

M12 Programación II FIN A

ACTIVIDAD 1

| **Tutor:** | **Karla Ramos Garcia** |
| --- | --- |
| **Estudiante:** | **José Ramón Ibáñez Posadas** |
| **Matricula:** | **BNL098377** |

| Monterrey, Nuevo León | Jueves, 28 de Noviembre de 2024 |
| --- | --- |

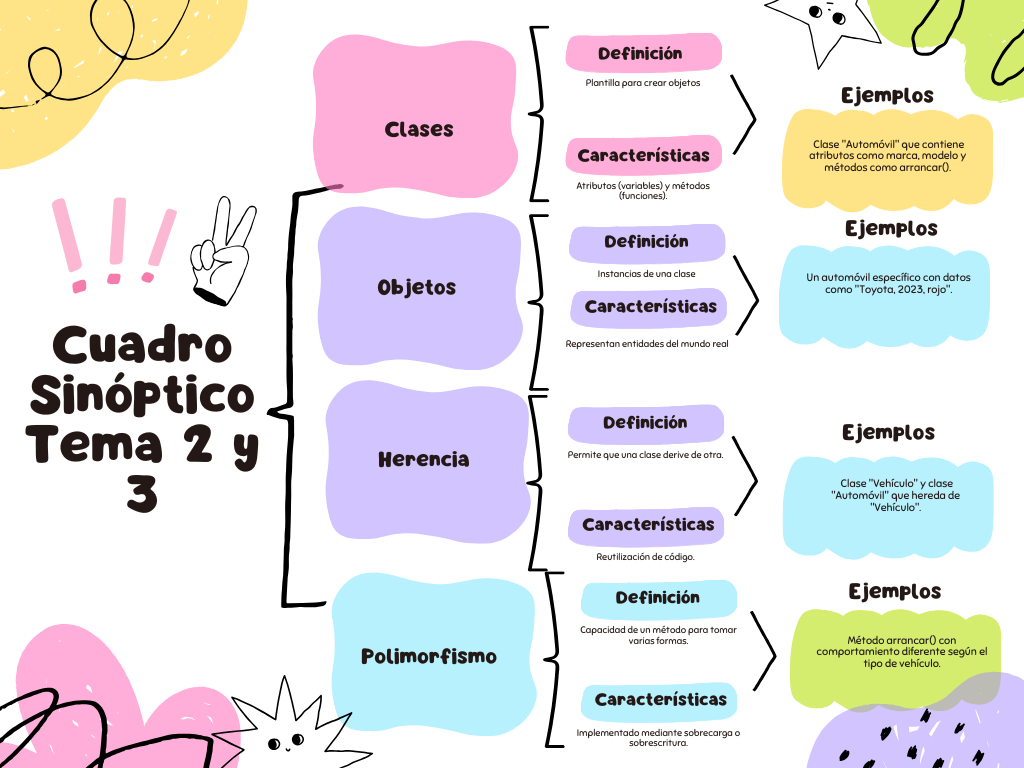
INTRODUCCIÓN

En este proyecto, abordaremos los aspectos fundamentales de la programación orientada a objetos en Java, específicamente en relación con los temas de **Clases, Objetos, Herencia y Polimorfismo**. El propósito es comprender cómo estos conceptos forman la base del desarrollo en Java, permitiendo la creación de soluciones robustas y escalables, como el caso práctico planteado sobre el inventario de automóviles.

Para ello, hemos organizado un cuadro sinóptico que sintetiza los temas 2 y 3 del curso, destacando los elementos clave de cada uno. Además, se identifican y aplican los elementos básicos de Java al caso de estudio propuesto, asegurando una conexión directa entre la teoría y la práctica. Este enfoque busca facilitar el entendimiento y fomentar el aprendizaje significativo a través de ejemplos claros y concretos.

DESARROLLO

CUADRO SINÓPTICO



TABLA

| Aspectos | Respuesta |
| --- | --- |
| **Clases** | Clase "Automóvil". Representa a un vehículo en el inventario digital. |
| **Objetos** | Instancias de la clase Automóvil, como un automóvil específico con sus datos. |
| **Atributos** | Marca, placa, modelo, kilometraje. |
| **Métodos** | Métodos como registrarAutomóvil(), actualizarKilometraje(), mostrarDatos(). |

CONCLUSIÓN

Al finalizar este proyecto, hemos logrado consolidar conocimientos clave sobre la programación orientada a objetos en Java y su aplicación práctica. Entre las ideas más relevantes que emergen de este aprendizaje destacan:

1. **Importancia de las clases y objetos**: Entender que las clases son plantillas esenciales para modelar problemas del mundo real en el ámbito de la programación nos permite diseñar soluciones organizadas y reutilizables. Los objetos, como instancias de estas clases, hacen que las aplicaciones sean dinámicas y versátiles.
2. **Relevancia de la herencia y el polimorfismo**: Estos conceptos no solo promueven la reutilización del código y la reducción de redundancias, sino que también permiten crear aplicaciones más flexibles al adaptar comportamientos según las necesidades específicas de cada caso.
3. **Aplicación práctica**: Relacionar la teoría con casos prácticos, como el inventario digital de automóviles, demuestra cómo estos conceptos no son solo abstractos, sino herramientas esenciales para resolver problemas reales de manera eficiente y escalable.

Este ejercicio refuerza la importancia de comprender los fundamentos de Java, ya que constituyen una base sólida para abordar desafíos más complejos en el desarrollo de software. Además, fomenta una visión integral del diseño y la implementación, fortaleciendo nuestras habilidades tanto en el análisis como en la ejecución.

BIBLIOGRAFÍA

