

DISEÑO DE SOFTWARE

**ACTIVIDAD 3 – GUIA 1.**

**RESOLUCION DE PROBLEMAS USANDO HERENCIA**

JOAN SEBASTIAN GUERRA ARANGO

DILSA TRIANA MARTINEZ

2024

Contenido

Objetivos …………………………………………………………………………………………

Marco Teórico ……………………………………………………………………………………

Diagrama de clase UML ………………………………………………………………………..

Link Repositorio …………………………………………………………………………………

Conclusiones …………………………………………………………………………………….

Bibliografía ……………………………………………………………………………………….

# Objetivos:

* Proporcionar una representación gráfica de los elementos principales de un sistema y sus relaciones en este caso un Parqueadero.
* Aplicar los principios de herencia y polimorfismo de la programación orientada a Objetos.
* Crear un diagramas de clase UML para el caso práctico.
* Almacenar el código fuente en un repositorio (GitHub, GitLab o BitBucket).

# Marco Teórico:

La herencia es un mecanismo fundamental en la programación orientada a objetos (POO) que permite que una clase (subclase) herede atributos y métodos de otra clase (superclase).

Dentro de las ventajas que tenemos al aplicar la herencia y el polimorfismo del paradigma orientado a objetos son:

Permite la reutilización de código

Facilita la extensibilidad.

Mejora la organización del código en una jerarquía de clases

Simplifica el mantenimiento

# Diagrama de clase UML:

Diagrama de clases UML

Ejercicio de modelado Parqueadero



*Elaboración propia*

# Decisiones:

Para resolver los ejercicios que hemos abordado, se han tomado varias decisiones clave en el diseño y la implementación.

1. Crear funciones específicas para tareas concretas (como calcular, convertir fechas) permite una mejor organización.
2. Implementar una función que use una Lista(arrayLIst) para guardar los Vehiculos.
3. Scanner es una herramienta estándar en Java para la entrada de datos, y permite una interacción fácil con el usuario para introducir datos en tiempo real.
4. Asegurar que las entradas estén dentro de los rangos válidos y manejar errores como divisiones por cero o entradas no válidas.
5. Utilizar estructuras if y switch para controlar la lógica del programa.

# Link Repositorio:

https://github.com/jsguerra07/DDS\_Guia1.git

# Conclusiones:

La herencia y el polimorfismo son conceptos fundamentales en Java que permiten a los desarrolladores escribir código más modular, reutilizable y mantenible. La herencia organiza el código en jerarquías de clases, mientras que el polimorfismo permite a un solo método operar en diferentes formas de datos, promoviendo la flexibilidad y extensibilidad del software.

# Bibliografía

Villalobos, J. & Casallas R. (2006). *Fundamentos de Programación*. Editorial Pearson, Prentice Hall.