

# Modelagem de Sistemas

## Tipos

# Introdução

É o processo de **representar graficamente ou de forma estruturada** como um sistema funciona ou deverá funcionar.

A ideia é **facilitar a comunicação** entre desenvolvedores, usuários e clientes, antes da implementação.

Ela responde a perguntas como:

- O que o sistema precisa fazer?
- Quem vai usar o sistema?
- Como as partes interagem entre si?
- Quais são os fluxos de informação e de processos?

# Objetivos

- **Visualizar:** mostrar de forma clara a lógica do sistema.
- **Comunicar:** alinhar equipe técnica, cliente e usuários.
- **Documentar:** registrar requisitos e decisões.
- **Planejar:** antecipar problemas antes da codificação.

# 1. Modelagem de Processos

**O que é:** Representa o passo a passo de um processo dentro da empresa ou do sistema, mostrando **atividades, decisões e fluxo de trabalho**.

**Por que usar:** ajuda a entender como o trabalho é feito antes mesmo de existir um software.

**Exemplo prático:** Processo de compra online:

- Cliente escolhe produto
- Cliente paga
- Sistema confirma pagamento
- Produto é enviado

**Notação mais comuns:** Fluxogramas.



# Modelagem de Processos

## 2. Modelagem de Dados

**O que é:** Mostra como as informações serão organizadas e armazenadas no sistema.

**Por que usar:** facilita a criação do **banco de dados** e garante que os dados tenham sentido e consistência.

**Exemplo prático:** Um cadastro de alunos com nome, matrícula, e-mail e telefone.

**Notações mais comuns:** Modelo Conceitual (MER), Modelo Lógico, Modelo Físico.



## Modelagem de Dados

### 3. Modelagem de Entidades e Relacionamentos

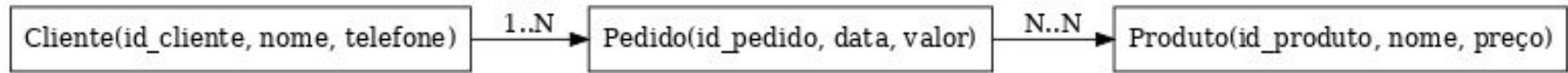
**O que é:** É um tipo específico de modelagem de dados, focada em **entidades (tabelas)**, **atributos (colunas)** e **relacionamentos (ligações entre tabelas)**.

**Por que usar:** permite enxergar a ligação entre diferentes conjuntos de dados.

**Exemplo prático:**

- Cliente faz vários pedidos.  
Pedido contém vários produtos.

**Notação mais comum:** DER (Diagrama Entidade-Relacionamento).



### Relacionamentos entre entidades:

- Um **Cliente** pode fazer vários **Pedidos** (1..N).
- Um **Pedido** pode conter vários **Produtos** (N..N).

## Modelagem de Entidades e Relacionamentos

## 4. Modelagem de Casos de Uso

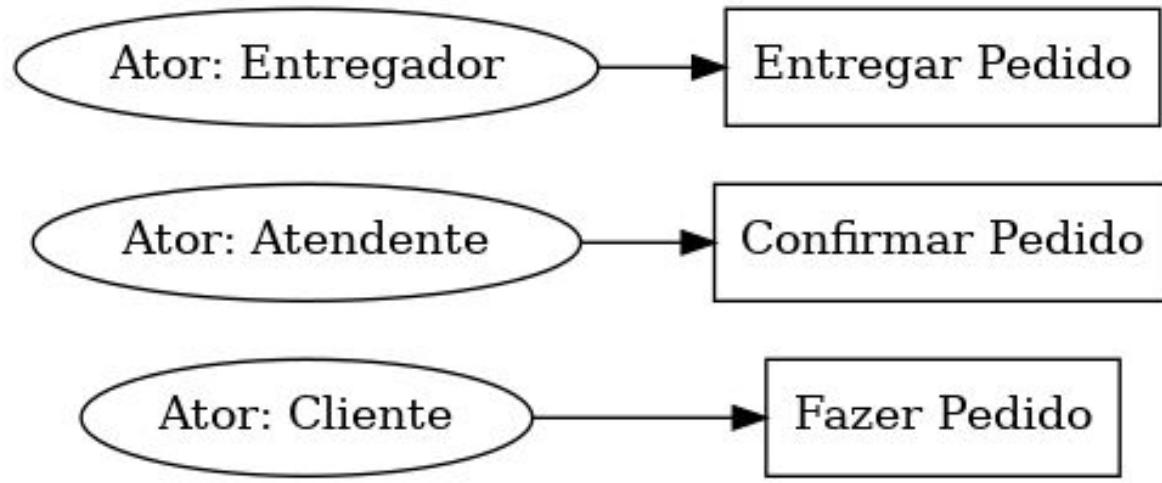
**O que é:** Representa quem vai usar o sistema (atores) e o que ele pode fazer (funcionalidades).

**Por que usar:** é a porta de entrada para entender os requisitos do sistema.

**Exemplo prático:**

- Cliente → Fazer pedido
- Atendente → Confirmar pedido
- Entregador → Entregar pedido

**Notação mais comum:** Diagrama de Casos de Uso.



## Modelagem de Caso de Uso

# 5. Modelagem de Sequência

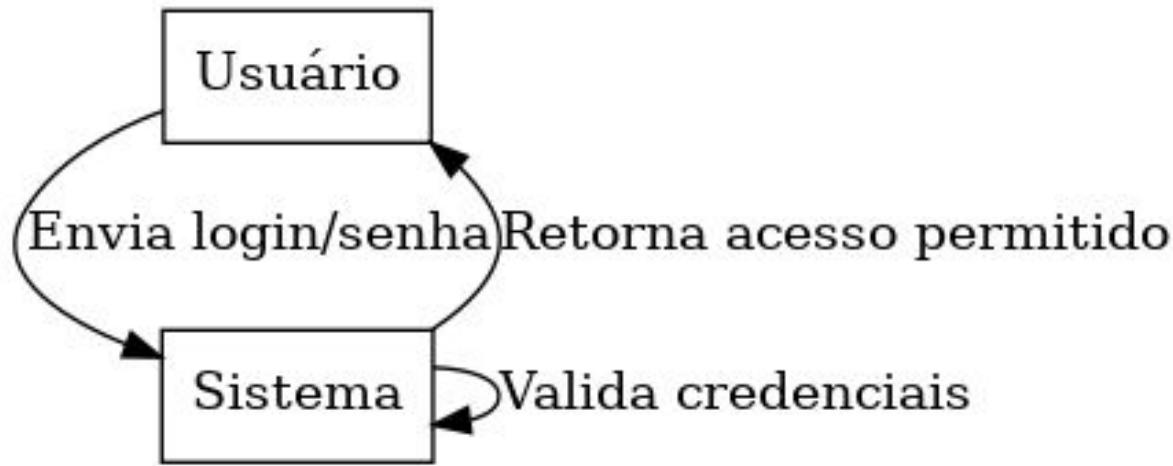
**O que é:** Mostra a **ordem em que acontecem as interações** entre objetos, usuários e sistema, ao longo do tempo.

**Por que usar:** ajuda a entender o **fluxo de mensagens e chamadas** dentro do sistema.

**Exemplo prático (login):**

1. Usuário envia login e senha
2. Sistema valida os dados
3. Sistema retorna “Acesso permitido”

**Notação mais comum:** Diagrama de Sequência.



## Modelagem de Sequência

## 6. Modelagem de Estado

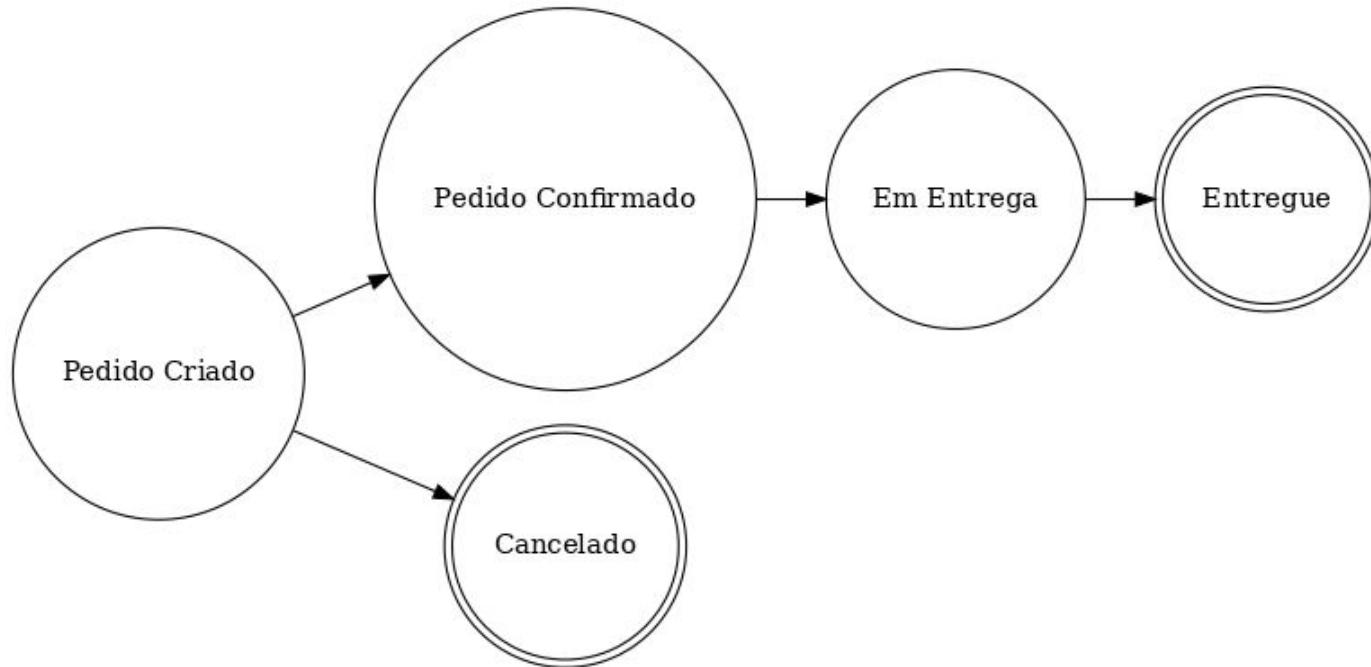
**O que é:** Mostra os **diferentes estados de um objeto** e como ele muda de estado a partir de eventos.

**Por que usar:** facilita o entendimento do **ciclo de vida de um objeto**.

**Exemplo prático:** Pedido no sistema de compras:

- Estado inicial: Criado
- Depois: Confirmado
- Depois: Em entrega
- Estado final: Entregue ou Cancelado

**Notação mais comum:** Diagrama de Estados.



## Modelagem de Estado

# 7. Modelagem de Componentes

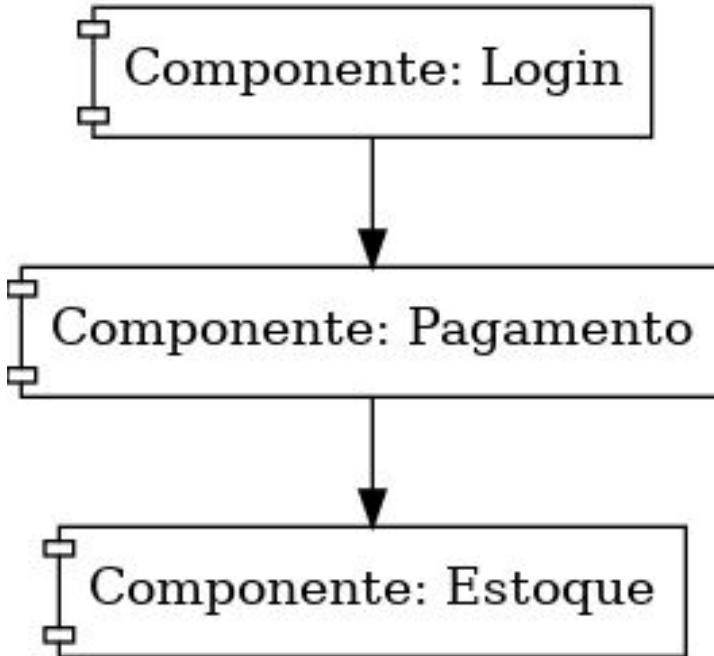
**O que é:** Mostra o sistema dividido em **partes independentes (componentes)** e como eles se relacionam.

**Por que usar:** ajuda a planejar a construção do sistema de forma **modular e organizada**.

**Exemplo prático:**

- Sistema de e-commerce:
  - Componente de Login
  - Componente de Pagamento
  - Componente de Estoque

**Notação mais comum:** Diagrama de Componentes.



## Modelagem de Componentes

## 8. Modelagem de Arquitetura

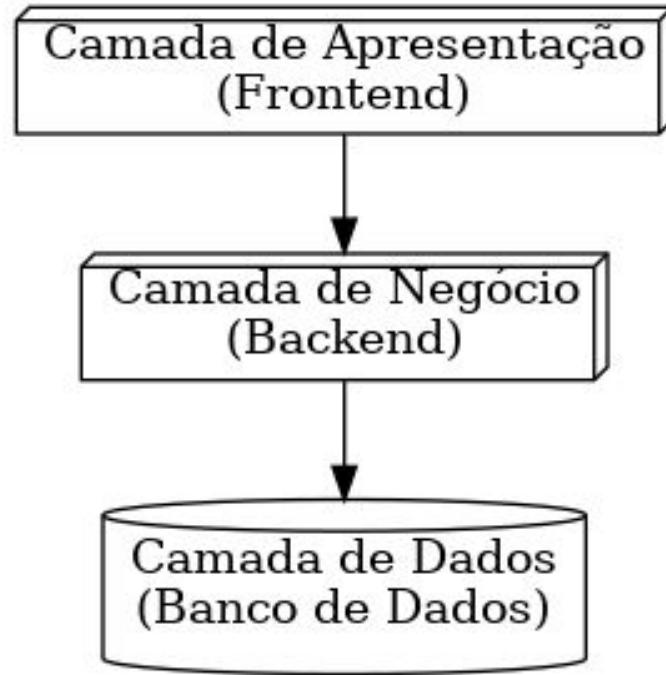
**O que é:** Dá uma **visão geral do sistema**, mostrando camadas, servidores, bancos de dados e integrações.

**Por que usar:** permite entender a **estrutura tecnológica** necessária para o sistema funcionar.

**Exemplo prático:** Arquitetura em 3 camadas:

- **Camada de Apresentação (Frontend):** telas que o usuário vê.
- **Camada de Negócio (Backend):** regras e lógica.
- **Camada de Dados:** banco de dados.

**Notações mais comuns:** Diagrama de Arquitetura.



## Modelagem de Arquitetura

# Resumo

**Processos** → mostra o fluxo de atividades.

**Dados** → como as informações são organizadas.

**Entidades e Relacionamentos** → como tabelas se conectam.

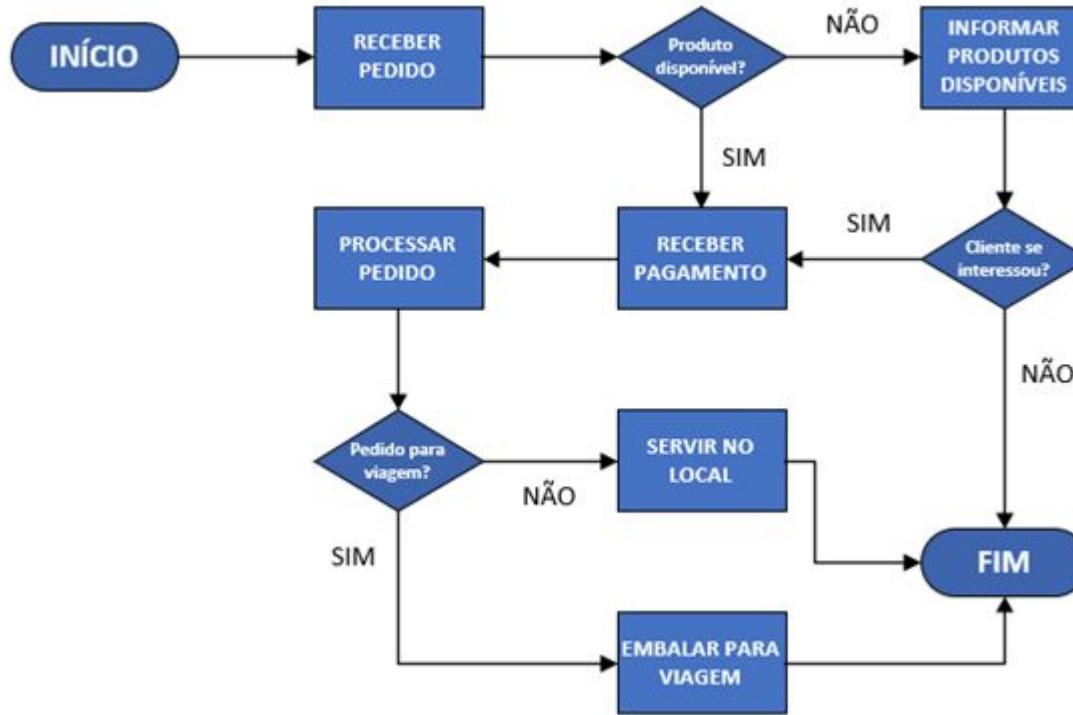
**Casos de Uso** → o que o sistema faz para cada ator.

**Sequência** → ordem das interações.

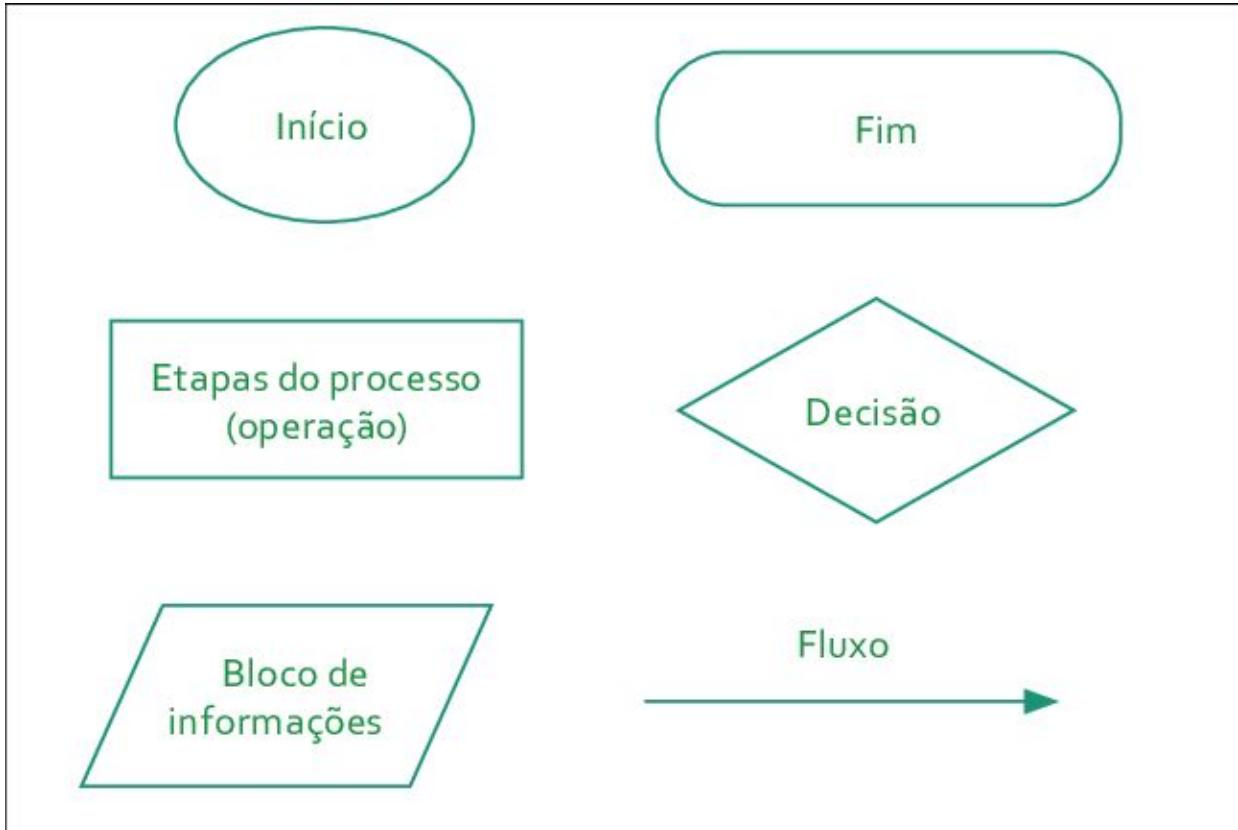
**Estado** → mudanças de status de um objeto.

**Componentes** → como o sistema é dividido em partes.

**Arquitetura** → visão geral do sistema.



## Modelagem de Processo - Fluxograma



## Modelagem de Processo - Fluxograma