

## Introdução à UML

### Linguagem de Modelagem Unificada

Disciplina: Engenharia de Software I  
Prof. Me. Fernando Roberto Proença

## Perguntas...

- O que é um modelo?
    - Um modelo é uma **simplificação** da realidade.
  - Por que construímos modelos?
    - Construímos modelos para **compreender** melhor o sistema que estamos desenvolvendo.
- Um modelo pode ser **estrutural** ou **comportamental**.

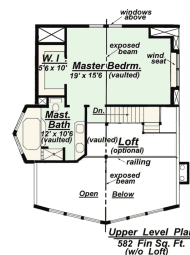
2

## O que é um modelo?



3

## O que é um modelo?



4

## Por que modelar software?

- Ajuda a ter uma **visão geral** do sistema
- Permite especificar a **estrutura** e o **comportamento** do sistema
- Proporciona um **guia** para a construção do sistema
- Documenta as **decisões** tomadas.

5

## Modelagem visual

- Modelagem Visual permite que você construa da forma correta na primeira vez:
  - Entender os requisitos do usuário
  - Verificar se o projeto (design) atende as necessidades
  - Visualizar a interface, a lógica do negócio e os dados separadamente
  - Separar os domínios do negócio conforme necessário
  - Visualizar todas as dependências
  - Validar a performance antes de implementar o código-fonte.

6

## O que é Linguagem de Modelagem Unificada (UML)?



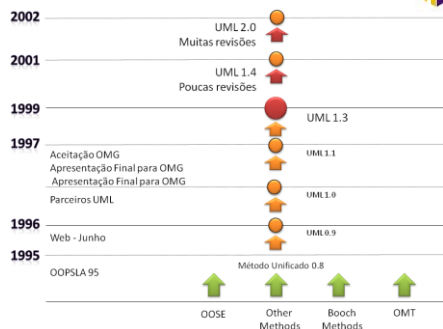
## O que é a UML?



- Linguagem de Modelagem Unificada (*Unified Modeling Language*):
  - É uma **linguagem gráfica** para **visualizar, especificar, construir** e **documentar** os **artefatos de um sistema** de software.
  - É o resultado da **unificação das notações** utilizadas nos métodos OMT (*Object Modeling Technique*) e OOSE (*Object-Oriented Software Engineering*).
  - É adotada por grande parte da indústria de software e por fornecedores de ferramentas CASE como **linguagem padrão de modelagem**.
  - É utilizada com **qualquer** processo de desenvolvimento.

8

## Contexto Histórico



9

## Motivação da UML



- Descobrir entre as **etapas** a serem percorridas, aquelas **mais importantes** do ponto de vista do **cliente**;
- Necessidade de estabelecer um **modelo visual** que permita representar (especificar) os **requisitos do software**.
- Necessidade de estabelecer uma **padronização** para facilitar a **comunicação** entre **analistas** e **programadores**.

10

## A UML não é...



- Um Processo de Desenvolvimento de Software
- Uma Metodologia
- Análise e Projeto Orientado a Objetos
- Regras do Projeto / Negócio
- Dependente de ferramentas CASE
- Dependente do domínio de aplicação

11

## A UML é...



- A UML é uma linguagem para
  - visualização
  - especificação
  - construção
  - documentação
- dos artefatos de um sistema.

12

## UML



- A UML é uma linguagem para **visualização**
  - No processo de desenvolvimento de sistemas de software, é quase **impossível** a visualização de toda a estrutura de um sistema sem o uso de modelos
  - A UML fornece os **símbolos gráficos** para a representação de artefatos de software.
  - Por trás de cada símbolo empregado na notação da UML, existe uma **sintaxe** e uma **semântica** bem-definidas.
  - Assim, o desenvolvedor pode usar a UML para **escrever modelos**, diminuindo a ambiguidade em sua interpretação.

13

## UML

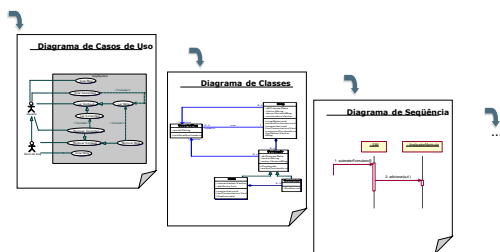
- A UML é uma linguagem para **construção**
  - Os modelos de UML podem ser diretamente “traduzidos” para várias linguagens de programação
  - É possível **mapear** os **modelos da UML** para **linguagens de programação** tais como, Java, C++, C#, Visual Basic, Python, entre outras.
  - Realização de uma **engenharia de produção**: geração de código a partir de um modelo em UML.
  - O processo inverso, a **engenharia reversa**, também é possível, com a reconstrução de um modelo a partir de sua implementação.

14

## UML



- A UML é uma linguagem para **documentação**
  - Cada modelo criado é um artefato de software



15

## UML



- A UML basicamente permite modelar:
  - elementos
  - relacionamentos
  - diagramas.

16

## UML – Elementos



- Estruturais
  - Classes, objetos, interfaces, componentes, colaboração
- Comportamentais
  - Casos de uso, interações, atividades, máquinas de estado
- Grupos de elementos
  - Pacotes, subsistemas
- Outros
  - Anotações.

17

## UML – Relacionamentos



- Associações
- Generalizações
- Agregação
- Composição

18

## UML – Diagramas



- Um modelo é representado por um ou mais diagramas. Desta forma, um diagrama pode ser visto como uma visão dentro de um modelo.
- Um diagrama pode ser representado de várias formas, dependendo de quem irá interpretá-lo.

## Principais Diagramas UML

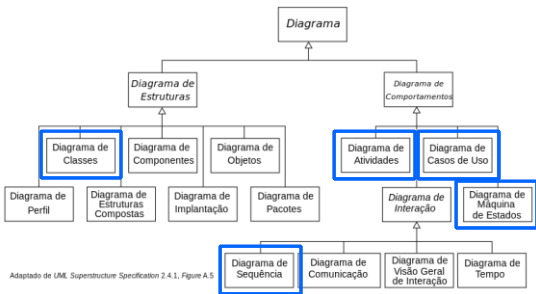
### Ponto de Vista Estático

Diagrama de Casos de Uso
Diagrama de Classes
Diagrama de Objetos
Diagrama de Componentes
Diagrama de Implantação

### Ponto de Vista Dinâmico

Diagrama de Atividades
Diagrama de Estados
Diagramas de Sequência
Diagramas de Colaboração

## Conjunto de Diagramas UML



## UML – Conceitos



- Ator
- Atividade
- Interface
- Pacote
- Classe
- Evento

## UML – Conceitos



- Ator
  - Define o papel do usuário do sistema modelado.
  - Pode ser:
    - Uma pessoa;
    - Outro sistema informático;
    - Um equipamento hardware especializado;
  - Uma entidade pode ser representada por vários atores, já que pode estar a assumir diferentes papéis.

## UML – Conceitos



- Atividade
  - Representa os fluxos conduzidos por processamentos;
  - Modelagem de aspectos dinâmicos de um sistema.
- Interface
  - Abstração de comportamentos a serem implementados pelas classes.
- Pacote:
  - Agrupamentos de elementos.

## UML – Conceitos



- Classe
  - Estrutura que abstrai um conjunto de objetos com características similares;
  - Define o comportamento de seus objetos através de métodos e os estados possíveis destes objetos através de atributos.
- Evento
  - Resultado de uma ação.

25

## UML – Algumas Ferramentas CASE

- Astah
- Microsoft Office Visio
- Dia
- Visual UML
- Jude UML
- Argo UML

26

## Quem deve modelar?



27

## Dúvidas?

Prof. Me. Fernando Roberto Proença

[fernando.proenca@uemg.br](mailto:fernando.proenca@uemg.br)



28