Diagrama de Classe

Definições

- **Classe**:
  - Representa um conjunto de objetos de mesmo tipo; Ex: Classe = {obj1, obj2, obj3, …, objN}
- **Objeto**:
  - Corresponde a cada instância derivada da classe; É um elemento do conjunto representado pela classe Entidade que descreve uma realidade.

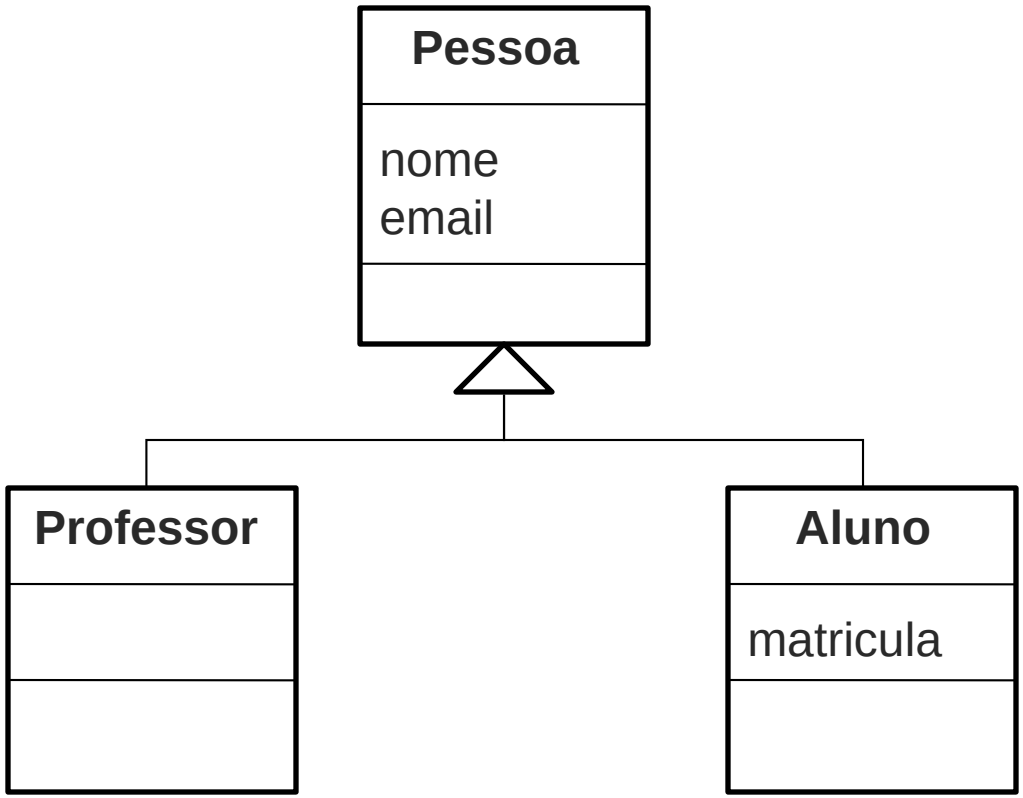
Características de Objetos

- Objetos possuem **atributos**
  - Tamanho, forma, cor, peso, etc.
- Objetos exibem **comportamentos (métodos)**
  - Uma bola rola, um avião voa
  - Uma pessoa anda, fala, pensa, etc.

Diagrama de Classes

- O mais importante e o mais utilizado diagrama da UML
- Permite a visualização das classes que compõem o sistema
- Representa
  - Atributos e métodos de uma classe
  - Os relacionamento entre classes.

Meu Primeiro Diagrama



Outro Diagrama de Classes

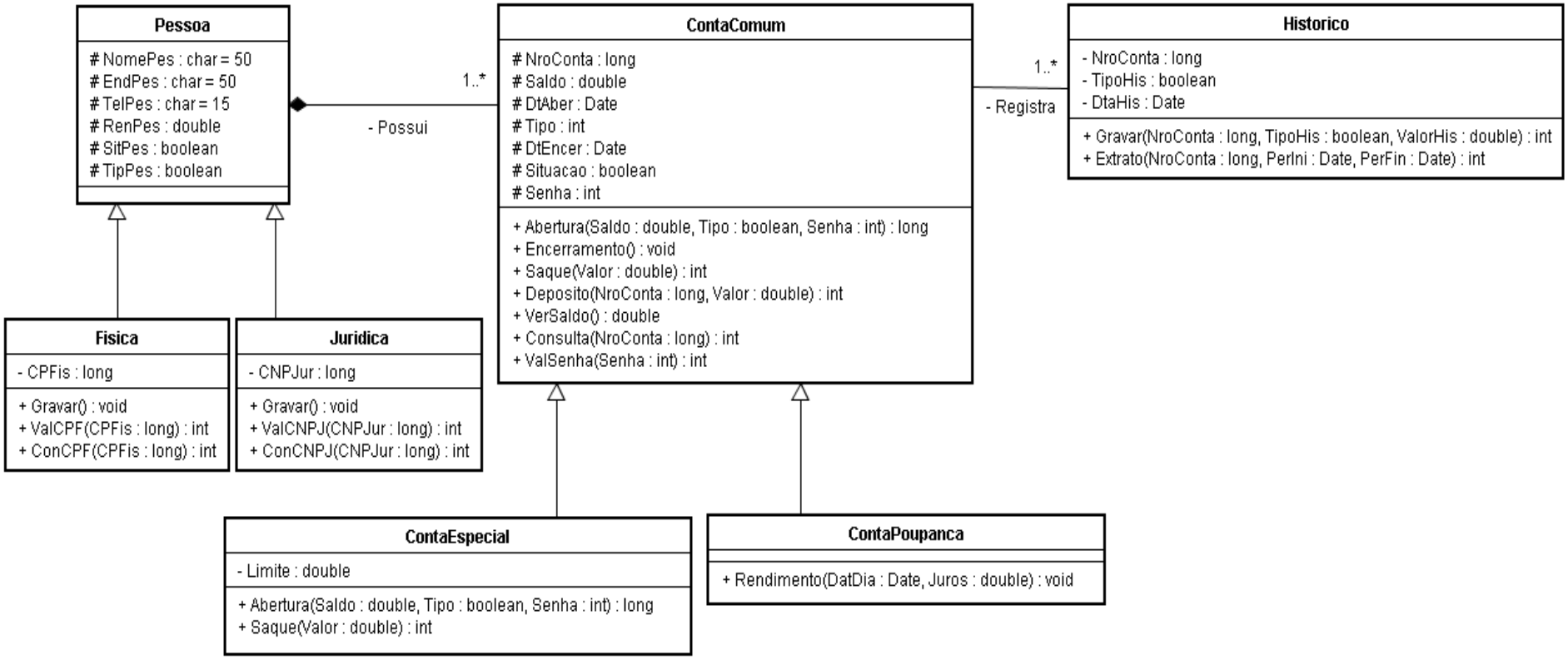


Diagrama de Classes

- Apresenta uma visão estática de como as classes estão organizadas
- Preocupação com a estrutura lógica

Atributos

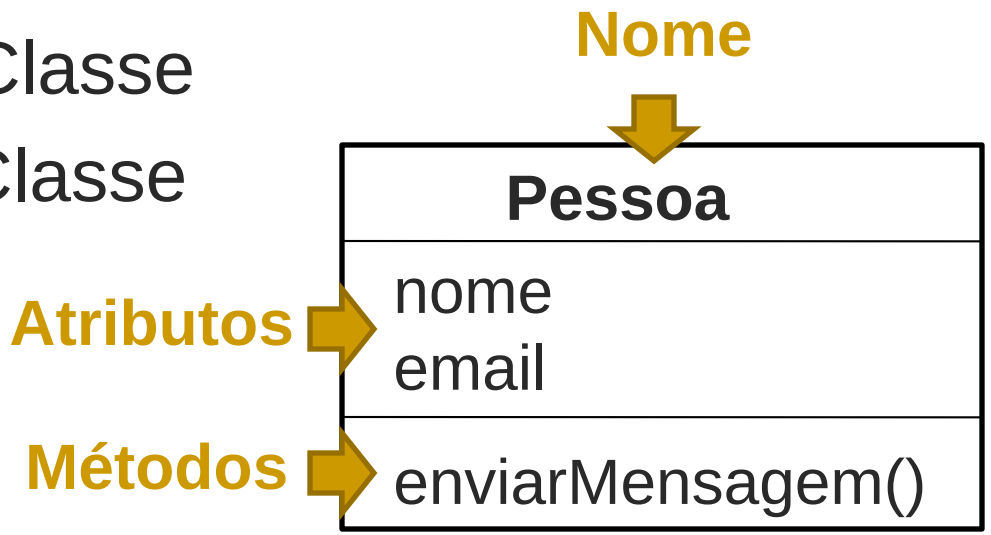
- Permite a identificação de cada objeto de uma classe
- Os valores dos atributos podem variar de instância para instância
- Atributos devem conter o tipo de dados a ser armazenado
  - Byte, boolean, int, double, char, String, etc.

Métodos

- São apenas declarados neste diagrama
  - Diagrama de Classes não define a implementação
- Correspondem a comportamentos ou operações dos objetos.

Representação de uma Classe

- Uma classe é representada por um retângulo com três divisões:
  - Nome da Classe
  - Atributos da Classe
  - Métodos da Classe

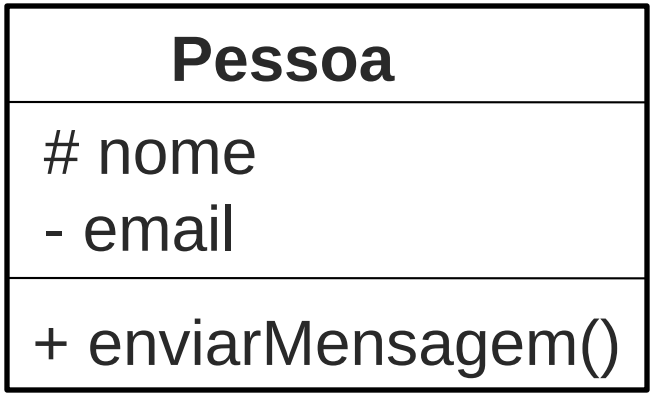


Tipos de visibilidade

- Pública (+)
  - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
  - Somente a classe ou sub-classes terão acesso
- Privada (-)
  - Somente a classe terá acesso

Tipos de visibilidade

- Pública (+)
  - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
  - Somente a classe ou sub-classes terão acesso
- Privada (-)
  - Somente a classe terá acesso



Relacionamento

- Classes possuem relacionamentos entre elas
  - Compartilham informações
  - Colaboram umas com as outras
- Principais tipos de relacionamentos
  - Associação
  - Agregação / Composição
  - Herança
  - Dependência

Comunicação entre Objetos (I)

- Conceitualmente, objetos se comunicam através da troca de mensagens.
- Mensagens definem:
  - O nome do serviço requisitado
  - A informação necessária para a execução do serviço
  - O nome do requisitante.

Comunicação entre Objetos (II)

- Na prática, mensagens são implementadas como chamadas de métodos
  - Nome = o nome do método
  - Informação = a lista de parâmetros
  - Requisitante = o método que realizou a chamada

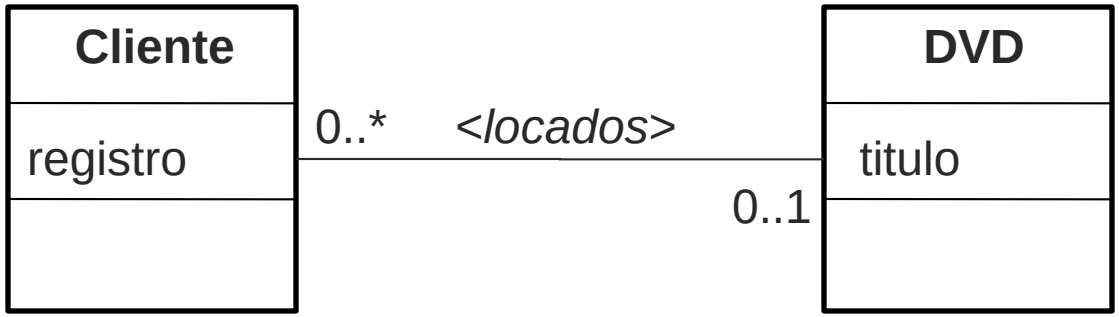
Associações

- Descreve um vínculo entre duas classes
  - Chamado **Associação Binária**
- Determina que as instâncias de uma classe estão de alguma forma ligadas às instâncias da outra classe

Multiplicidade

0..1	No máximo um. Indica que os Objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	Muitos. Indica que podem haver muitos objetos da classe envolvidos no relacionamento
1..*	Um ou muitos. Indica que há pelo menos um objeto envolvido no relacionamento.
3..5	Valores específicos.

Representação de Associação



Agregação

- Tipo especial de associação
- Demonstra que as informações e um objeto precisam ser complementadas por um objeto de outra classe
- Associação Todo-Parte
  - objeto-todo
  - objeto-parte

Representação de Agregação

- Um losango na extremidade da classe que contém os *objetos-todo*

Composição

- Uma variação do tipo agregação
- Representa um vínculo mais forte entre objetos-todo e objetos-parte
- Objetos-parte **têm** que pertencer ao objeto-todo
  - O todo não existe (ou não faz sentido) sem a parte

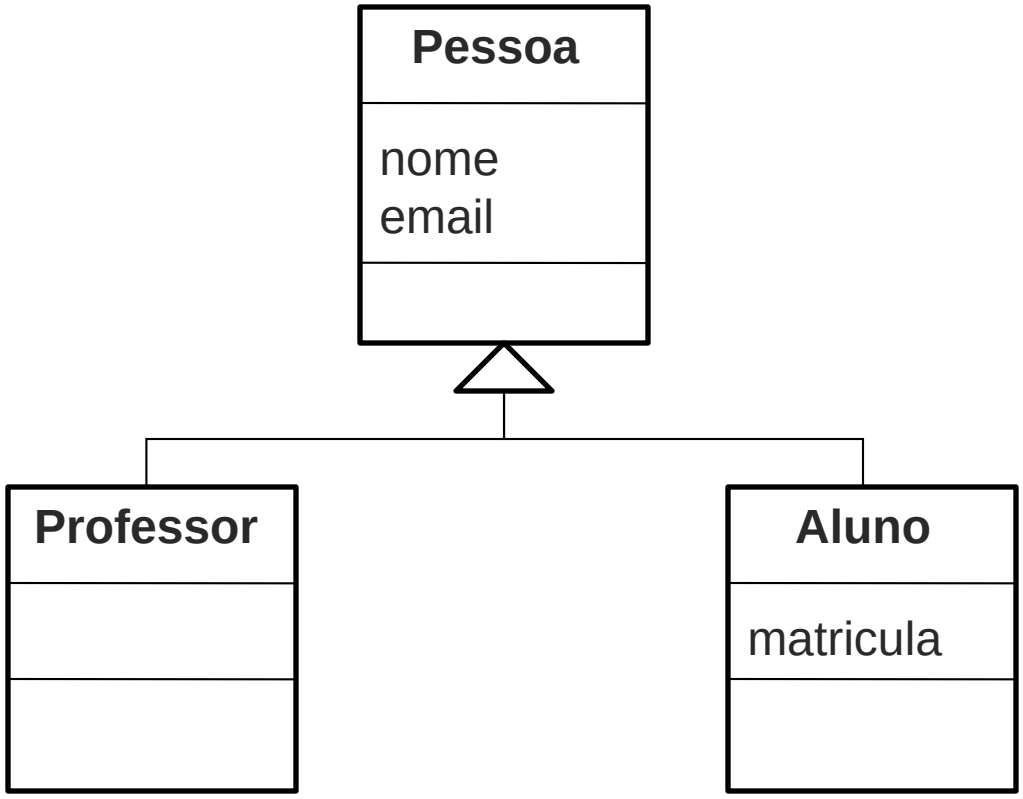
Representação da Composição

- Um losango preenchido, e da mesma forma que na Agregação, deve ficar ao lado do objeto-todo

Especialização / Generalização

- Identificar classes-mãe (gerais) e classes-filhas (especializadas)
- Atributos e métodos definidos na classe-mãe são **herdados** pelas classes-filhas

Especialização / Generalização

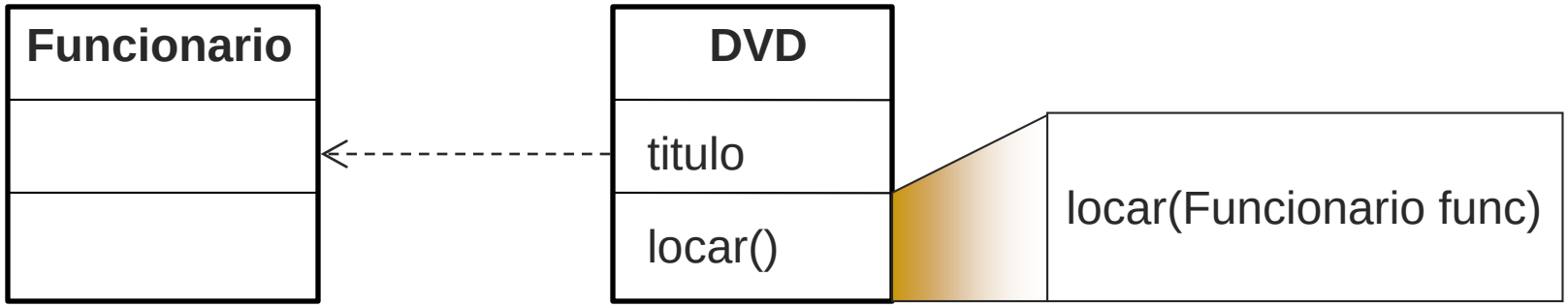


Dependência

- Tipo menos comum de relacionamento
- Identifica um baixo grau de dependência de uma classe em relação a outra

Dependência

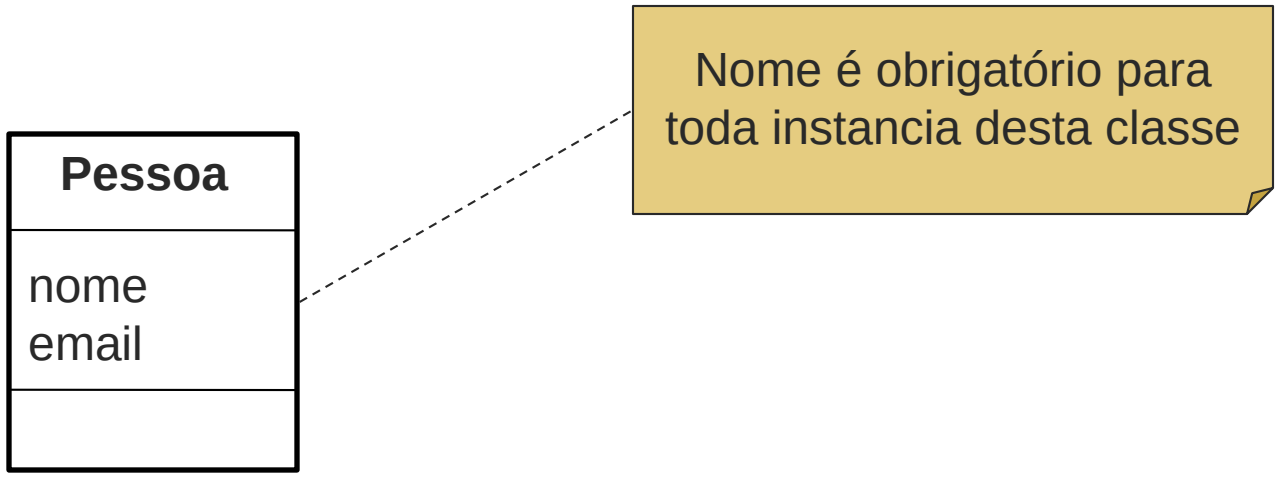
- Representado por uma reta tracejada entre duas classes
- Uma seta na extremidade indica o dependente



Notas

- Informativos
  - Algum comentário na classe, método ou atributo
  - Alguma restrição de funcionalidade
- Objetivo é informa como o objeto se comporta

Notas


Pessoa
nome
email
Nome é obrigatório para toda instancia desta classe

Referências

- DEITEL, H. M.; DEITEL P. J. **Java: Como Programar**, 6a. Edição. Pearson, 2005.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.

Atividade

Cenário:

Vamos supor que estamos desenvolvendo um sistema para uma biblioteca. Esse sistema deve controlar livros, clientes, empréstimos e funcionários. O sistema precisa gerenciar o cadastro dos livros, o empréstimo de livros pelos clientes e o registro de funcionários responsáveis por controlar as transações de empréstimos.

Passo 1: Identificação das Classes

Analisando o cenário, podemos identificar algumas classes principais:

1. **Livro**: Representa os livros disponíveis na biblioteca.
2. **Cliente**: Representa os usuários da biblioteca que podem pegar livros emprestados.
3. **Empréstimo**: Representa a ação de um cliente pegar um livro emprestado.
4. **Funcionário**: Representa os funcionários que realizam o controle dos empréstimos.

Atividade

Passo 2: Definir Atributos e Métodos das Classes

1. **Classe Livro**:
    - Atributos:
        - titulo: String
        - autor: String
        - isbn: String
        - disponivel: Boolean (indica se o livro está disponível para empréstimo)
    - **Métodos:**
        - emprestar(): Método para marcar o livro como emprestado.
        - devolver(): Método para marcar o livro como devolvido

Atividade

Passo 2: Definir Atributos e Métodos das Classes

Classe Cliente:

- Atributos:
  - nome: String
  - id: Integer
  - telefone: String
  - endereco: String

- Métodos:
  - realizarEmprestimo(): Método para o cliente pegar um livro emprestado.
  - devolverLivro(): Método para o cliente devolver um livro.

Atividade

Passo 2: Definir Atributos e Métodos das Classes

Classe Empréstimo:

- Atributos:
  - dataEmprestimo: Date
  - dataDevolucao: Date
  - livro: Livro (Associação com a classe Livro)
  - cliente: Cliente (Associação com a classe Cliente)

- Métodos:
  - calcularMulta(): Método para calcular multa caso o livro não seja devolvido no prazo.
  - finalizarEmprestimo(): Método para finalizar o empréstimo (realizar a devolução).

Atividade

Passo 2: Definir Atributos e Métodos das Classes

Classe Funcionário:

- Atributos:
  - nome: String
  - matricula: Integer
  - cargo: String
- Métodos:
  - registrarEmprestimo(): Método para registrar o empréstimo de um livro.
  - registrarDevolucao(): Método para registrar a devolução de um livro.

Atividade

Passo 3: Identificar Relacionamentos

•**Cliente** *tem* muitos **Empréstimos**. Ou seja, um cliente pode fazer vários empréstimos.

•**Empréstimo** *está relacionado* a um único **Livro** e a um único **Cliente**. Ou seja, um empréstimo está vinculado a um livro e a um cliente específicos.

•**Funcionário** *realiza* os **Empréstimos**. Ou seja, o funcionário registra e controla os empréstimos dos livros.

Atividade

Passo 5: Explicação dos Relacionamentos e Visibilidade

1.Relacionamento entre Cliente e Empréstimo:

1. A relação é de **um para muitos** (1:N). Ou seja, um cliente pode ter vários empréstimos, mas um empréstimo pertence a apenas um cliente.

2.Relacionamento entre Livro e Empréstimo:

1. A relação é de **um para muitos** (1:N). Um livro pode estar presente em vários empréstimos (em diferentes momentos), mas um empréstimo está vinculado a um único livro.

3.Relacionamento entre Funcionário e Empréstimo:

1. A relação é de **um para muitos** (1:N), ou seja, um funcionário pode registrar vários empréstimos, mas um empréstimo é registrado por um único funcionário.