

Facultad de Ingeniería - Universidad Nacional de Cuyo			
Asignatura:	Programación II		
Carrera:	Licenciatura en Ciencias de la Computación		
Año: 2024	Semestre: 2	Trabajo Práctico n° 2	

Trabajo Práctico n° 2: Implementación de Clases en Java a partir de Diseños en UML

Objetivos:

- Implementar en código Java los diseños de clases realizados en UML.
- Desarrollar habilidades para traducir diseños de clases en UML.
- Consolidar los conocimientos de abstracción, encapsulamiento, y relaciones entre clases.
- Familiarizarse con la sintaxis y estructura de un proyecto en Java orientado a objetos.

Metodología

- Trabajo individual.
- Tiempo de realización estimado: 2 semanas.

Contenido

1. Implementación en Java:

- **Clases y Objetos:**
Implementar en Java las clases diseñadas en el diagrama UML.
- **Atributos y Métodos:**
Definir los atributos y métodos de cada clase según lo modelado en UML.
- **Encapsulamiento:**
Aplicar los modificadores de acceso (private, public, protected) según lo definido en el diseño UML.
- **Getters y Setters:**
Implementar métodos getters y setters para cada atributo privado de las clases.
- **Relaciones entre Clases:**
Implementar relaciones de asociación, herencia, agregación y composición según lo indicado en el diagrama UML.

2. Validación y Testeo:

- Crear una clase principal (Main) donde se instancian los objetos y se prueben las interacciones entre ellos.
- Incluir métodos de prueba que validan el correcto funcionamiento de los getters y setters.

3. Documentación:

- Añadir comentarios en el código explicando la funcionalidad de los métodos y las decisiones de diseño tomadas.

Actividades:

1. Implementación de Clases en Java:

- A partir del diagrama UML creado en el trabajo práctico anterior, implementar las clases, atributos, y métodos en Java. El sistema debe contar con un mínimo de 8 a 10 clases.
- Implementar las relaciones entre clases según lo definido en UML: herencia, agregación, composición, y asociación.

2. Creación de Getters y Setters:

- Para cada atributo privado en las clases, crear los métodos `get` y `set` correspondientes.

3. Desarrollo de una Aplicación Principal:

- Crear una clase main que sirva como punto de entrada al programa.
- Instanciar y utilizar los objetos para demostrar el funcionamiento del sistema, incluyendo el uso de getters y setters.
- Probar los métodos y relaciones definidos.

4. Testeo y Validación:

- Asegurarse de que el código funcione correctamente y cumpla con los requisitos establecidos en el diagrama UML.
- Identificar y corregir posibles errores en la implementación.