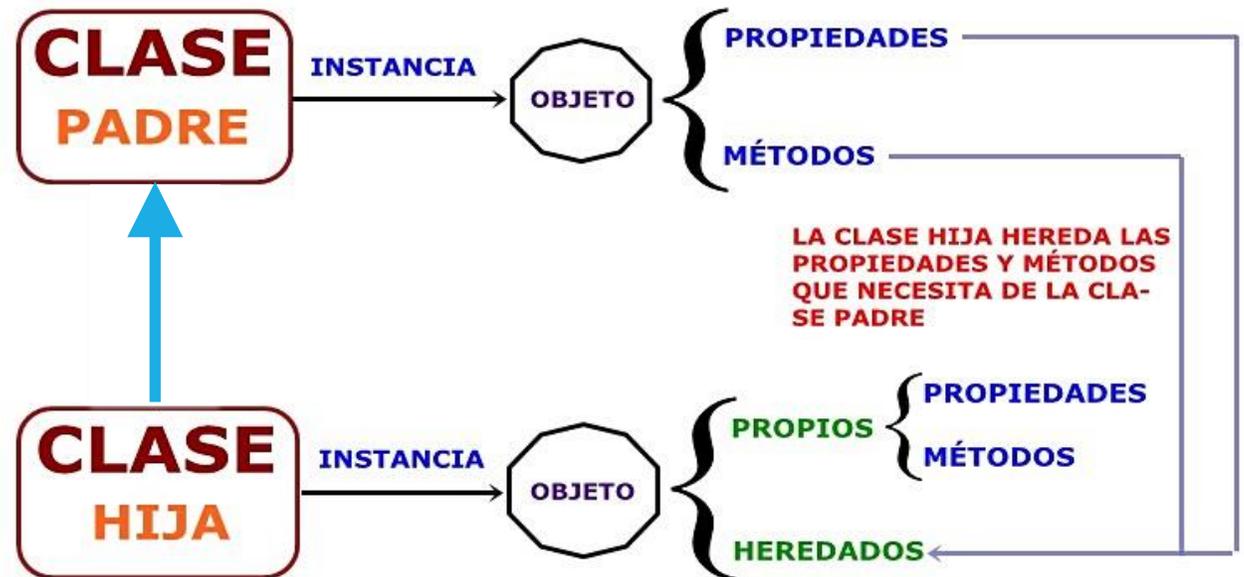
Relaciones

En general las relaciones expresan las conexiones lógicas que tienen los objetos y sus clases, como resultado del proceso de abstracción. Dentro de UML podemos encontrar:

- Herencia
- Enlace
- Asociación
- Agregación
- Composición

Relación - Herencia

Es la capacidad que tienen la clase mediante la cual se relaciona con otras clases, heredando atributos y operaciones



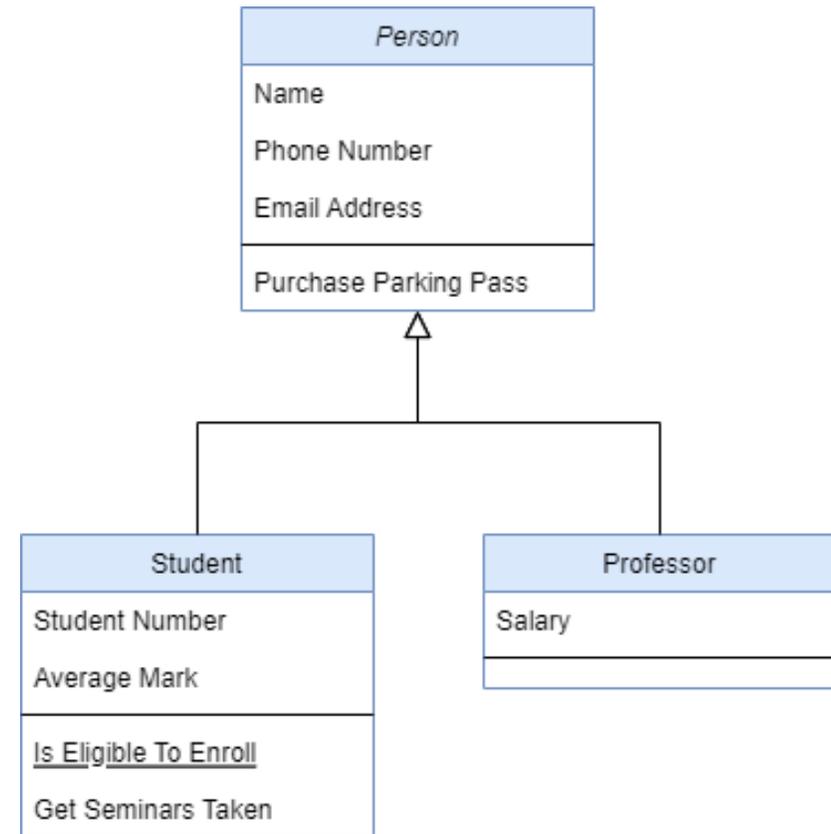
Esta foto de Autor desconocido está bajo licencia [CC BY-NC](#)

Relación - Herencia

Ejemplo de herencia.

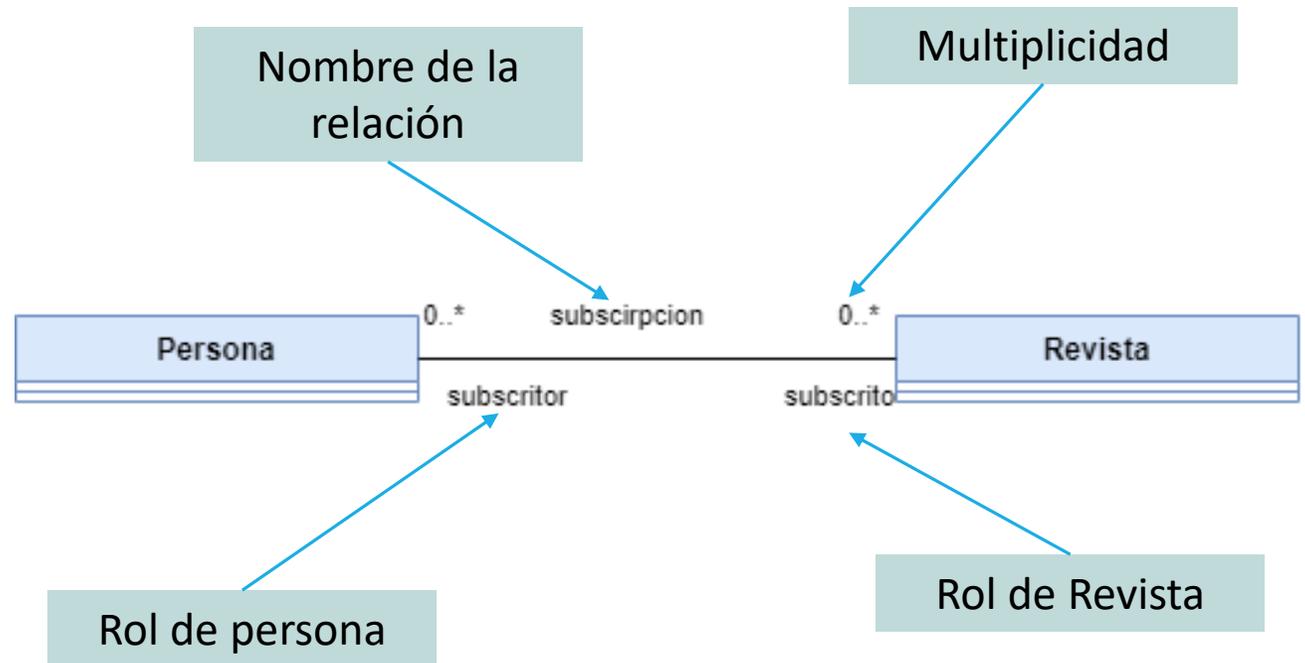
Es también conocido como generalización o especialización, según se de el caso.

Su símbolo es una flecha (generalmente recta), con punta de flecha serrada.



Relación – Asociación

Esta forma de relación se puede presentar entre cualquier número de clases y se puede presentar de manera bidireccional (como en el caso del ejemplo) o direccionada, en caso de que se le agrega la punta de flecha abierta \rightarrow , este tipo de relación puede tener opcionalmente un nombre para la relación y también indicar los roles que desempeñan sus participantes. También es adecuado indicar la multiplicidad según sea el caso.



Relación – Asociación – Multiplicidad

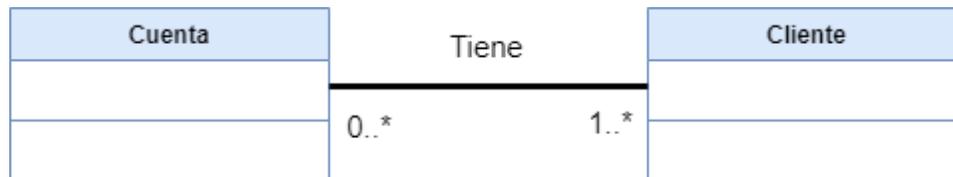
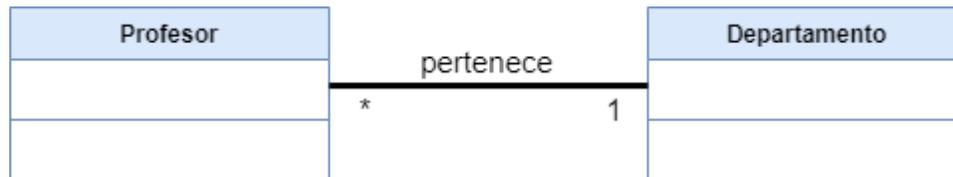
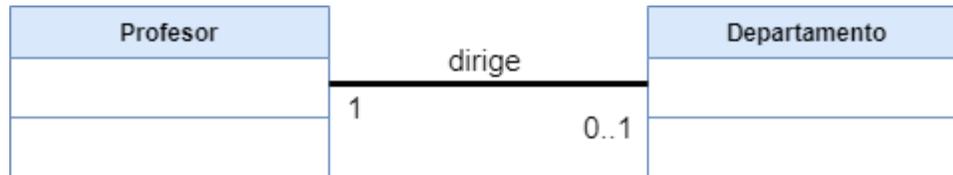
Multiplicidad	Significado
1	Uno y solo uno
0..1	Cero o uno
N..M	De N hasta M
*	Cero o muchos
0..*	Cero o muchos
1..*	Uno o muchos

Multiplicidad:

Esta determina cuantos objetos del tipo indicado participan de la relación. La cual se indica en cada lado de la asociación, especificándola de la siguiente manera.

Cuando el primer valor establecido es 0, significa que la relación es opcional, en cambio si es 1 o mas indica que esta es obligatoria.

Ej: Relación – Asociación – Multiplicidad

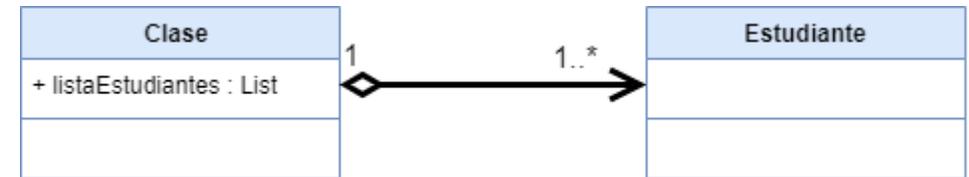


- Un profesor puede dirigir (opcional) un departamento – Un departamento lo dirige (obligatorio) un profesor
- Un profesor pertenece a un departamento (obligatorio), a un departamento pertenecen muchos docentes (opcional)
- Un cliente puede tener cero o múltiples cuentas – una cuenta tiene un mínimo 1 o muchos clientes relacionados.

Relación – Agregación

Su representación grafica, en UML es mediante un rombo hueco junto a la clase contenedora.

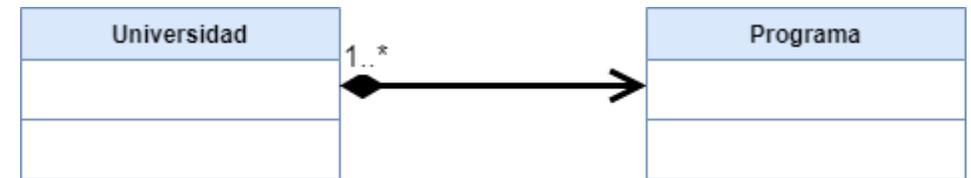
Esta relación es un tipo de asociación, pero mas fuerte ya que una de las clases contiene a la otra, pero la clase contenida, no depende su existencia de la relación. En el ejemplo es claro que el estudiante existe aun cuando no este en una clase.



Relación – Composición

La representación grafica en UML es un rombo relleno en el lado de la clase contenedora.

Este tipo de relación es mucho mas fuerte, ya que se trata a la entidad contenedora, como un todo y la contenida son sus partes. Por lo que existe una dependencia de existencia.



Ejemplo:

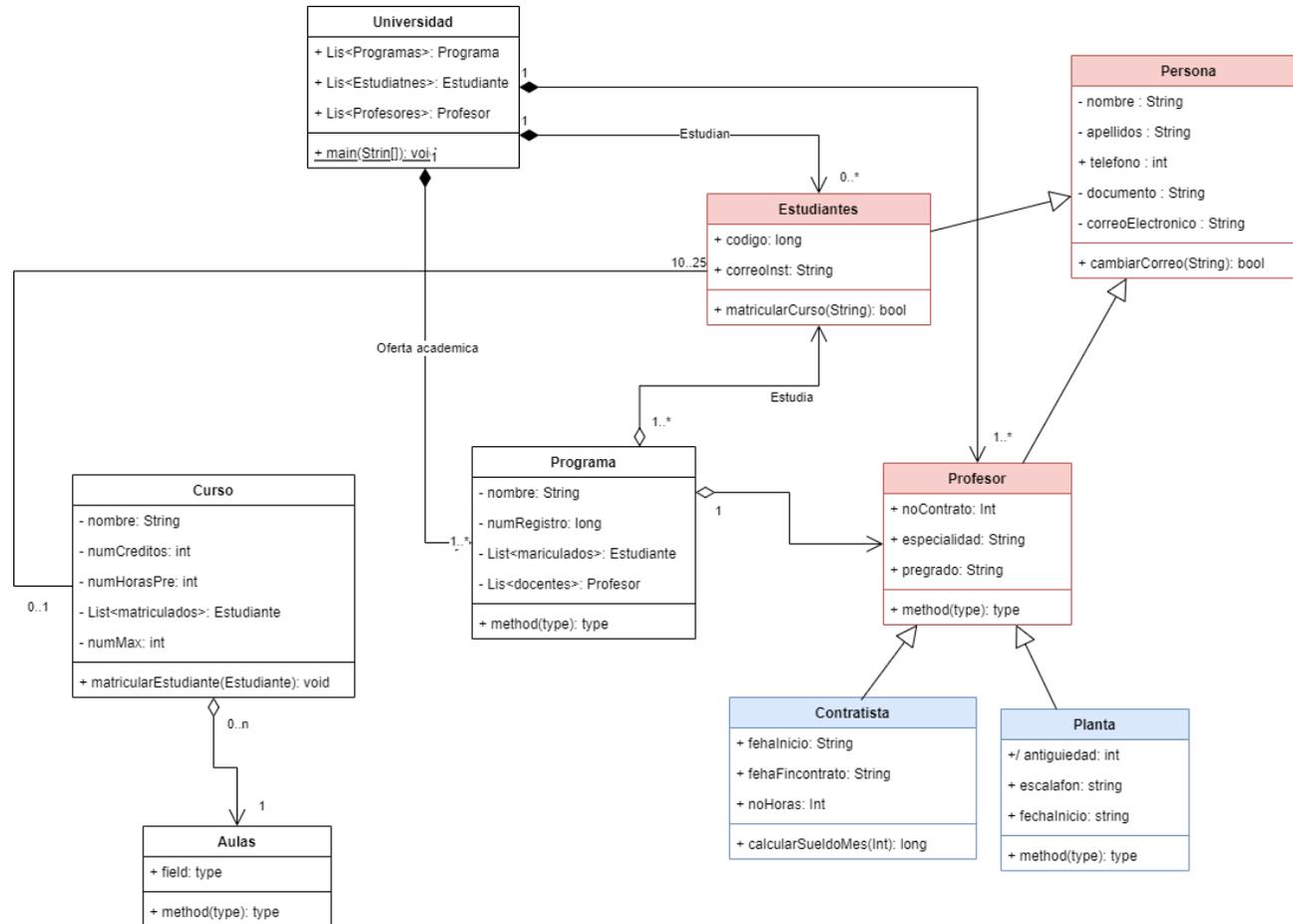
- La universidad esta compuesta de programas
- Los programas dependen de la existencia de la universidad.

Ejemplo

Considere el siguiente ejercicio:

Realice el modelado de los elementos principales de una universidad, dentro de lo que es adecuado que analice la relación que existe entre los programas, estudiantes, profesores, cursos y aulas.

Resultado del ejercicio



Ejercicio propuesto

Enunciado: Se requiere un programa que modele varias figuras geométricas: el círculo, el rectángulo, el cuadrado y el triángulo rectángulo. Estas figuras tiene en común que todas tienen un área y un perímetro

- El círculo tiene como atributo su radio en centímetros.
- El rectángulo, su base y altura en centímetros.
- El cuadrado, la longitud de sus lados en centímetros.
- El triángulo, su base y altura en centímetros.

Se requieren métodos para determinar el área y el perímetro de cada figura geométrica. Además, para el triángulo rectángulo se requiere:

- Un método que calcule la hipotenusa.

Se debe desarrollar una clase que permita almacenar las figuras y con un método principal para crear las cuatro figuras y probar los métodos respectivos.