

CENTRO PAULA SOUZA  
ETEC PROFESSOR CAMARGO ARANHA  
Técnico em Desenvolvimento de Sistemas

Alan Neves Ribeiro

DESENVOLVIMENTO DE SISTEMAS: Linguagem de Modelagem Unificada

2025  
São Paulo

Alan Neves Ribeiro

## DESENVOLVIMENTO DE SISTEMAS: Linguagem de Modelagem Unificada

Trabalho de pesquisa sobre UML para disciplina de Desenvolvimento de Sistemas ministrada pelo professor Juarez Brandão, o trabalho é requisito parcial para obtenção de menção avaliativa da disciplina.

2025

São Paulo

## **RESUMO**

Este trabalho apresenta uma pesquisa sobre a Linguagem de Modelagem Unificada (UML), abordando seus conceitos fundamentais, tipos de classes, relações entre classes e suas especificidades. A UML é uma linguagem padronizada amplamente utilizada para modelagem de sistemas orientados a objetos, auxiliando no planejamento, concepção e documentação de projetos de software. Além disso, explora-se a relação da UML com metodologias ágeis, como Scrum e Kanban, e sua adaptação ao Manifesto Ágil de 2021.

**PALAVRAS-CHAVE:** UML, Classes, Relacionamentos, Modelagem, POO, Scrum, Kanban, Metodologias Ágeis.

## **ABSTRACT**

This paper presents research on the Unified Modeling Language (UML), covering its fundamental concepts, types of classes, relationships between classes, and specific features. UML is a standardized language widely used for modeling object-oriented systems, assisting in the planning, design, and documentation of software projects. Additionally, this study explores the relationship between UML and agile methodologies, such as Scrum and Kanban, as well as its adaptation to the 2021 Agile Manifesto.

**KEYWORDS:** UML, Classes, Relationships, Modeling, OOP, Scrum, Kanban, Agile Methodologies.

## **SUMÁRIO**

### **1. INTRODUÇÃO**

### **2. HISTÓRICO DA UML**

### **3. TIPOS DE CLASSES NA UML**

### **4. RELACIONAMENTOS ENTRE CLASSES**

### **5. UML E O MANIFESTO ÁGIL DE 2021**

### **6. SCRUM E KANBAN**

### **7. ESPECIFICIDADES DA UML**

### **8. CONCLUSÃO**

## **REFERÊNCIAS**

## 1. INTRODUÇÃO

A Linguagem de Modelagem Unificada (UML - Unified Modeling Language) é uma notação padronizada utilizada para modelar sistemas baseados na Programação Orientada a Objetos (POO). Criada pela Object Management Group (OMG), a UML permite a representação visual dos componentes e relações entre elementos de um sistema, auxiliando no desenvolvimento de software. Este trabalho tem como objetivo explorar os tipos de classes, os diferentes relacionamentos entre classes e as especificidades da UML, bem como sua relação com metodologias ágeis, como Scrum e Kanban, e a adaptação ao Manifesto Ágil de 2021.

## 2. HISTÓRICO DA UML

A UML surgiu na década de 1990 como uma tentativa de unificar as diversas metodologias de modelagem que existiam na época. Três dos principais pesquisadores da área, Grady Booch, James Rumbaugh e Ivar Jacobson, desenvolveram essa linguagem com o apoio da Rational Software Corporation. Em 1997, a UML foi adotada pelo Object Management Group (OMG) como um padrão para modelagem de software.

Desde então, a UML passou por diversas atualizações para refletir as necessidades do mercado e da engenharia de software. Atualmente, a UML é amplamente utilizada para modelagem de sistemas complexos, sendo adotada em áreas como desenvolvimento de software, engenharia de sistemas e negócios.

Com a crescente adoção de metodologias ágeis, algumas organizações passaram a utilizar abordagens mais flexíveis para modelagem, combinando UML com práticas ágeis para manter a documentação concisa e eficaz.

## 3. TIPOS DE CLASSES NA UML

As classes são elementos fundamentais da UML, representando abstrações de objetos do mundo real. Elas podem ser classificadas em diferentes tipos:

- **Classes Concretas:** Representam entidades comuns com atributos e métodos definidos.
- **Classes Abstratas:** Não podem ser instanciadas diretamente, servindo como base para outras classes.
- **Interfaces:** Definem um contrato de implementação para outras classes.
- **Classes de Associação:** Representam relações entre outras classes.
- **Classes de Implementação:** Utilizadas para representar detalhes específicos da lógica do sistema.

## 4. RELACIONAMENTOS ENTRE CLASSES

Os relacionamentos entre classes na UML são fundamentais para definir as interações entre os componentes de um sistema. Os principais tipos de relacionamento são:

- **Associação:** Representa um vínculo entre duas classes. Pode ser unidirecional ou bidirecional.
- **Agregação:** Um tipo de associação que indica uma relação "tem-um", onde um objeto contém outro, mas pode existir independentemente.
- **Composição:** Similar à agregação, mas o objeto contido depende totalmente do objeto contêiner.
- **Herança (Generalização):** Define uma relação pai-filho, onde uma classe filha herda atributos e métodos da classe pai.
- **Dependência:** Indica que uma classe depende temporariamente de outra para executar uma função.

## 5. UML E O MANIFESTO ÁGIL DE 2021

O Manifesto Ágil, criado em 2001, revolucionou o desenvolvimento de software ao enfatizar a colaboração, flexibilidade e entregas incrementais. Em 2021, novas discussões emergiram sobre a necessidade de adaptação contínua das práticas ágeis às novas realidades tecnológicas e empresariais.

Os 12 princípios do Manifesto Ágil enfatizam pontos como satisfação do cliente, entregas frequentes, colaboração entre times, e simplicidade nos processos. Entre os principais princípios, destacam-se:

- **Priorizar a satisfação do cliente através de entregas contínuas de software funcional.**
- **Aceitar mudanças de requisitos, mesmo em fases tardias do desenvolvimento.**
- **Promover a colaboração diária entre desenvolvedores e stakeholders.**

A UML, sendo uma linguagem de modelagem detalhada, muitas vezes foi considerada incompatível com metodologias ágeis, que priorizam menos documentação e mais interação contínua. No entanto, com a evolução do Manifesto Ágil, a UML passou a ser utilizada de forma mais flexível, focando em diagramas simplificados e diretamente aplicáveis ao desenvolvimento iterativo.

## 6. SCRUM E KANBAN

Duas metodologias ágeis amplamente utilizadas são o Scrum e o Kanban.

- **Scrum:** Baseia-se em ciclos curtos de desenvolvimento chamados Sprints, nos quais são planejadas, desenvolvidas e entregues partes do software. Há papéis bem definidos, como Product Owner, Scrum Master e Time de Desenvolvimento.
- **Kanban:** Utiliza um fluxo contínuo de trabalho visualmente organizado em colunas (como "A fazer", "Em andamento" e "Concluído"), permitindo uma adaptação flexível da equipe às demandas do projeto.

A UML pode ser integrada ao Scrum e ao Kanban para melhorar a comunicação e documentação de projetos ágeis, garantindo que a equipe tenha uma visão clara da arquitetura e do funcionamento do sistema.

## 7. ESPECIFICIDADES DA UML

A UML é composta por diversos diagramas, cada um com uma finalidade específica. Os principais são:

- **Diagrama de Classes:** Representa a estrutura do sistema e seus relacionamentos.
- **Diagrama de Casos de Uso:** Define as interações entre usuários e o sistema.
- **Diagrama de Sequência:** Modela a ordem das interações entre os objetos.
- **Diagrama de Estado:** Representa os estados e transições de um objeto.
- **Diagrama de Atividades:** Mostra o fluxo de atividades dentro do sistema.

## 8. CONCLUSÃO

A UML é uma ferramenta essencial para a modelagem de sistemas baseados em POO. Além de auxiliar na documentação e estruturação de software, sua flexibilidade permite aplicação em ambientes ágeis, como os que utilizam Scrum e Kanban.

## REFERÊNCIAS

- Notas de aula de Análise e Projeto de Sistemas, professor Davi Vilar.
- <https://www.luiztools.com.br/post/os-12-principios-do-manifesto-agil-e-dinamicas/>
- <https://focusnfe.com.br/blog/scrum-x-kanban/>
- OMG. *Unified Modeling Language (UML) Specification*. Disponível em: <https://www.omg.org/spec/UML>.
- PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 8. ed. McGraw-Hill, 2014.