

# UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CAMPUS CAMPO MOURÃO

## RESENHA CAPÍTULO 1 - LIVRO “UML Essencial FOWLER, Martin”

### 1 O que é a UML?

UML é uma família de notações gráficas, sua sigla vem de *Unified Modeling Language* (Linguagem de Modelagem Unificada) [1]. Este tipo de linguagem existe há muito tempo na indústria de software, uma de suas principais funções é a ajuda na descrição e no projeto de sistemas de software, especificamente no paradigma orientado a objetos (OO).

### 2 Maneiras de usar a UML

Existem varias maneiras diferentes de utilizar a **UML** como ferramenta, de acordo com **Steve Mellor** e **Martin Fowler** três modos são principais. Sendo eles: esboço, projeto e linguagem de programação [1].

#### 2.1 Esboço

È principalmente usado para transmitir informações sobre um sistema, sendo elas diferentes em cada objetivo que deseja atingir. A fim de exemplificar, diferenciaremos entre desenvolvimento e engenharia reversa:

- **Desenvolvimento:** destacamos ideias e alternativas que utilizaremos no código, ou seja, visualizamos tudo antes de começar a programar.
- **Engenharia Reversa:** utilizamos para explicar o funcionamento de uma parte do sistema.

#### 2.2 Projeto

Sua principal característica está em sempre construir um modelo completo, ou seja, um projeto detalhado para o programador codificar, possibilitando-o a seguir o desenvolvimento como uma tarefa simples e direta.

## 2.3 Linguagem de Programação

Neste caso programadores utilizam a **UML** como linguagem de programação, ou seja, os diagramas são compilados diretamente para código-fonte, utilizando ferramentas sofisticadas (MDA e UML) [1].

A partir da **UML 2** podemos utilizar três tipos de diagramas a fim de modelar lógicas comportamentais, sendo eles: diagramas de interação, estado e atividade.

## 2.4 Perspectivas de Modelagem

Podemos enxergar a **UML** em duas variações de modelagem, sendo elas: conceitual e modelagem de software.

- **Conceitual:** representa uma descrição dos conceitos a respeito de um domínio de estudo.
- **Software:** os elementos de um sistema de software dizem a respeito diretamente dos elementos da UML.

## 3 Como chegamos à UML?

A partir da década de oitenta, com a popularização do paradigma orientado a objeto, muitas pessoas começaram a pensar sobre linguagens gráficas a fim de modelar problemas utilizando tal paradigma [1].

Existiam muitos profissionais que estudavam sobre o assunto na época, porém, havia algumas divergências sobre seus estudos. Somente com a aliança de **Booch** e **Rumbaugh** foi possível a publicação de um método padrão para a utilização e aperfeiçoamento das linguagens gráficas [1].

Somente após o ano de 1997, com a liderança de **Mary Loomis** e **Jim Odell**, começou-se a unificação de várias metodologias em favor de um padrão. Desta forma surgiu-se então a primeira versão do que conhecemos hoje como **UML** [1].

## 4 Notações e Metamodelos

As notações na UML são basicamente a sintaxe gráfica da linguagem de modelagem, com objetivo de exemplificar, citamos a notação de diagrama de classes, que são representados os itens e conceitos, associação e multiplicidade.

Existem algumas definições informais para explicar associação, multiplicidade ou classe. Então com intuito de atribuir significados formais para tais características, é possível indicar um **metamodelo**: diagrama que define conceitos da linguagem. Veja na Figura 1 um exemplo de metamodelo UML.

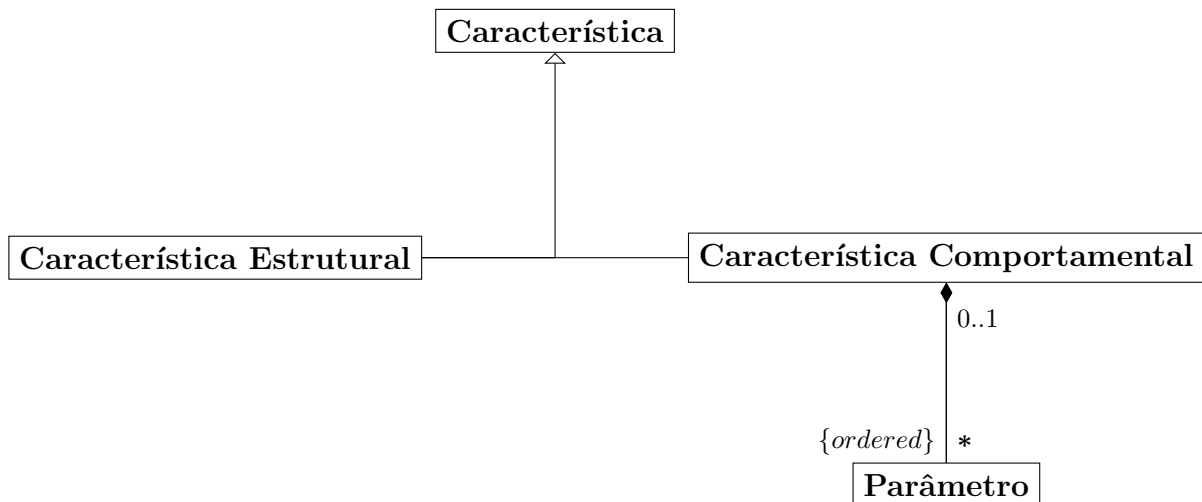


Figura 1: Uma pequena parte do metamodelo UML

Fonte: Martin, FOWLER,. UML Essencial. Grupo A, 2011.

## 5 Diagramas UML

Existem 13 tipos de diagramas oficiais presentes na **UML 2**, que serão classificados ao decorrer da matéria [1]. Porém, como os tipos de diagramas não são totalmente rígidos, podemos utilizar um tipo de diagrama em outro. Veja na Figura 2 os tipos de diagramas UML.

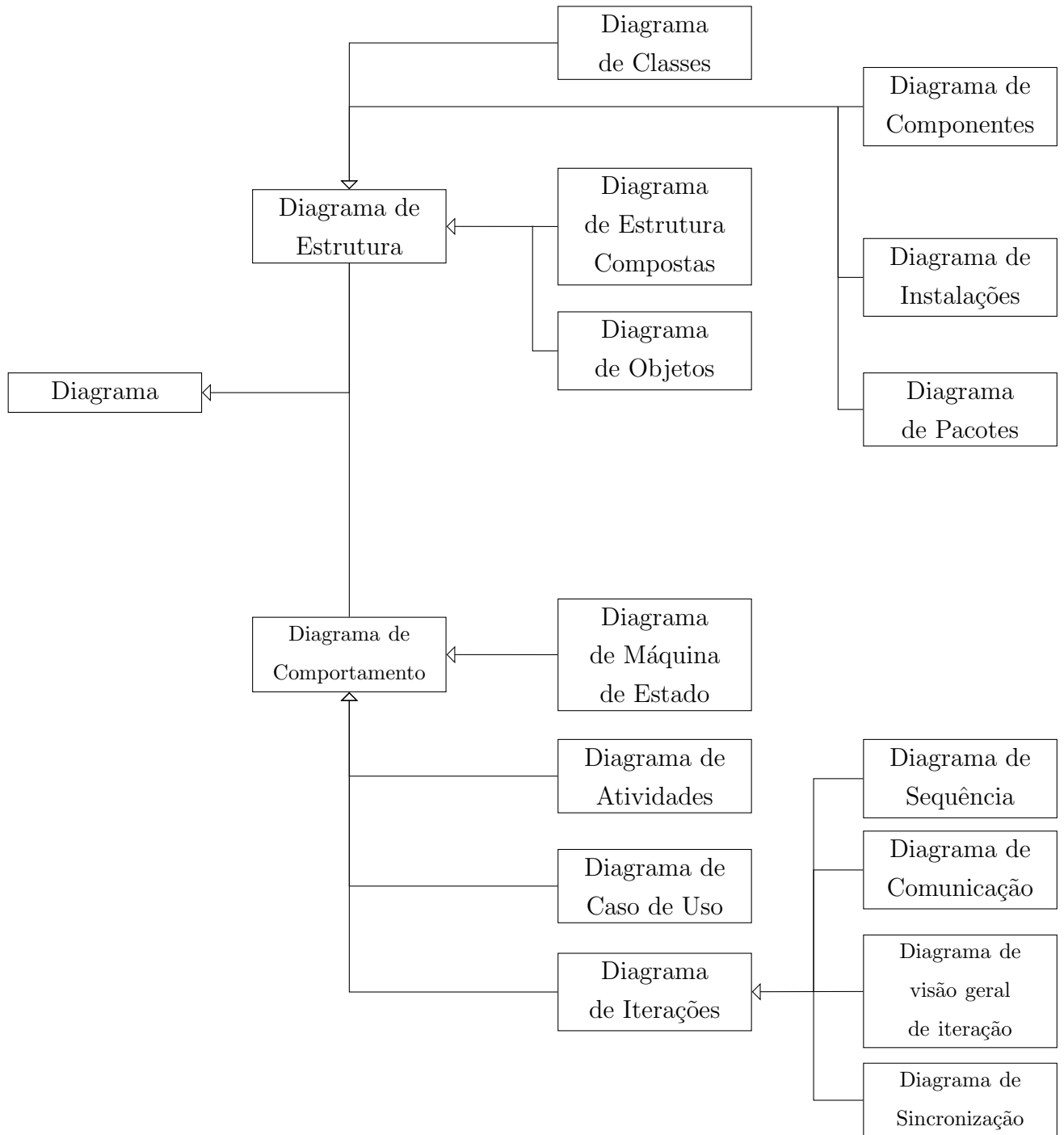


Figura 2: Classificação dos tipos de diagrama da UML

Fonte: Martin, FOWLER,. UML Essencial. Grupo A, 2011.

A **UML** é uma linguagem que você pode esperar bastante regras prescritivas (estabelecidas por um comitê formador de padrões). Porém, como a mesma é muito complexa, acaba por ser frequentemente aberta à múltiplas interpretações.

## 6 Considerações Finais

Apesar de todas as qualidades da utilização de **UML**, um de seus defeitos é que o programador ao analisar o diagrama não irá saber exatamente como ficará o código final, somente ter ideias aproximadas.

Desta forma fica evidente que não devemos ficar presos aos diagramas **UML**, ou seja, caso o mesmo não for suficiente para descrever nossas ideias, podemos recorrer a outras alternativas de diagramas e utiliza-los em conjunto.

## Referências

- [1] Martin. Fowler. *UML Essencial: Um Breve Guia para Linguagem Padrão*. Martin Fowler; trad. João Tortello. – 3.ed. Porto Alegre: Bookman, 2005. ISBN: 8536304545.