



Universidad Nacional de Villa Mercedes

ESCUELA DE INGENIERÍA Y CIENCIAS AMBIENTALES

Ejercicio N°4 – Aplicando los aprendido 3

Carrera: Ingeniería en Sistemas de Información.

Materia: Paradigmas de Programación.

Profesor/res: Walter Molina, Diego Puertas.

Alumno: Germán Adrián Muñoz.

Fecha de entrega: 16/10/2023

Ejercicio N°4

Explica en un texto, con ejemplos y fundamentación qué características de la OOP utilizaste para resolver los programas de los Ejercicios 2 y 3. Si hay alguna que no utilizaste o no implementaste, indica cuál y por qué crees que no fue necesario.

- Calculadora:
 - **Clases y Objetos:** se utiliza la orientación a objetos mediante una clase llamada Calculadora. Esta clase tiene métodos que representan operaciones matemáticas como suma, resta, multiplicación y división. Cada método encapsula una operación específica.
 - **Encapsulación:** los métodos de la clase Calculadora encapsulan la lógica de las operaciones matemáticas. Los detalles de cómo se realizan las operaciones están ocultos al usuario y encapsulados en la clase. El usuario solo interactúa con los métodos públicos para realizar cálculos.
 - **Abstracción:** La clase Calculadora proporciona una interfaz abstracta para realizar operaciones matemáticas. Los detalles de cómo se implementan las operaciones se mantienen ocultos al usuario.
 - **Herencia:** en JavaScript, todos los objetos heredan métodos y propiedades del objeto Object. El objeto Object es el objeto raíz en JavaScript, y todos los demás objetos se derivan de él.
- Aplicación de Tareas:
 - **Clases y Objetos:** En el programa de gestión de tareas, se utilizan las clases Task y TaskList para modelar tareas individuales y una lista de tareas. La clase Task contiene propiedades y métodos para gestionar información sobre una tarea, y la clase TaskList gestiona la lista de tareas.
 - **Encapsulación:** La clase Task encapsula los datos de una tarea, como el título, la descripción y el estado. Además, la clase TaskList encapsula la lógica relacionada con la gestión de la lista de tareas.

- **Abstracción:** La clase Task representa una abstracción de una tarea, con propiedades que describen sus atributos. La clase TaskList abstracta la lista de tareas y proporciona métodos para interactuar con ella.
- **Herencia:** las instancias de Task y TaskList heredan propiedades y métodos comunes a través de sus prototipos. Esto es parte del sistema de prototipos de JavaScript, donde los objetos pueden heredar de otros objetos a través del mecanismo de __proto__.
- Características no utilizadas:
 - **Polimorfismo:** no se utilizó el polimorfismo en estos programas. No se necesitaba que objetos de diferentes clases compartieran un método con el mismo nombre, pero con diferentes implementaciones.