

Informe de Programación en Java

Tema: Encapsulamiento, Relaciones entre Clases y Validaciones en Setters

Alumno: Cristian Emanuel Cáceres

Materia: Programación Orientada a Objetos

Fecha: Septiembre 2025

1. Encapsulamiento

En la solución se aplicó encapsulamiento declarando los atributos de cada clase como privados (private). De esta forma, el acceso directo a los datos internos queda restringido y solo es posible interactuar con ellos mediante métodos públicos (getters y setters). Esto garantiza que los valores no puedan ser modificados de manera incontrolada y que cada cambio pase por reglas definidas en los métodos de acceso.

2. Relaciones entre clases

Las relaciones entre las clases se implementaron de la siguiente forma: - Estudiante – Materia: un estudiante puede estar inscrito en varias materias, utilizando un ArrayList. Esto refleja una relación de composición uno a muchos. - Estudiante – Carrera: cada estudiante está asociado a una carrera, utilizando un atributo de tipo Carrera. Esto refleja una relación de asociación uno a uno. - App: clase principal utilizada para crear instancias y probar la interacción entre estudiantes, materias y carreras.

3. Validaciones en métodos setter

Se agregaron validaciones en los métodos setter para asegurar la consistencia de los datos: - En el atributo edad de Estudiante, se valida que sea mayor a 16. - En atributos como nombre o apellido, se valida que no sean nulos ni vacíos. - En Materia, se valida que el nombre no sea vacío y que los créditos sean mayores a cero. - En Carrera, se valida que el nombre no sea nulo o vacío. De esta forma, se evitan datos inválidos y se mantiene la integridad de los objetos creados.

Conclusión

La solución desarrollada demuestra la correcta aplicación de los principios de la programación orientada a objetos. El uso de encapsulamiento protege los atributos, las relaciones entre clases modelan adecuadamente el dominio del problema, y las validaciones en los setters garantizan la integridad de los datos.