

# Análise e Modelagem de Sistemas

## Métodos orientados a objetos

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nesta webaula vamos conhecer os principais diagramas da UML (*Unified Modeling Language*).

## UML (Linguagem Unificada de Modelagem)

Com a introdução do paradigma orientado a objeto, surgiu a necessidade de métodos específicos voltados para análise e projetos orientados a objetos (A/POO).

A UML (Linguagem Unificada de Modelagem ou *Unified Modeling Language*) foi a linguagem que unificou os métodos para modelagem de software orientado a objetos. Essa linguagem permitiu aos desenvolvedores e engenheiros de software um padrão único para elaborar a modelagem de programas e projetos de desenvolvimento.

## Diagramas

A UML faz uso de uma linguagem gráfica, o que nos permite visualizar com mais facilidade os objetos e suas interações (relacionamentos), bem como construir, especificar e documentar os artefatos gerados por um software.

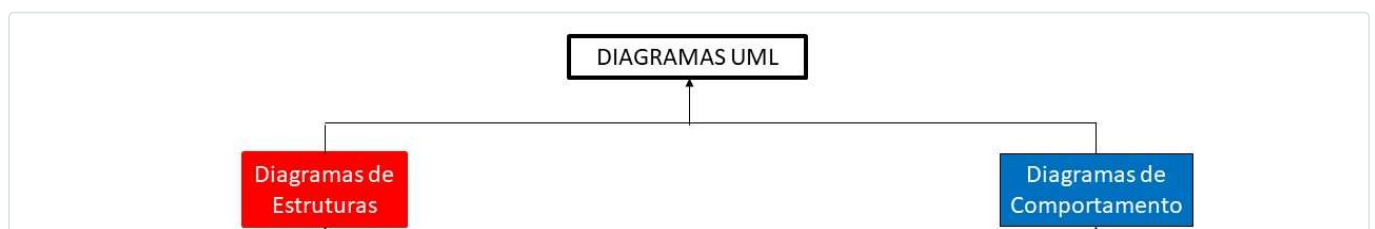
A versão 2.5 apresenta 14 diagramas subdivididos em duas categorias: **diagramas de estrutura** e **diagramas de comportamento**.

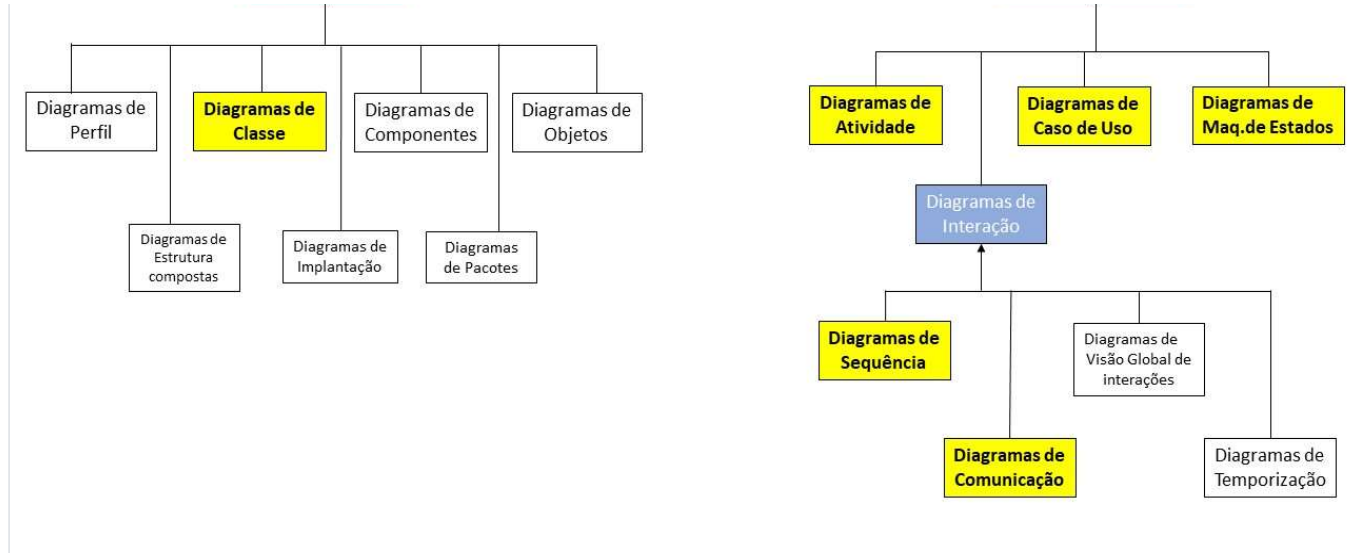
### Diagramas de estrutura

Representam as estruturas estáticas do sistema por meio de objetos, relações e atributos. Eles proporcionam uma visão da arquitetura e os aspectos funcionais globais do sistema, além de conceitos de implementação. Seu foco não são os detalhes nem o tempo, pois seu objetivo é mostrar como os objetos se relacionam.

### Diagramas de comportamento

Representam os aspectos dinâmicos do sistema, que são as mudanças que ocorrem no sistema com o passar do tempo, por meio da comunicação entre os objetos e das mudanças de estados internos e/ou externos no sistema. Entre os diagramas de comportamento há uma subdivisão, denominada diagrama de interação, em que encontramos quatro diagramas.





Fonte: adaptada de OMG® (2017).

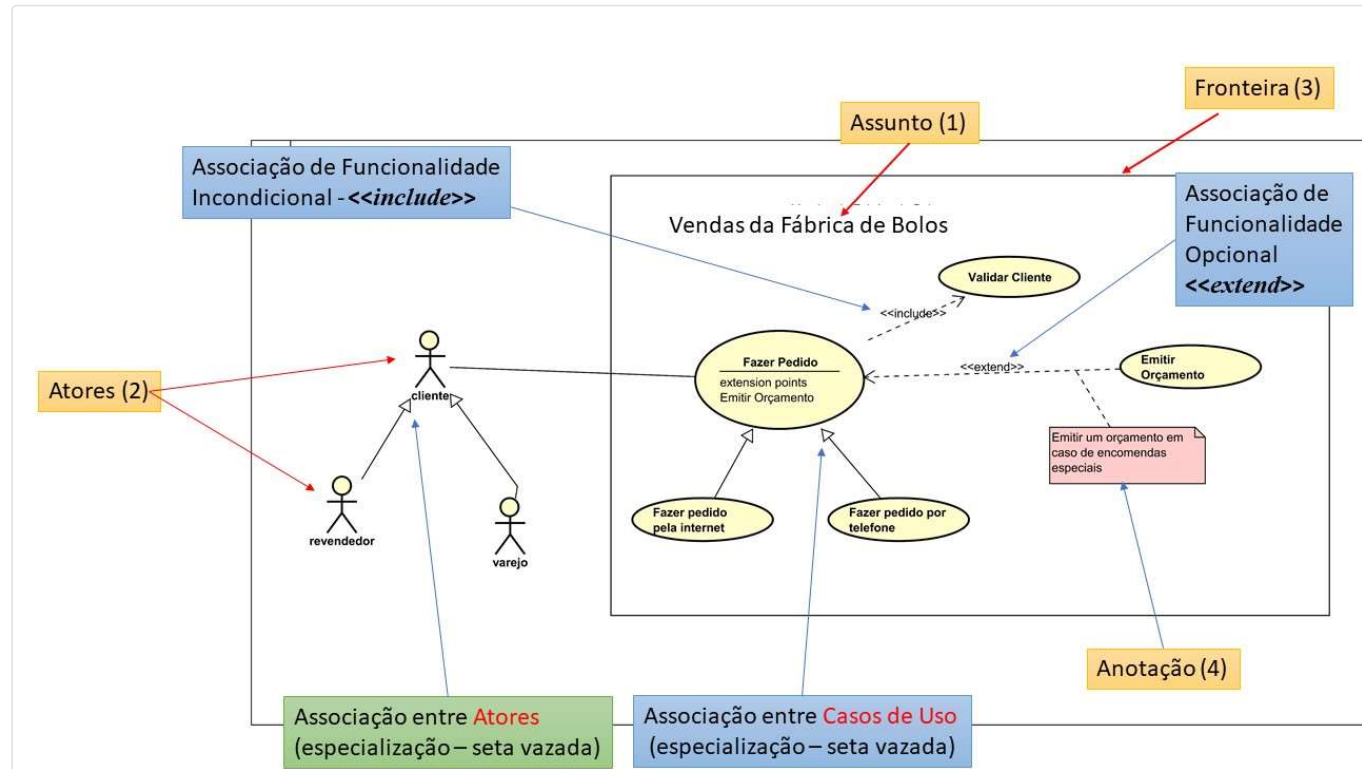
A existência de tantos diagramas tem como objetivo permitir visões múltiplas do sistema a ser modelado, como afirma Guedes (2011).

Vamos conhecer um pouco mais sobre alguns dos diagramas mais importantes da UML.

## Diagrama de caso de uso

O diagrama de caso de uso fornece uma visão geral dos objetivos que os usuários (os atores) desejam alcançar utilizando o sistema. Os elementos mais importantes são os atores, os relacionamentos e o fluxo de eventos.

### Elementos básicos do diagrama de caso de uso

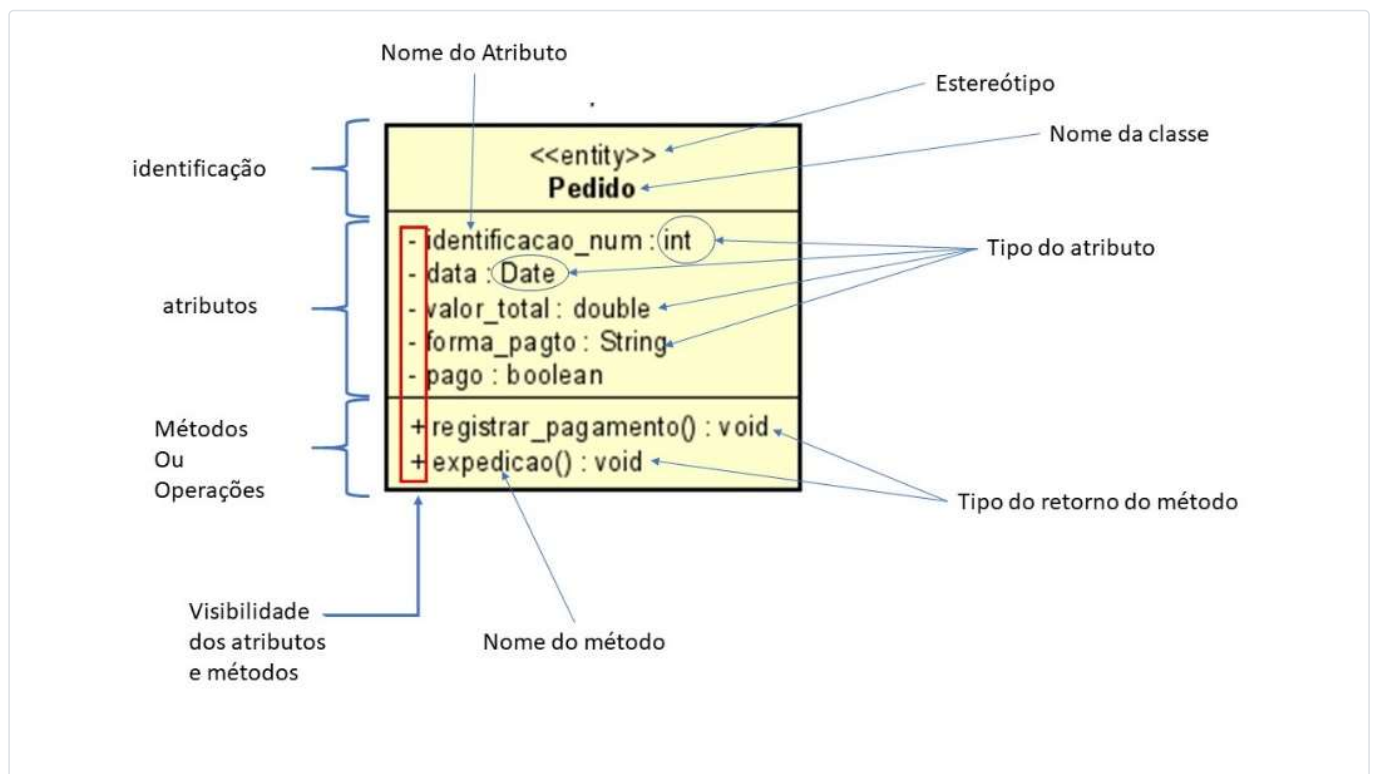


Fonte: elaborada pela autora.

## Diagrama de classe

O diagrama de classe, segundo Guedes (2011, p. 31), "[...] define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si".

A representação da classe é uma retângulo dividido em três partes: o nome da classe, os atributos e os métodos.



Fonte: elaborada pela autora.

Um mecanismo de extensibilidade da UML bastante utilizado nos diagramas de classe são os **estereótipos**. A finalidade de um estereótipo é permitir classificar elementos do diagrama que tenham algo em comum entre si. Os estereótipos podem ser definidos pelo desenvolvedor ou predefinidos. Os estereótipos predefinidos são nativos na linguagem. A UML dispõe de 50 estereótipos predefinidos, segundo a especificação OMG® (2017), mas os estereótipos predefinidos mais comuns são três:

## <<entity>>

Estereótipo identidade, que identifica classes de persistências. Essas classes armazenam dados recebidos pelo sistema.

## <<boundary>>

Estereótipo fronteira, o qual identifica uma classe de fronteira. Essas classes servem de comunicação entre atores externos e o sistema.

## <<control>>

Estereótipo de controle. Esse estereótipo geralmente é formado por classes que indicam regras de negócio.

As classes relacionam-se entre si, e os tipos de **relacionamentos** possíveis especificados na UML (OMG®, 2017) são:

## Dependência

É o tipo de relacionamento mais fraco entre duas classes, chamado de relação semântica entre duas classes, na qual uma alteração na classe independente pode afetar a classe dependente.

## Associação

Este é o tipo de relacionamento mais comum, e indica que a classe A tem uma relação com a classe B. É um relacionamento genérico.

## Agregação

É uma associação específica, em que a classe filha pode existir independentemente da classe pai.

## Composição

É uma associação específica em que, se o objeto da classe pai é destruído o outro objeto associado, não fará sentido a existência da classe filha.

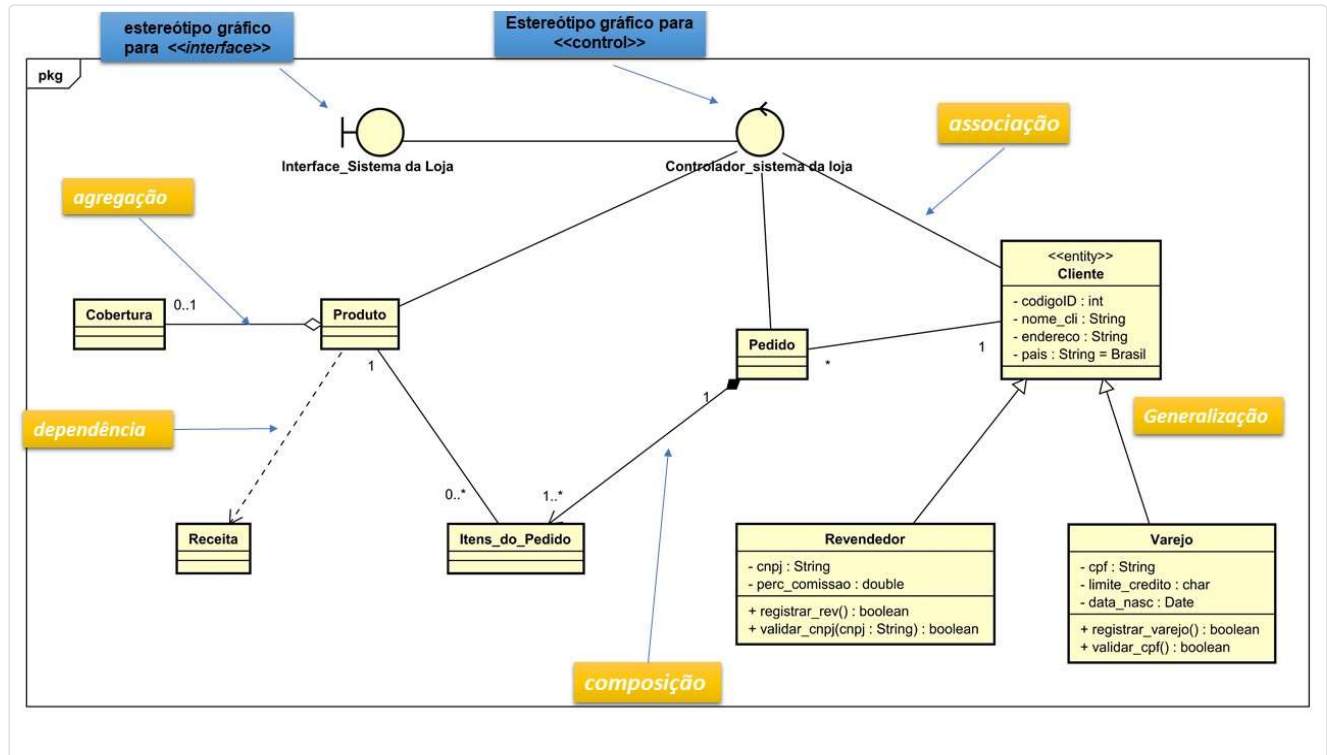
## Generalização/especialização

Indica que a classe filha herda as características da classe pai, também conhecida como especialização da classe.

## Multiplicidade

Indica quantas instâncias dos objetos estão envolvidos na associação.

Diagrama de classe representando a venda de bolos com as indicações dos tipos de relacionamentos entre as classes



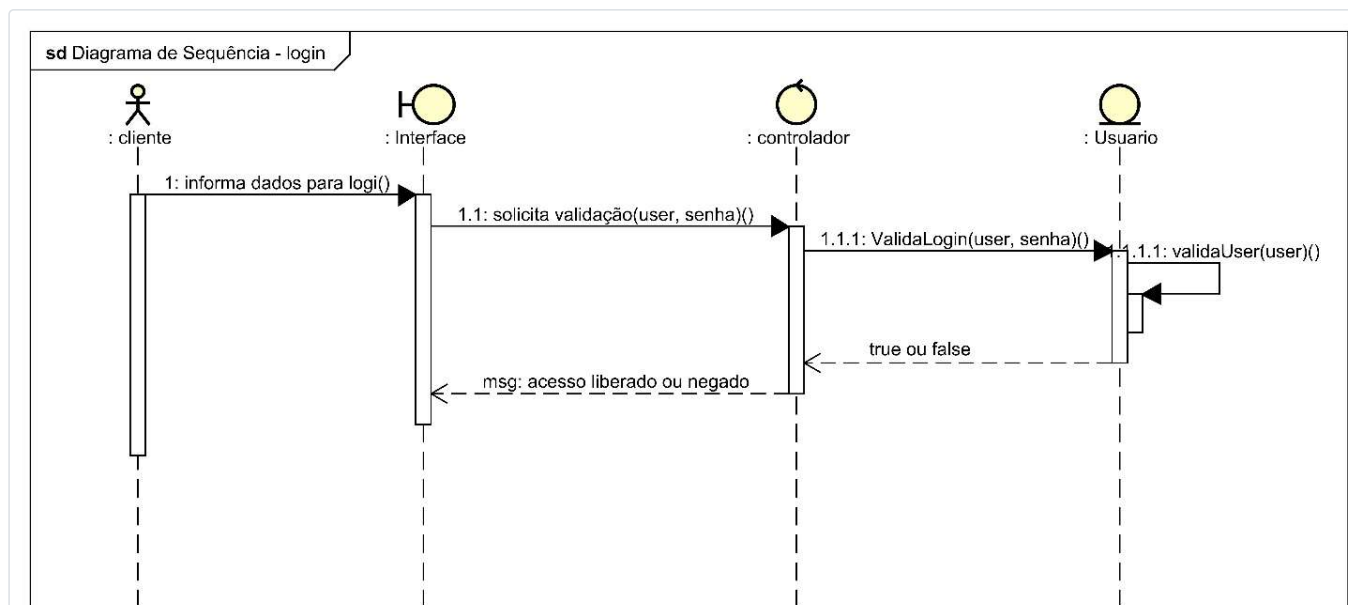
Fonte: elaborada pela autora.

[Saiba Mais](#)

## Diagrama de sequência

O diagrama de sequência mostra a interação entre os participantes do cenário ao longo da vida. Ele apresenta, como o próprio nome diz, a sequência dos passos realizados pelos objetos.

## Diagrama de sequência

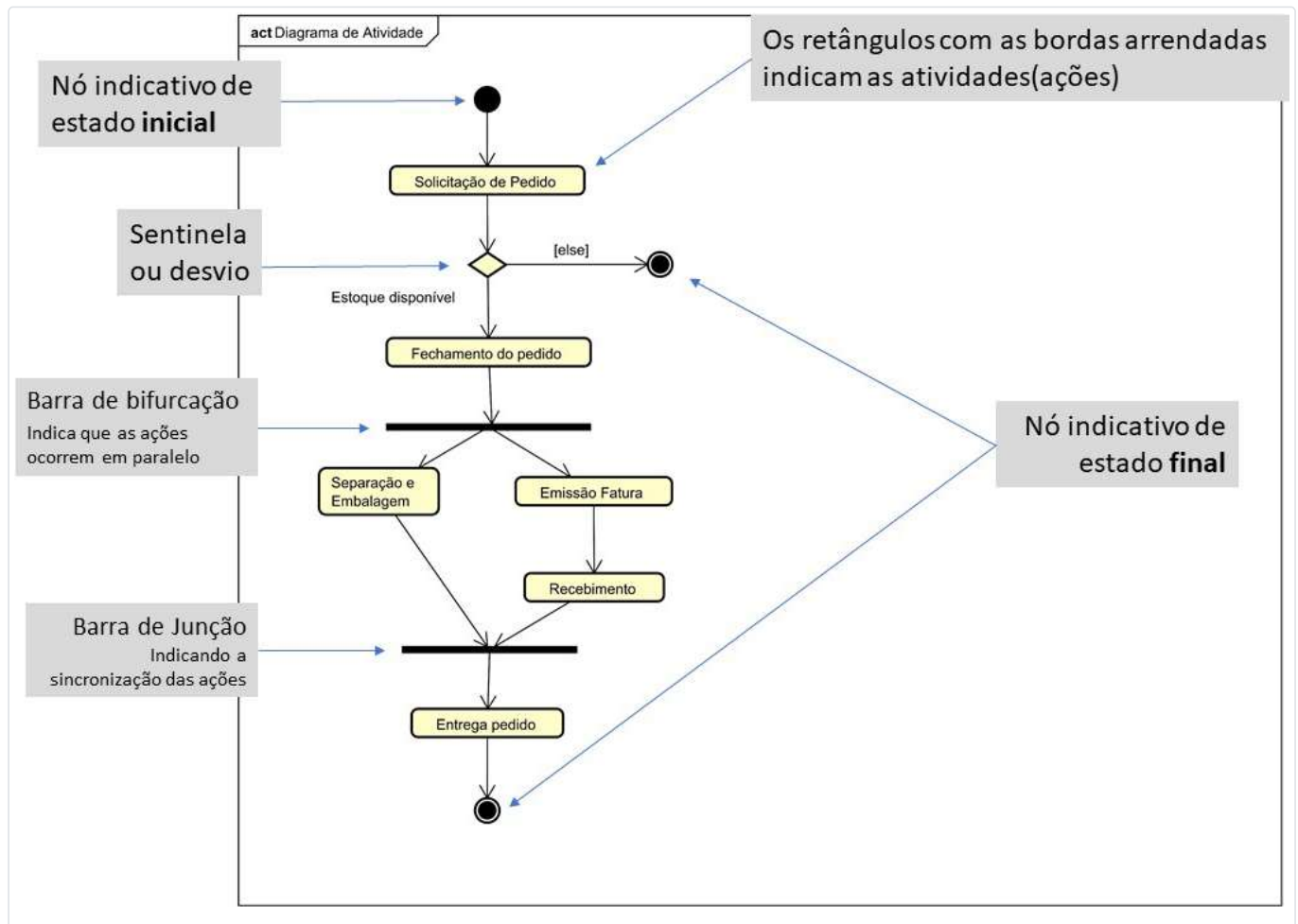


## Diagrama de atividade

O diagrama de atividade tem por objetivo descrever os passos que devem ser seguidos para a execução de uma determinada atividade. Esse diagrama assemelha-se muito com as funções de um fluxograma.

Os elementos básicos do diagrama de atividade são: ações (atividades), sentinela (desvios), estados inicial e final, barra de bifurcação e barra de junção.

### Diagrama de atividades



Fonte: elaborada pela autora.

## Diagrama de máquina de estados

Esse diagrama visa mostrar a transição de um estado a outro dos objetos do sistema. É muito semelhante ao diagrama de atividade.

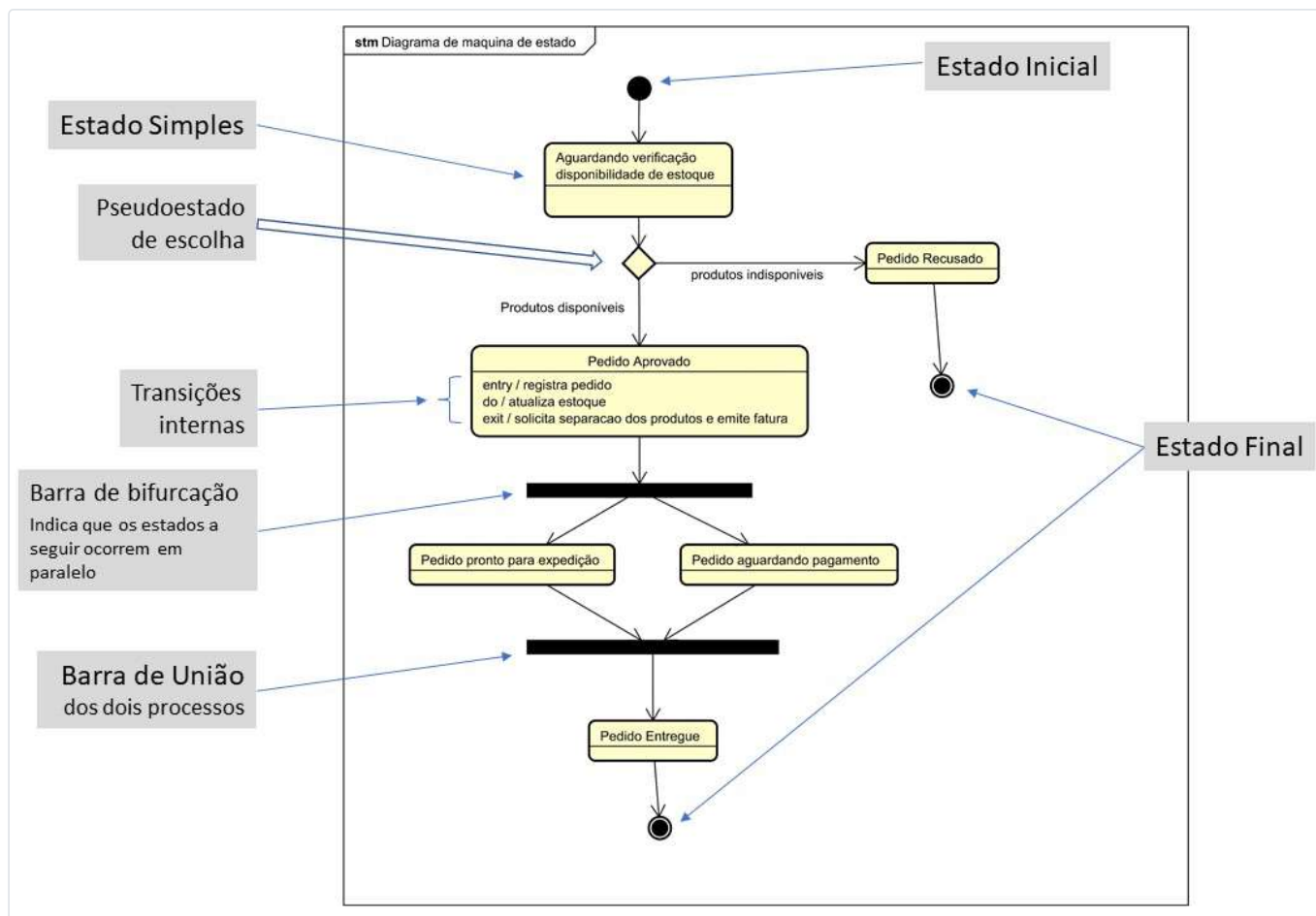
As transições internas de estado são três:

**Entry** - Indica ações internas do estado executada pelo objeto ao assumir um estado.

**Do** - São ações internas do estado executadas durante o período que o objeto se encontra em um estado.

**Exit** - Neste caso, são ações executadas pelo objeto quando ele sai de um estado.

### Diagrama de máquina de estado



Fonte: elaborada pela autora.

Vimos os principais diagramas propostos pela UML, que trouxeram para engenharia de software o instrumental para modelar os programas e projetos de acordo com o paradigma orientado a objeto.

### Pesquise mais!

Para saber mais sobre UML, consulte o livro *UML Essencial*, de Fowler (disponível na Biblioteca Virtual), um dos desenvolvedores que assinou o Manifesto Ágil.

Sugerimos os seguintes excertos:

Capítulo 9, sobre diagrama de caso de uso.

Capítulo 3, sobre diagrama de classe, em especial sobre as associações.

Capítulo 4, sobre diagrama de sequência.

FOWLER, M. **UML Essencial**: um breve guia para a linguagem de modelagem de objetos. Tradução: João Tortello. 3. ed. Porto Alegre, Bookman, 2005.



Para visualizar o vídeo, acesse seu material digital.