

# **Aula 6**

## **Notação Básica dos Diagramas de Comunicação**

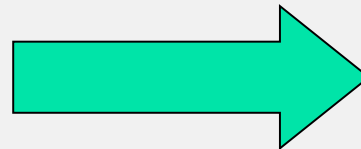
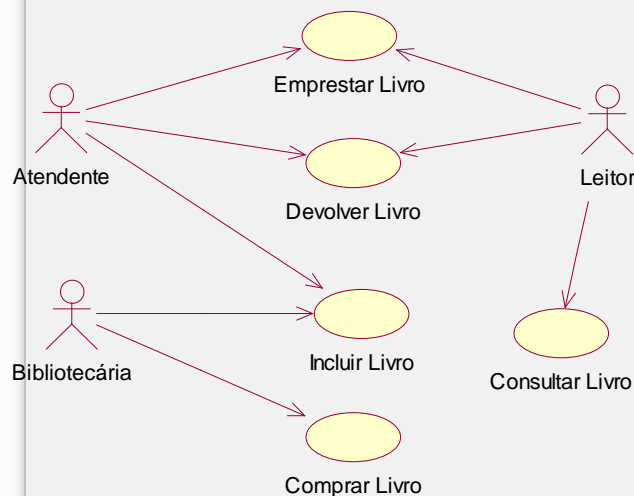
**Responsável**

Prof. Seiji Isotani ([sisotani@icmc.usp.br](mailto:sisotani@icmc.usp.br))

# O que já foi visto até agora

## Casos de Uso Completo Abstrato

### Diagrama de Casos de Uso



#### Caso de Uso: Emprestar Livro

**Ator Principal:** Atendente

**Interessados e Interesses:**

- Atendente: deseja registrar que um ou mais livros estão em posse de um leitor, para controlar se a devolução será feita no tempo determinado.
- Leitor: deseja emprestar um ou mais livros, de forma rápida e segura.
- Bibliotecário: deseja controlar o uso dos livros, para que não se percam e para que sempre se saiba com que leitor estão no momento.

**Pré-Condições:** O Atendente é identificado e autenticado.

**Garantia de Sucesso (Pós-Condições):** Os dados do novo empréstimo estão armazenados no Sistema. Os livros emprestados possuem status "emprestado"

**Cenário de Sucesso Principal:**

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para realizar um novo empréstimo.
3. O Atendente solicita ao leitor sua carteira de identificação, seja de estudante ou professor.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.
9. Se necessário, o Atendente desbloqueia os livros para que possam sair da biblioteca.
10. O Leitor sai com os livros.

**Fluxos Alternativos:**

- (1-8). A qualquer momento o Leitor informa ao Atendente que desistiu do empréstimo.
3. O Leitor informa ao Atendente que esqueceu a carteira de identificação.
  1. O Atendente faz uma busca pelo cadastro do Leitor e pede a ele alguma informação pessoal para garantir que ele é mesmo quem diz ser.
4. O Leitor está impedido de fazer empréstimo, por ter não estar apto.
  1. Cancelar a operação.
- 7a. O Livro não pode ser emprestado, pois está reservado para outro leitor.
  1. O Atendente informa ao Leitor que não poderá emprestar o livro e pergunta se deseja reservá-lo.
  2. Cancelar a operação (se for o único livro)
- 7b. O Livro não pode ser emprestado, pois é um livro reservado somente para consulta.
  1. Cancelar a operação (se for o único livro)

# O que já foi visto até agora

Casos de Uso com substantivos e verbos sublinhados

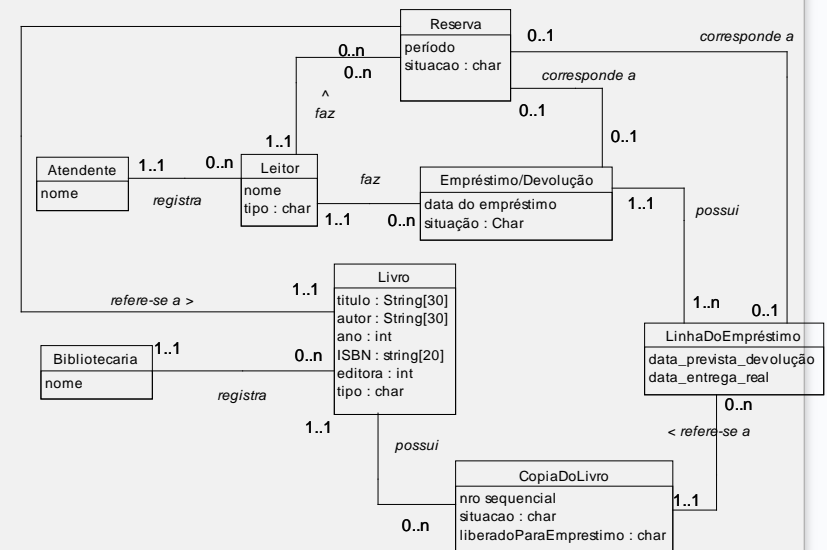
## Caso de Uso 1

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para adicionar um novo empréstimo.
3. O Atendente solicita ao leitor sua carteirinha, seja de estudante ou professor.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.
9. O Atendente desbloqueia os livros para que possam sair da biblioteca.
10. O Leitor sai com os livros.

## Caso de Uso n

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para adicionar um novo empréstimo.
3. O Atendente solicita ao leitor sua carteirinha, seja de estudante ou professor.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.
9. O Atendente desbloqueia os livros para que possam sair da biblioteca.
10. O Leitor sai com os livros.

## Modelo Conceitual



# O que já foi visto até agora

Modelo Conceitual  
+  
Casos de Uso

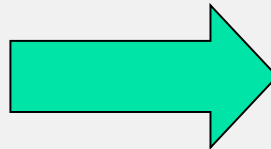
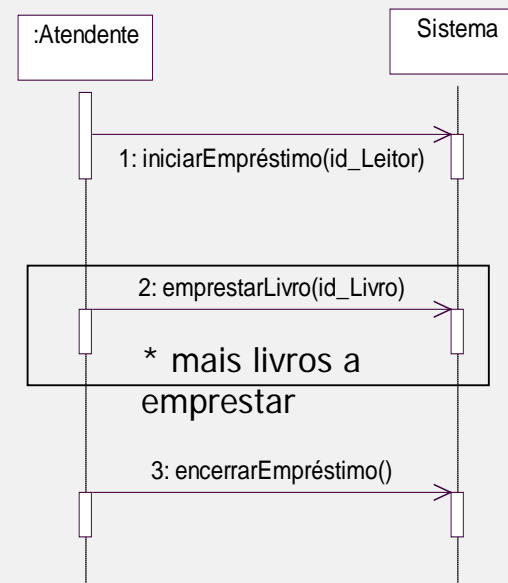
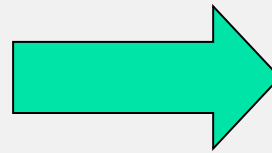
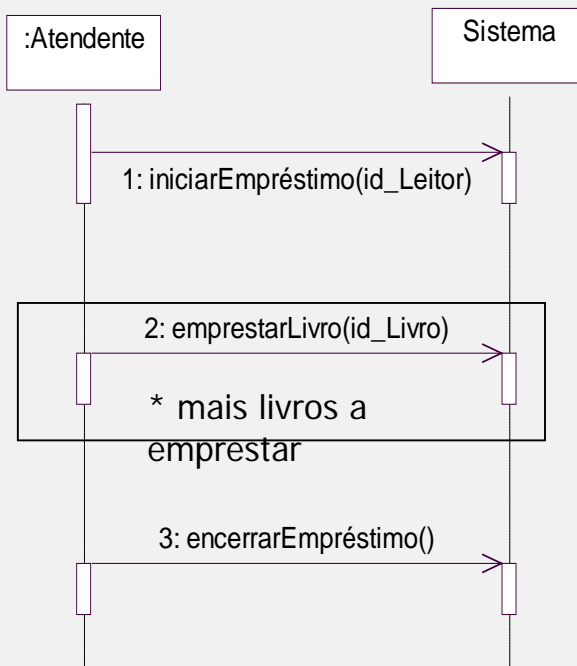


Diagrama de Seqüência do  
Sistema  
(para cada caso de uso)



# O que já foi visto até agora

## Diagrama de Seqüência do Sistema (para cada caso de uso)



## Contrato da Operação (para cada operação)

**Operação:** encerrarEmpréstimo()

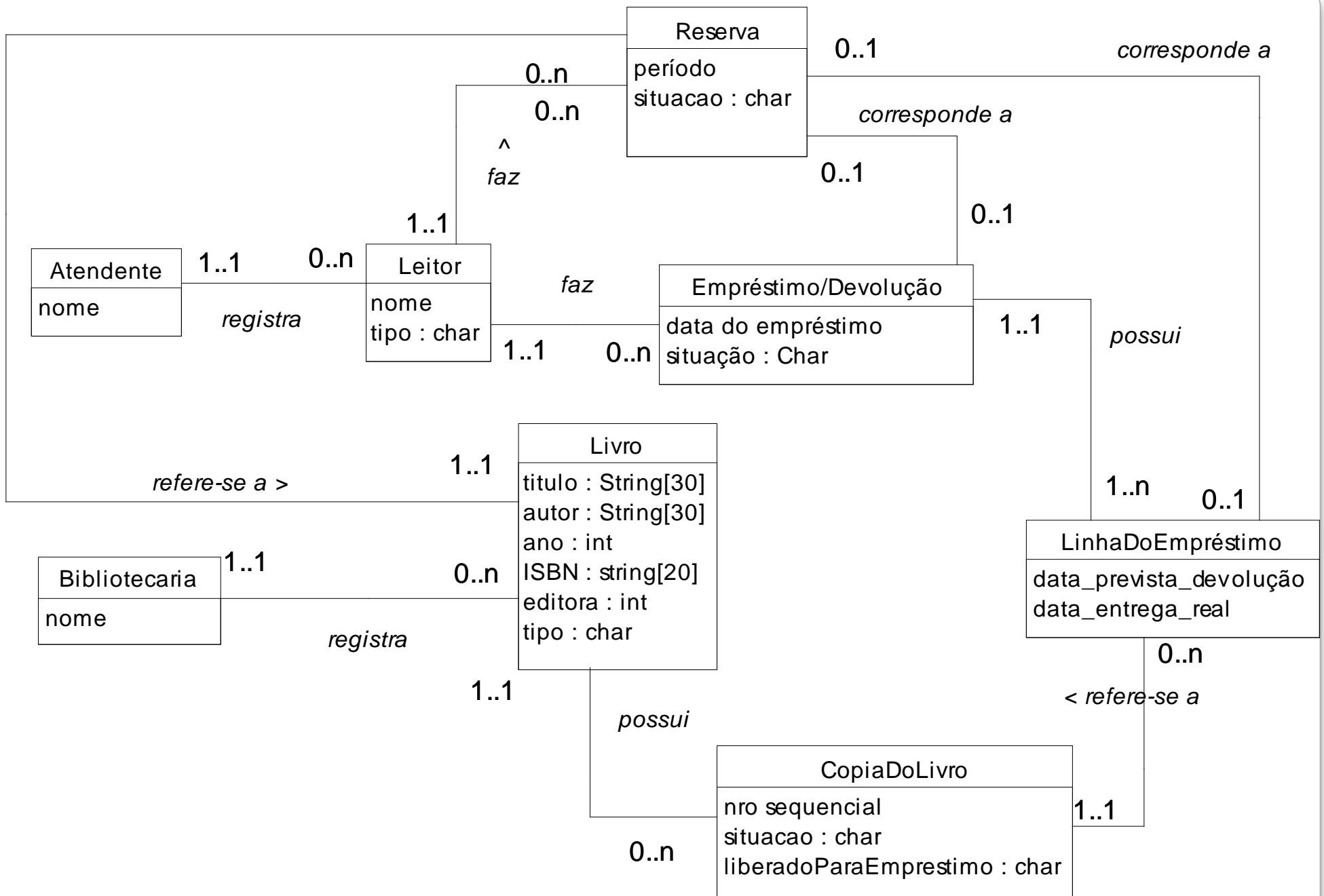
**Referências Cruzadas:** Caso de uso: “Emprestar Livro”

**Pré-Condições:** Um leitor apto a emprestar livros já foi identificado; pelo menos um livro já foi identificado e está disponível para ser emprestado.

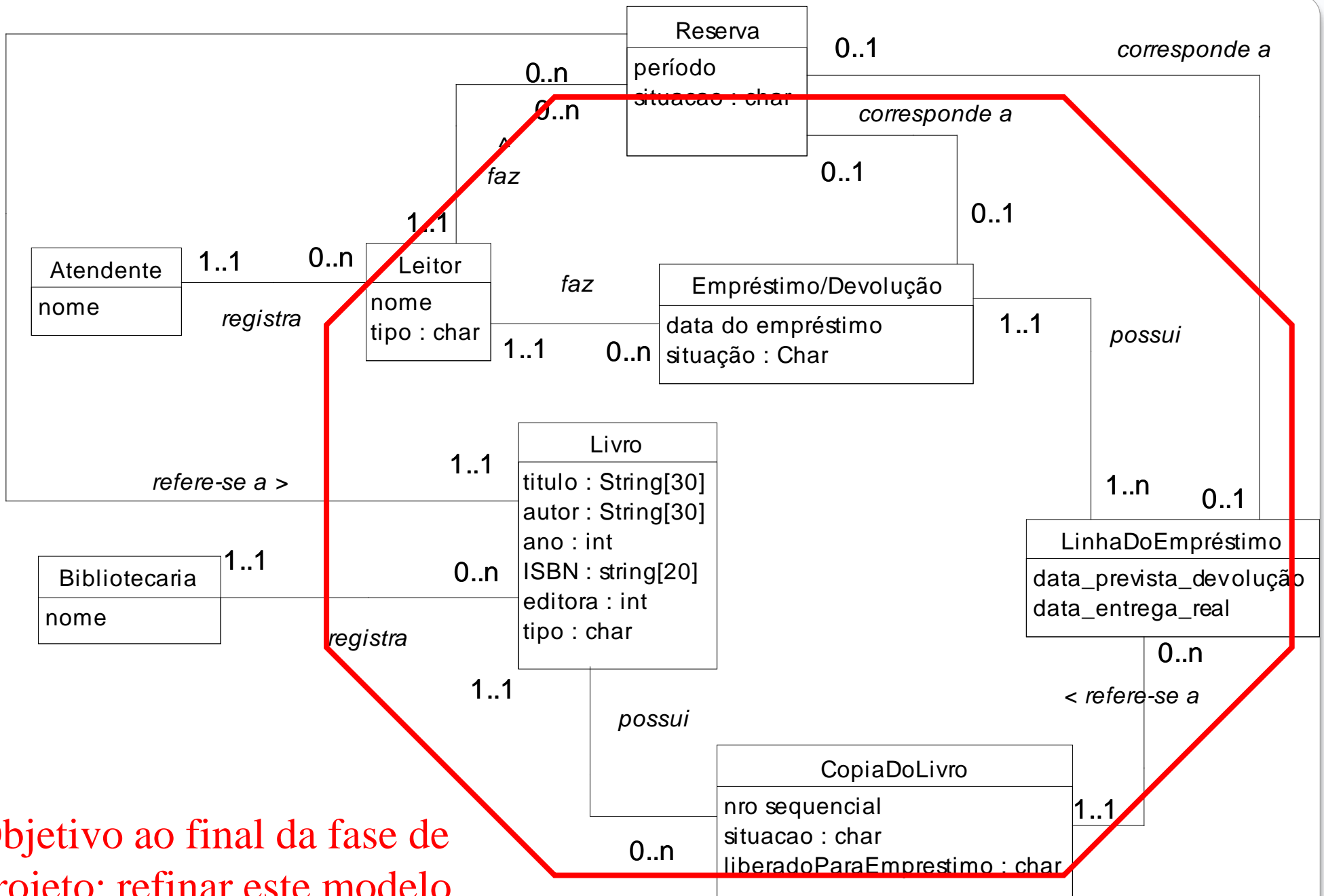
**Pós-Condições:** um novo empréstimo foi registrado; o novo empréstimo foi relacionado ao leitor já identificado na operação “iniciar o empréstimo”; a situação dos livros emprestados foi alterada para “emprestado”.

# Projeto Orientado a Objetos

- Finalizada a análise, inicia-se o **Projeto** do sistema
- Artefatos de projeto contém detalhes de **como** o sistema poderá ser implementado utilizando um computador.
  - detalhar as informações sobre as classes que comporão o sistema → diagrama de classes
  - comportamento esperado de cada objeto e a colaboração entre os objetos → diagramas de interação

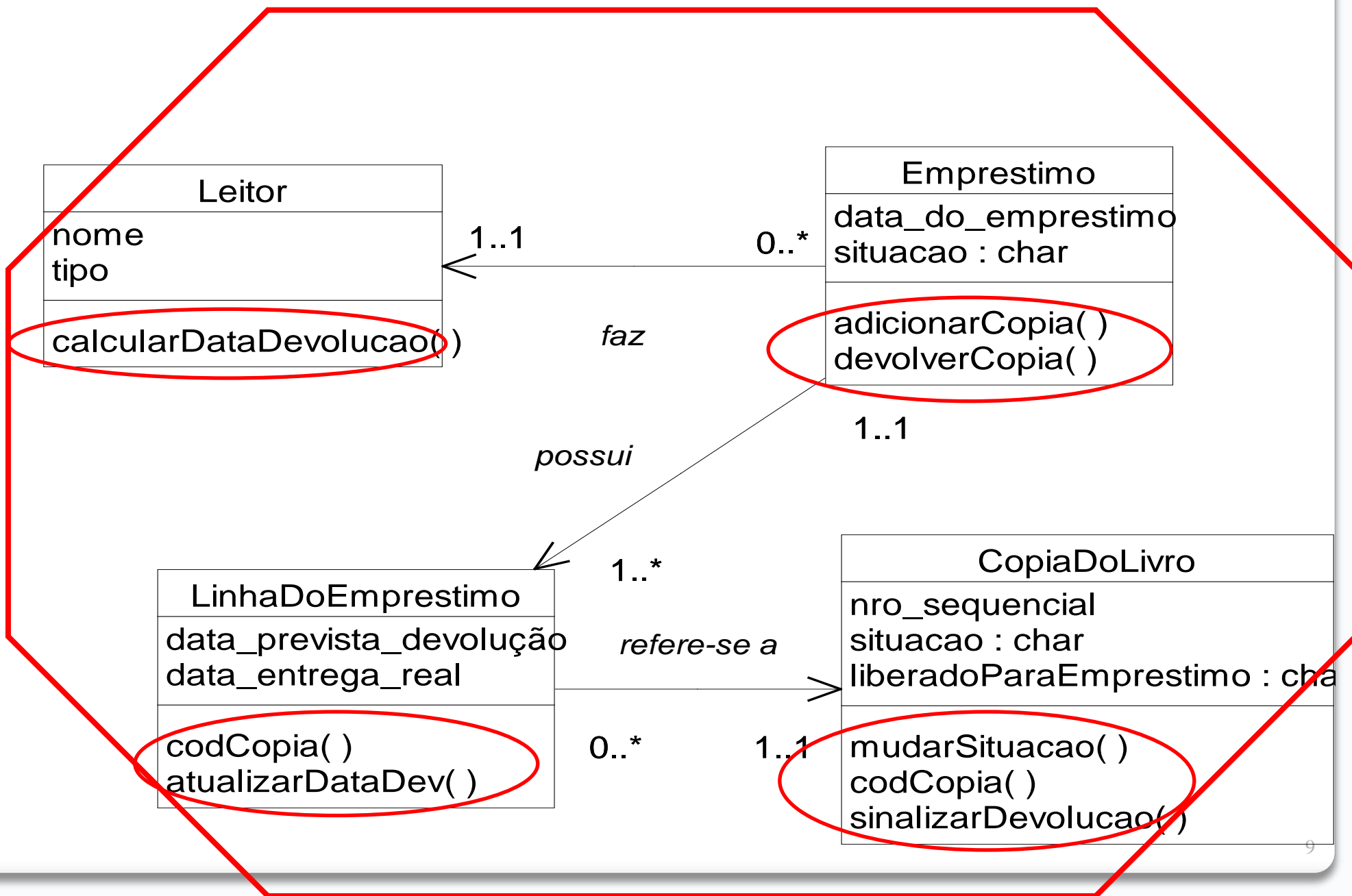


Modelo Conceitual produzido na análise



Objetivo ao final da fase de projeto: refinar este modelo e incluir métodos (interface)





# Como projetar as responsabilidades de cada objeto?

- Sabemos que os objetos precisam se comunicar
- Os Diagramas de comunicação mostram escolhas de atribuição de responsabilidade a objetos
- Mas quem é o melhor candidato para realizar/implementar/materializar cada uma das operações e dos métodos do sistema?

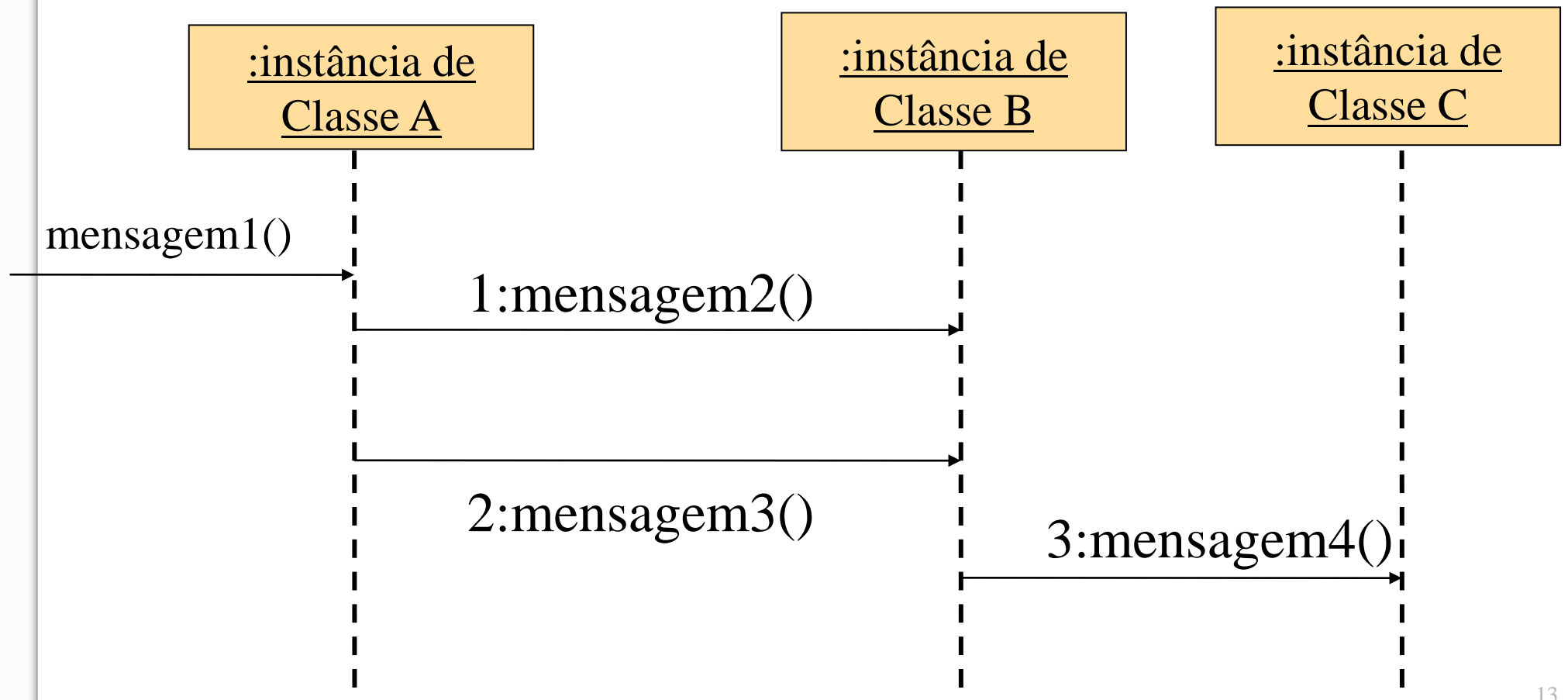
# Como projetar as responsabilidades de cada objeto?

- Responsabilidade:
  - um contrato ou obrigação de um tipo ou classe
  - serviços fornecidos por um elemento (classe ou subsistema)
- Dois tipos de responsabilidades básicas:
  - Fazer
    - fazer algo (criar um objeto, executar uma operação,...)
    - iniciar ações em outros objetos (delegação)
    - coordenar e controlar atividades em outros objetos
  - Saber
    - conhecer dados privados encapsulados
    - conhecer objetos relacionados
    - conhecer dados/atributos que podem ser derivados ou calculados

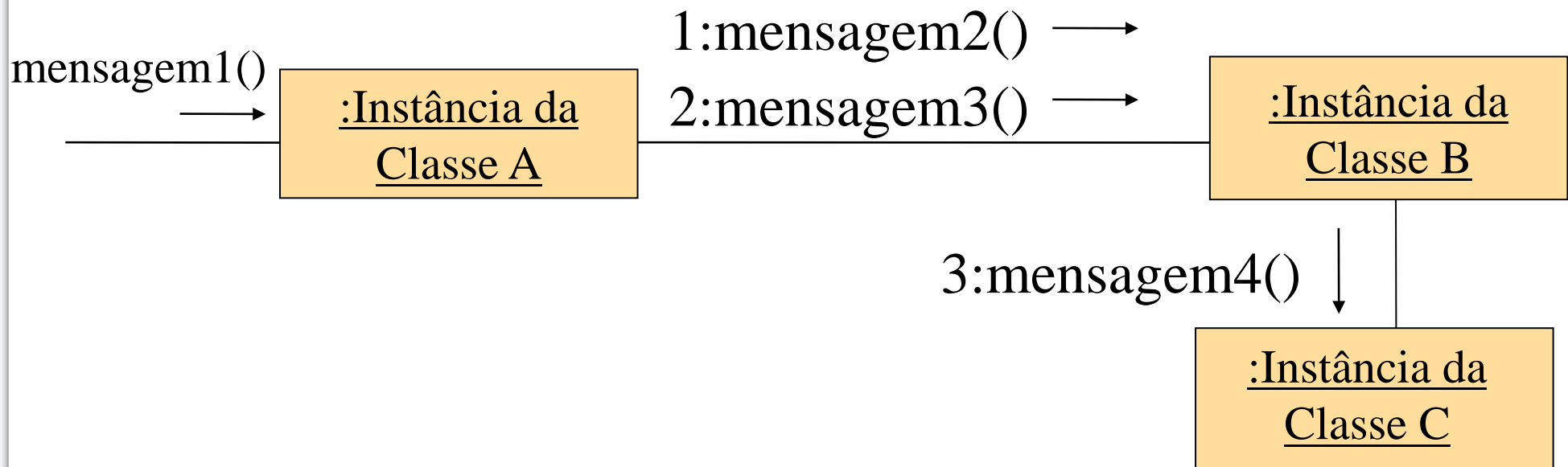
# Diagramas de Interação

- A UML oferece dois tipos de diagramas para mostrar a interação entre os objetos do sistema, por meio de troca de mensagens:
  - diagrama de seqüência e
  - diagrama de comunicação (anteriormente à UML 2.0 chamado de diagrama de colaboração)
    - Atenção: algumas ferramentas CASE ainda usam o termo “diagrama de colaboração”

# Diagrama de Sequência



# Diagrama de Comunicação

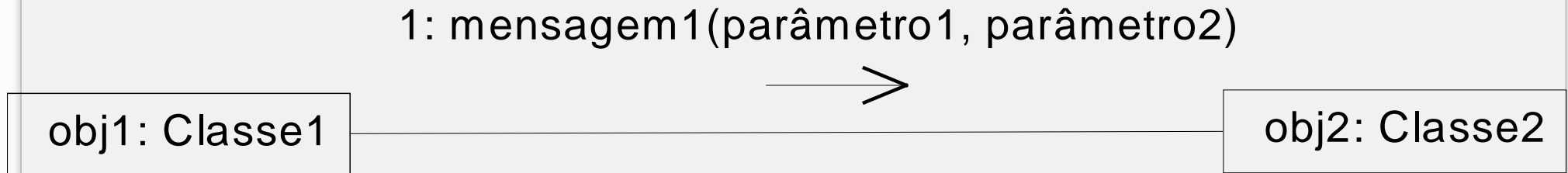


- Os diagramas de comunicação têm melhor capacidade de expressar informações contextuais e podem ser mais econômicos em termos de espaço

# Contribuições dos Diagramas de Comunicação

- *Adição dos métodos*
- *Adição da direção das associações*
- *Possível detalhamento dos atributos e associações*
- *Possível alteração na estrutura das classes e associações*
- *Possível criação de atributos privados ou protegidos*

# Mensagem entre dois objetos no diagrama de comunicação



- obj1 é uma instância da classe Classe1
- obj2 é uma instância da classe Classe2
- A mensagem1 está sendo enviada pelo objeto obj1 ao objeto obj2
- A mensagem1 passa dois parâmetros para o objeto obj2



# Classes X Instâncias

Classe

Livro

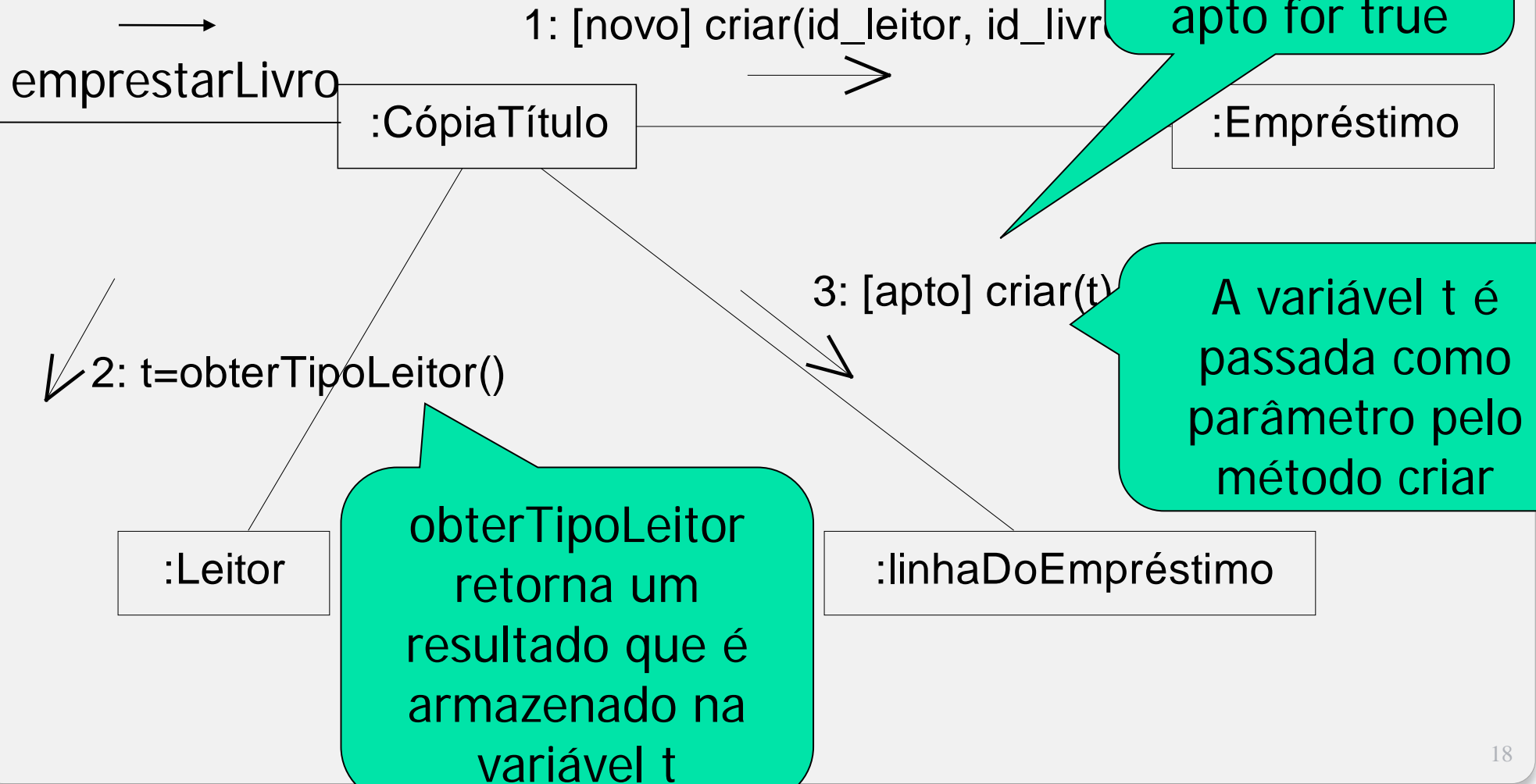
Instância

:Livro

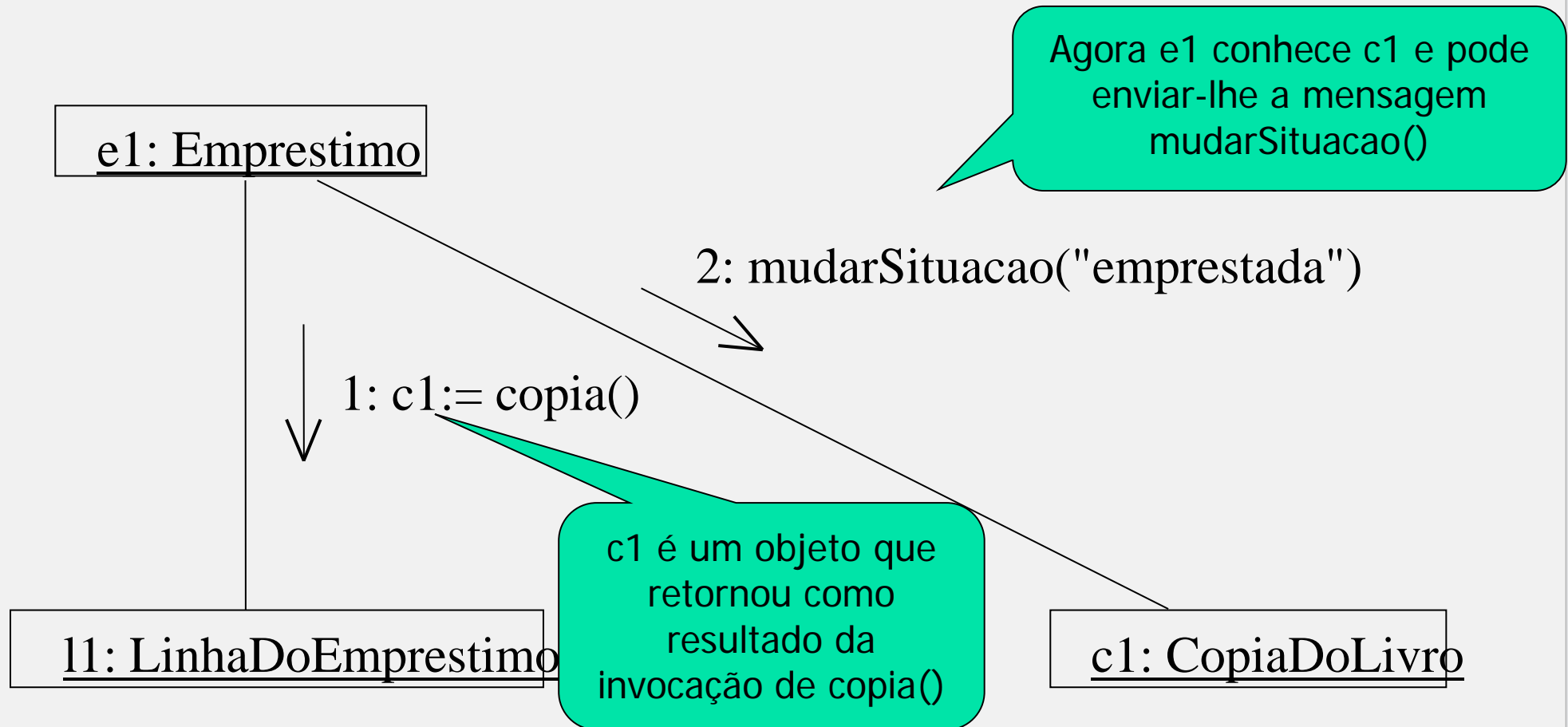
Instância  
nomeada

livro1: Livro

# Ordem das mensagens, condicionais e atribuição a variáveis



# Objeto como retorno de método



# Condicionais e Operadores lógicos e relacionais

e1: Emprestimo



:linhaDoEmprestimo

1: [naoEstaEmAtraso] and [nroLivros < maximoPermitido]

1: [condicao1] mensagem1()

c2: Classe2

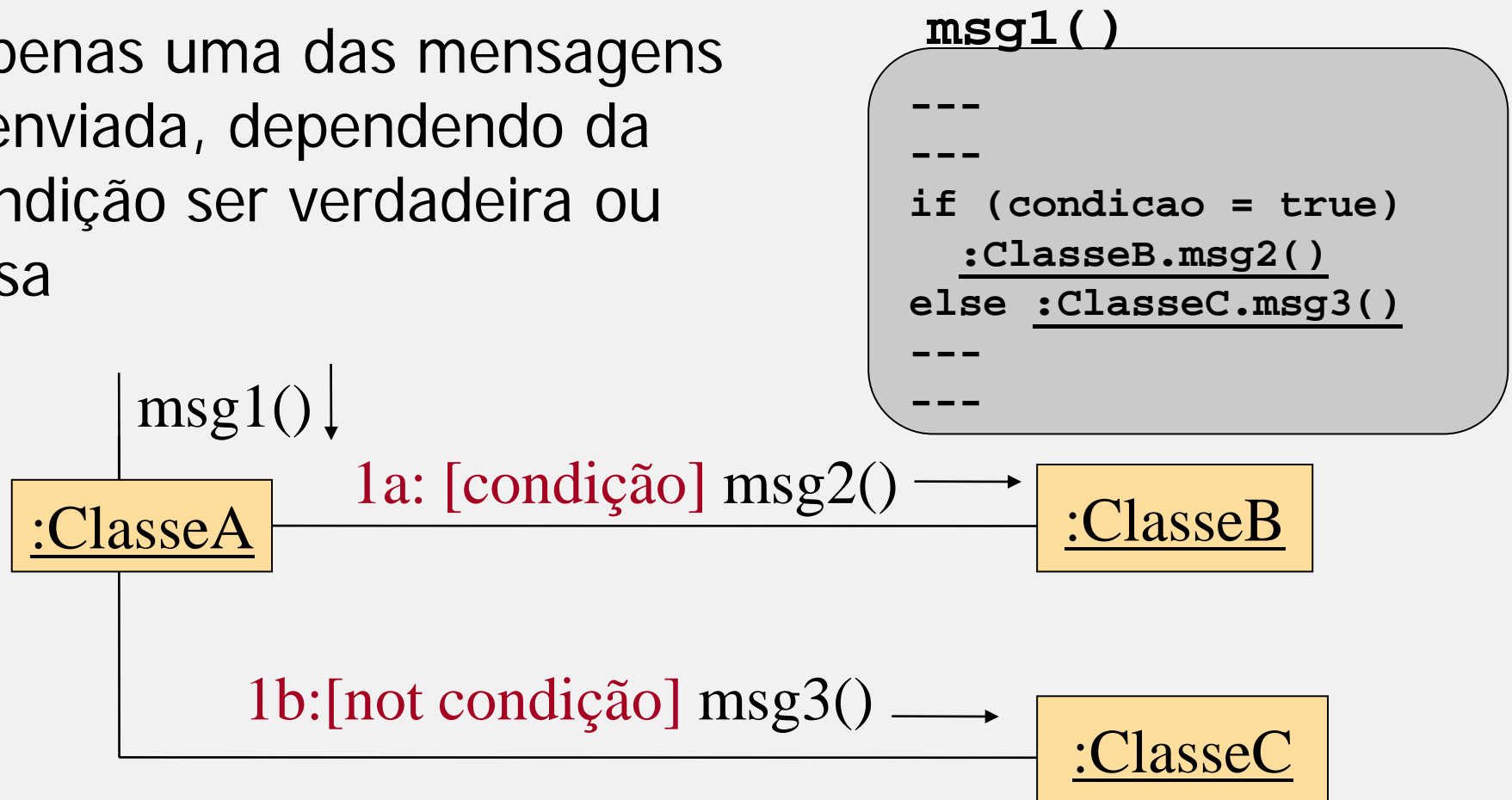
c1: Classe1

2: [not condicao1] mensagem2()

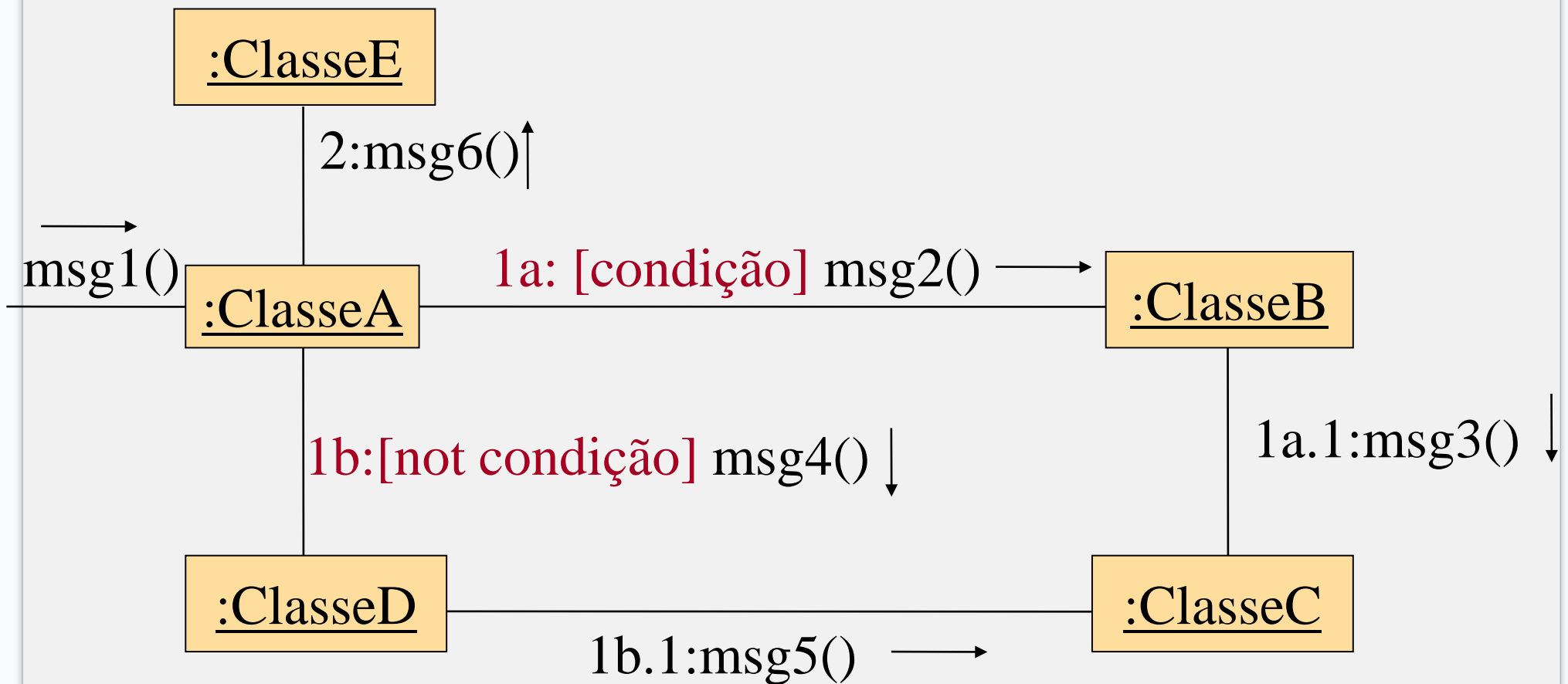
c3: Classe3

# Caminhos Condicionais Mutuamente Exclusivos

Apenas uma das mensagens é enviada, dependendo da condição ser verdadeira ou falsa

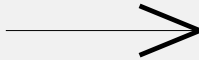


# Caminhos Condicionais Mutuamente Exclusivos



# Repetição de mensagem

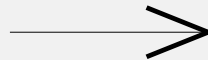
1: [repita para i de 1 a 100] mensagem1(i)



c1: Classe1

c2: Classe2

1: [para cada registro] mensagem1()

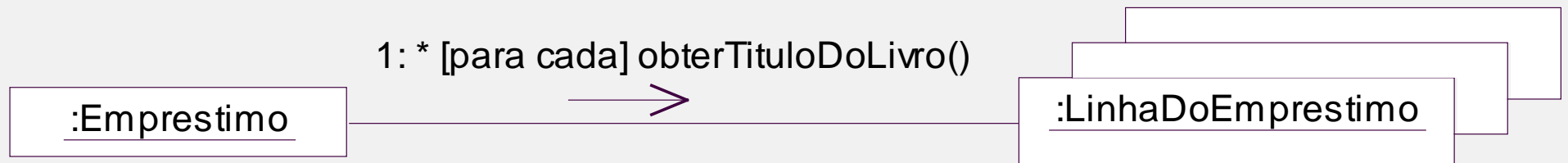


c1: Classe1

c2: Classe2

Em ambos os casos, a mensagem é enviada várias vezes,  
sempre para o mesmo objeto!!

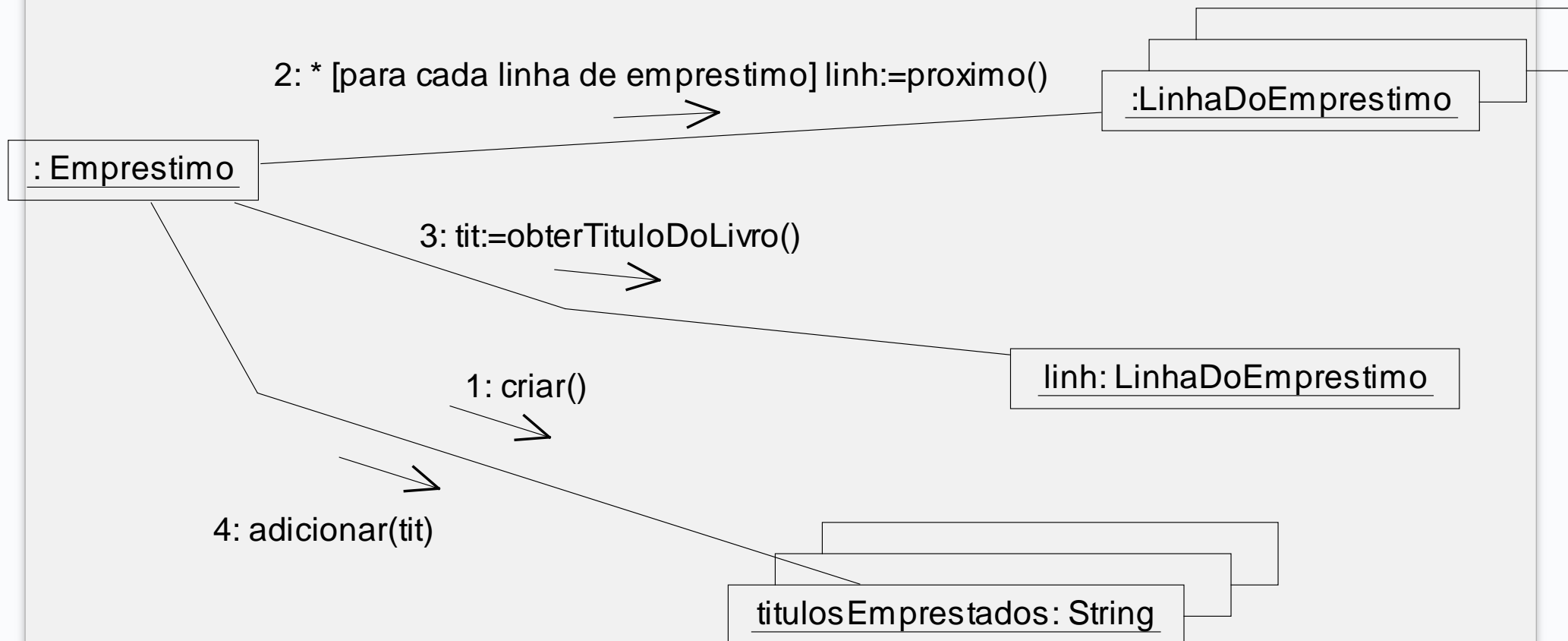
# Mensagem para coleção



A mensagem é enviada uma vez para cada objeto da coleção !!



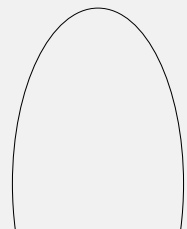
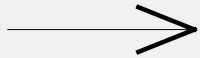
# Criação de uma coleção, percurso de uma coleção



A coleção é criada (1) para depois ser preenchida com títulos (4)  
O método `proximo()` é utilizado para percorrer a coleção

# Auto-mensagem

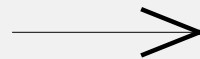
1: cons := ehDeConsulta()



l1: Livro



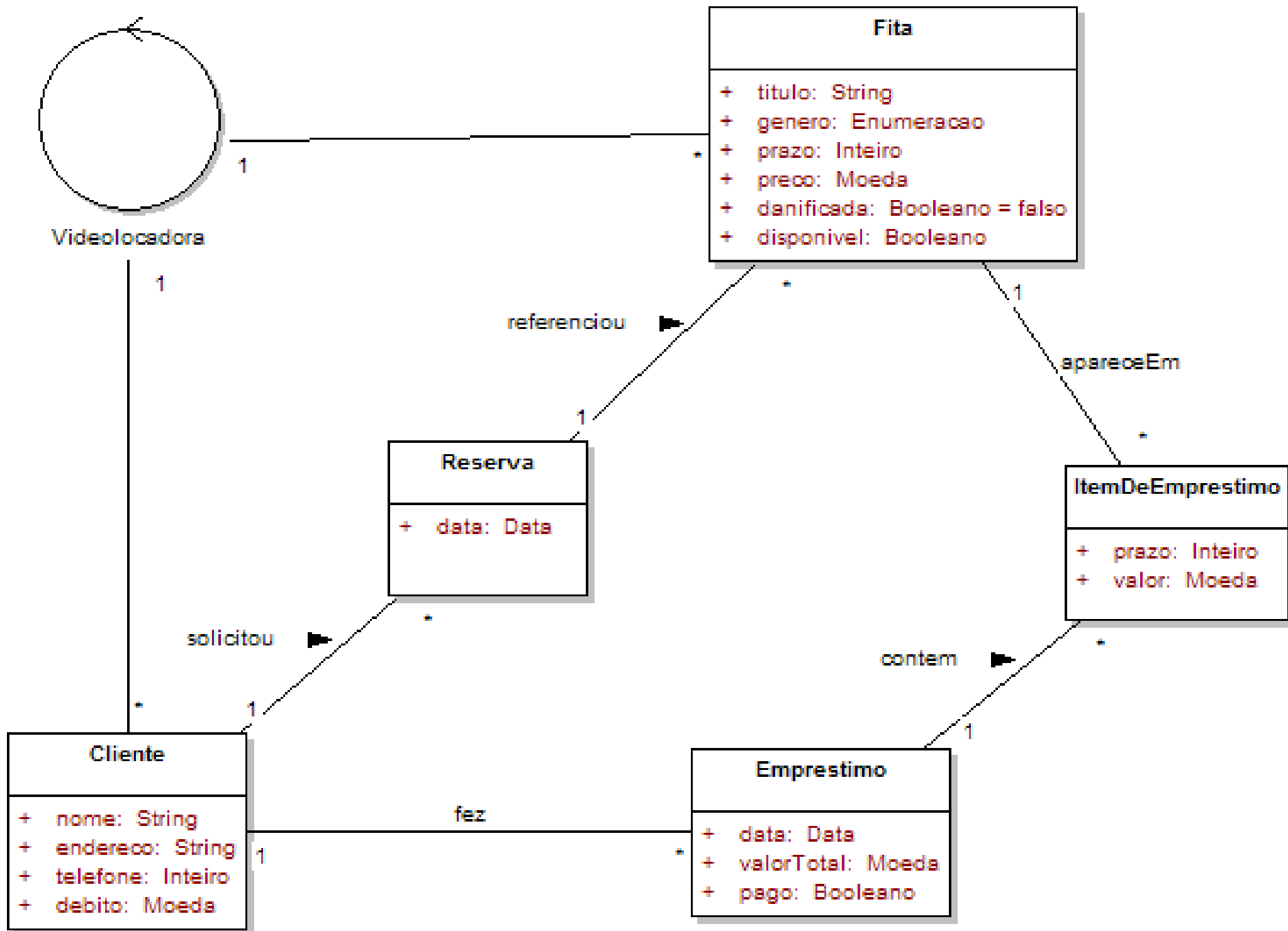
cl1: CopiaDoLivro



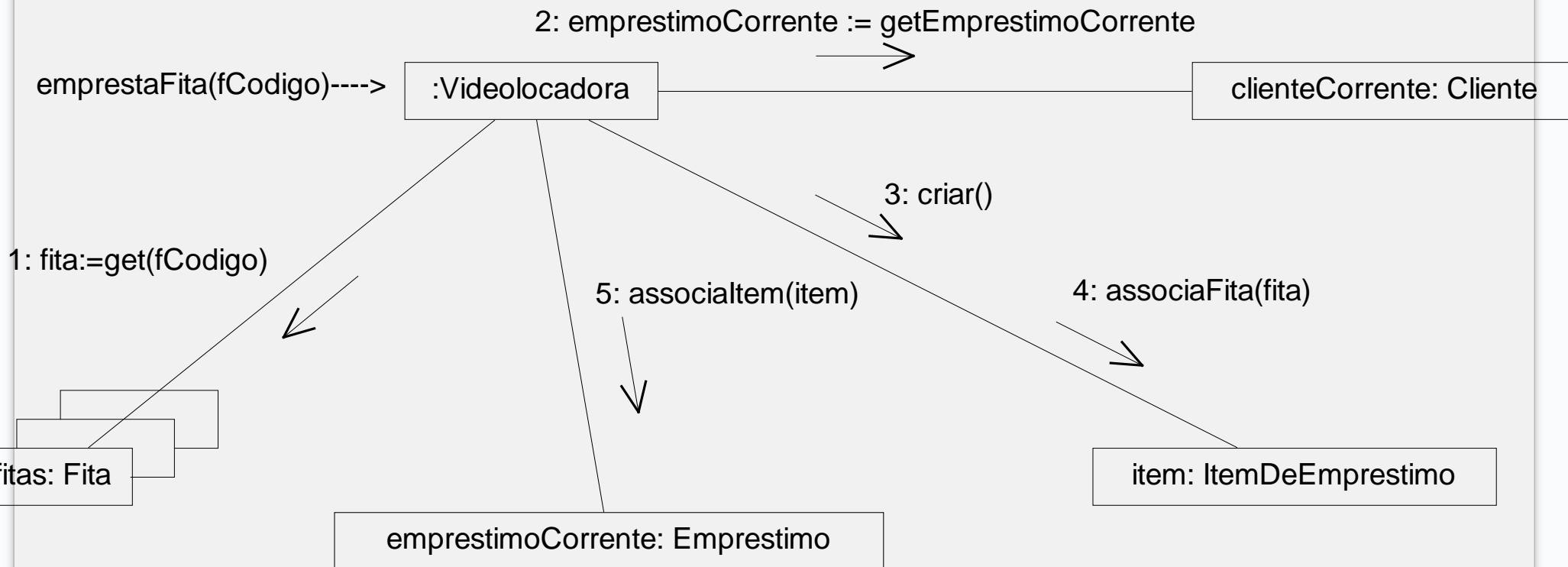
2: [not cons] mudarSituacao("emprestada")

# Exemplo

- Implementação inchada ou concentradora,
- Versus
- Implementação leve, distribuída



# Comunicação entre os objetos



# Pseudocódigo concentrador

## **Classe VideoLocadora**

*fitas : Conjunto;*

*clienteCorrente : Cliente;*

### **Método emprestaFita(fCodigo: String)**

*fita : Fita;*

*emprestimoCorrente : Emprestimo;*

*item : ItemDeEmprestimo;*

*fita := fitas.get(fCodigo);*

*emprestimoCorrente := clienteCorrente.getEmprestimoCorrente();*

*item := ItemDeEmprestimo.new();*

*item.associaFita(fita);*

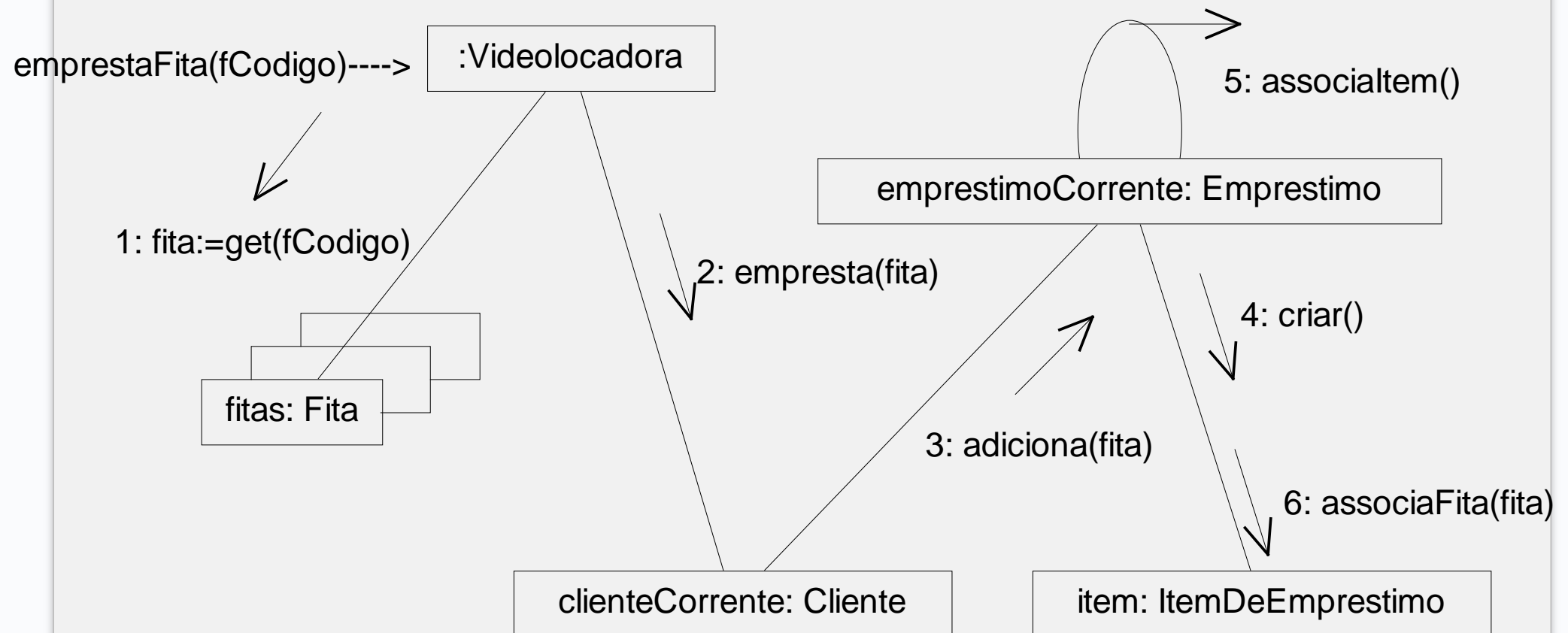
*emprestimoCorrente.associaItem(item);*

**Fim Método;**

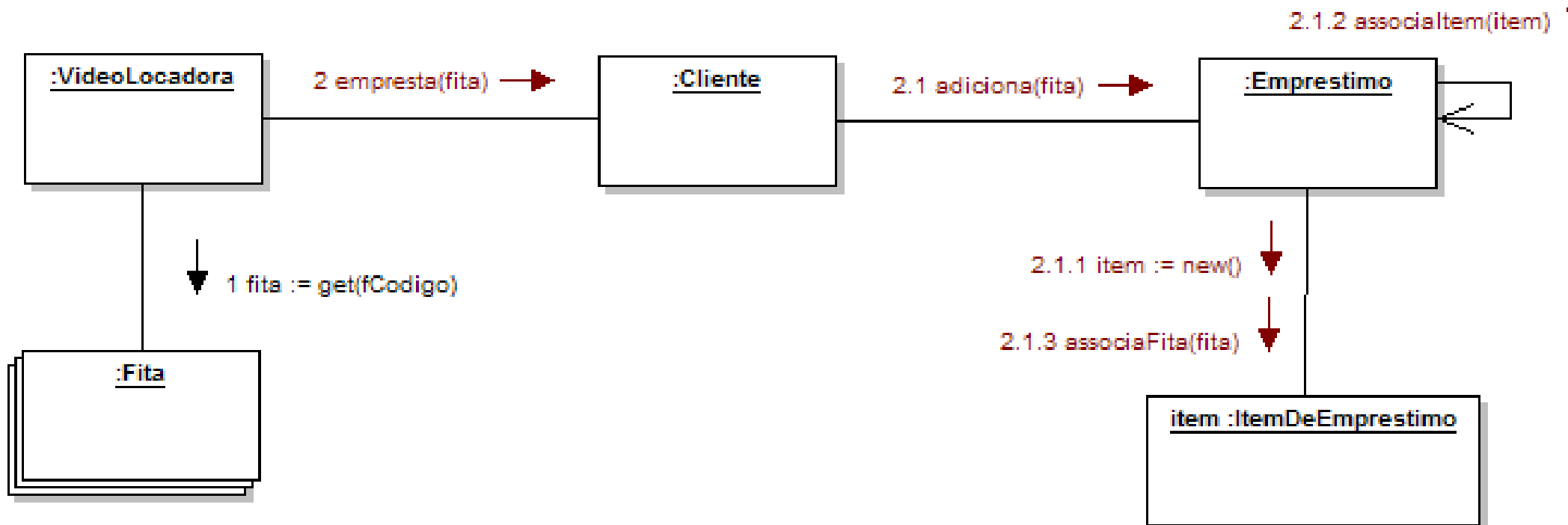
**Fim Classe.**

*(by Raul Wazlawick)*

# Diagrama de Comunicação não concentrador



# Exemplo de um DC





# Código com Responsabilidades Distribuídas

## Classe VideoLocadora

*fitas : Conjunto ;*  
*clienteCorrente : Cliente;*

Metodo emprestaFita(fCodigo : String);  
*fita : Fita;*

*fita := fitas.get(fCodigo);*  
*clienteCorrente.empresta(fita)*

Fim Metodo;  
Fim Classe.

## Classe Cliente

*emprestimoCorrente : Emprestimo;*

Metodo empresta(fita : Fita);  
*emprestimoCorrente.adiciona(fita);*

Fim Metodo;  
Fim Classe.

## Classe Emprestimo

*itens : Conjunto;*

Metodo adiciona(fita : Fita);  
*item : ItemDeEmprestimo;*

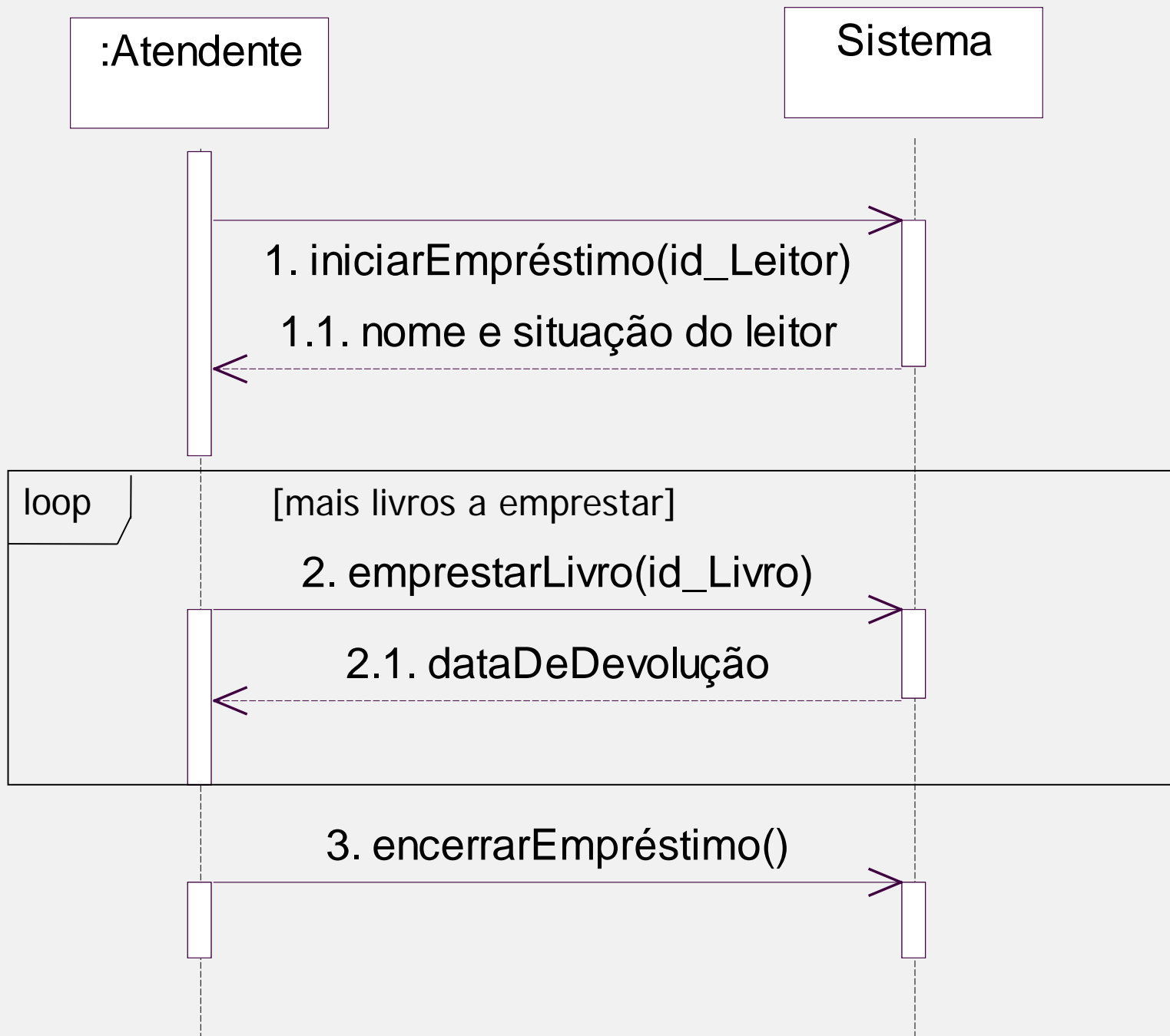
*item := ItemDeEmprestimo.new();*  
*self.associaItem(item);*  
*item.associaFita(fita);*

Fim Metodo;  
Fim Classe.

*(de Raul Wazlawick)*

# Discussão

- Qual dos códigos é mais fácil de entender e manter?
- Em qual dos códigos as responsabilidades das classes parecem mais intuitivas?
- Qual código é mais rápido?
- Para desenvolver um bom projeto, precisamos de princípios que nos guiem na atribuição de responsabilidades



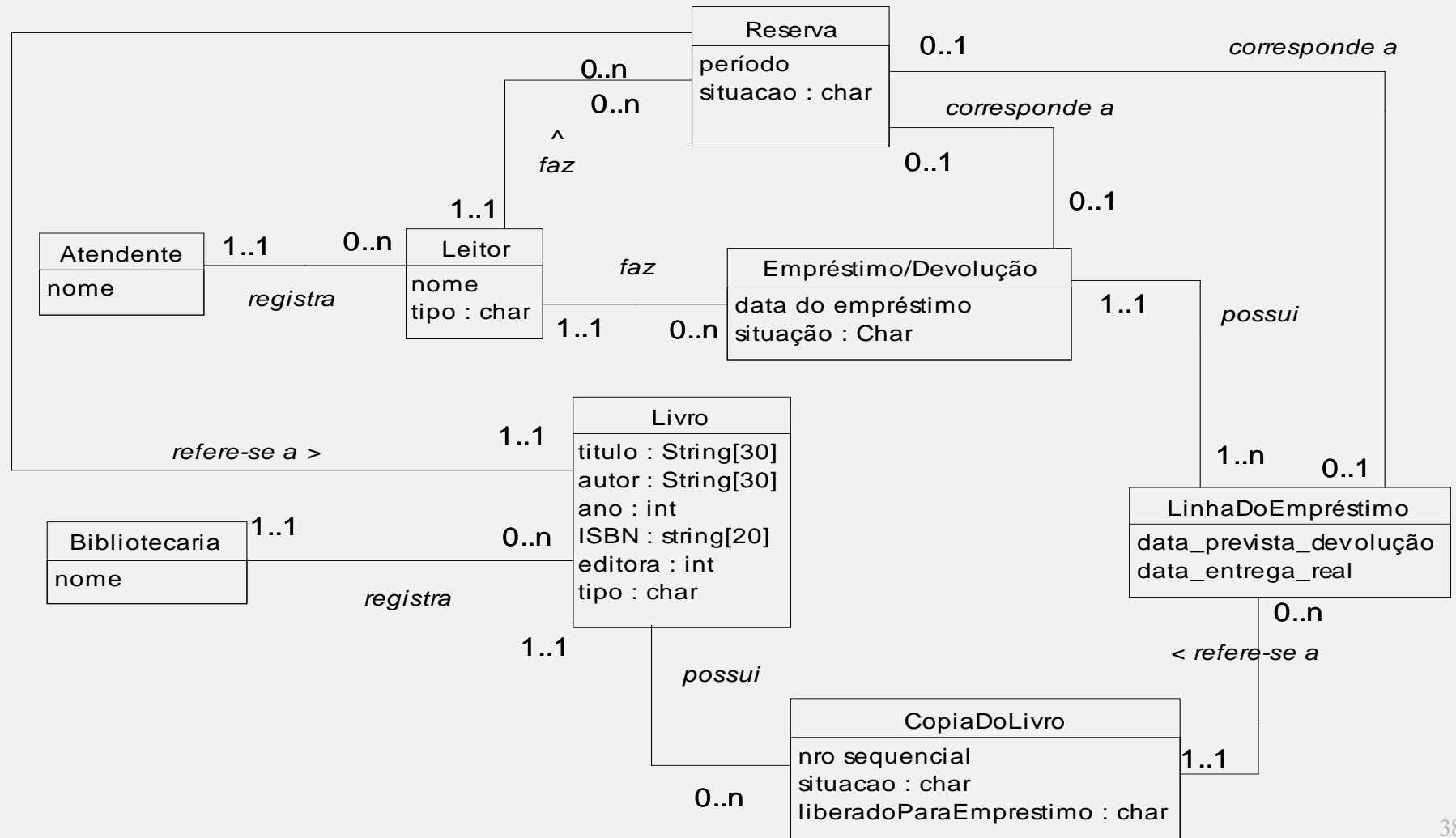
# Exemplo: Fazer um DC para a operação `iniciarEmprestimo()`

- *Operação*: `iniciarEmprestimo()`
- *Referências Cruzadas*: Caso de uso: "Emprestar Livro"
- *Pré-Condições*:
  - O leitor é registrado na biblioteca e sabe-se seu ID.
- *Pós-Condições*:
  - Um leitor apto a emprestar livros foi identificado.
  - Um objeto Empréstimo/Devolução foi criado e associado ao leitor

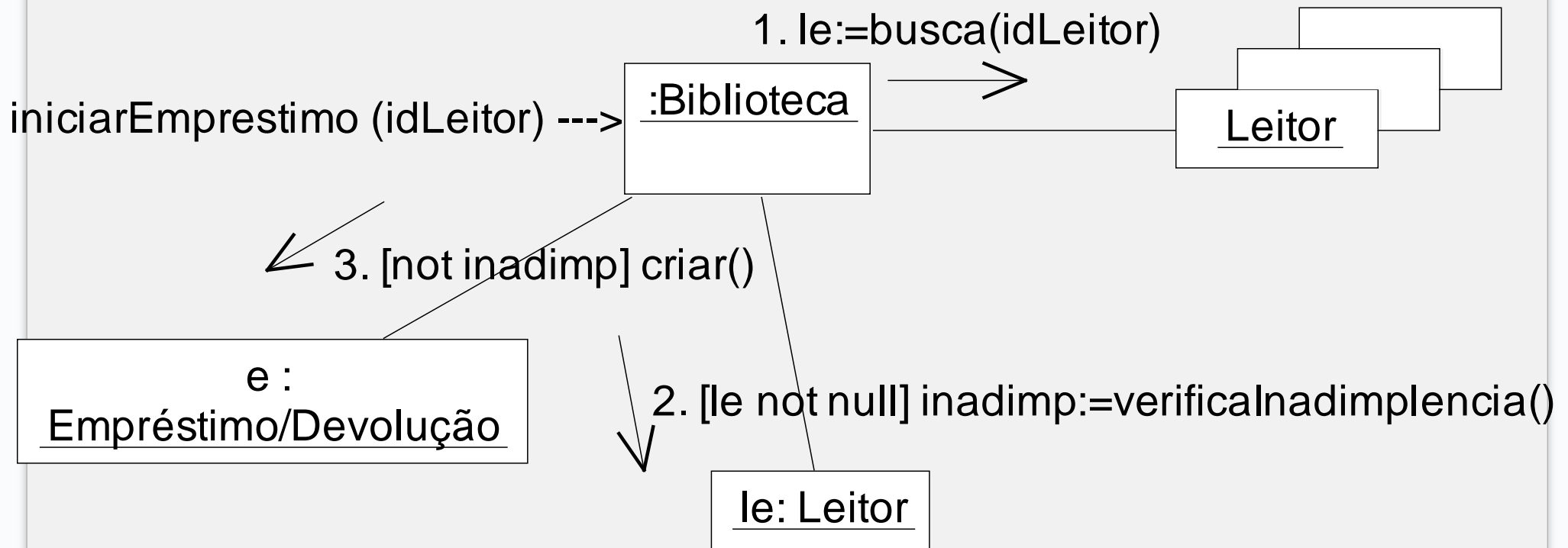
# Exemplo: Fazer um DC para a operação `iniciarEmprestimo()`

- Por onde começar?
  - Analisar as pós-condições: elas dizem qual é o estado do sistema após a invocação da operação.
  - Veja que parâmetros tem em mãos, eles podem dar uma indicação de como começar
  - Olhar os requisitos do sistema para entender a lógica da operação
  - Olhar as operações que vieram antes e as que virão depois, para saber o que tem em mãos no contexto do DSS
  - Olhar o modelo conceitual do sistema

# Exemplo : Fazer um DC para a operação `iniciarEmprestimo()`



# Diagrama de Comunicação

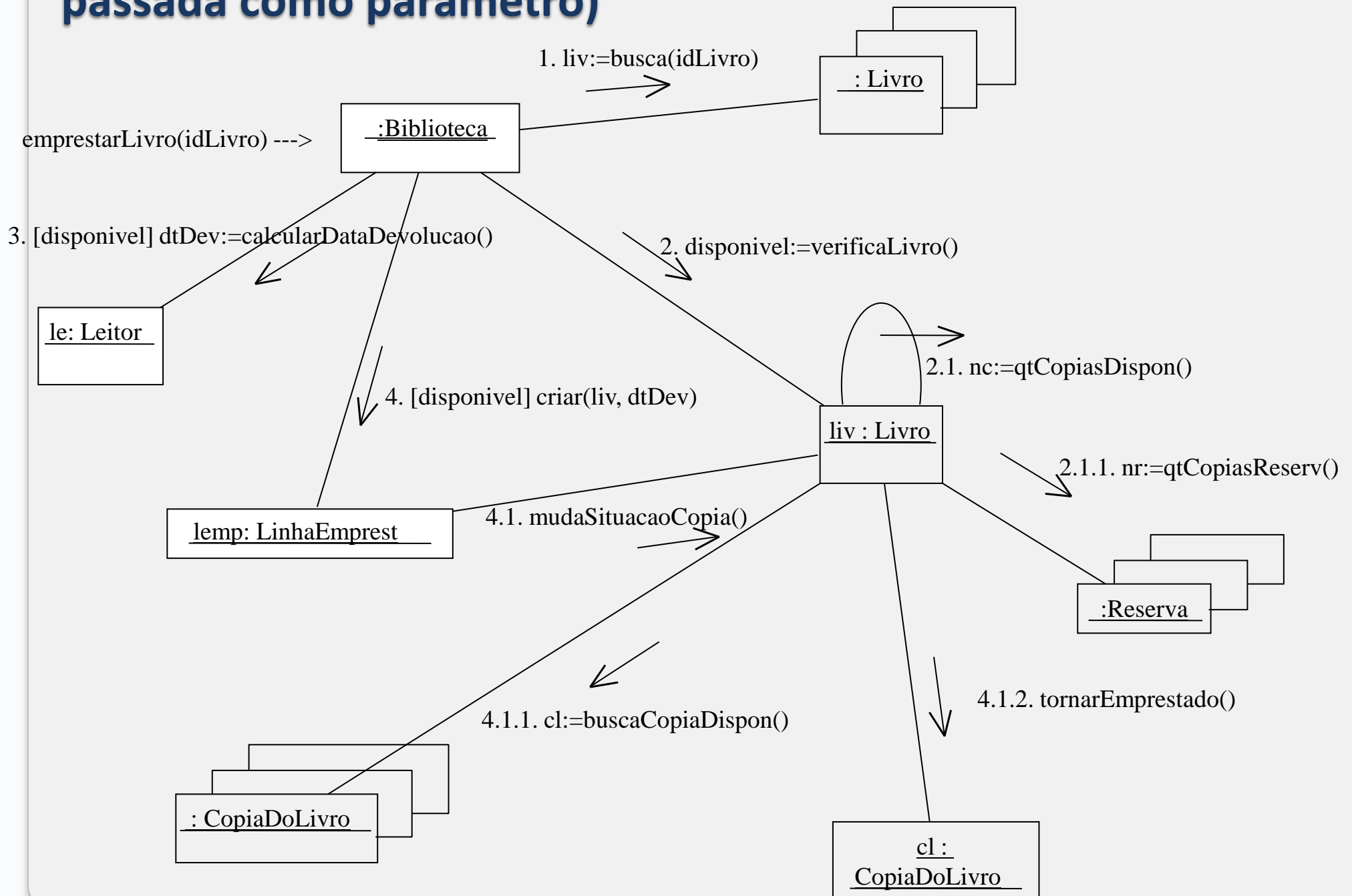


# Exemplo : Fazer um DC para a operação `emprestarLivro()`

- **Operação:** `emprestarLivro(id_Livro)`
- **Referências Cruzadas:** Caso de uso: "Emprestar Livro"
- **Pré-Condições:**
  - Um leitor apto a emprestar livros já foi identificado e um empréstimo já foi criado;
- **Pós-Condições:**
  - um objeto `LinhaDoEmpréstimo` foi criado;
  - a `LinhaDoEmpréstimo` foi associada à cópia do livro e ao empréstimo criado para o leitor na operação anterior "iniciarEmprestimo";
  - O atributo "situação" da cópia do livro foi alterado para "emprestada".
  - a data de devolução foi calculada, armazenada na "`LinhaDoEmprestimo`" e informada ao leitor.

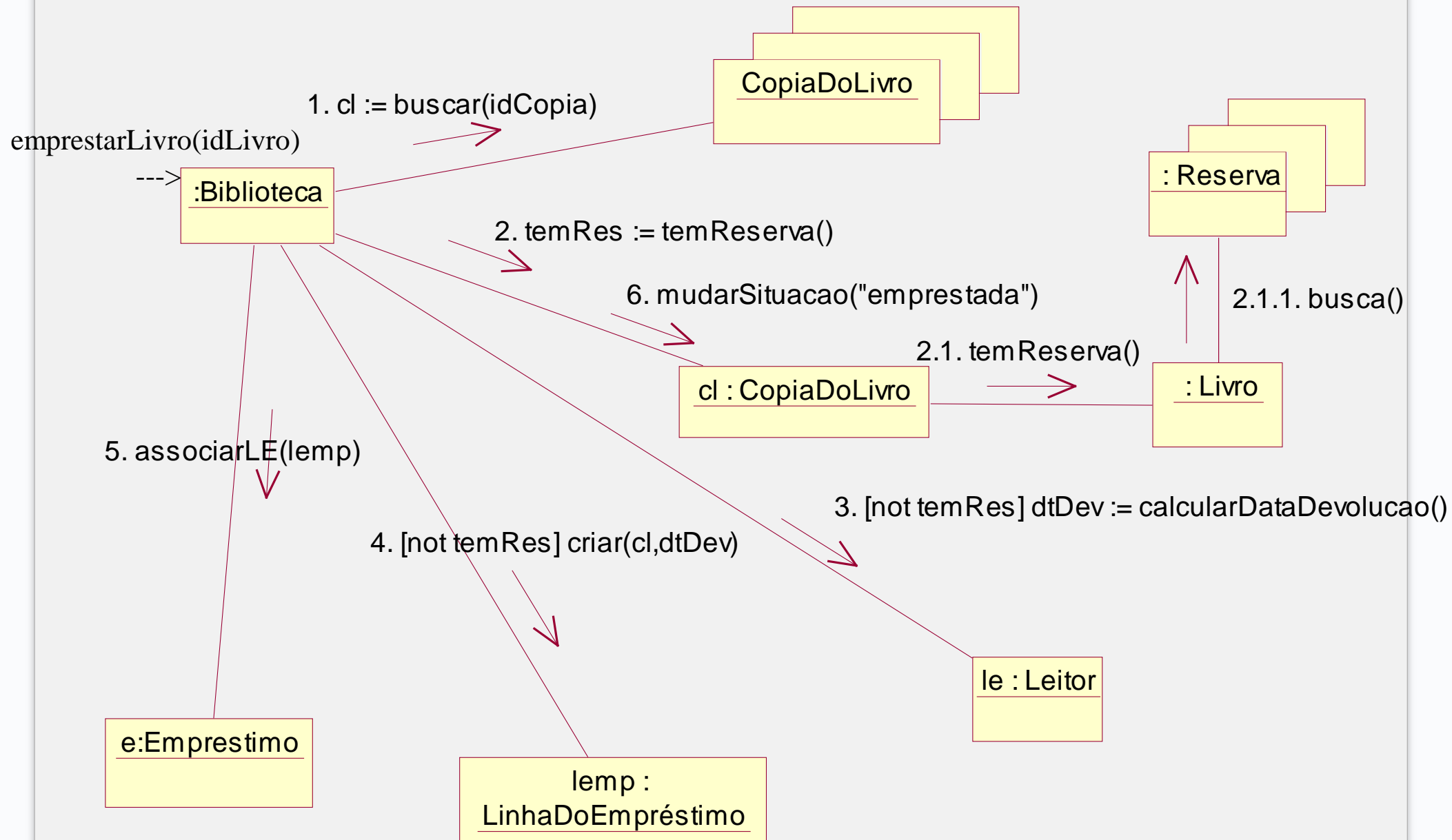


# Diagrama de Comunicação (solução 1 considerando idLivro passada como parâmetro)



E para verificar se leitor já tem o número máximo permitido de livros em seu poder?

# Diagrama de Comunicação (solução 2 considerando idCopia passada como parâmetro)



# Exemplo: Fazer um DC para a operação `encerrarEmprestimo()`

- *Operação*: `encerrarEmpréstimo()`
- *Referências Cruzadas*: Caso de uso: “Emprestar Livro”
- *Pré-Condições*:
  - Existe “empréstimo/devolução” e ao menos uma linha do empréstimo
- *Pós-Condições*:
  - O sistema foi informado que o empréstimo atual foi concluído.
  - O atributo “situação” de “Empréstimo/Devolução” foi alterado para “vigente”

# Diagrama de Comunicação

