

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

Trabajo Práctico integrador

Objetivos:

- Aplicar los principios fundamentales de POO como encapsulamiento, abstracción, herencia y polimorfismo, en el diseño y desarrollo de un sistema.
- Utilizar encapsulamiento y modulación para organizar el código de manera eficiente, gestionando adecuadamente los modificadores de acceso.
- Implementar herencia y polimorfismo para la reutilización de código y la creación de relaciones entre clases. Emplear composición y agregación para representar relaciones entre objetos.
- Implementar una base de datos SQLite para gestionar y persistir datos, utilizando operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre las tablas. Implementar iteraciones sobre las listas de objetos utilizando estructuras como for, while, y el uso de Iterator para recorrer y manipular los elementos de las listas.
- Permitir el ingreso de información por parte del usuario mediante la consola, capturando y validando los datos de entrada.
- Integrar el manejo de excepciones personalizadas y utilizar excepciones para el control de errores.
- Incorporar principios de diseño como los principios SOLID, el principio DRY, y patrones de diseño.

Metodología

- Trabajo grupal. 5 estudiantes por equipo.
- Tiempo de realización estimado: 5 semanas.

Cronograma de actividades

27/09/2024:

- Conformación de grupos.
- Selección y aprobación de tema de desarrollar
- Planteo y comienza el desarrollo del proyecto

04/10/2024:

- Desarrollo del proyecto

18/10/2024:

- Desarrollo del proyecto y primera revisión

25/10/2024:

- Desarrollo del proyecto

01/11/2024:

- Desarrollo del proyecto y revisión

08/11/2024:

- Revisión de proyectos

15/11/2024:

- Coloquio final y cierre.

Requisitos del Proyecto:

- El sistema deberá implementar al menos tres niveles de herencia que permitan demostrar el uso de clases abstractas y concretas, junto con la sobrecarga de métodos.
- Deberá conectarse a una base de datos SQLite para almacenar y gestionar información, permitiendo a los usuarios:
 - Agregar nuevos registros a las tablas de la base de datos.
 - Modificar los registros existentes.
 - Eliminar registros de las tablas.
 - Buscar y filtrar registros utilizando consultas SQL.
- El sistema debe permitir el ingreso de información por consola, validando la entrada del usuario y manejando excepciones para gestionar entradas incorrectas.
- El uso de excepciones personalizadas será necesario para controlar errores como intentos de acceso a posiciones inválidas dentro de la lista o entradas no válidas por parte del usuario.
- Los grupos deberán demostrar el uso de encapsulamiento, gestionando los atributos con modificadores de acceso adecuados y proporcionando getters y setters.
- El sistema deberá incluir interfaces que implementen polimorfismo y permitan extender la funcionalidad de las clases mediante la implementación de diferentes comportamientos.
- Deben utilizarse composición y agregación para representar relaciones entre clases, con al menos un ejemplo de cada tipo.

Evaluación:

- La presentación de este trabajo será en forma oral donde realizarán una demostración del sistema funcionando. Además, deberán explicar el diseño del sistema, justificando las decisiones tomadas en cuanto a la organización de clases, herencia, encapsulamiento, polimorfismo, y manejo de excepciones.