

Informe de Programación

Temas

- Encapsulamiento
 - Relaciones entre clases y validaciones en setters
-

1. Encapsulamiento

En el proyecto, se aplicó encapsulamiento declarando los atributos de cada clase como privados. Esto impide el acceso directo a los datos internos, permitiendo modificarlos únicamente a través de métodos públicos (getters y setters). Así se asegura que los valores sean consistentes y que cualquier cambio cumpla con reglas definidas.

2. Relaciones entre clases

Las clases se relacionan de manera que un estudiante puede tener varias materias (composición uno a muchos) y cada estudiante pertenece a una carrera (asociación uno a uno). La clase `App` permite crear y probar instancias, mostrando cómo interactúan estudiantes, materias y carreras.

3. Validaciones en métodos setter

Se agregaron validaciones en métodos setter para asegurar la consistencia de los datos (edad, nombre, crédito, etc.). Por ejemplo, en Edad se valida que sea mayor a 16; en Nombre no debe ser nulo ni vacío; en Apellidos y Créditos se imponen reglas para evitar datos inválidos. La conclusión destaca que el uso de

encapsulación protege atributos y mantiene la integridad de los objetos, validando por setter.

Conclusión

La solución demuestra una correcta aplicación de la programación orientada a objetos.

El encapsulamiento protege los datos y asegura su integridad, mientras que las relaciones entre clases reflejan fielmente la estructura del problema. De esta manera, se facilita la gestión de estudiantes, materias y carreras, manteniendo la información organizada y confiable.

Alumna: Ruiz Gabriela
Materia: Programación Orientada a Objetos
Fecha: 15/09/2025