



**Curso:** Desenvolvimento de Software Multiplataforma

**Disciplina:** Engenharia de Software II

**Professor:** André Olímpio

# Engenharia de Software II

 **LOADING** 

# UML

- Linguagem clássica e mais completa para modelagem de sistemas.
- Atualmente está na versão 2.5
- Total de 14 diagramas.
- Linguagem universal para modelagem de sistemas.
- Qualquer desenvolvedor de software no mundo atualmente tem por obrigação entender uma modelagem desenvolvida através da UML.



Fonte: UML.org

# Diagramas da UML

- Diagramas Estruturais
- Diagramas Comportamentais
- Diagramas de Interação



Fonte: UML.org

# Diagramas Estruturais

- Diagrama de classes
- Diagramas de objetos
- Diagramas de componentes
- Diagrama de instalação
- Diagramas de pacotes
- Diagramas de estrutura
- Diagrama de perfil



Fonte: Pinterest.com

# Diagramas Comportamentais

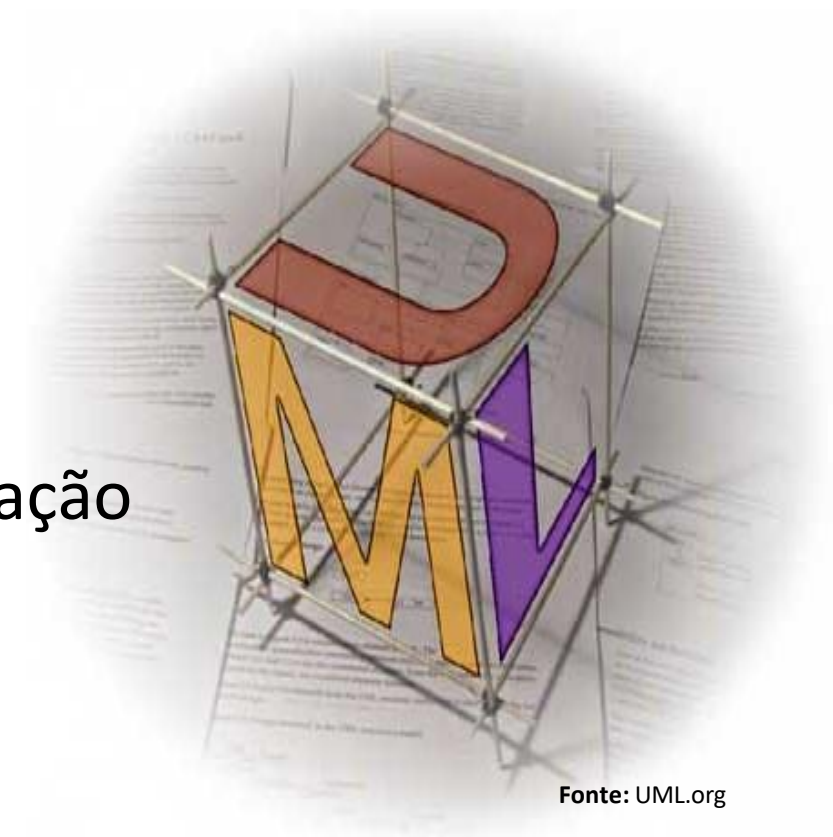
- Diagrama de casos de uso
- Diagramas máquina de estados
- Diagramas de atividades



Fonte: Pinterest.com

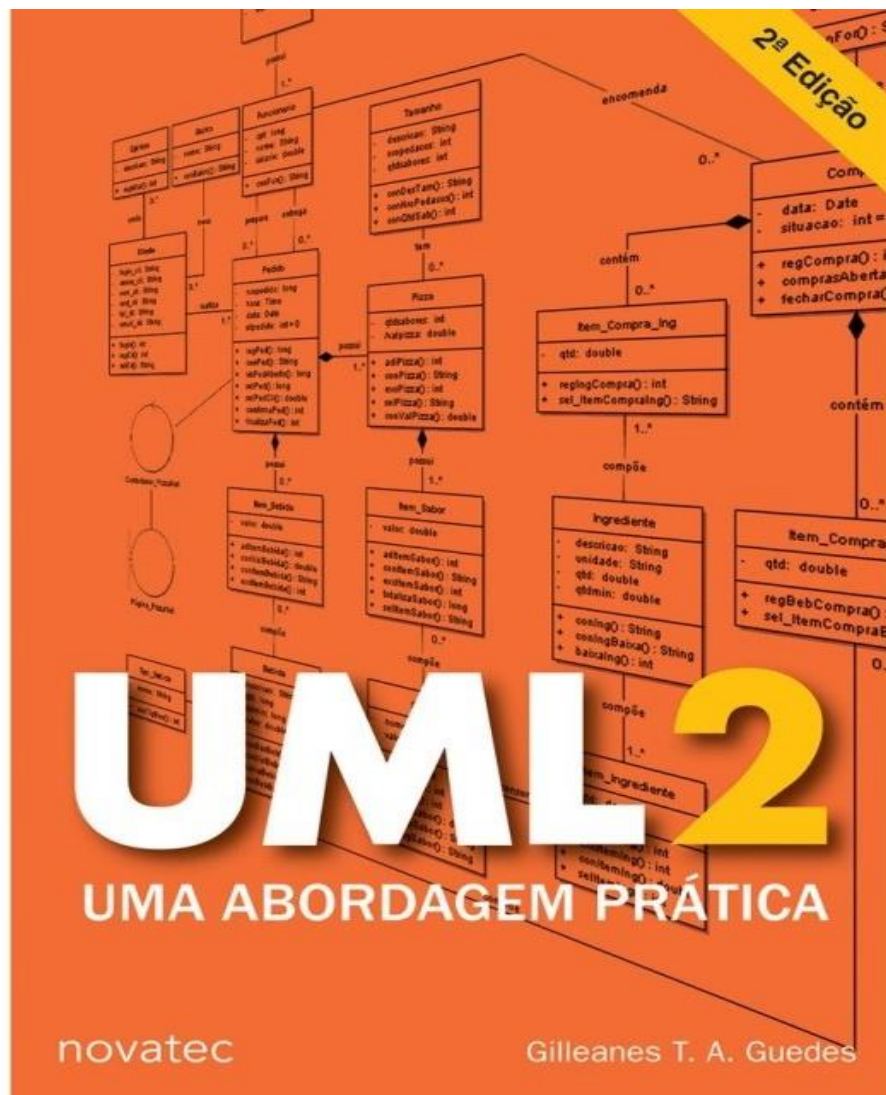
# Diagramas de Interação

- Diagrama de sequência
- Diagrama de interatividade
- Diagrama de comunicação ou colaboração
- Diagrama de tempo



Fonte: UML.org





- **Título:** UML 2 – Uma abordagem prática
- **Autor:** Gilleanes Guedes
- **Editora:** Novatec
- **Ano:** 2011

Fonte: NovaTec.com.br



# Ferramenta CASE

- Computer-Aided Software Engineering.
- *Engenharia de Software Auxiliada por Computador.*
- É um tipo de software que permite criar diagramas da UML.
- Utilizável em qualquer plataforma (Multiplataforma).
- Ferramenta imprescindível para desenvolvimento de software.
- Gera resultados muito satisfatórios, com ótima qualidade de visualização e de fácil compreensão.

## JUDE / ASTAH

- **JUDE** é um software para modelagem UML.
- É desenvolvido na plataforma JAVA, o que garante sua portabilidade para qualquer plataforma que possui a máquina virtual Java instalada.
- O nome JUDE significa:  
*Java and UML Developers Environment*  
(Ambiente para Desenvolvedores UML e Java).
- Atualmente chama-se **ASTAH**.



Fonte: Astah.net

# Diagrama de Use Case

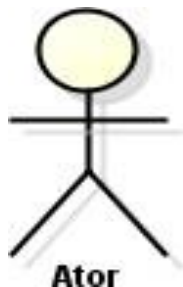


Fonte: UML.org

# Diagrama de Casos de Uso

- É o diagrama mais geral da UML.
- Representa o sistema no ponto de vista do usuário, ou seja, de fora para dentro.
- Um caso de uso representa **uma ação a ser realizada no sistema de software e não ações realizadas pelo usuário.**
- Caso de uso é a representação de uma funcionalidade do sistema.

# Diagrama de Casos de Uso



**Ator** – é uma pessoa, um equipamento ou um outro sistema que irá interagir diretamente com o software a ser desenvolvido.

**Caso de uso** – é uma ação realizada pelo sistema. Sempre deve ser escrita com um verbo no infinitivo.

**Associação** – é a ligação entre os itens que compõem o diagrama (ator e caso de uso).

Fonte: Elaborado pelo autor

# Diagrama de Casos de Uso

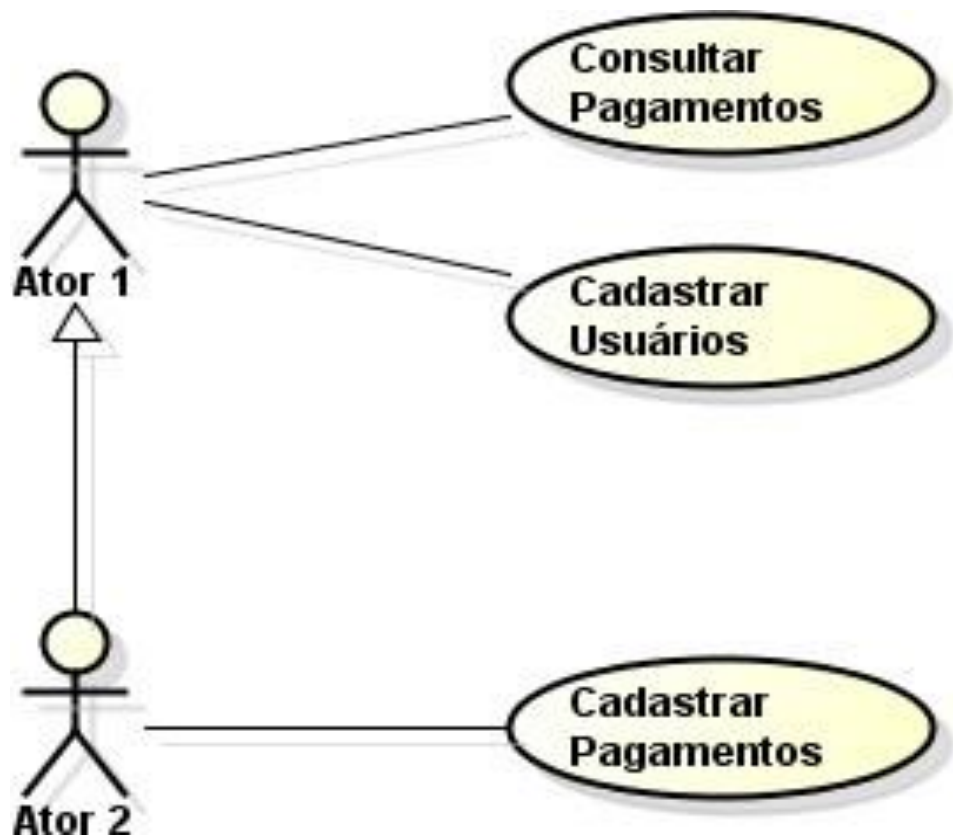
- Mesmo sendo o diagrama mais geral dos existentes na UML, também possui recursos específicos e mais refinados.
- A orientação a objetos prega o estudo de cada item de maneira separada, porém permitindo a junção destes itens quando assim for necessário.
- Assim, podemos deixar a estruturação do software mais consistência e eficiente.

# Herança

- É o princípio no qual um objeto pode herdar características de outro(s) objetos ancestrais, além de possuir suas próprias características.
- No diagrama de caso de uso, esta representação pode ser realizada tanto para o caso de uso em si quanto para os atores.
- Normalmente, a herança é definida na representação dos atores e suas respectivas interações com o sistema.



# Herança



Fonte: Elaborado pelo autor

- No exemplo ao lado, podemos verificar que o **Ator 1** pode realizar duas ações no sistema e, através do conceito da herança, o **Ator 2** pode realizar três ações neste mesmo sistema.

# Relacionamento entre Casos de Usos

- Existem duas formas de identificarmos o relacionamento entre dois ou mais casos de usos existente em um diagrama.
- São eles:
  - **Include**
  - **Extends**
- Veja o exemplo a seguir:



# Include

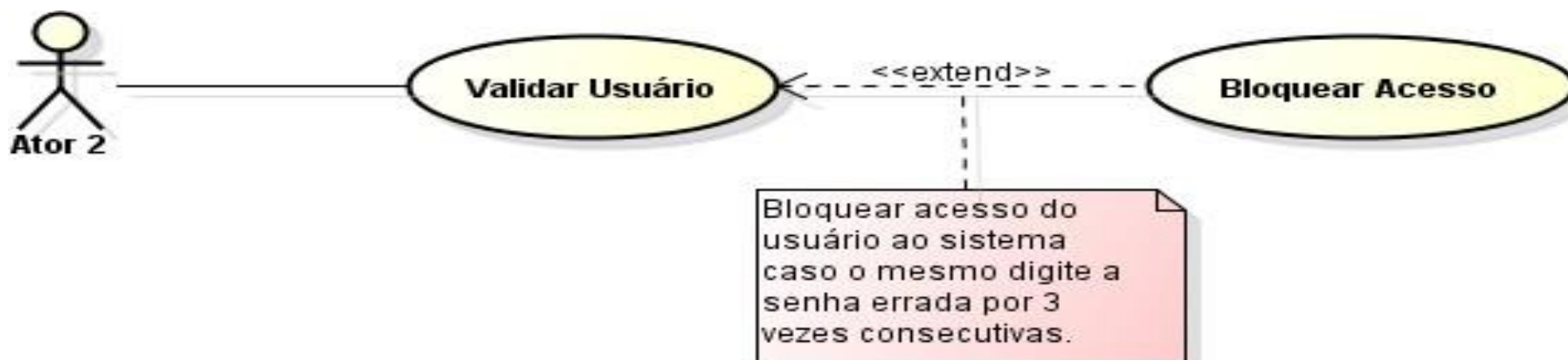
- Quando um caso de uso “A” inclui outro caso de uso “B”.
- Isto implica que ao executar um caso de uso, o outro também será executado.
- Ou seja, toda vez em que “A” for executada “B” também será.



Fonte: Elaborado pelo autor

# Extends

- Quando um caso de uso “A” tem um relacionamento do tipo extends com um caso de uso “B”.
- Implica que ao se executar “A”, a ação “B” será executada somente em uma situação específica.



Fonte: Elaborado pelo autor

# Leitura Complementar

- **Modelagem de sistemas através de UML: uma visão geral**

<https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913>

- **Introduzindo os Conceitos de Modelagem e Diagramação**

<https://spaceprogrammer.com/uml/introduzindo-o-conceito-de-modelagem-e-diagramacao/>

- **SOMOS TODOS T.I. – Diagrama da UML – Casos de Uso**

<https://youtu.be/fk1JdFjKneE>