

Linguagem UML

(Unified Modeling Language)

UML é uma linguagem de modelagem padrão para visualizar, especificar, construir e documentar artefatos de um sistema. Ela ajuda a representar de forma clara a estrutura e o comportamento de um software.

Antes de construir um programa, os programadores desenham o que ele vai fazer e como será por dentro, como se fosse o projeto de uma casa, mas para sistemas. Ela serve para facilitar o entendimento e a comunicação entre desenvolvedores, analistas, clientes e outros stakeholders.

Existem vários tipos de diagramas, divididos em duas categorias principais:

1. Diagramas Estruturais (representam a estrutura estática do sistema):

- Diagrama de Classes
- Diagrama de Objetos
- Diagrama de Componentes
- Diagrama de Implantação (Deployment)
- Diagrama de Pacotes

2. Diagramas Comportamentais (representam o comportamento e interações):

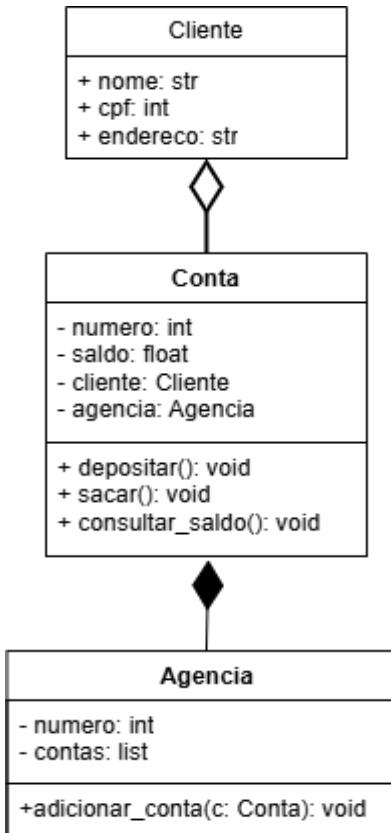
- Diagrama de Casos de Uso
- Diagrama de Sequência
- Diagrama de Atividades
- Diagrama de Estados
- Diagrama de Comunicação
- Diagrama de Interação Geral
- Diagrama de Tempo

1. Diagramas Estruturais

Esses diagramas mostram a **estrutura estática** do sistema, ou seja, os elementos que o compõem e seus relacionamentos.

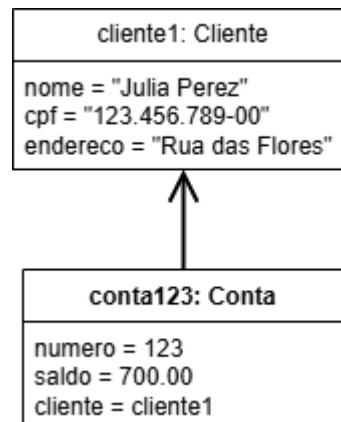
1.1 Diagrama de Classes

- Representa as **classes** do sistema, seus respectivos **atributos** e **métodos**, e os **relacionamentos** entre elas.
- É o principal diagrama usado para modelagem orientada a objetos.
- Exemplo: modelar um sistema bancário com as classes **Conta**, **Cliente**, **Agência**.



1.2 Diagrama de Objetos

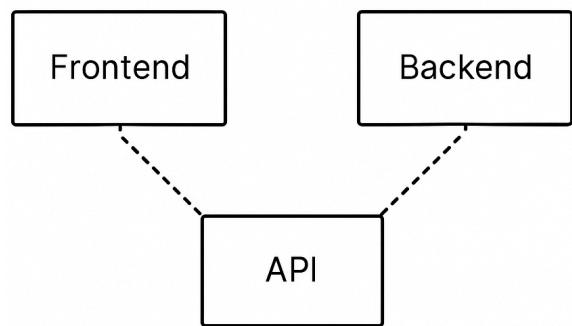
- Mostra **instâncias reais** das classes em um dado momento.
- Útil para ilustrar **situações específicas** de execução.
- Exemplo: um objeto cliente1:Cliente associado a conta123:Conta.



Por: Alice Dantas

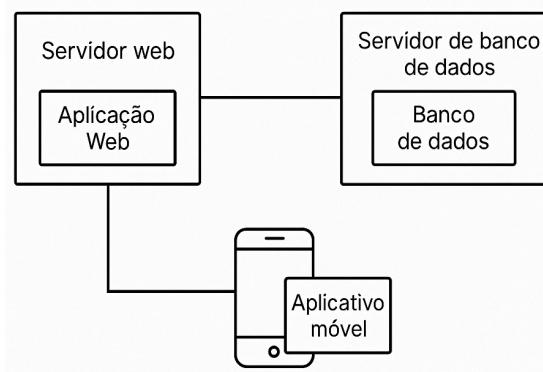
1.3 Diagrama de Componentes

- Representa os **módulos físicos** do sistema (componentes de software).
- Mostra **dependências e interfaces** entre os componentes.
- Exemplo: um sistema dividido em **Frontend**, **Backend**, **API**.



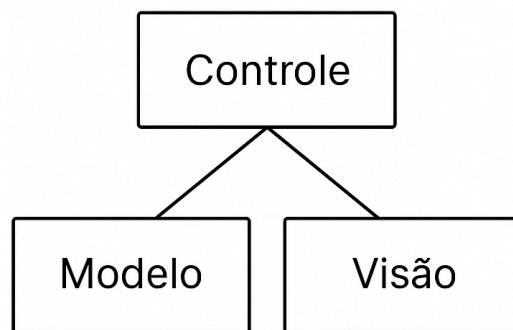
1.4 Diagrama de Implantação (Deployment)

- Mostra como o sistema será **implantado fisicamente** em servidores e dispositivos.
- Inclui **nós** (máquinas, dispositivos), **artefatos de software** e suas conexões.
- Exemplo: servidor web, servidor de banco de dados, dispositivos móveis.



1.5 Diagrama de Pacotes

- Organiza o sistema em **pacotes** (conjuntos de classes ou componentes).
- Ajuda na **modularização** e na **visão geral da arquitetura**.
- Exemplo: pacotes **Controle**, **Modelo**, **Visão** num padrão MVC.

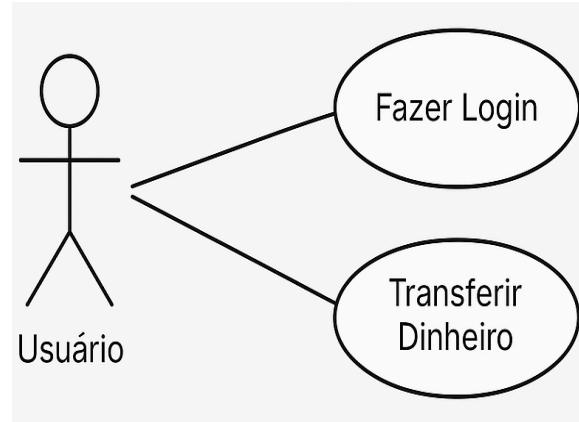


2. Diagramas Comportamentais

Mostram o **comportamento dinâmico** do sistema e as **interações** entre seus elementos.

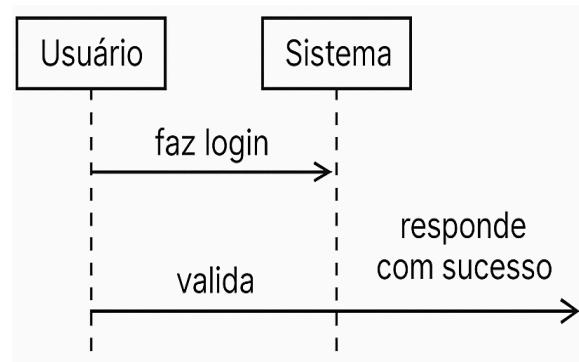
2.1 Diagrama de Casos de Uso

- Representa **funcionalidades** do sistema sob o ponto de vista do **usuário**.
- Mostra os **atores** e os **casos de uso** (ações que eles realizam).
- Exemplo: Usuário → Fazer Login, Transferir Dinheiro.



2.2 Diagrama de Sequência

- Mostra a **ordem temporal das mensagens** trocadas entre objetos para realizar uma função.
- Útil para entender **fluxos de execução**.
- Exemplo: um usuário faz login → o sistema valida → responde com sucesso.



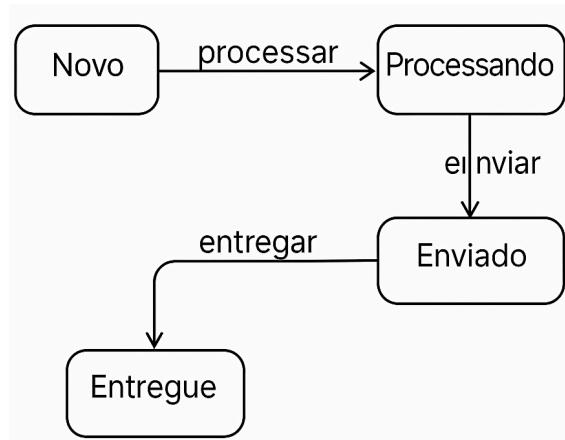
2.3 Diagrama de Atividades

- Representa **fluxos de trabalho** ou **processos**, similar a um fluxograma.
- Útil para modelar **processos de negócios** ou **algoritmos**.
- Exemplo: processo de compra online desde a seleção até o pagamento.



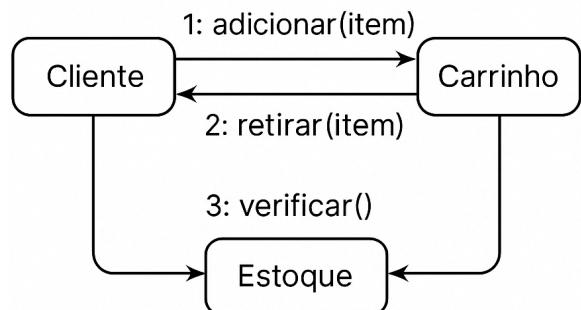
2.4 Diagrama de Estados

- Mostra os **estados possíveis** de um objeto e as **transições** entre eles.
- Indicado quando o comportamento depende fortemente do estado.
- Exemplo: um pedido que pode estar em **Novo**, **Processando**, **Enviado**, **Entregue**.



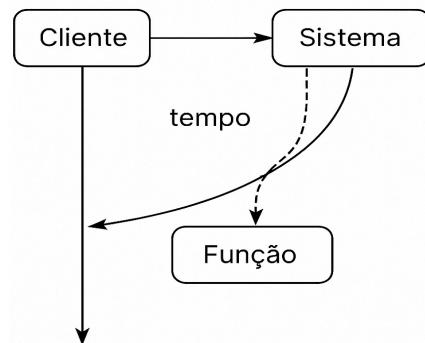
2.5 Diagrama de Comunicação

- Mostra **interações entre objetos**, destacando as **ligações e mensagens trocadas**.
- Exemplo: como **Cliente**, **Carrinho**, **Estoque** se comunicam numa compra.



2.6 Diagrama de Interação Geral

- Uma visão **mais ampla e resumida** das interações.
- Pode combinar múltiplas interações (sequência, tempo, etc.) em um só diagrama.



2.7 Diagrama de Tempo

- Representa **variações no estado dos objetos ao longo do tempo**.
- Foco nos **prazos e sincronização**.
- Exemplo: comunicação em tempo real entre sensores e um servidor.

