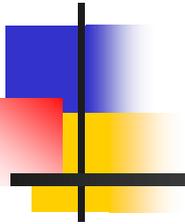


# Notação Básica dos Diagramas de Comunicação

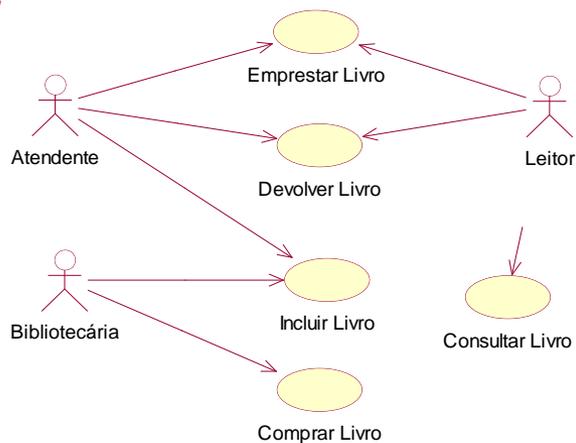


Análise e Projeto Orientados a Objetos

Profa Rosana

# O que já foi visto até agora

## Diagrama de Casos de Uso



## Casos de Uso Completo Abstrato

### 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

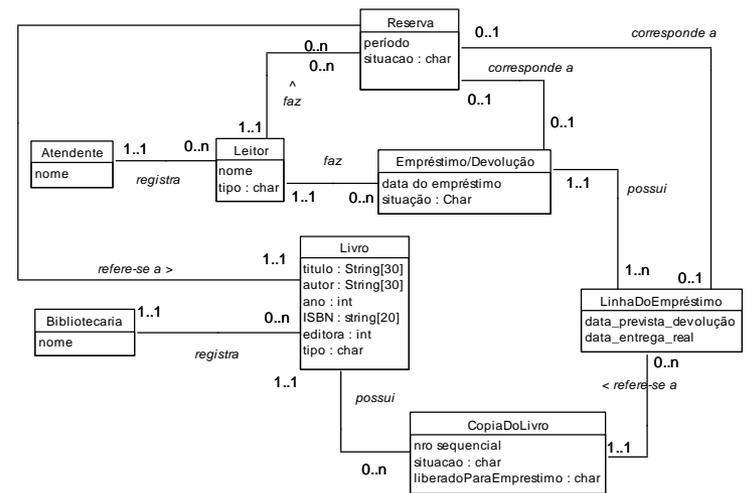
## Caso de Uso 1

- 1 Consultar a lista de títulos em catálogos e fazer a consulta de reservas e empréstimos de livros da biblioteca
- 2 Consultar o status de uma reserva ou empréstimo
- 3 Consultar o status de uma reserva em uma biblioteca específica
- 4 Consultar o status de uma identificação de livro
- 5 Consultar o status de uma situação
- 6 Consultar o status de uma prestação
- 7 Consultar o status de uma identificação de livro
- 8 Consultar o status de uma identificação de livro
- 9 Consultar o status de uma situação de uma biblioteca
- 10 Consultar o status de uma situação

## Caso de Uso n

- 1 Consultar a lista de títulos em catálogos e fazer a consulta de reservas e empréstimos de livros da biblioteca
- 2 Consultar o status de uma reserva ou empréstimo
- 3 Consultar o status de uma reserva em uma biblioteca específica
- 4 Consultar o status de uma identificação de livro
- 5 Consultar o status de uma situação
- 6 Consultar o status de uma prestação
- 7 Consultar o status de uma identificação de livro
- 8 Consultar o status de uma identificação de livro
- 9 Consultar o status de uma situação de uma biblioteca
- 10 Consultar o status de uma situação

## Modelo Conceitual

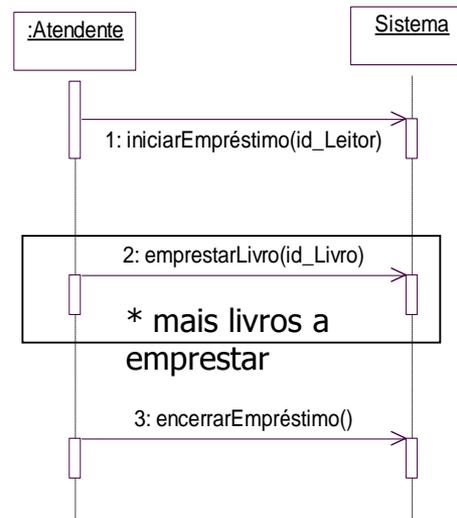


# O que já foi visto até agora

Modelo Conceitual  
+  
Casos de Uso

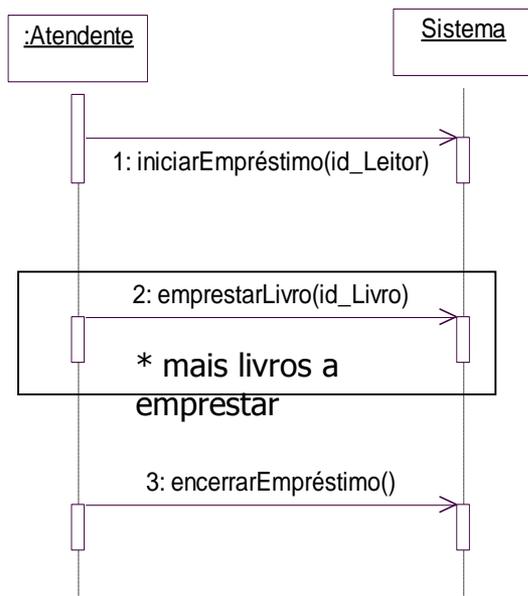


Diagrama de Seqüência do Sistema  
(para cada caso de uso)



# O que já foi visto até agora

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



Contrato da Operação  
(para cada operação)

**Operação:** `criarEmpréstimo()`

**Referências Gerais:** Código de '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.

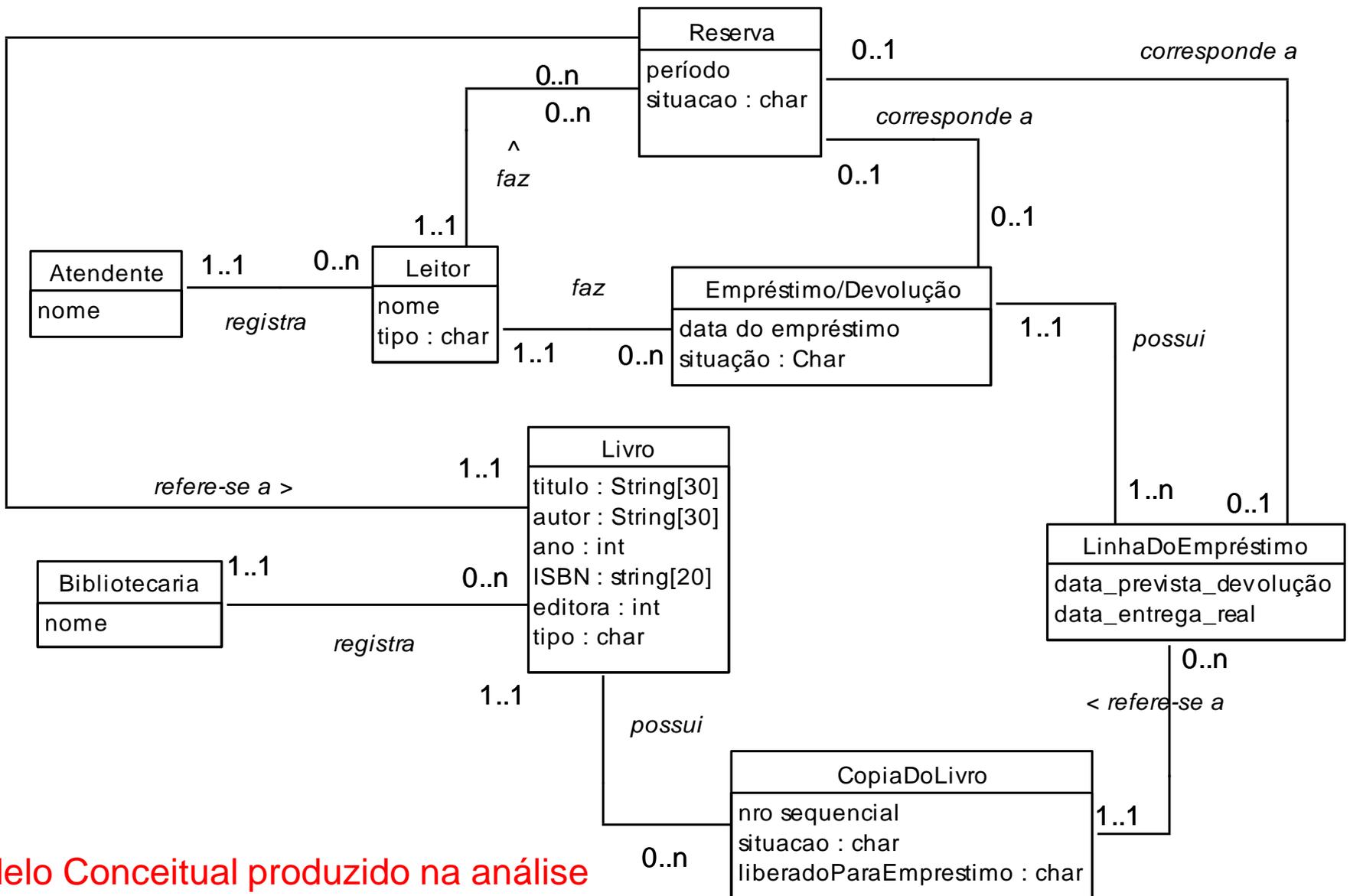
**Pós-Condições:** um novo empréstimo foi registrado, outro empréstimo foi relacionado ao leitor já identificado na operação 'iniciar o empréstimo', a situação dos livros emprestados foi alterada para 'emprestado'.



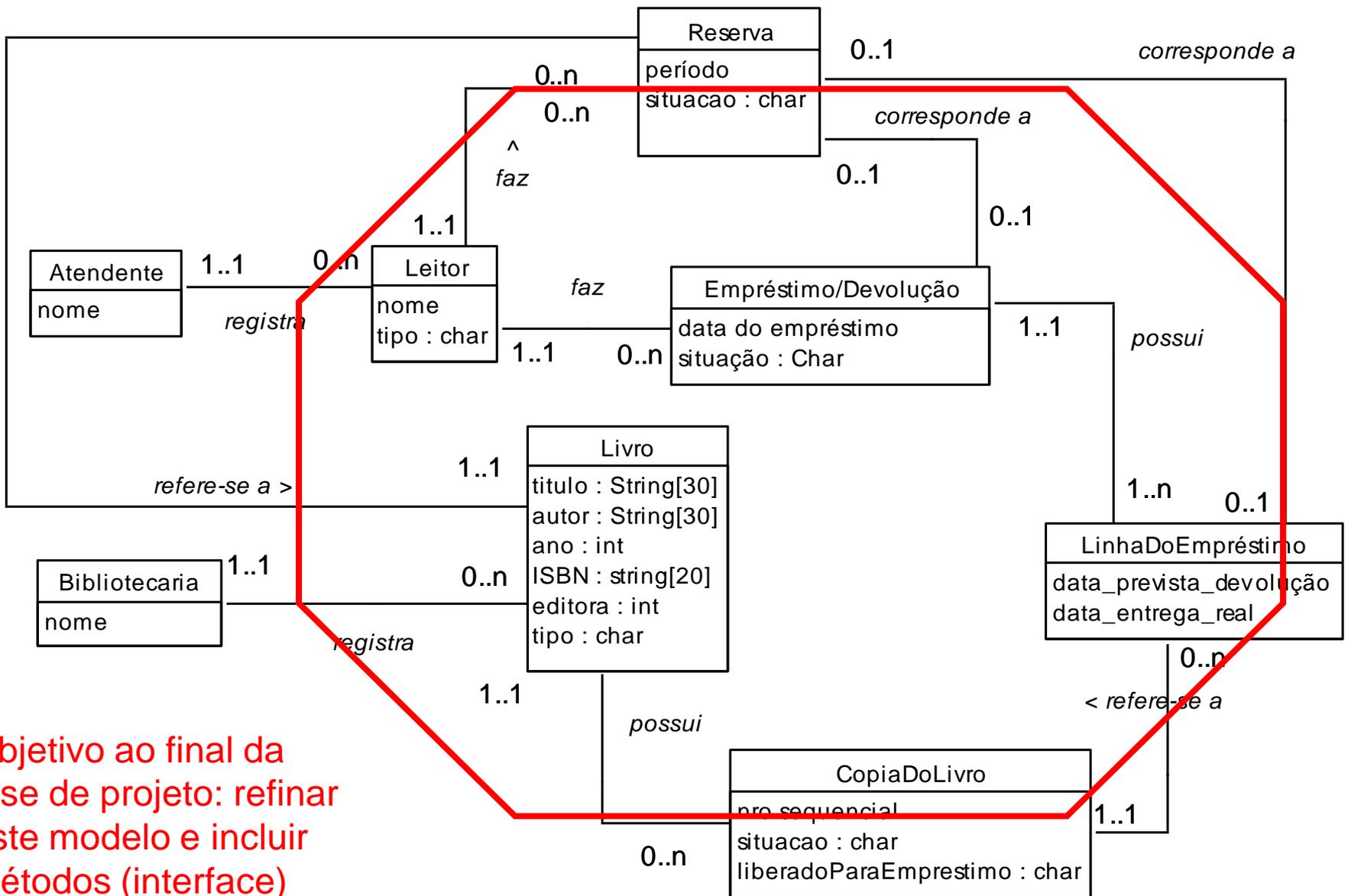
# Projeto Orientado a Objetos

---

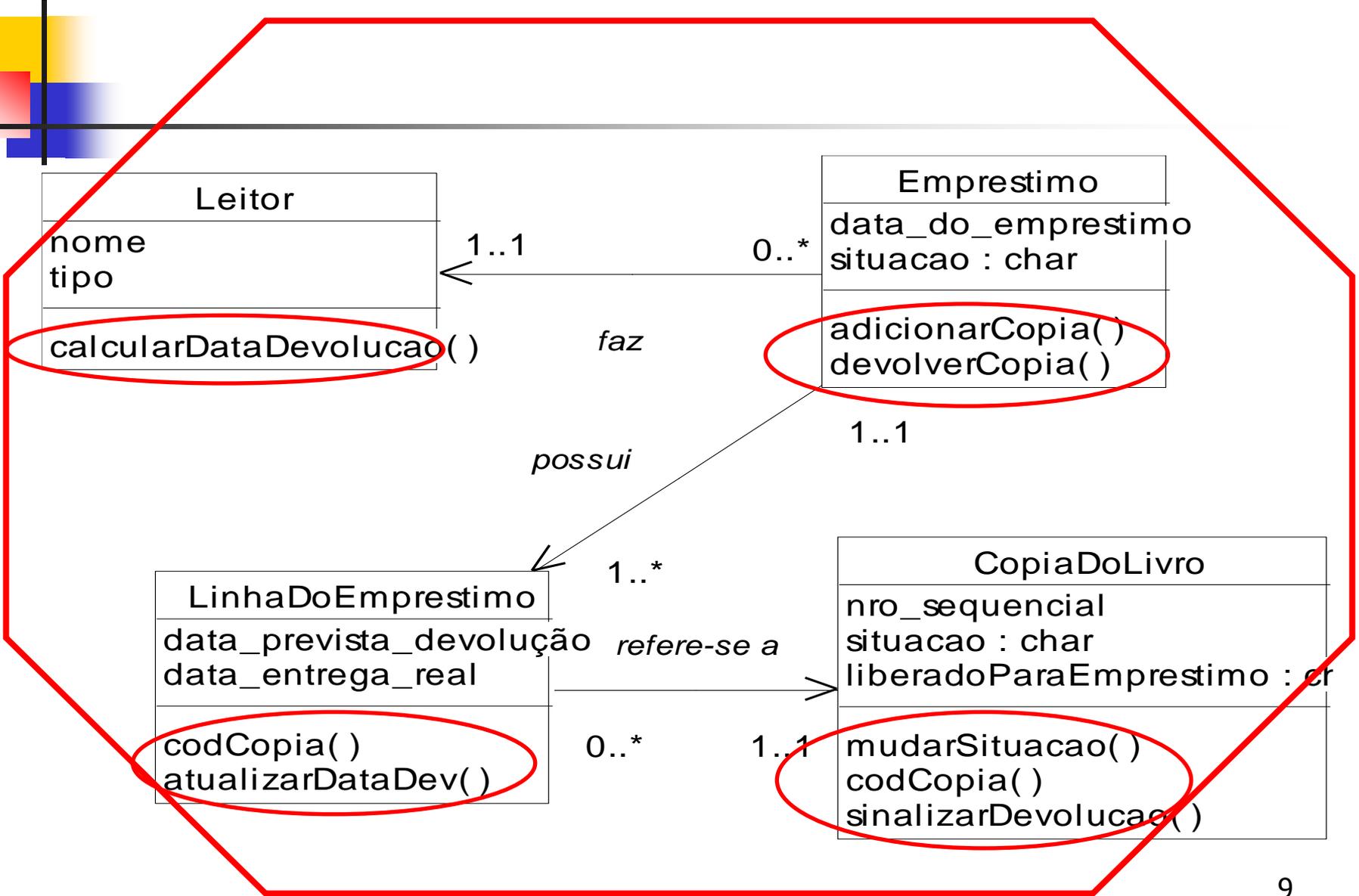
- Finalizada a análise, inicia-se o **Projeto** do sistema
- Artefatos de projeto contém detalhes de **como** o sistema poderá ser implementado utilizando um computador.
  - detalhar as informações sobre as classes que comporão o sistema → diagrama de classes
  - comportamento esperado de cada objeto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



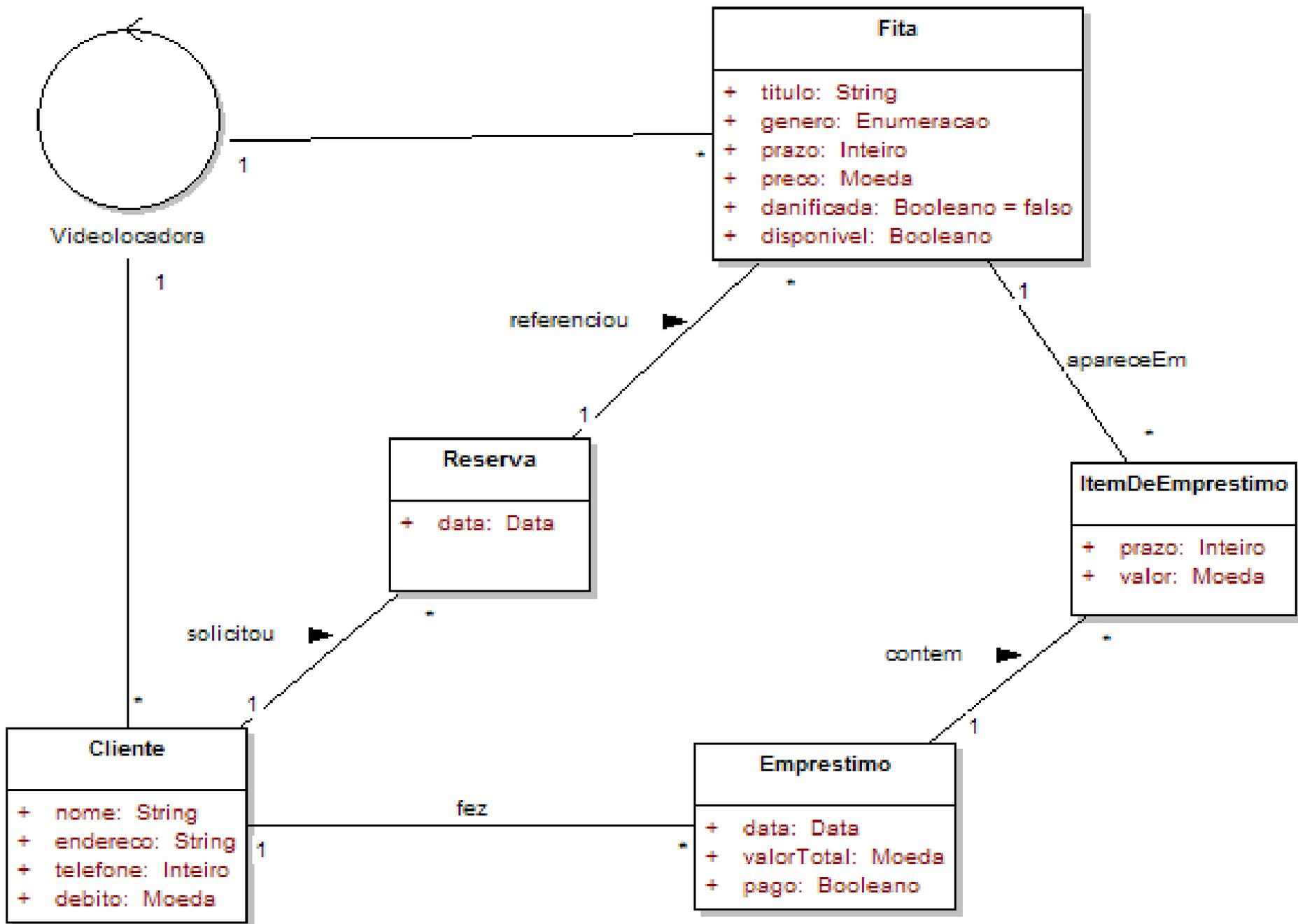
# Exemplo

---

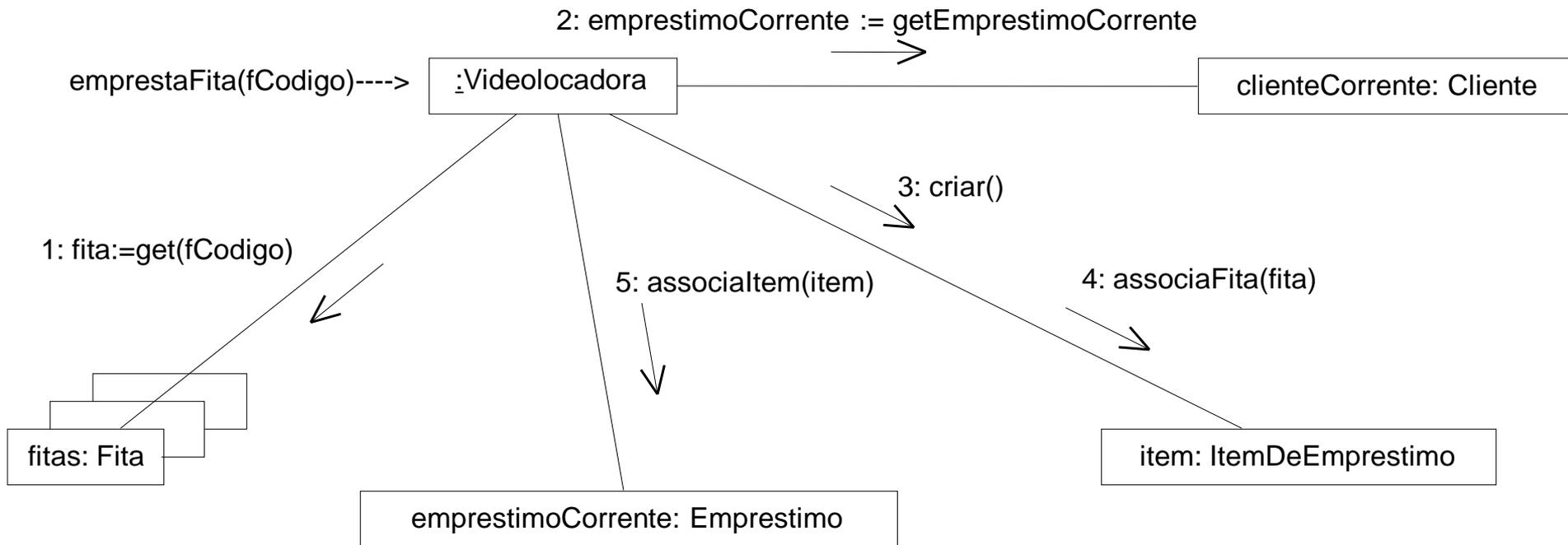
Implementação inchada ou  
concentradora,

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*);

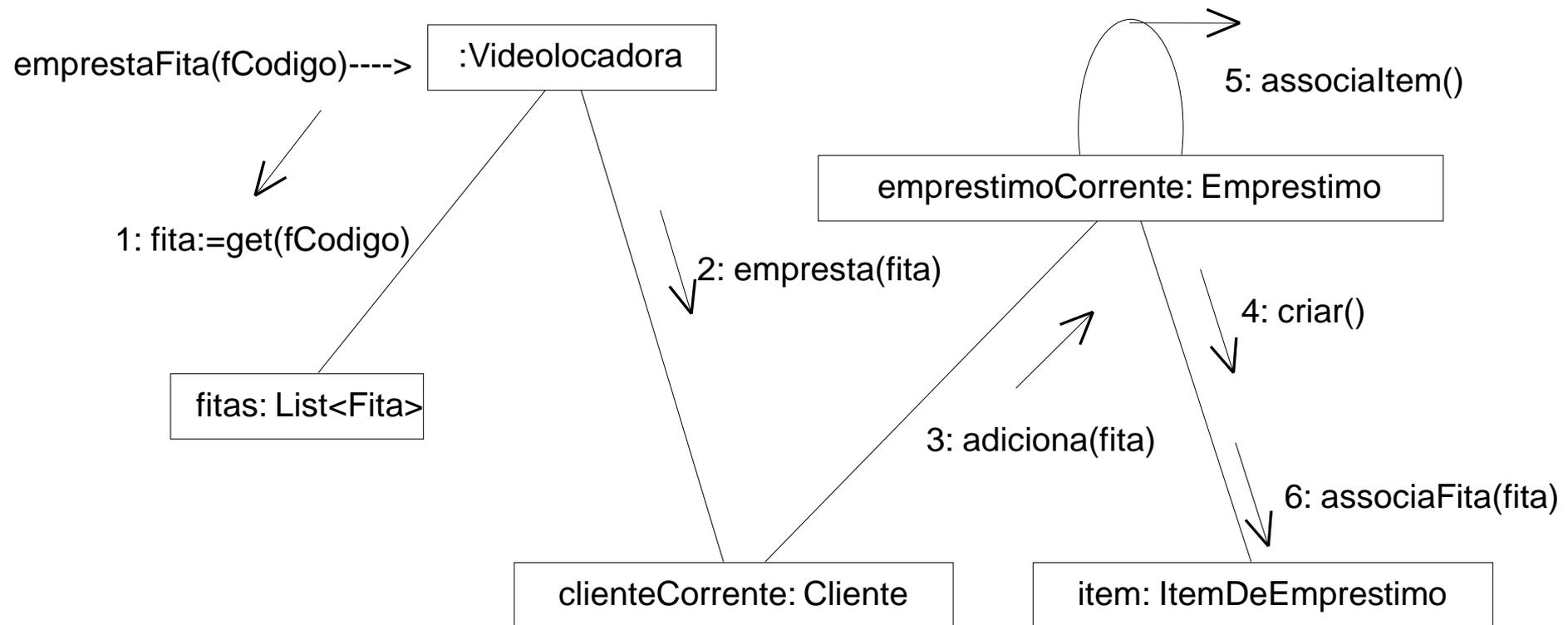
*emprestimoCorrente*.associaItem(*item*);

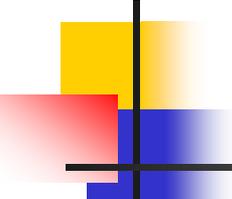
Fim Método;

Fim Classe.

(by Raul Wazlawick)

# Diagrama de Comunicação não concentrador





# 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?

Para desenvolver um bom projeto, precisamos de princípios que nos guiem na atribuição de responsabilidades → padrões GRASP (próxima aula)

Aula de hoje: notação dos diagramas de comunicação

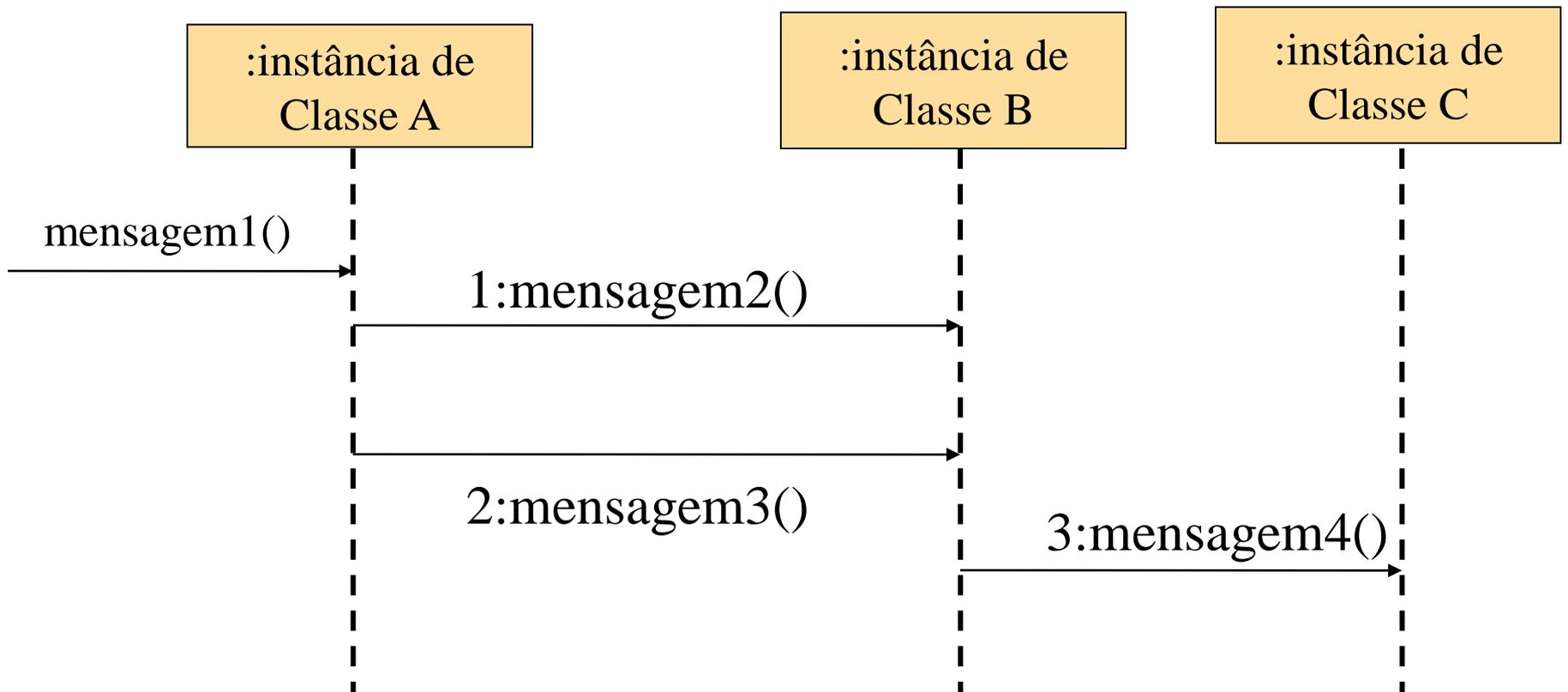


# Diagramas de Interação

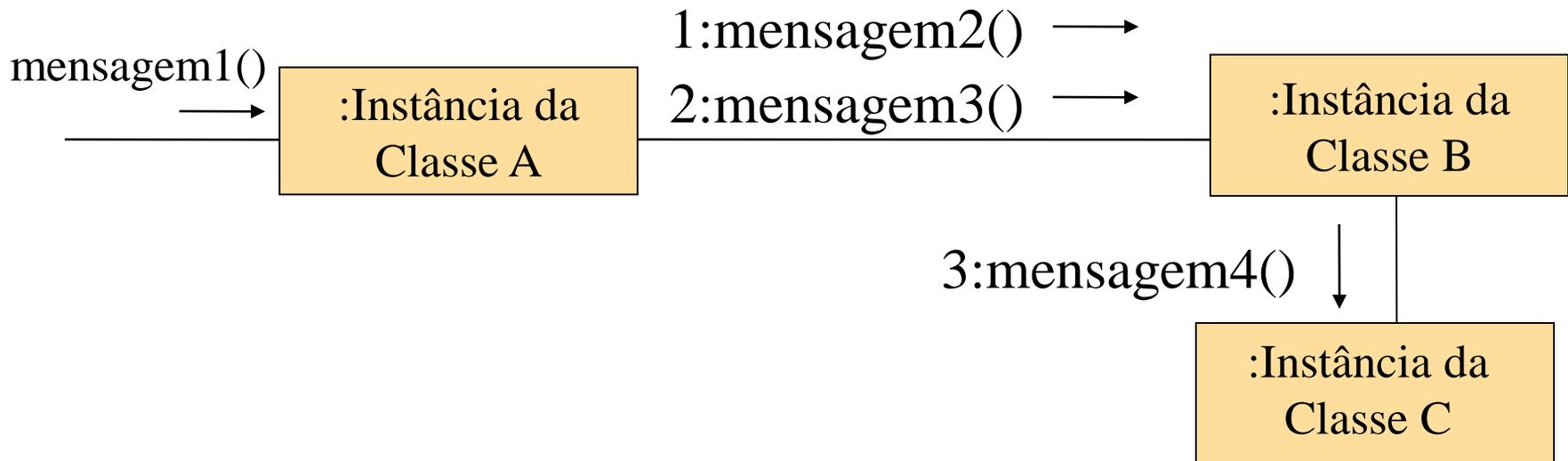
---

- 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

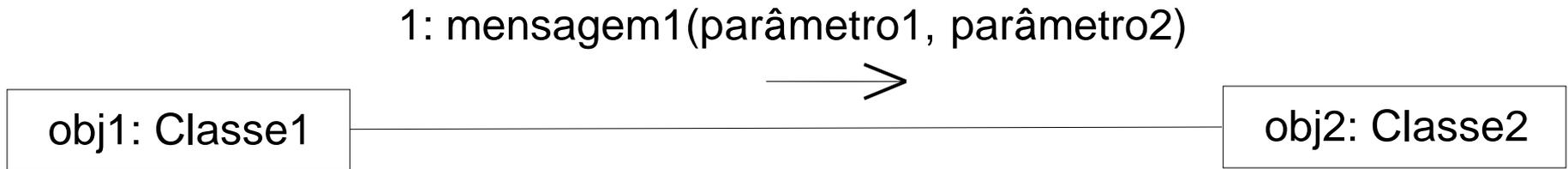


# Contribuições dos Diagramas de Comunicação

---

- *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

Coleção

: List<Livro>

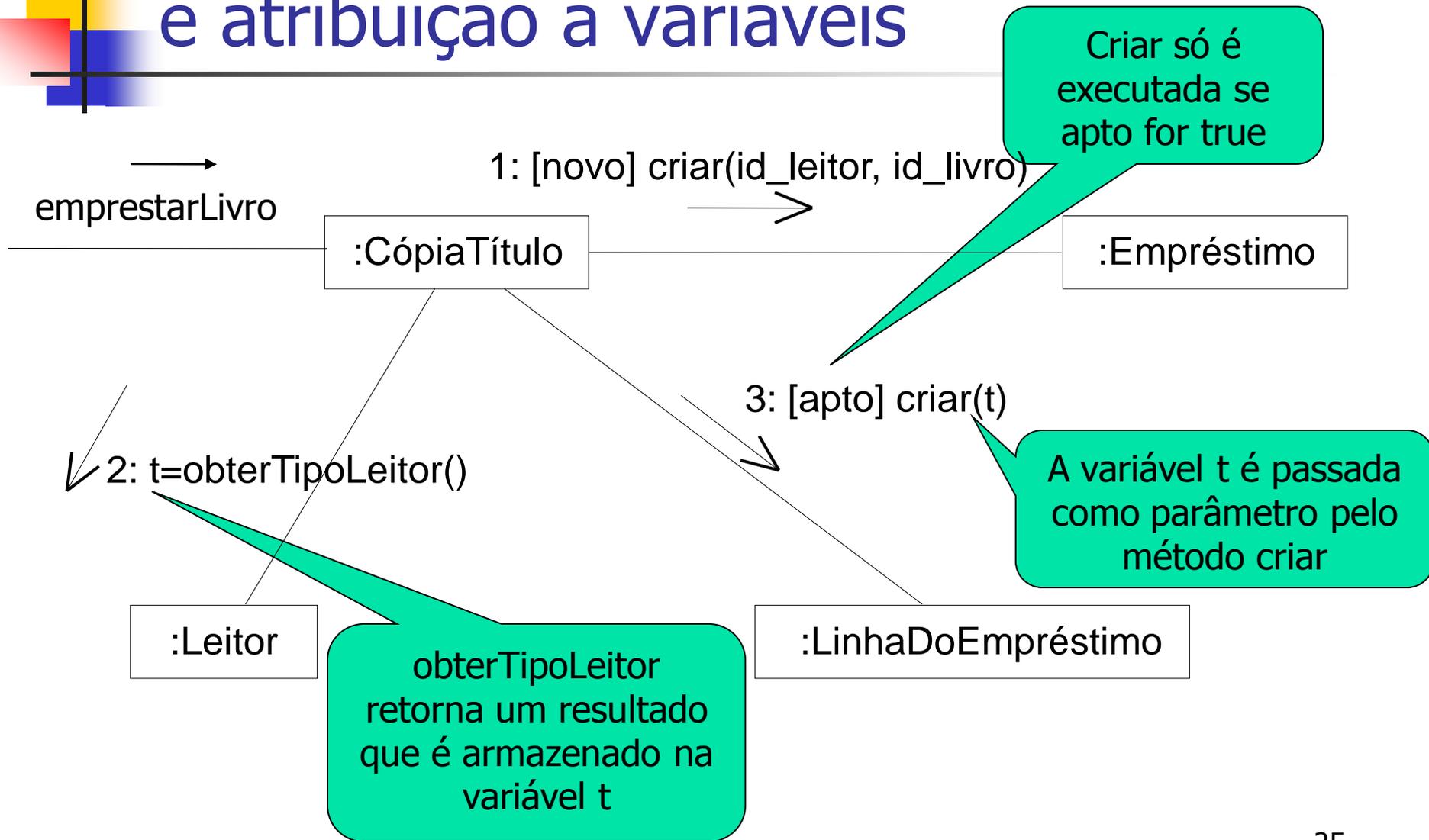
Coleção nomeada

livros: List<Livro>

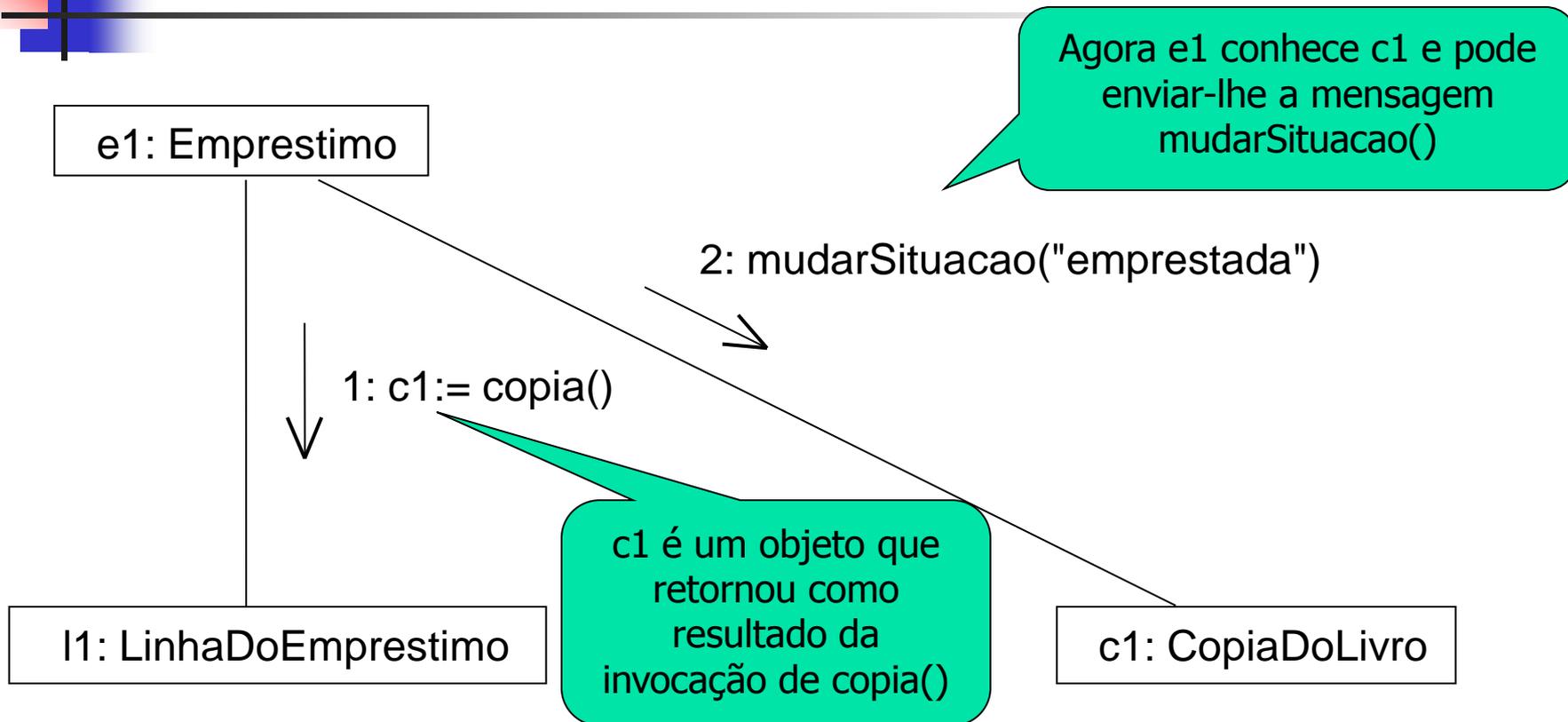
Instância da coleção

liv[i]: Livro

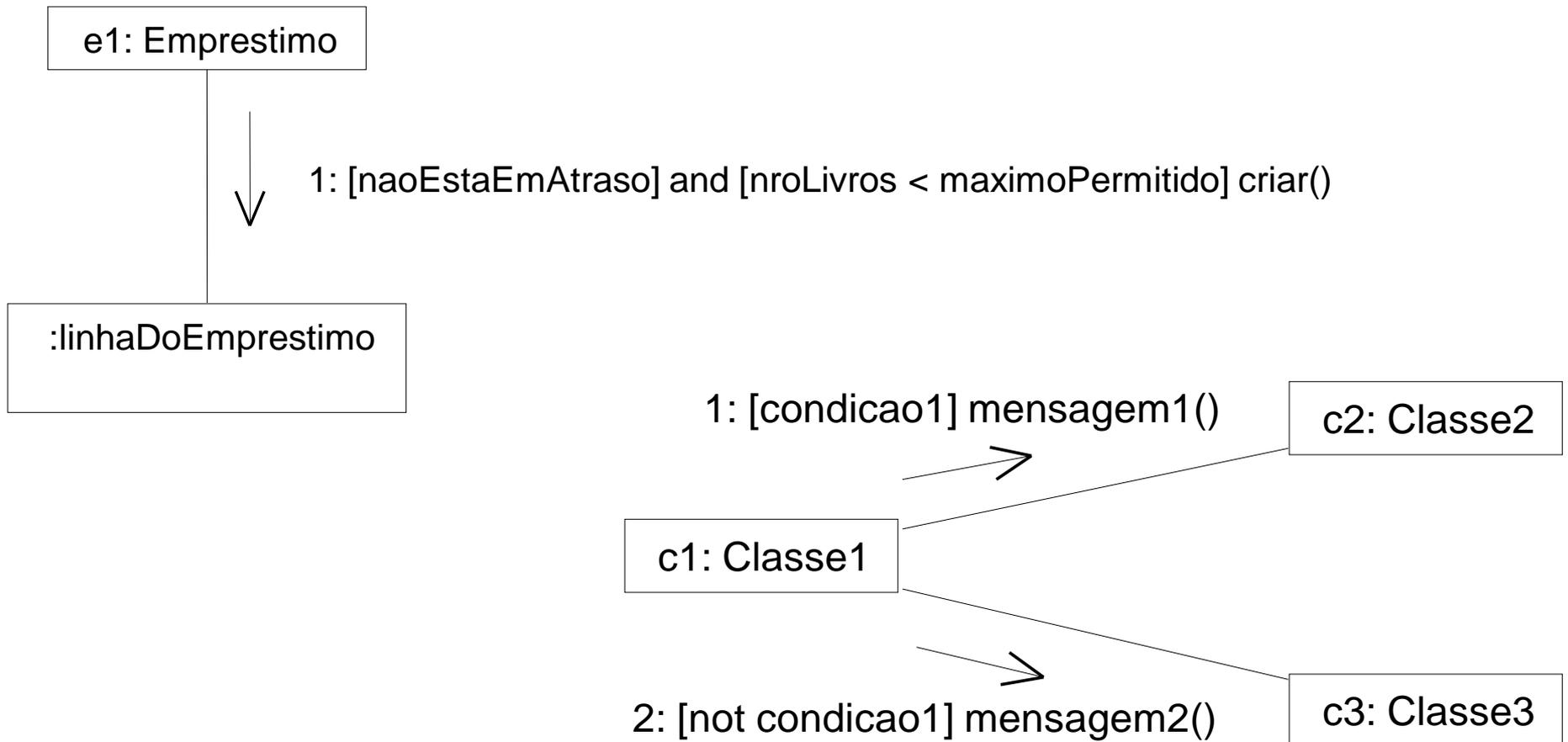
# Ordem das mensagens, condicionais e atribuição a variáveis



# Objeto como retorno de método



# Condicionais e Operadores lógicos e relacionais

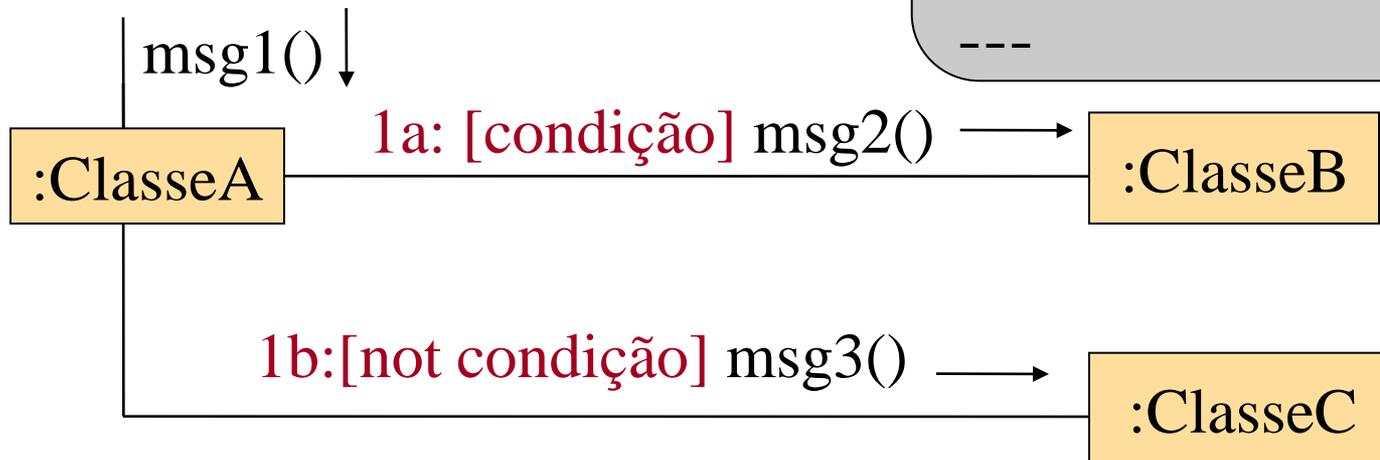


# Caminhos Condicionais Mutuamente Exclusivos

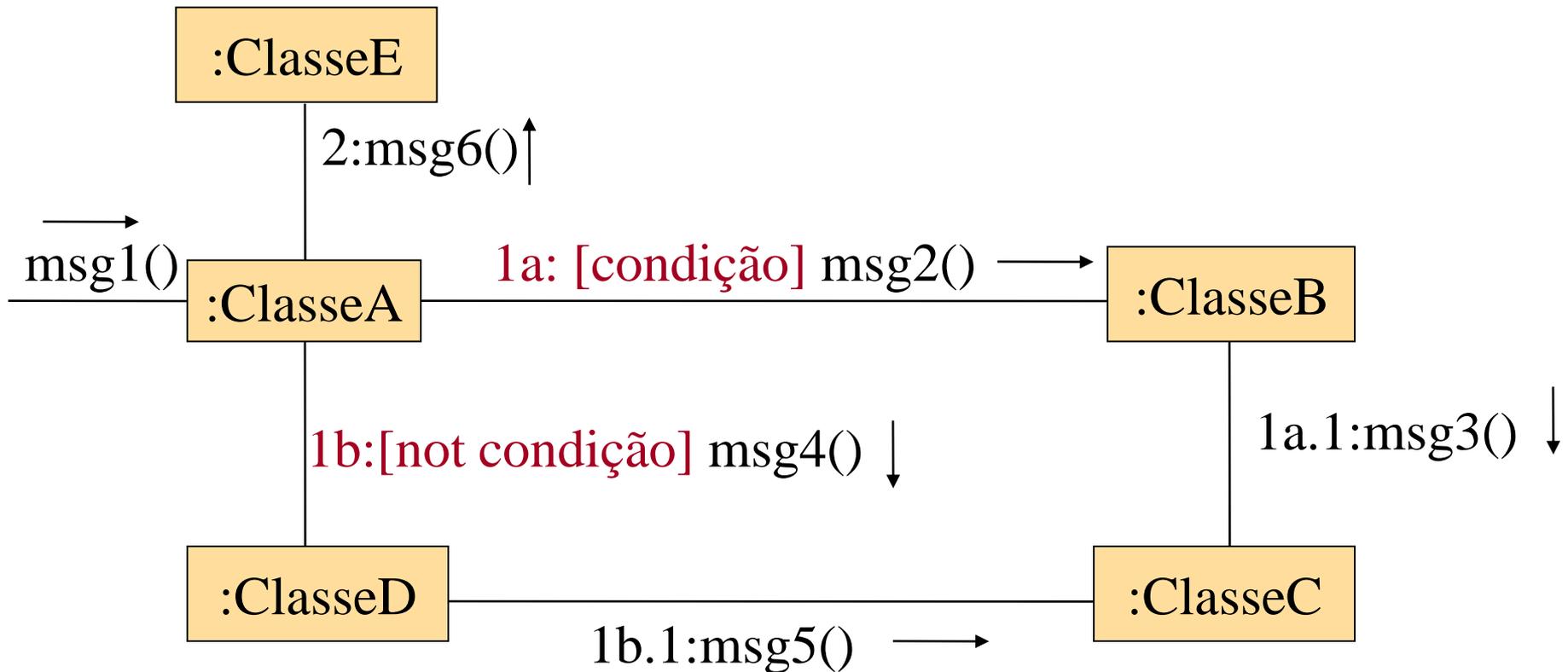
Apenas uma das mensagens é enviada, dependendo da condição ser verdadeira ou falsa

msg1 ()

```
---  
---  
if (condicao = true)  
    :ClasseB.msg2 ()  
else :ClasseC.msg3 ()  
---  
---
```



# 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

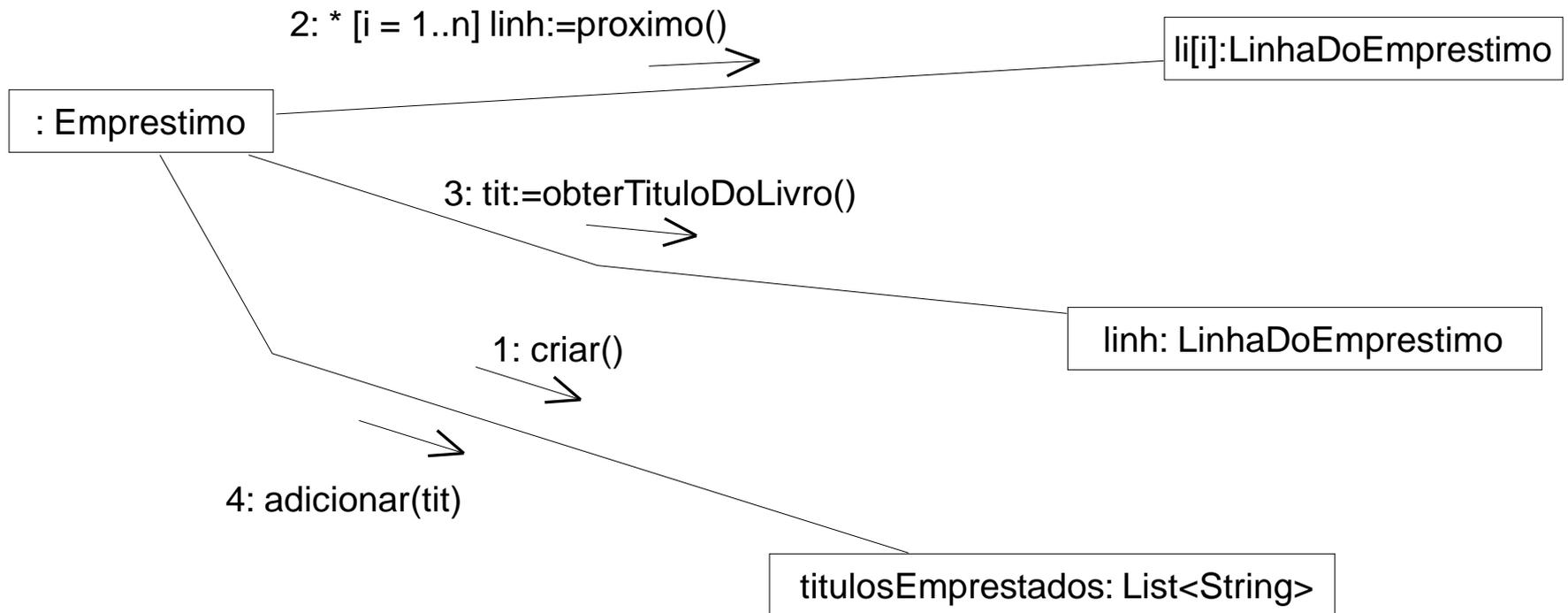
Em ambos os casos, a mensagem é enviada várias vezes,  
sempre para o mesmo objeto!!

# 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

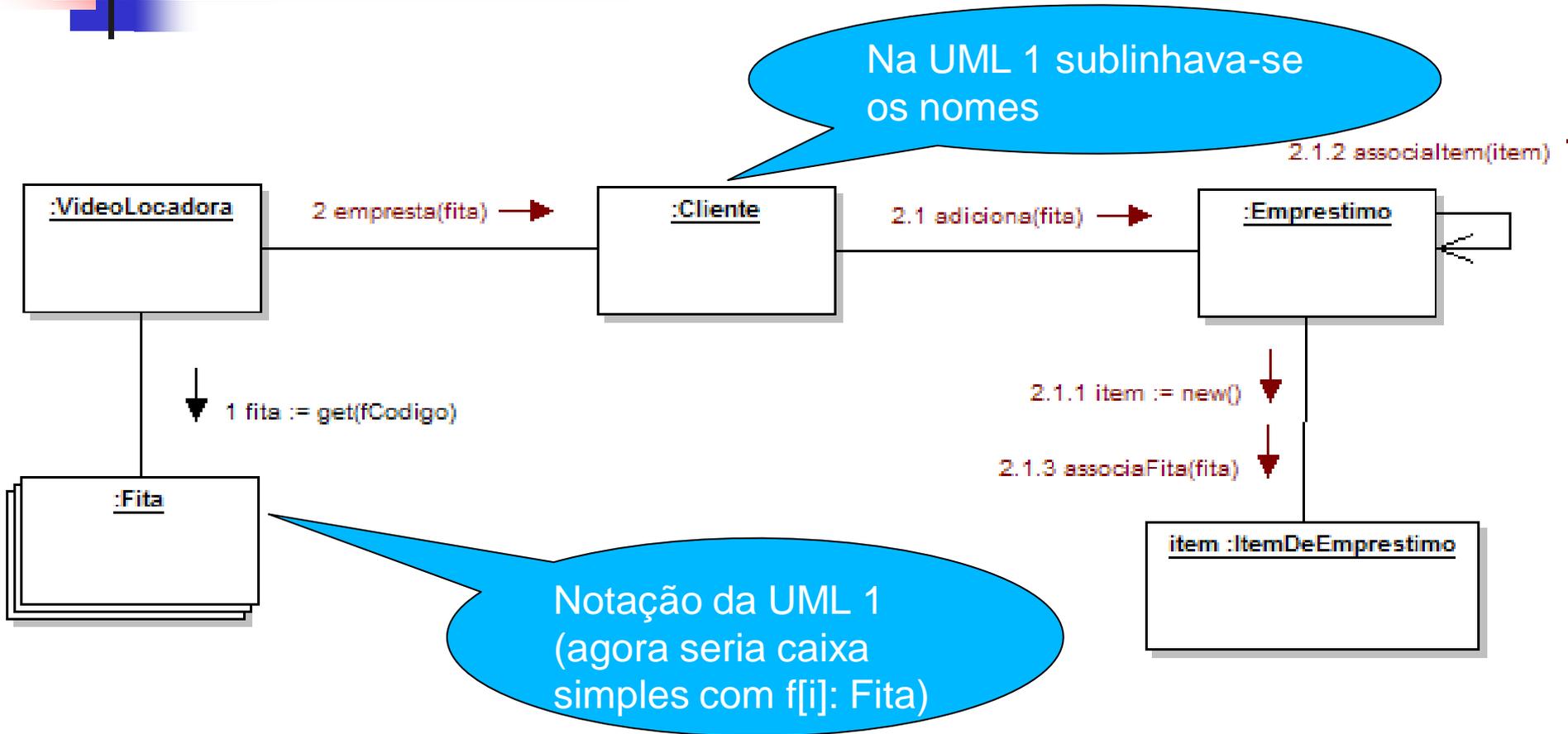
# Auto-mensagem

1: cons := ehDeConsulta()

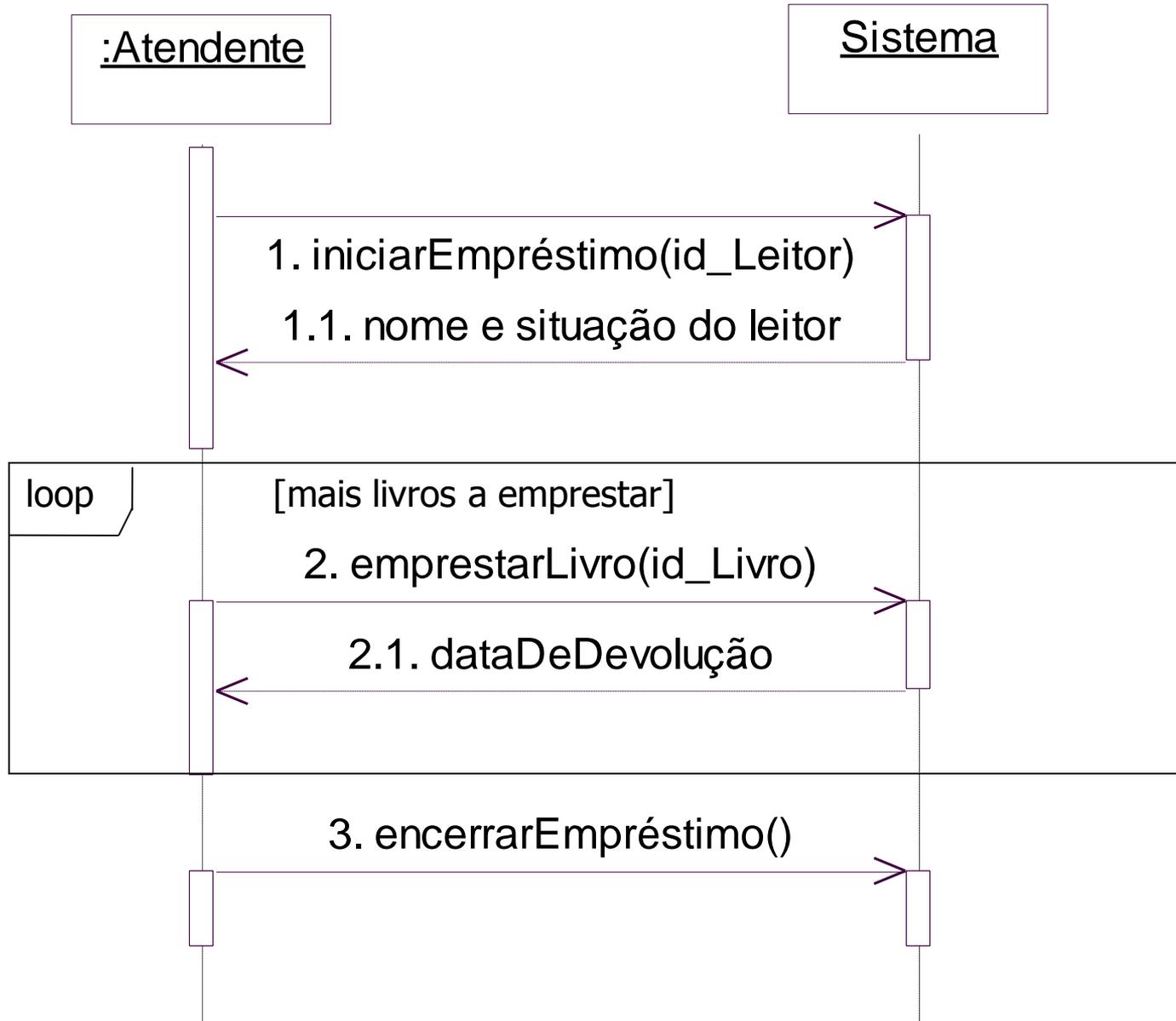


2: [not cons] mudarSituacao("emprestada")

# Exemplo de um DC – UML 1



OBS: algumas ferramentas ainda usam a notação da UML 1





# 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

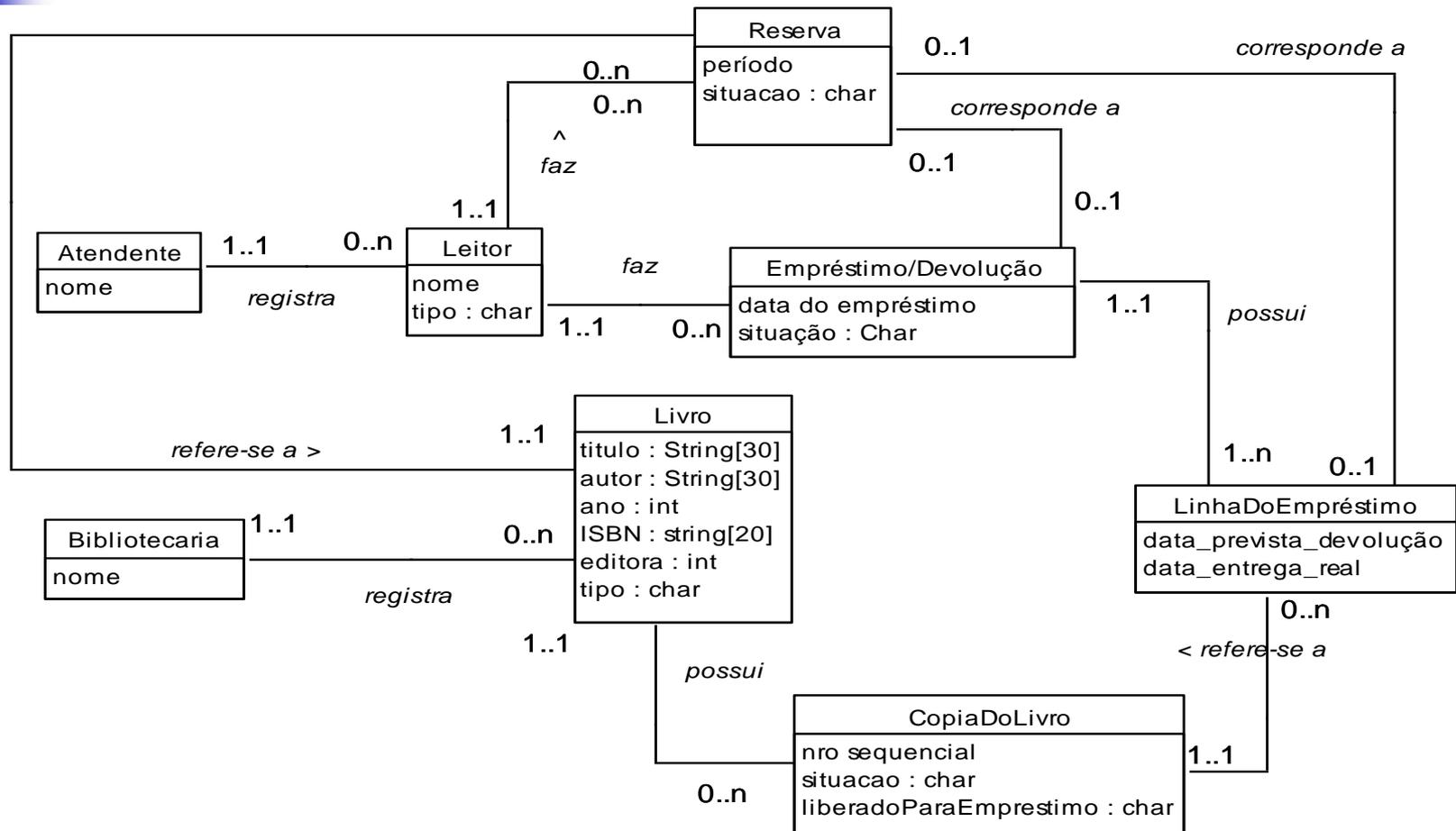


# Exemplo: Fazer um DC para a operação `iniciarEmprestimo()`

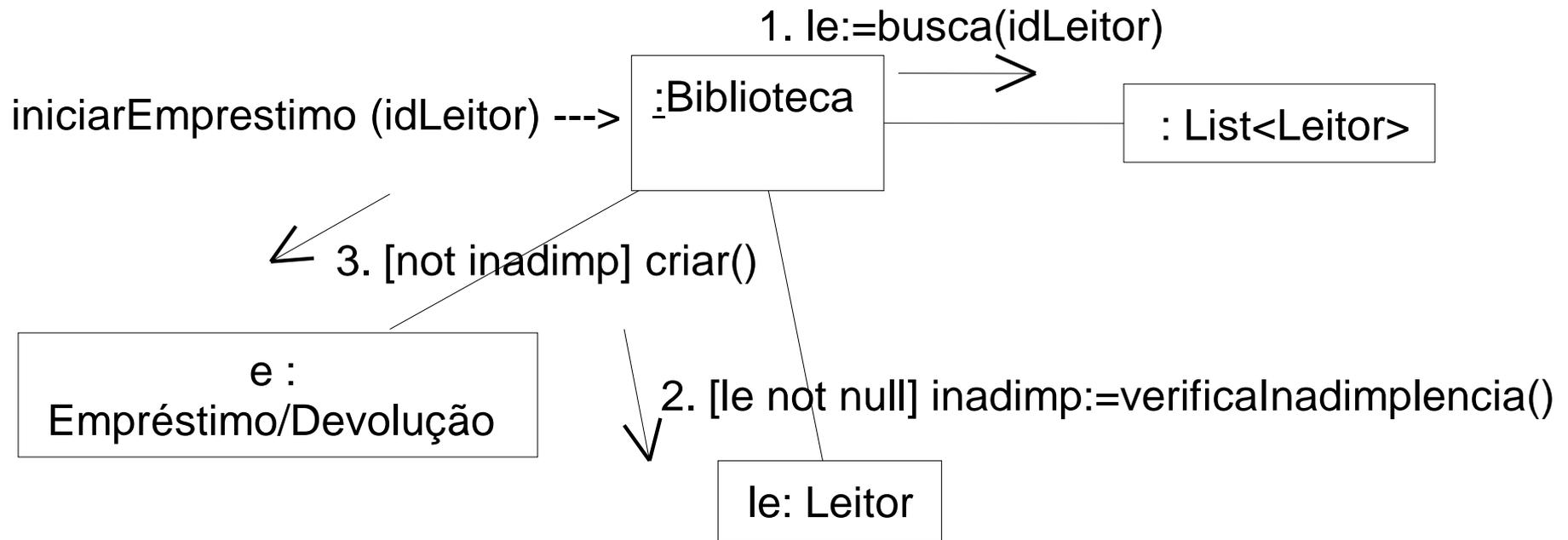
---

- 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



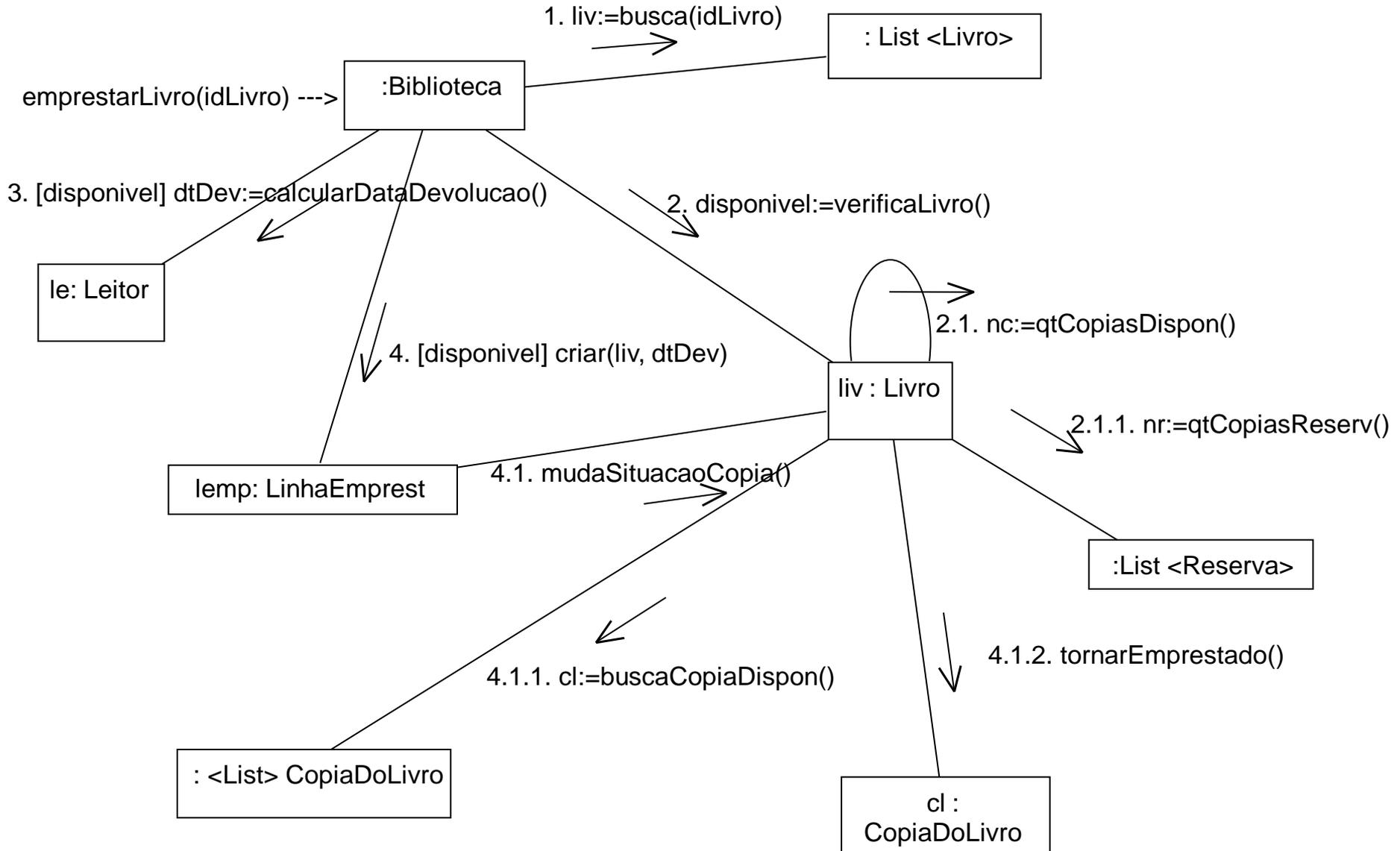


# Exemplo : Fazer um DC para a operação `emprestarLivro()`

---

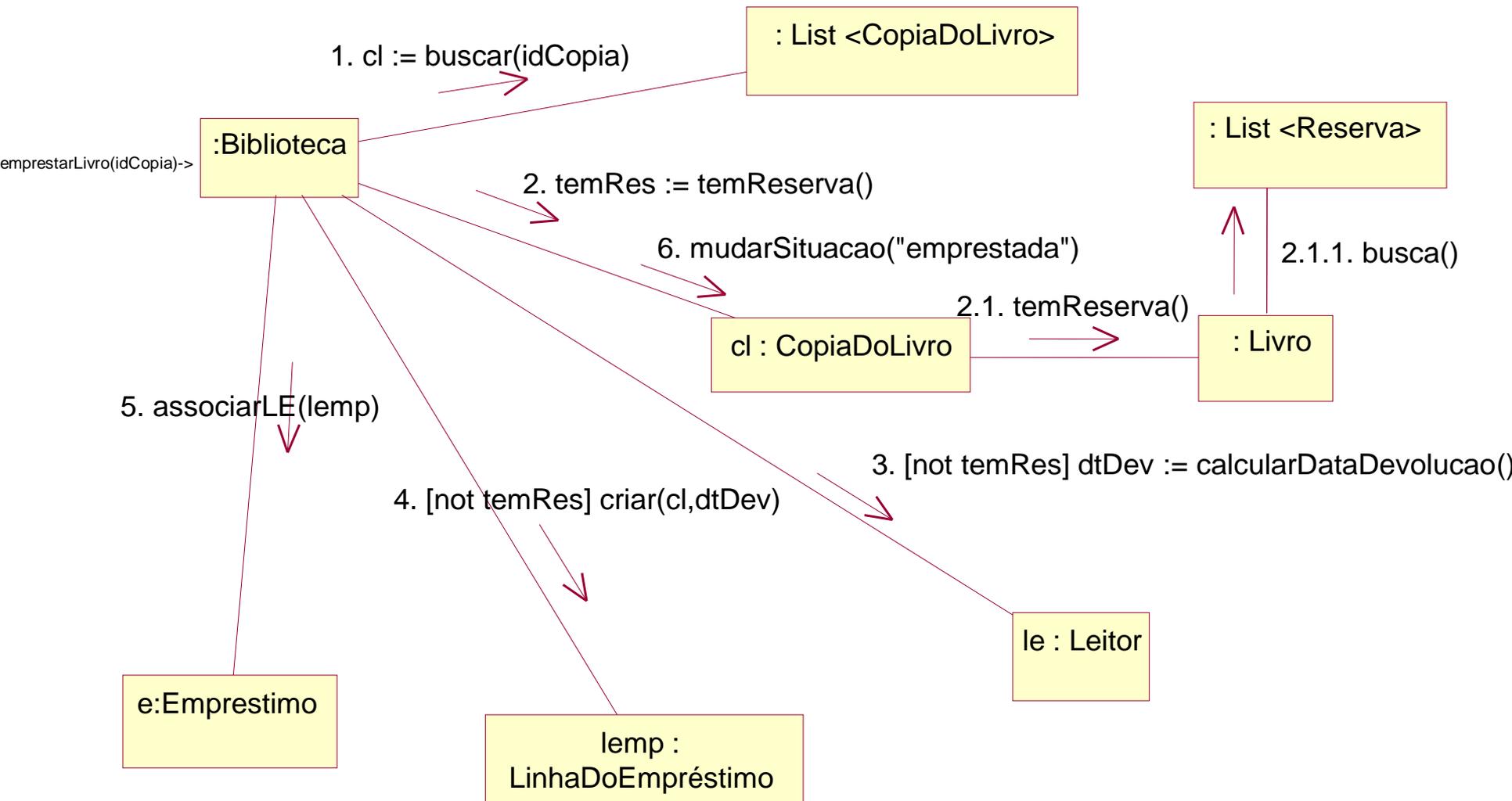
- **Operação:** `emprestarLivro(id_Livro)`
- **Referências Cruzadas:** Caso de uso: “Emprestar Livro”
- **Pré-Condições:**
  - Um leitor apto a emprestar livros já foi identificado e um empréstimo já foi criado;
- **Pós-Condições:**
  - um objeto `LinhaDoEmpréstimo` foi criado;
  - a `LinhaDoEmpréstimo` 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.

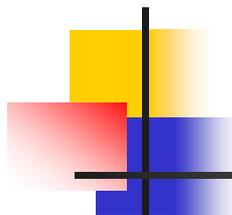
# Diagrama de Comunicação



Exercício: verificar se leitor já não tem o número máximo permitido de livros em seu poder

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



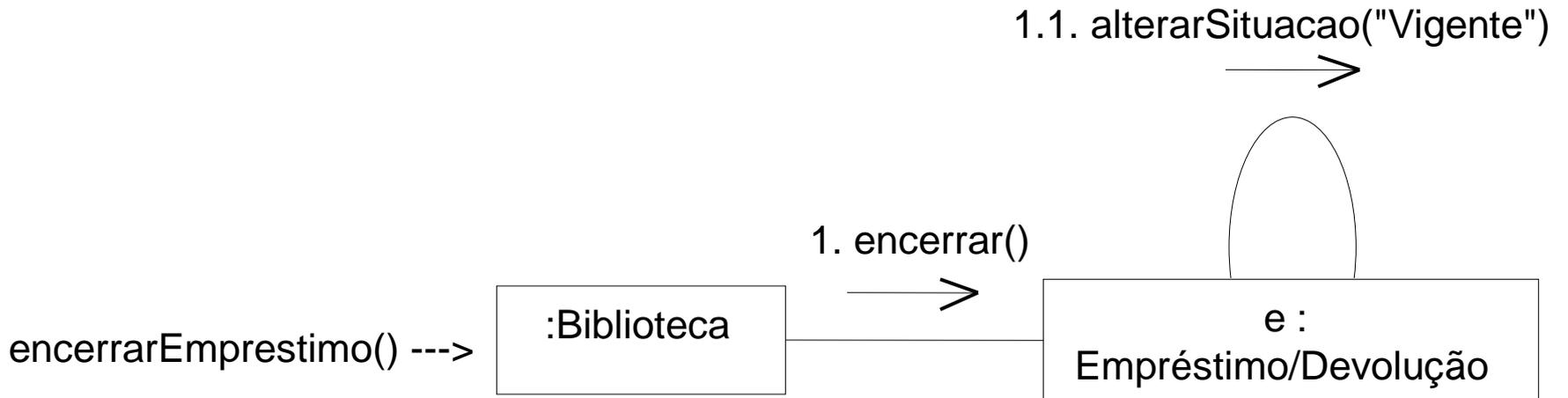


# Exemplo: Fazer um DC para a operação encerrarEmpréstimo()

---

- **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

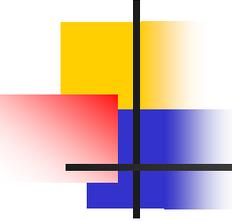




# Dicas ferramentas CASE

---

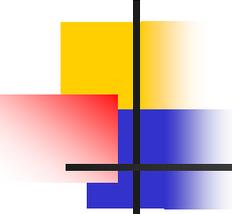
- Ferramenta StarUML:
  - as mensagens devem ser colocadas por cima do link entre objetos. Clique no link, daí clique no forwardStimulus e clique novamente sobre o link. Ele cria a mensagem e a seta, já com o número da mensagem
  - Se você esquecer uma mensagem e precisar reordenar, dê um clique duplo sobre a mensagem e aparecerá um quadradinho com o número, basta muda-lo.
- Ferramenta Rose:
  - Se precisar reordenar, alterne para o diagrama de sequência (F5) mude a ordem e depois volte ao diag. de comunicação



# Dicas adicionais

---

- Variáveis usadas no contexto do caso de uso podem ser consideradas conhecidas pelas próximas operações do mesmo caso de uso.
  - Pode-se passá-las como parâmetro para as próximas operações (melhor opção, já que a operação poderá ser invocada por diversos casos de uso)
  - Ou considerar que serão variáveis declaradas numa operação maior referente ao caso de uso em si.



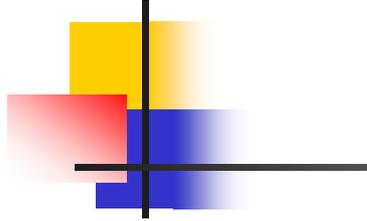
# Dicas adicionais

---

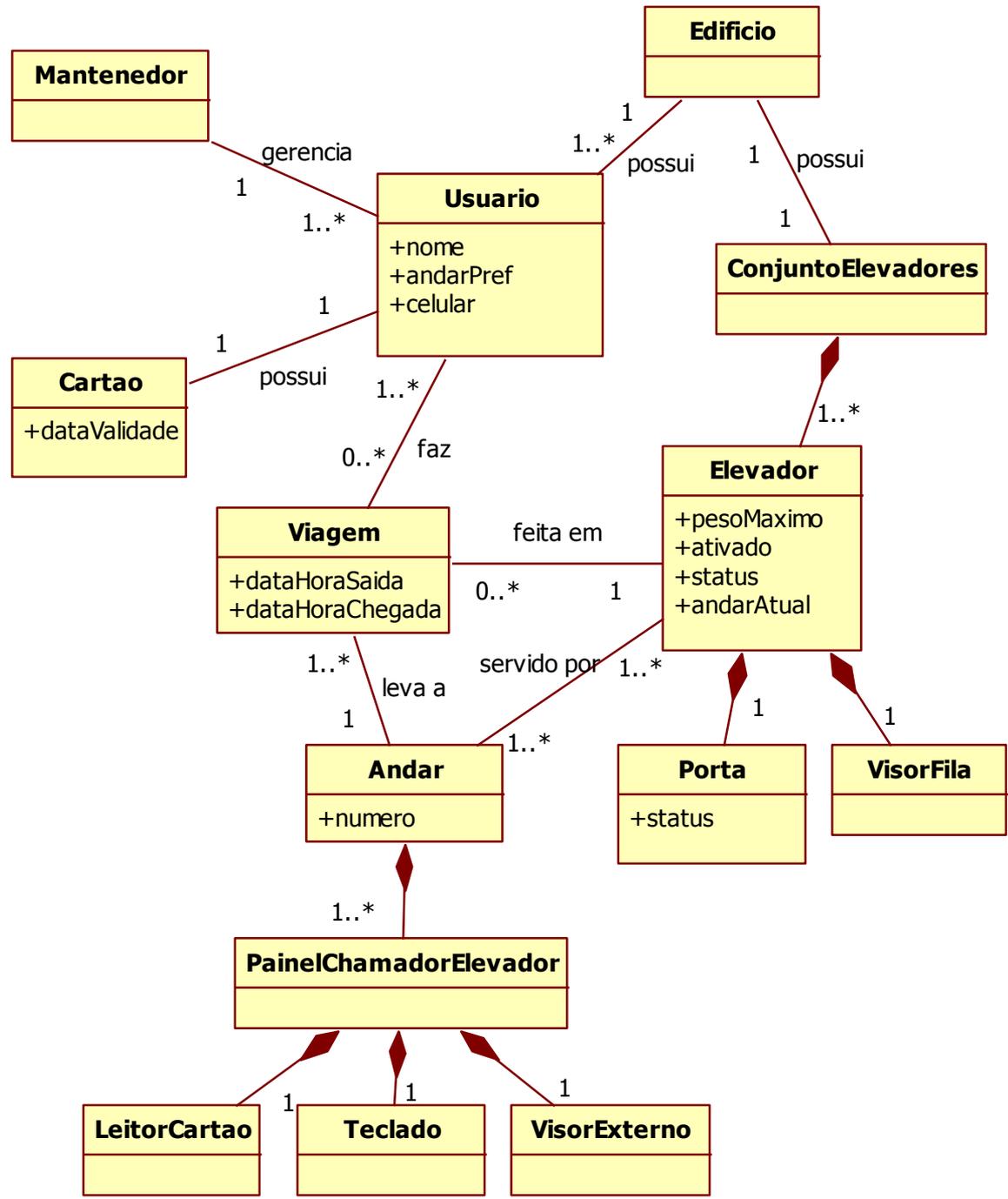
- Para associar um objeto a outro, há duas formas:
  - uma mensagem `associaObjeto(x)`, em que o objeto passado por parâmetro é associado ao objeto que recebeu a mensagem
  - Ou na criação do objeto já se passa o objeto a ser associado como parâmetro
- Respeite o encapsulamento: qualquer mudança de atributos tem que ser feita por método `set`.
- Lembrete: um diagrama de comunicação deve ser feito para cada uma das operações, e não um só diagrama para todo o caso de uso.

# Diagramas de comunicação – Sistema Elevador

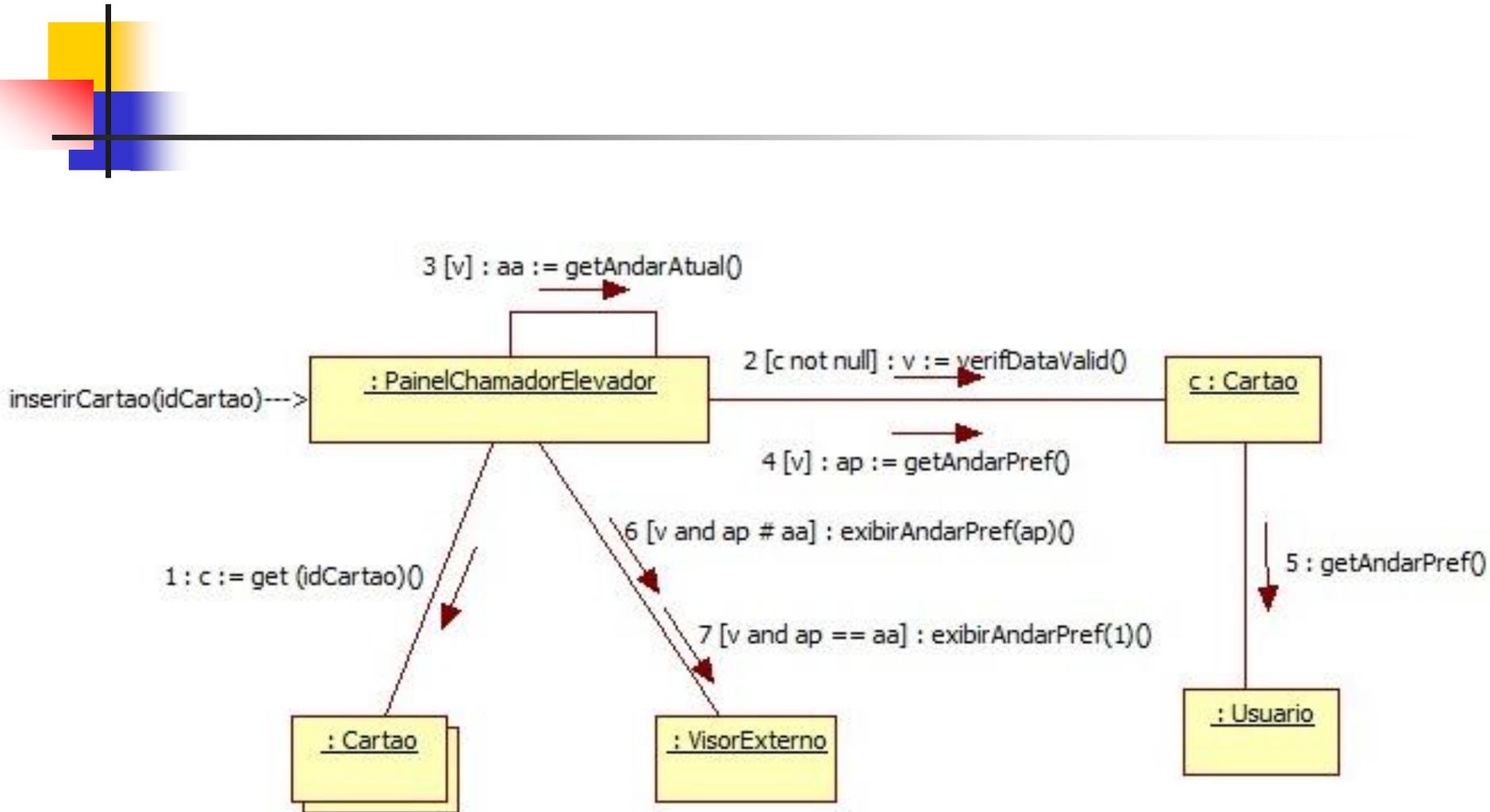
- Para iniciar, temos que:
  - olhar o contrato e ver os parâmetros, pré e pós condições
  - Olhar o modelo conceitual e identificar o que muda no estado do sistema: objetos criados, associações feitas, atributos modificados
- Contrato
  - Operação: inserirCartao (idCartao)
  - Pré-Condições: O usuário é cadastrado no sistema e possui um cartão de acesso.
  - Pós-Condições: Se o cartão tiver não tiver expirado então:
    - O usuário associado ao cartão foi identificado no sistema.
    - O visorExterno exibiu o andar preferencial do usuário.



# Modelo conceitual



# Diag. de comunicação: inserirCartao



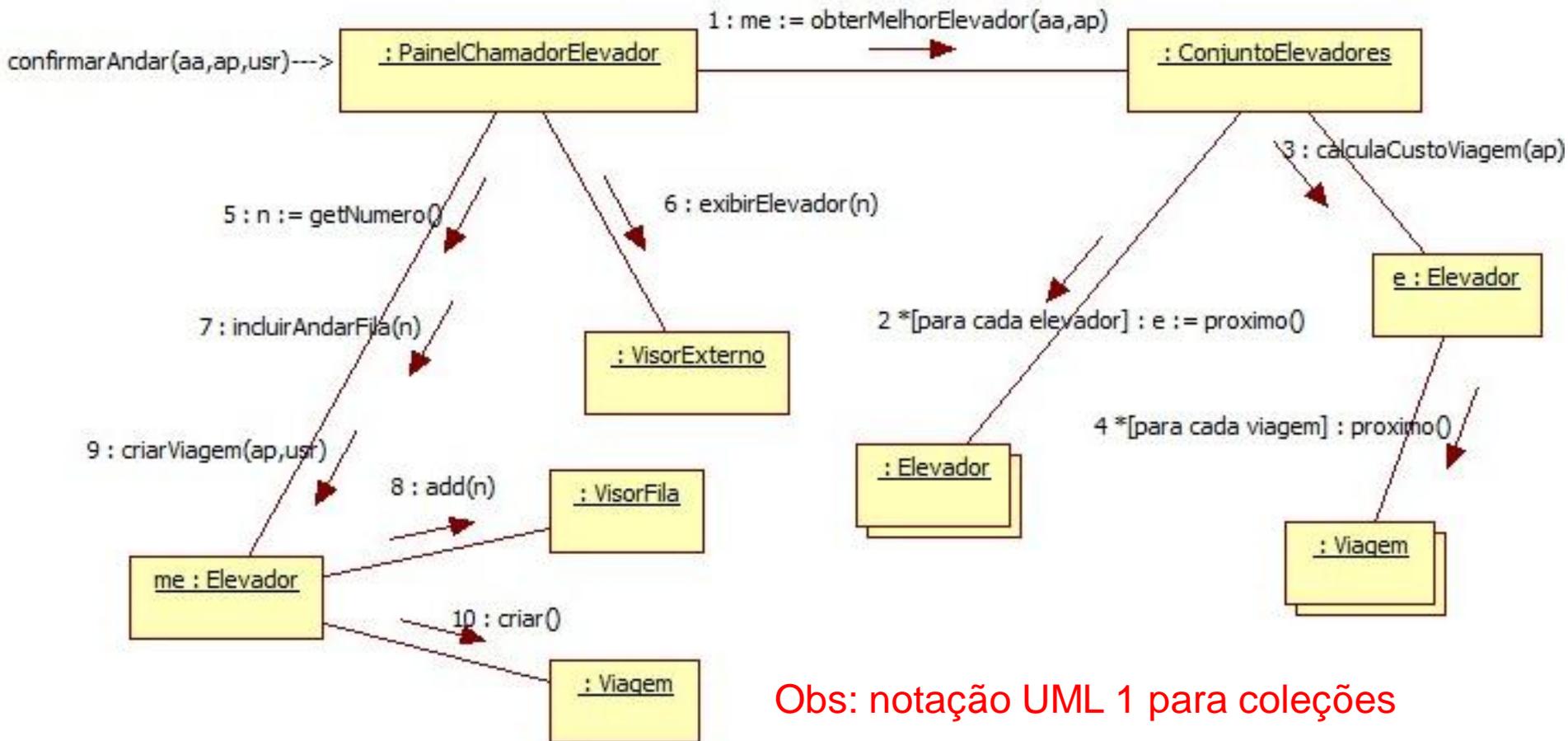
Obs: notação UML 1 para coleções

# Diagramas de comunicação – Sistema Elevador

## ■ Contrato

- Operação: confirmarAndar
- Parâmetros: andarOrigem, andarDestino
- Referências Cruzadas: Caso de Uso: Requisitar Elevador
- Pré-Condições: O Usuário já foi identificado no sistema e selecionou seu andar destino.
- Pós-Condições:
  - Foi obtido o elevador com maior disponibilidade de chegar antes ao andar destino, porém respeitando as restrições de peso máximo e economia de energia.
  - Foi criado um objeto Viagem e associado ao Usuario, ao Andar e ao Elevador.
  - Foi exibida no visor externo do chamador de elevador a letra do melhor elevador que atinja o andar destino.
  - O número andarDestino foi incluído no visor de fila do elevador.

# Diag. de comunicação: confirmarAndar



Obs: notação UML 1 para coleções