

Guía 1

Programación orientada por objetos

OscarEdgar Vargas Neira

Facultad de Ingeniería

Desarrollo de software

Tutor: Dilsa Enith Triana Martínez

19 de Agosto 2024

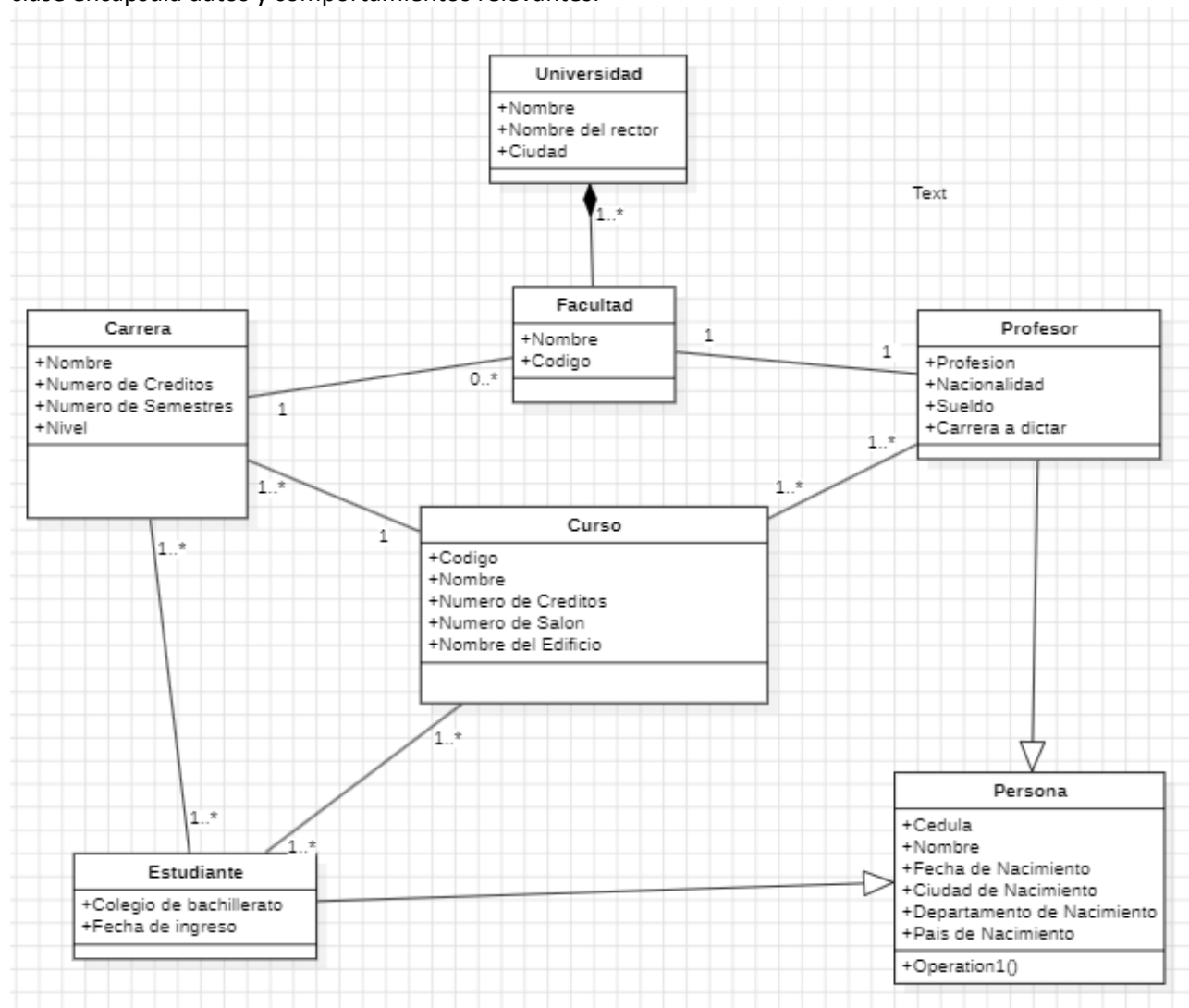
Repositorio Guia 1:

<https://github.com/GOLDENFLOWW/GUIA-1--DdS>

Informe del Diagrama de Clases en Java (Actividad 1)

Descripción General:

El diagrama de clases presentado modela un sistema de información para una universidad, representando las entidades principales y sus relaciones. Se utiliza una aproximación orientada a objetos, donde cada clase encapsula datos y comportamientos relevantes.



Clases Principales y sus Atributos:

- **Universidad:** Contiene información general de la institución como nombre, nombre del rector y ciudad.
- **Facultad:** Representa una facultad dentro de la universidad, con atributos como nombre y código. Incluye métodos para gestionar profesores (insertar, modificar, eliminar).

- **Carrera:** Describe una carrera ofrecida por la universidad, incluyendo nombre, número de créditos, número de semestres y nivel.
- **Curso:** Detalla un curso específico, con atributos como código, nombre, número de créditos, número de salón y nombre del edificio donde se imparte.
- **Profesor:** Representa a un profesor, con atributos como nombre, profesión, nacionalidad, sueldo y la carrera que dicta.
- **Persona:** Clase base para representar información personal, con atributos como cédula, nombre, fecha y lugar de nacimiento.
- **Estudiante:** Clase derivada de Persona, específica para estudiantes. Incluye atributos adicionales como colegio de bachillerato y fecha de ingreso, además de un método Matricular ().

Relaciones entre Clases:

- **Universidad - Facultad:** Relación de agregación (1 a muchos). Una universidad tiene varias facultades.
- **Facultad - Carrera:** Relación de agregación (1 a muchos). Una facultad ofrece varias carreras.
- **Facultad - Profesor:** Relación de agregación (1 a muchos). Una facultad tiene varios profesores.
- **Profesor - Curso:** Relación de asociación (1 a muchos). Un profesor puede dictar varios cursos.
- **Carrera - Curso:** Relación de asociación (1 a muchos). Una carrera tiene varios cursos asociados.
- **Persona - Estudiante:** Relación de herencia. Un estudiante es una persona con atributos adicionales.
- **Estudiante - Curso:** Relación de asociación (muchos a muchos) a través del método Matricular(). Un estudiante puede matricularse en varios cursos y un curso puede tener varios estudiantes matriculados.

Conclusión:

El diagrama proporciona una visión general de la estructura de un sistema de información universitario, pero requeriría detalles adicionales para su implementación completa. Es una base útil para el diseño de una aplicación Java que gestione información sobre estudiantes, profesores, carreras y cursos.