

TEORIA DIAGRAMA

ELABORADO POR: SERGIO NICOLAS JUYO PAMPLONA

INGENIERIA DE SOFTWARE

PROFESOR

WILLIAM ALEXANDER MATAALLANA PORRAS

UNIVERSIDAD DE CUNDINAMARCA EXTENSION CHIA

PROGRAMA DE INGENIERIA DE SISTEMAS Y COMPUTACION

11 DE FEBRERO DEL 2026

## INTRODUCCION

Los diagramas de clases son uno de los pilares fundamentales del Lenguaje Unificado de Modelado (UML), utilizados ampliamente en la ingeniería de software para representar la estructura estática de un sistema orientado a objetos. Este trabajo presenta una investigación completa sobre la teoría de diagramas de clases, los diferentes tipos de relaciones que existen entre clases, y una implementación práctica en Java que demuestra cada tipo de relación. El objetivo principal es comprender cómo los diagramas de clases sirven como puente entre el análisis y el diseño, permitiendo a los desarrolladores visualizar la arquitectura del sistema antes de la codificación

## TEORIA DE DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama estructural que muestra las clases del sistema, sus atributos, métodos y las relaciones entre ellas.

### Componentes Principales

*Clase:* Representada por un rectángulo dividido en 3 partes (nombre, atributos, métodos)

*Visibilidad:* + (público), - (privado), # (protegido)

### Tipos de Relaciones

relación	símbolo	descripción	Ejemplo
Herencia	→——	Es un tipo de	Perro → Animal
implementación	-- →	implementación	Círculo → Figura

<b>Asociación</b>	_____	relación estructural	Profesor — Universidad
<b>agregación</b>	◇_____	Todo-parte (débil)	Equipo ◇_____ Jugador
<b>composición</b>	◆_____	Todo-parte(fuerte)	Casa ◆_____ Habitación
<b>Dependencia</b>	- - ->	Uso temporal	Cliente - - -> Factura

## CONCLUSION

Los diagramas de clases UML constituyen una herramienta fundamental para el modelado de sistemas orientados a objetos, permitiendo visualizar la estructura estática del software antes de su implementación. Cada tipo de relación (herencia, implementación, asociación, agregación, composición y dependencia) cumple una función específica y se traduce de manera distinta en código Java, siendo la composición más fuerte que la agregación al ligar el ciclo de vida de las partes al todo.

La implementación práctica en Java demuestra que el modelado UML no es un ejercicio teórico, sino que se aplica directamente en el desarrollo de software real. El dominio de los diagramas de clases y sus relaciones mejora significativamente la calidad del diseño, facilita la comunicación entre los equipos de desarrollo y permite construir sistemas más mantenibles, escalables y alineados con los requisitos del negocio.

## Referencias

Booch, G. R. (2005). *El lenguaje unificado de modelado: Guía del usuario*. Pearson Educación.

Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley Professional.

Larman, C. (2004). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development*. Prentice Hall.

Object Management Group. . (2017). *Omg.org*. Obtenido de OMG Unified Modeling Language (OMG UML), Version 2.5.1.: <https://www.omg.org/spec/UML/>

Oracle Corporation. (2023). *Oracle* . Obtenido de Java Documentation: Object-Oriented Programming Concepts.: <https://docs.oracle.com/javase/tutorial/java/concepts/>