

# **PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML**

2ª EDIÇÃO

- Modelagem de Interações

Eduardo Bezerra  
Editora Campus/Elsevier



# CAPÍTULO 7

## MODELAGEM DE INTERAÇÕES

*“Somente após a construção de diagramas de interação para os cenários de um caso de uso, pode-se ter certeza de que todas as responsabilidades que os objetos devem cumprir foram identificadas”*

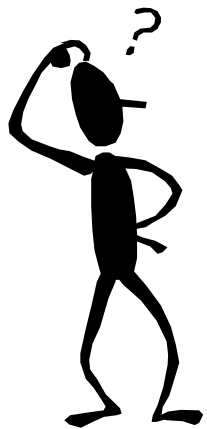
*-Ivar Jacobson.*

# TÓPICOS

- Introdução
- Diagrama de seqüência
- Construção do modelo de interações
- Modelo de interações em um processo iterativo

# INTRODUÇÃO

- O objetivo dos modelos vistos até agora é fornecer um entendimento do problema correspondente ao SSOO a ser desenvolvido.
- Entretanto, esses modelos deixam algumas perguntas sem respostas.
- No modelo de casos de uso:
  - Quais são as operações que devem ser executadas internamente ao sistema?
  - A que classes estas operações pertencem?
  - Quais objetos participam da realização deste caso de uso?



# INTRODUÇÃO

- No modelo de classes de análise:
  - De que forma os objetos colaboram para que um determinado caso de uso seja realizado?
  - Em que ordem as mensagens são enviadas durante esta realização?
  - Que informações precisam ser enviadas em uma mensagem de um objeto a outro?
  - Será que há responsabilidades ou mesmo classes que ainda não foram identificadas?
- Sessões CRC pode ajudar a identificar quais são as responsabilidades de cada objeto e com que outros objetos ele precisa colaborar.
- Mas sessões CRC não fornecem um modo de documentar essas interações.



# INTRODUÇÃO

- Para responder às questões anteriores, o **modelo de interações** deve ser criado.
- Esse modelo representa mensagens trocadas entre objetos para a execução de cenários dos casos de uso do sistema.
- A construção dos **diagramas de interação** é uma consolidação do entendimento dos aspectos dinâmicos do sistema, iniciado nas sessões CRC.
- A modelagem de interações é uma parte da **modelagem dinâmica** de um SSOO.

Diagramas de interação representam como o sistema age internamente para que um ator atinja seu objetivo na realização de um caso de uso. A modelagem de um SSOO normalmente contém diversos diagramas de interação. O conjunto de todos os diagramas de interação de um sistema constitui o seu **modelo de interações**.

# INTRODUÇÃO

- Os objetivos da construção do modelo de interação são:
  1. Obter informações adicionais para completar e aprimorar outros modelos (principalmente o modelo de classes)
    - Quais as operações de uma classe?
    - Quais os objetos participantes da realização de um caso de uso (ou cenário deste)?
    - Para cada operação, qual a sua assinatura?
    - Uma classe precisa de mais atributos?
  2. Fornecer aos programadores uma visão detalhada dos objetos e mensagens envolvidos na realização dos casos de uso.



# MENSAGEM

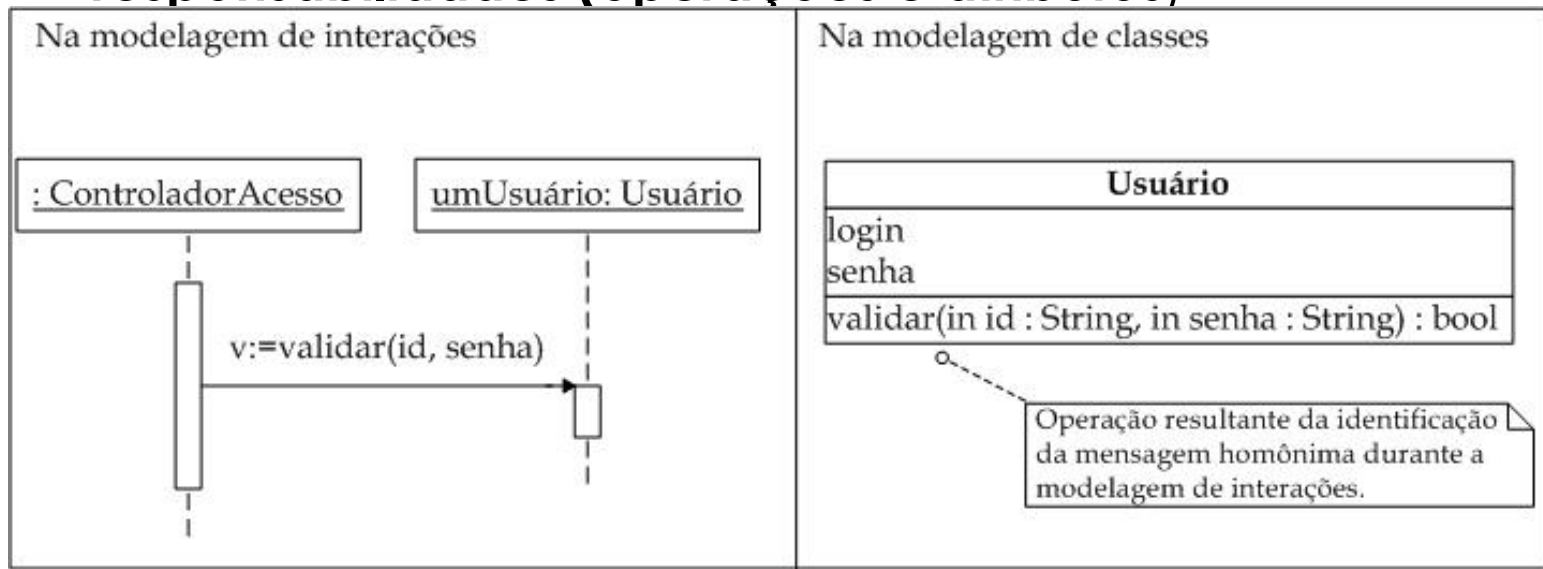
- O conceito básico da interação entre objetos é a *mensagem*.
- Um sistema OO é uma rede de objetos que trocam mensagens.
  - Funcionalidades são realizadas pelos objetos, que só podem interagir através de mensagens.
  - Um objeto envia uma mensagem para outro objeto quando o primeiro deseja que o segundo realize alguma tarefa.
- O fato de um objeto “precisar de ajuda” indica a necessidade de este enviar mensagens.
- Na construção de diagramas de interação, mensagens de um objeto a outro implicam em operações que classes devem ter.

Uma mensagem representa a requisição de um objeto remetente a um objeto receptor para que este último execute alguma operação definida para sua classe. Essa mensagem deve conter informação suficiente para que a operação do objeto receptor possa ser executada.



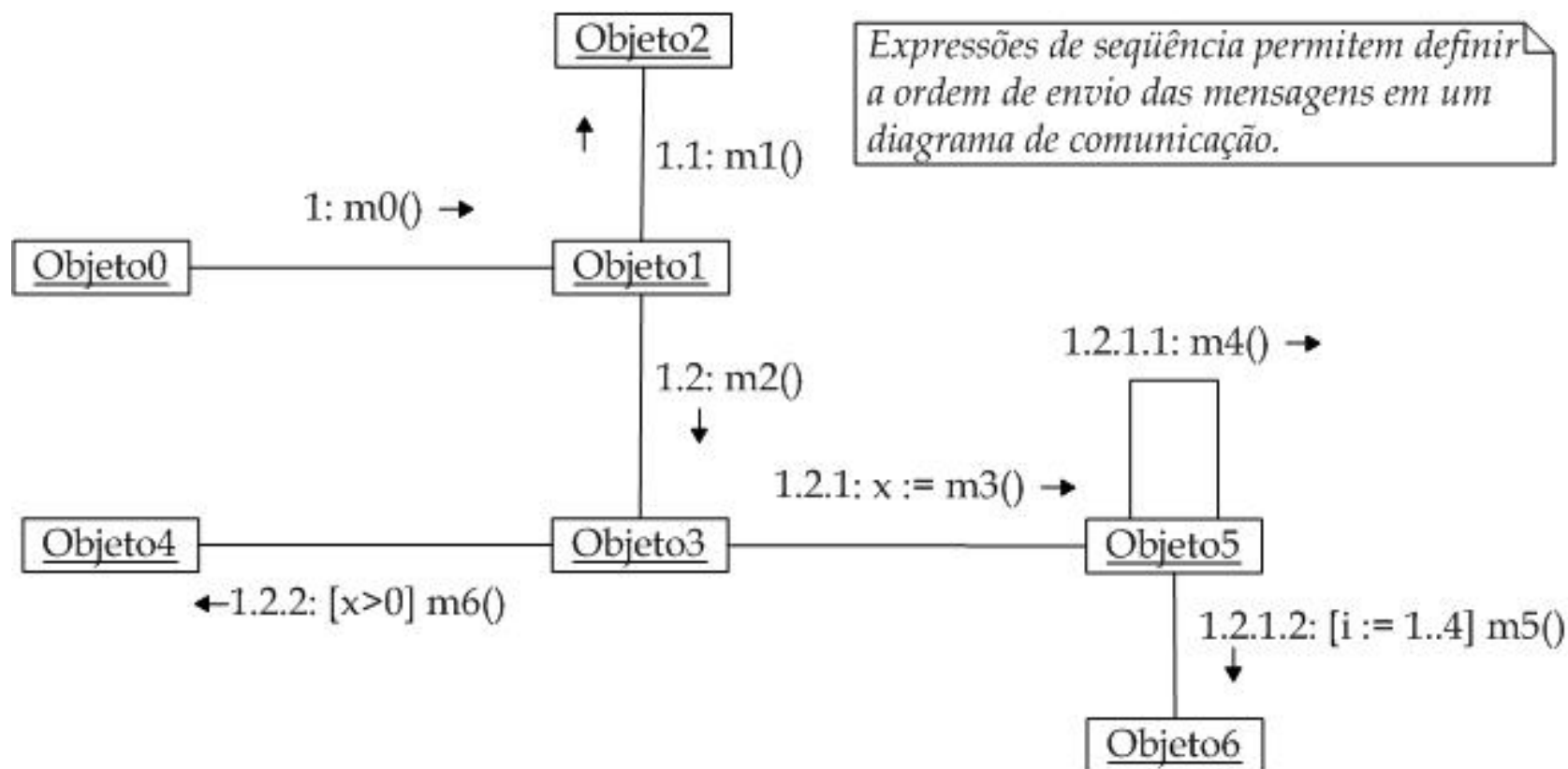
# MENSAGENS VERSUS RESPONSABILIDADES

- Qual o objetivo da construção dos diagramas de interação?
  - Identificar **mensagens** e, em última análise, **responsabilidades (operações e atributos)**

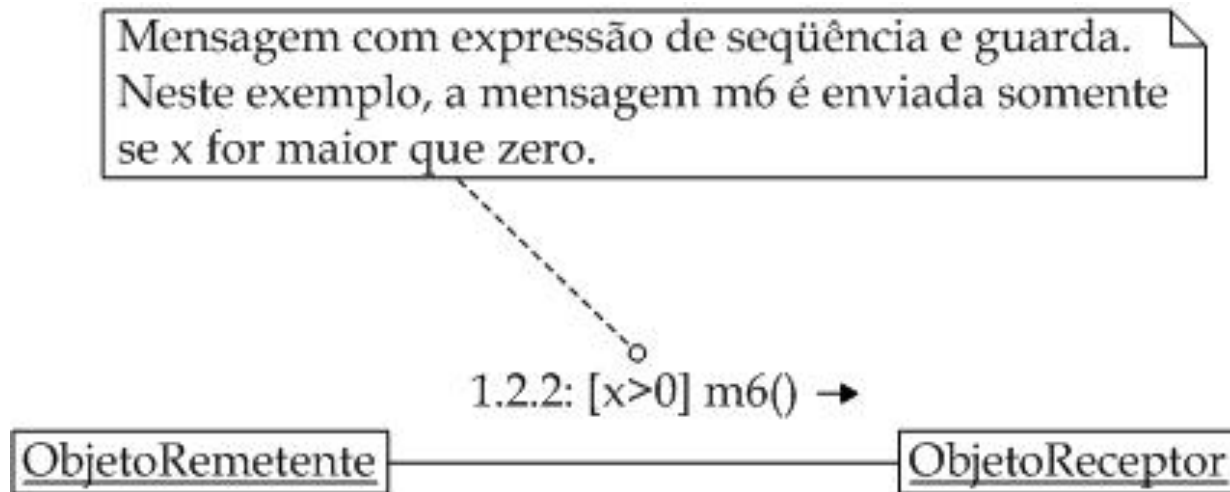


Uma mensagem implica na existência de uma operação no objeto receptor. A resposta do objeto receptor ao recebimento de uma mensagem é a execução da operação correspondente.

# EXEMPLOS (SINTAXE UML PARA MENSAGENS)



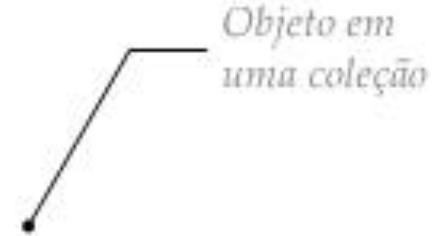
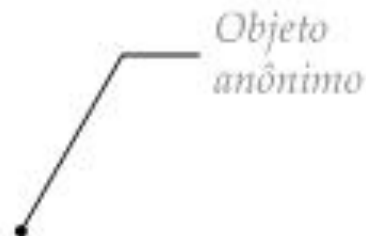
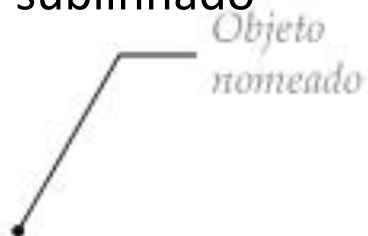
# EXEMPLOS (SINTAXE UML PARA MENSAGENS)



# NOTAÇÃO PARA OBJETOS

•12

- Objetos são representados em um diagrama de interação utilizando-se a mesma notação do diagrama de objetos.
- Pode-se representar objetos anônimos ou objetos nomeados, dependendo da situação.
- Elementos de uma coleção também podem ser representados.
- Classes também podem ser representadas.
  - Para o caso de mensagens enviadas para a classe.
  - Uma mensagem para uma classe dispara a execução de uma *operação estática*.
  - A representação de uma classe em um diagrama de seqüência é a mesma utilizada para objetos, porém o nome da classe não é sublinhado



# TIPOS DE DIAGRAMA DE INTERAÇÃO <sup>•13</sup>

- Há três tipos de diagrama de interação na UML 2.0: **diagrama de seqüência**, **diagrama de comunicação** e **diagrama de visão geral da interação**.
  - O diagrama de seqüência e o diagrama de comunicação são equivalentes.

***Diagrama de seqüência:*** foco nas mensagens enviadas no decorrer do tempo.

***Diagrama de comunicação:*** foco nas mensagens enviadas entre objetos que estão relacionados.

***Diagrama de visão geral de interação.*** Pode ser utilizado para apresentar uma visão geral de diversas interações entre objetos, cada uma delas representada por um diagrama de interação. Diagrama é útil para ***modularizar*** a construção do diagramas de seqüência (ou de comunicação).



## 7.2 DIAGRAMA DE SEQÜÊNCIA



# DIAGRAMA DE SEQUÊNCIA

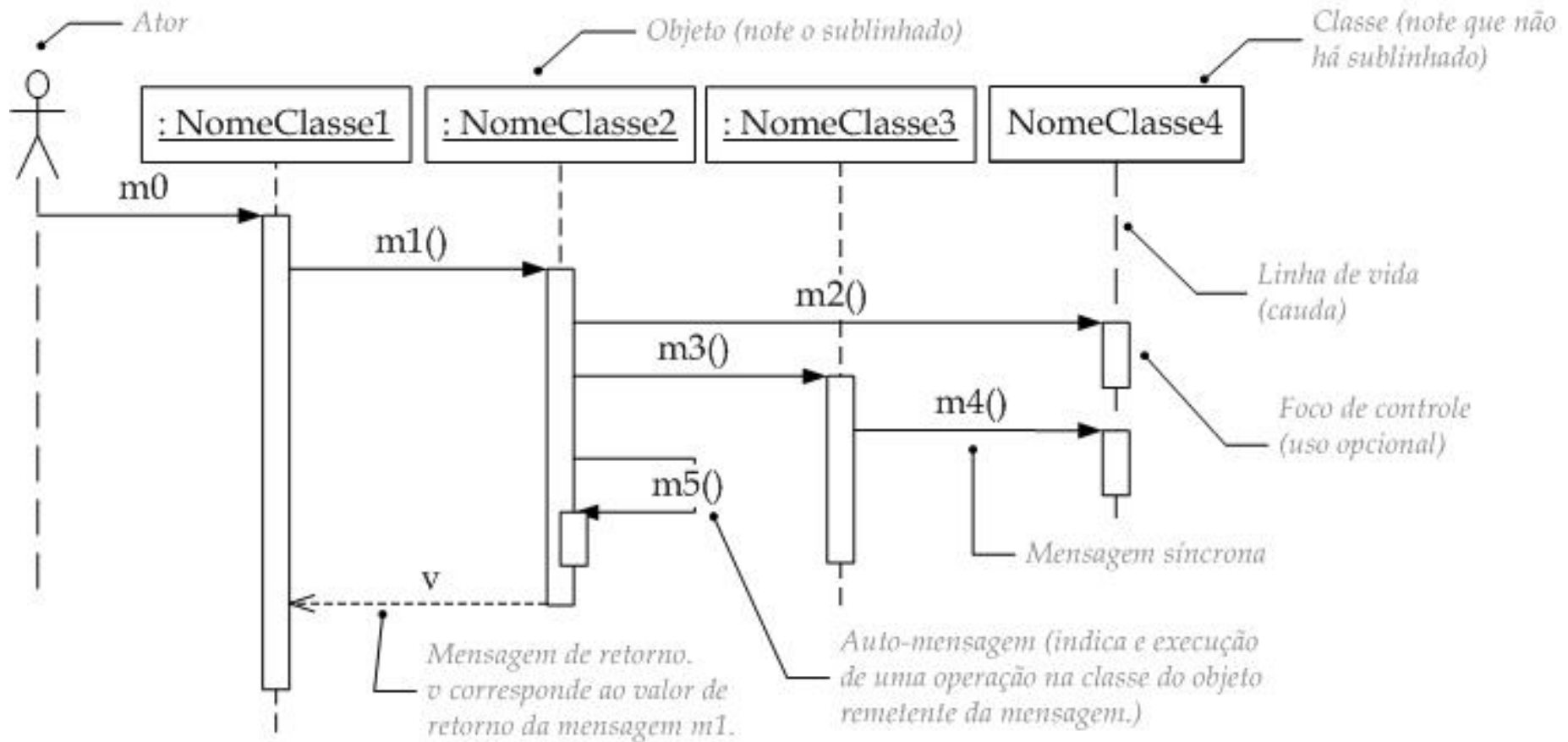
- Os objetos participantes da interação são organizados na horizontal.
- Abaixo de cada objeto existe uma linha (linha de vida)
- Cada linha de vida possui o seu foco de controle.
  - Quando o objeto está fazendo algo.
- As mensagens entre objetos são representadas com linhas horizontais rotuladas partindo da linha de vida do objeto remetente e chegando a linha de vida do objeto receptor.
- A posição vertical das mensagens permite deduzir a ordem na qual elas são enviadas.
- Ordem de envio de mensagens em um diagrama de sequência pode ser deduzida a partir das expressões de sequência.
- Criação e destruição de objetos podem ser representadas.



# ELEMENTOS GRÁFICOS DE UM DS

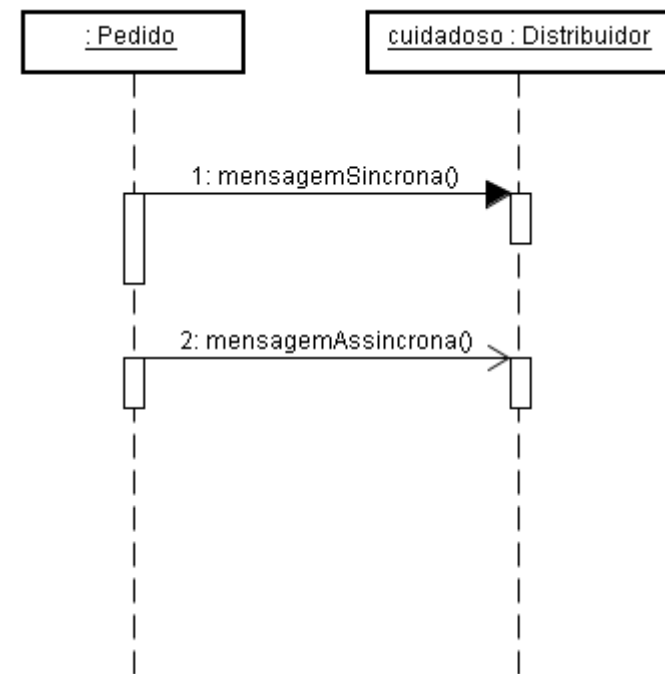
- Elementos básicos em um diagrama de seqüência:
  - Atores
  - Objetos, multiobjetos e classes
  - Mensagens
  - Linhas de vida e focos de controle
  - Criação e destruição de objetos
  - Iterações

# ELEMENTOS GRÁFICOS DE UM DS



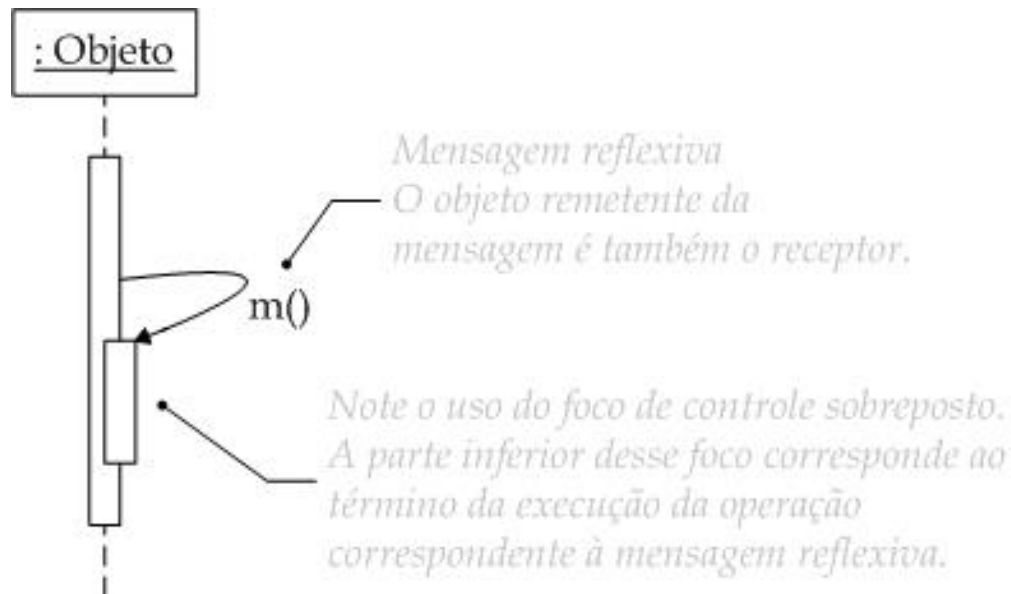
# CHAMADA SÍNCRONA X ASSÍNCRONA

- É possível utilizar dois tipos de chamada de métodos no diagrama de seqüência:
  - Chamada síncrona (seta cheia): a execução fica bloqueada até o retorno do método
  - Chamada assíncrona (seta vazia): a execução continua em paralelo ao método que foi chamado (*fork* implícito)

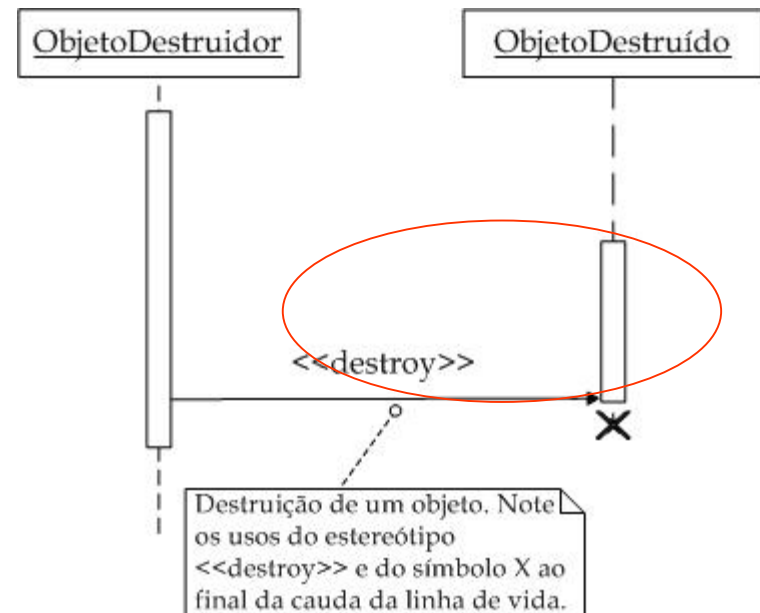
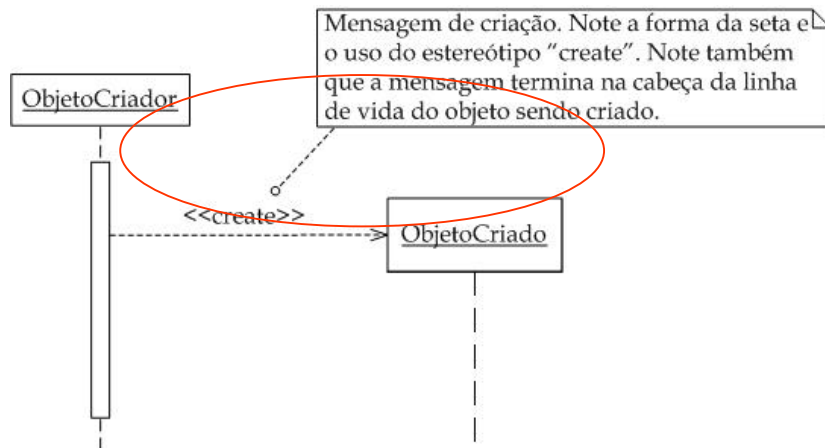


# MENSAGENS REFLEXIVAS EM UM DS

- Em uma mensagem reflexiva (ou **auto-mensagem**) o remetente é também o receptor.
  - Corresponde a uma mensagem para this (self).
  - O que isso significa na prática?

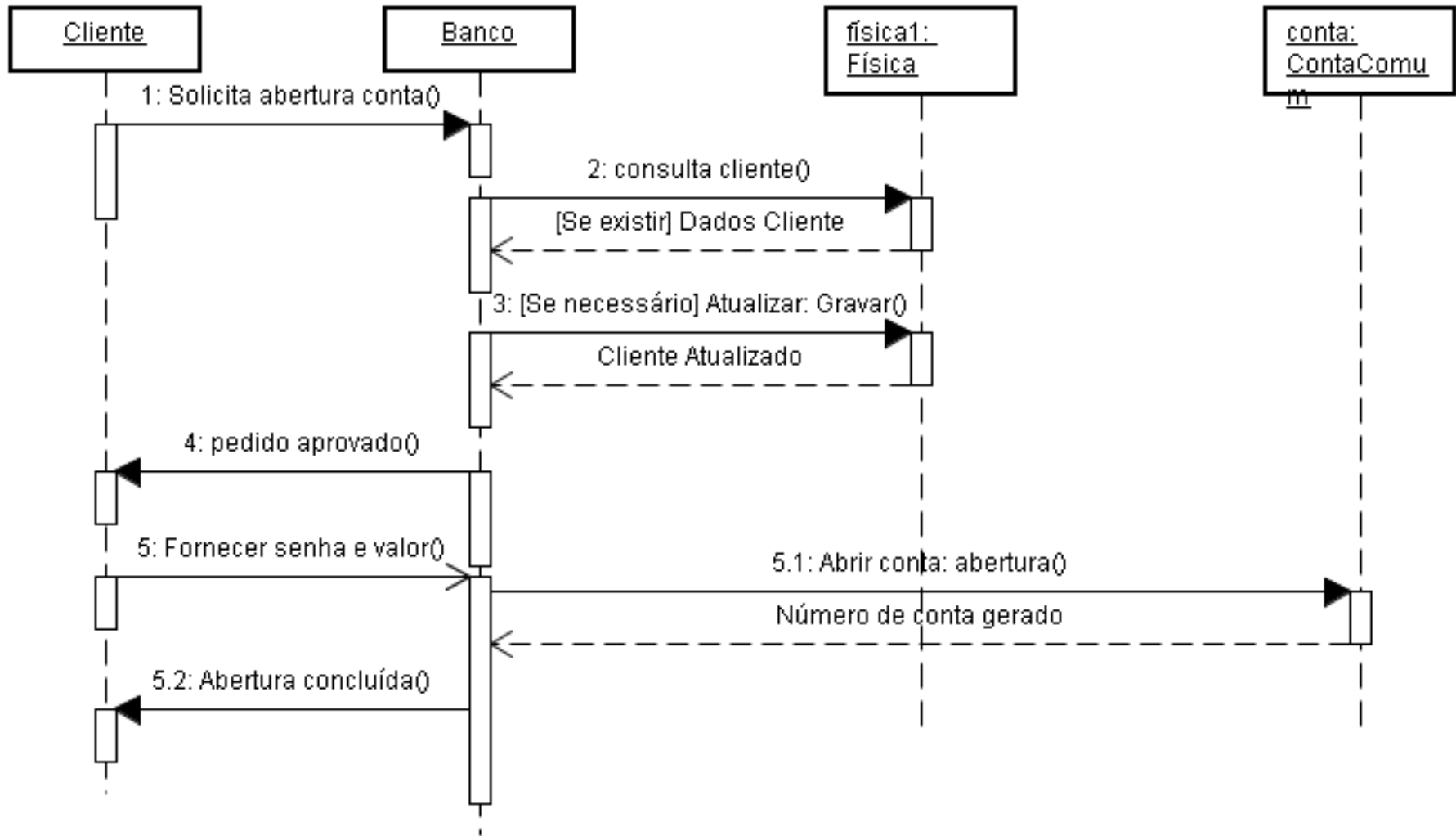


# CRIAÇÃO/DESTRUIÇÃO DE OBJETOS EM UM DS

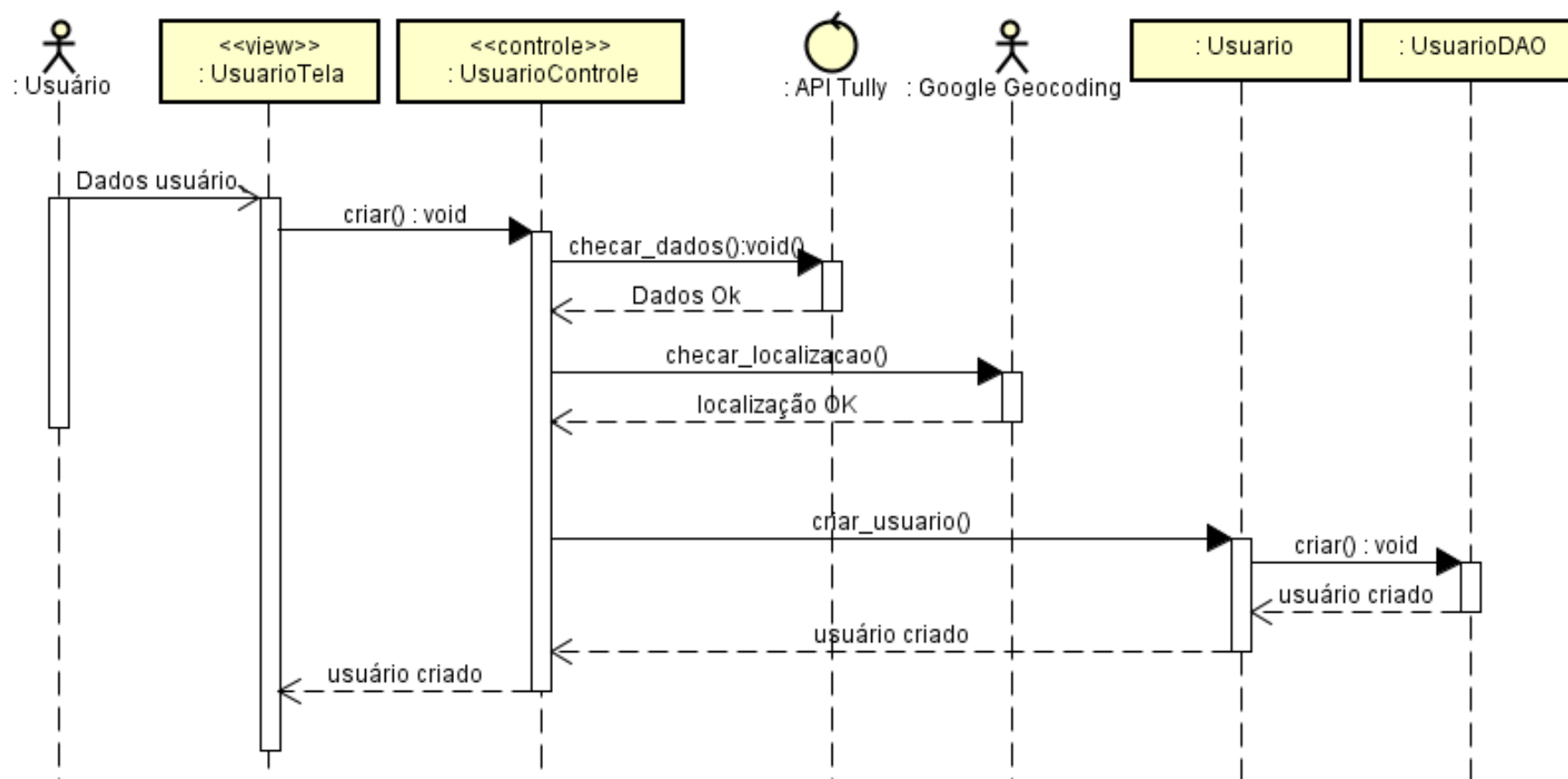


# EXEMPLO

•21



sd Diagrama de Sequência Fluxo Principal Cadastrar Usuário



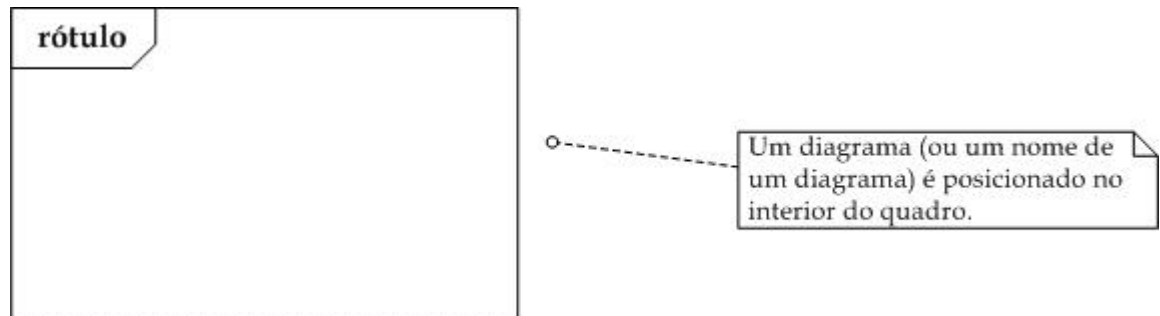




## 7.3 MODULARIZAÇÃO DE INTERAÇÕES

# QUADROS DE INTERAÇÃO

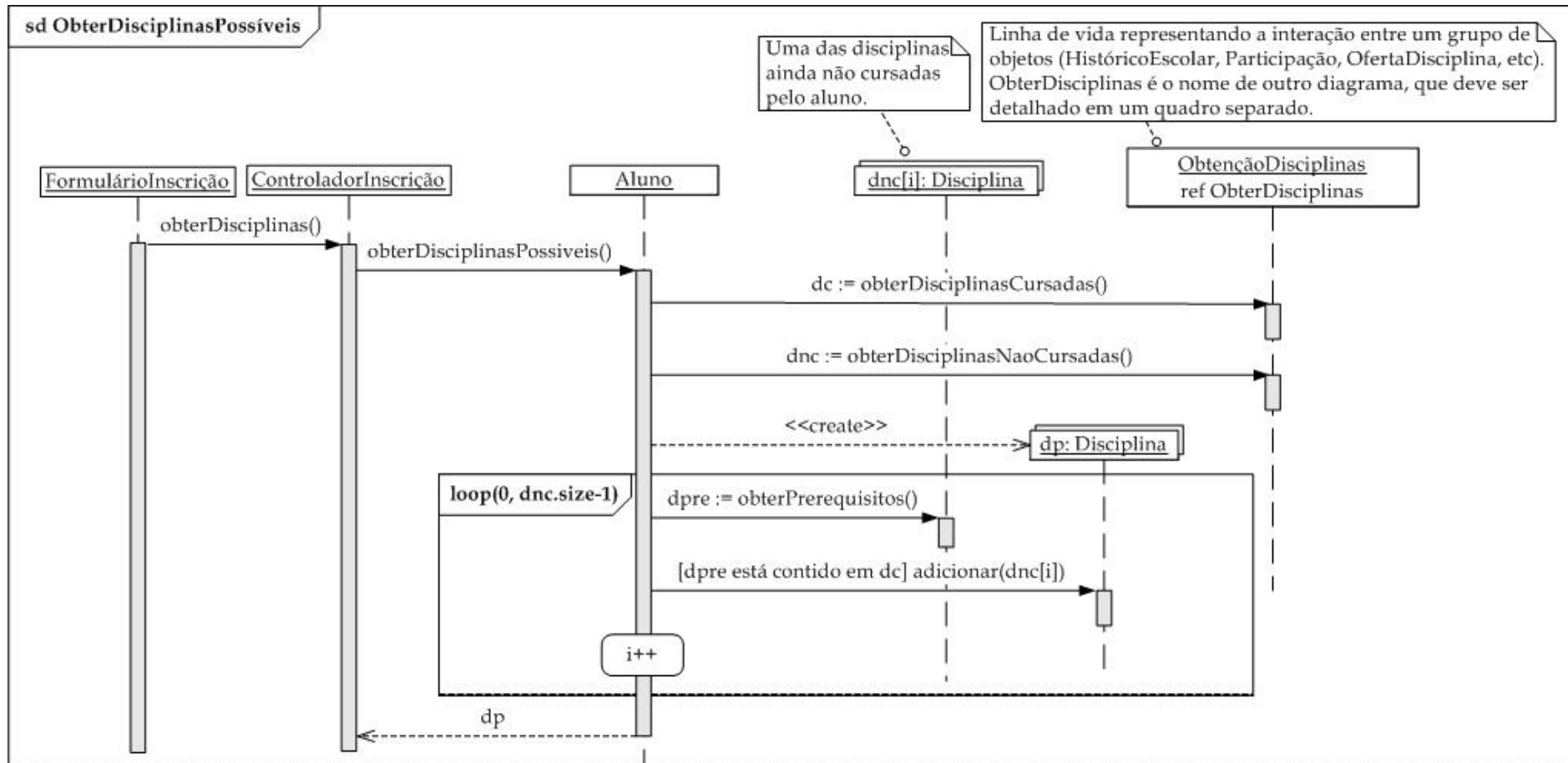
- Elemento gráfico, que serve para modularizar a construção de diagramas de seqüência (ou de comunicação).
- Objetivos específicos:
  - Dar um nome ao diagrama que aparece dentro do quadro;
  - Fazer referência a um diagrama definido separadamente;
  - Definir o fluxo de controle da interação.
- Notação:



# DIAGRAMAS NOMEADOS

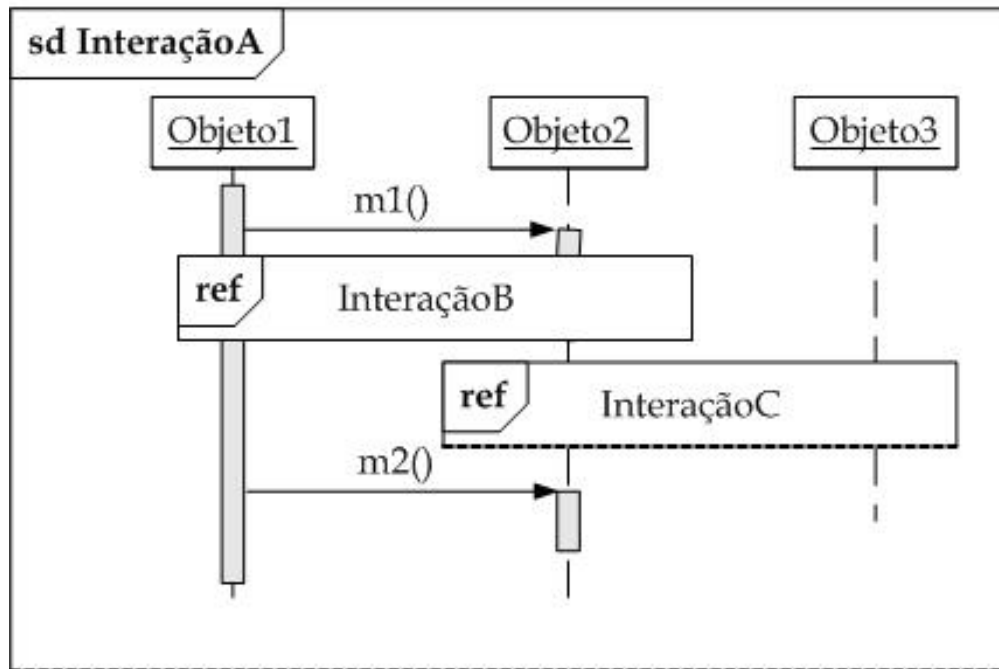
•25

*Dar um nome ao diagrama que aparece dentro do quadro*

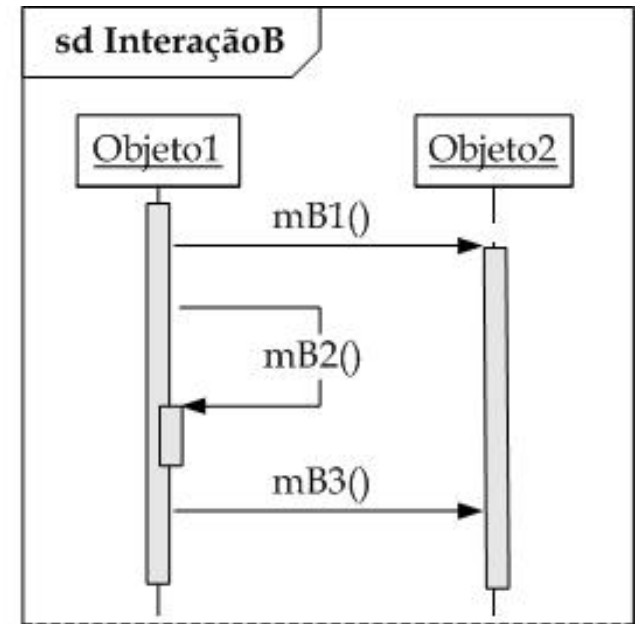


# DIAGRAMAS REFERENCIADOS

*Fazer referência a um diagrama definido separadamente.*

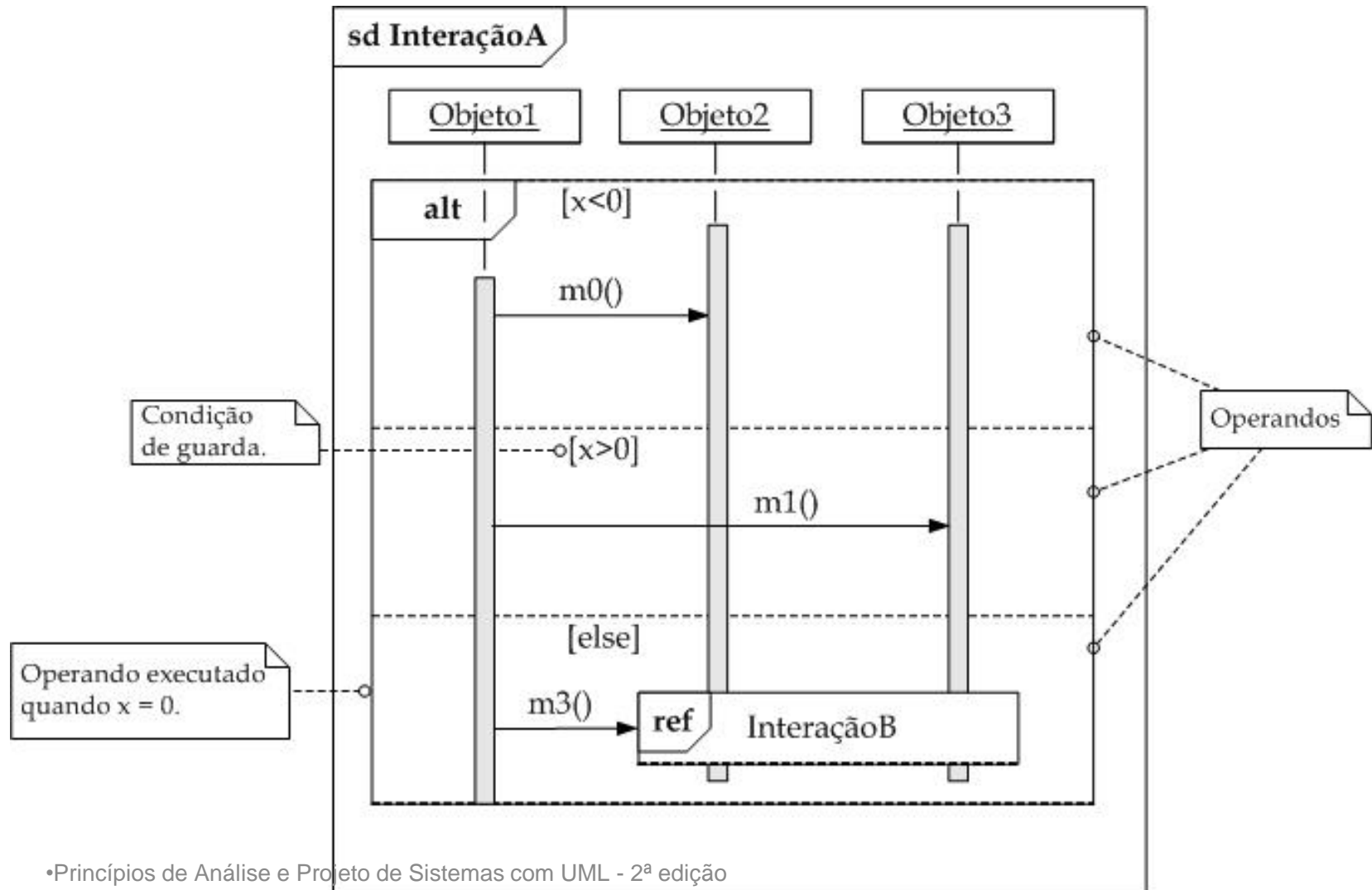


InteraçãoB e InteraçãoC são nomes de diagramas que apresentam mensagens trocadas entre os objetos Objeto1 e Objeto2. Note que os quadros correspondentes são rotulados com "ref" e posicionados sobre as linhas de vida dos objetos.

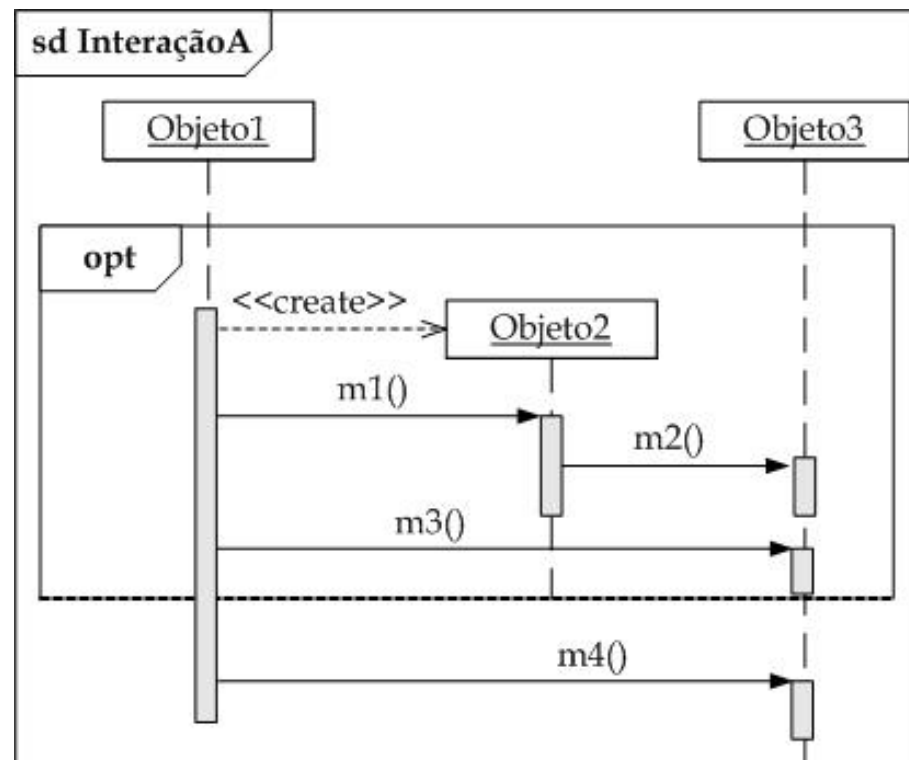


# FLUXO DE CONTROLE: ALTERNATIVAS

•27



# FLUXO DE CONTROLE: OPÇÃO



# FLUXO DE CONTROLE: ITERAÇÕES

