



# Iniciando com UML



## que é a Linguagem UML?


**UML**, abreviação de Unified Modeling Language, é uma linguagem de modelagem padronizada que consiste em um conjunto integrado de diagramas, desenvolvido para ajudar os desenvolvedores de sistemas e software a especificar, visualizar, construir e documentar os artefatos de sistemas de software, bem como modelagem de negócios e outros. A UML representa uma coleção das melhores práticas de engenharia de software que se mostraram bem-sucedidas na modelagem de sistemas grandes e complexos. A UML é uma parte muito importante do desenvolvimento de software orientado a objetos e do processo de desenvolvimento de software. A UML usa principalmente notações gráficas para expressar o design de projetos de software. O uso da UML ajuda as equipes de projeto a se comunicar, explorar possíveis projetos e validar o design arquitetônico do software.

## Quais são os diagramas de UML ?

Existem três classificações de diagramas UML:

A figura 1 apresenta todos os diagramas:

- **Diagramas de comportamento:** Um tipo de diagrama que descreve recursos comportamentais de um sistema ou processo de negócios. Isso

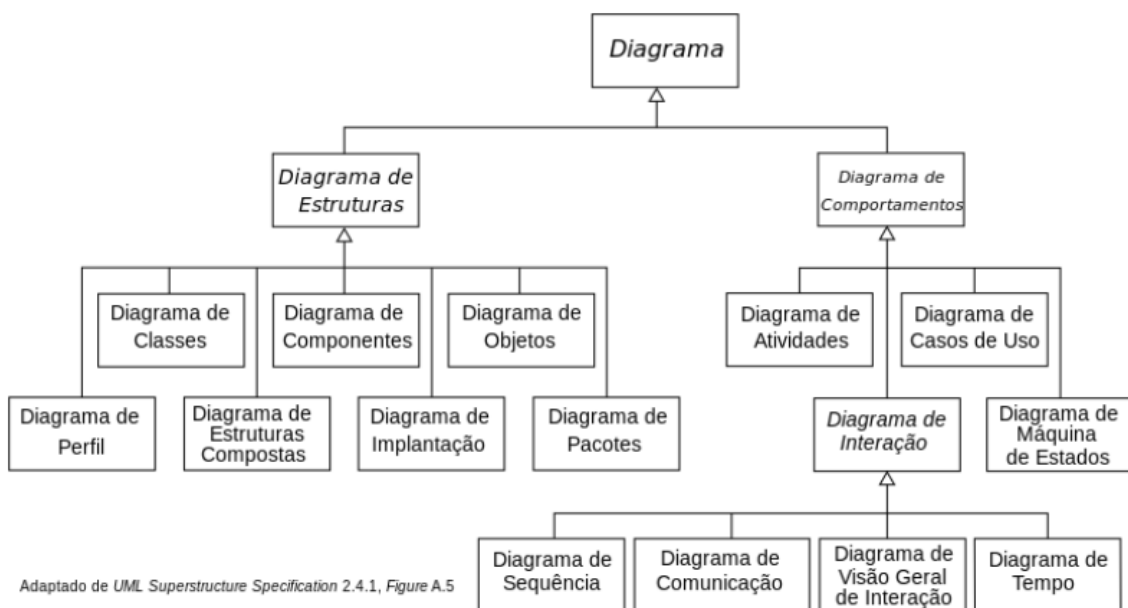
inclui atividade, máquina de estado e diagramas de casos de uso, bem como os quatro diagramas de interação. 

- **Diagramas de interação:** Um subconjunto de diagramas de comportamento que enfatizam as interações com objetos. Isso inclui comunicação, visão geral da interação, sequência e diagramas de tempo.
- **Diagramas de estrutura:** Um tipo de diagrama que descreve os elementos de uma especificação independentemente do tempo. Isso inclui diagramas de classe, estrutura composta, componente, implantação, objeto e pacote.



**Figura 1 – Divisão dos diagramas de UML**

A tabela 1 resume os treze diagramas e a coluna de prioridade indica a importância do aprendizado.



**Figura 1 – Divisão dos diagramas de UML**

A tabela 1 resume os treze diagramas e a coluna de prioridade (👤) indica a importância do aprendizado.

Fonte: Autoral

Diagrama	Descrição	Prioridade de Aprendizagem
Diagrama de atividades	Descreve processos de negócios de alto nível, incluindo fluxo de dados	Alto
Diagrama de Classes	Mostra uma coleção de elementos de modelo estático, como classes e tipos, seu conteúdo e seus relacionamentos.	Alto
Diagrama de Comunicação	Mostra instâncias de classes, suas inter-relações e o fluxo de mensagens entre elas. Os diagramas de comunicação geralmente se concentram na organização estrutural dos objetos que enviam e recebem mensagens. Anteriormente chamado de diagrama de colaboração.	Baixo
Diagrama de componentes	Descreve os componentes que compõem um aplicativo, sistema ou empresa. Os componentes, suas inter-relações, interações e suas interfaces públicas são representados.	Médio



Diagrama de estrutura composta

Descreve a estrutura interna de um classificador (como uma classe, componente ou caso de uso), incluindo os pontos de interação do classificador com outras partes do sistema.

Baixo

Diagrama de implantação

Mostra a arquitetura de execução dos sistemas. Isso inclui nós, ambientes de execução de hardware ou software, bem como o middleware que os conecta.

Baixo

Diagrama de controle

Uma variante de um diagrama de atividades que mostra o fluxo de atividades que mostra o fluxo de controle em um sistema ou processo visão geral da de negócios. Cada nó / atividade dentro do diagrama pode representar outro diagrama de interação.

Baixo

Diagrama de Objetos

Descreve objetos e seus relacionamentos em um determinado momento, normalmente um caso especial de um diagrama de classes ou de comunicação.

Baixo

Diagrama de Pacotes

Mostra como os elementos do modelo são organizados em

Médio

pacotes, bem como as dependências entre os pacotes.



**Diagrama de sequência** Modela a lógica sequencial, com efeito a ordem do tempo das mensagens entre os classificadores.

**Diagrama da Máquina de Estado** Descreve os estados em que um objeto ou interação pode estar, bem como as transições entre estados. Anteriormente chamado de Médio diagrama de estados, diagrama de estados ou diagrama de transição de estados.

**Diagrama de tempo** Descreve a alteração no estado ou condição de uma instância ou função do classificador ao longo do tempo. Normalmente usado para Baixo mostrar a alteração no estado de um objeto ao longo do tempo em resposta a eventos externos.

**Diagrama de Alto** Mostra casos de uso, atores e suas casos de uso inter-relações.

### Por que existem tantos diagramas de UML?

A primeira coisa a observar sobre a UML é que existem muitos diagramas diferentes para se acostumar. A razão para isso é que é possível olhar para um sistema de muitos pontos de vista diferentes. Um desenvolvimento de

software terá muitas partes interessadas desempenhando um papel. Por exemplo: Analistas de Negócios, Designers UX, Desenvolvedores, Testadores e o cliente. Todas essas pessoas estão interessadas em diferentes aspectos do sistema e cada uma delas exige um nível diferente de detalhes. Por exemplo, um desenvolvedor precisa entender o design do sistema e poder convertê-lo em uma linguagem de programação. Por outro lado, o analista de negócio está interessado no comportamento do sistema como um todo e precisa entender como o produto funciona. A UML tenta fornecer uma linguagem tão expressiva que todas as partes interessadas possam se beneficiar de pelo menos um diagrama UML.

### **Diagrama Estrutural, Diagrama de Comportamento, Diagrama de Interação**

Diagramas estruturais devem ser utilizados para especificar detalhes da estrutura do sistema (parte estática), por exemplo: classes, métodos, interfaces, namespaces, serviços, como componentes devem ser instalados, como deve ser a arquitetura do sistema etc.

Diagramas comportamentais devem ser utilizados para especificar detalhes do comportamento do sistema (parte dinâmica), por exemplo: como as funcionalidades devem funcionar, como um processo de negócio deve ser tratado pelo sistema, como componentes estruturais trocam mensagens e como respondem às chamadas etc.

Diagramas de interação mostram como os objetos interagem uns com os outros. Permitem assim modelar os aspectos dinâmicos de um sistema.

### **Atividade extra**



Ler o texto: “ Modelagem de sistemas por meio da UML”

Link para a atividade: <https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913> (**acesso em 19/09/2022**)

## Referências Bibliográficas

BOOCH, Grady. **Uml - Guia do Usuário**. Rio de Janeiro: Editora Campus, 2018.

GUEDES, Gilleanes T. A. **UML 2 - Uma Abordagem Prática**. São Paulo: NovaTec, 2018.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**. Porto Alegre: Bookman, 2010.

SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Pearson, 2015

**Ir para exercício**