Projeto detalhado
Diagramas de interação

Bruna Diirr

brunadiirr@ic.uff.br

Divisão da fase de Projeto

Projeto geral ou preliminar

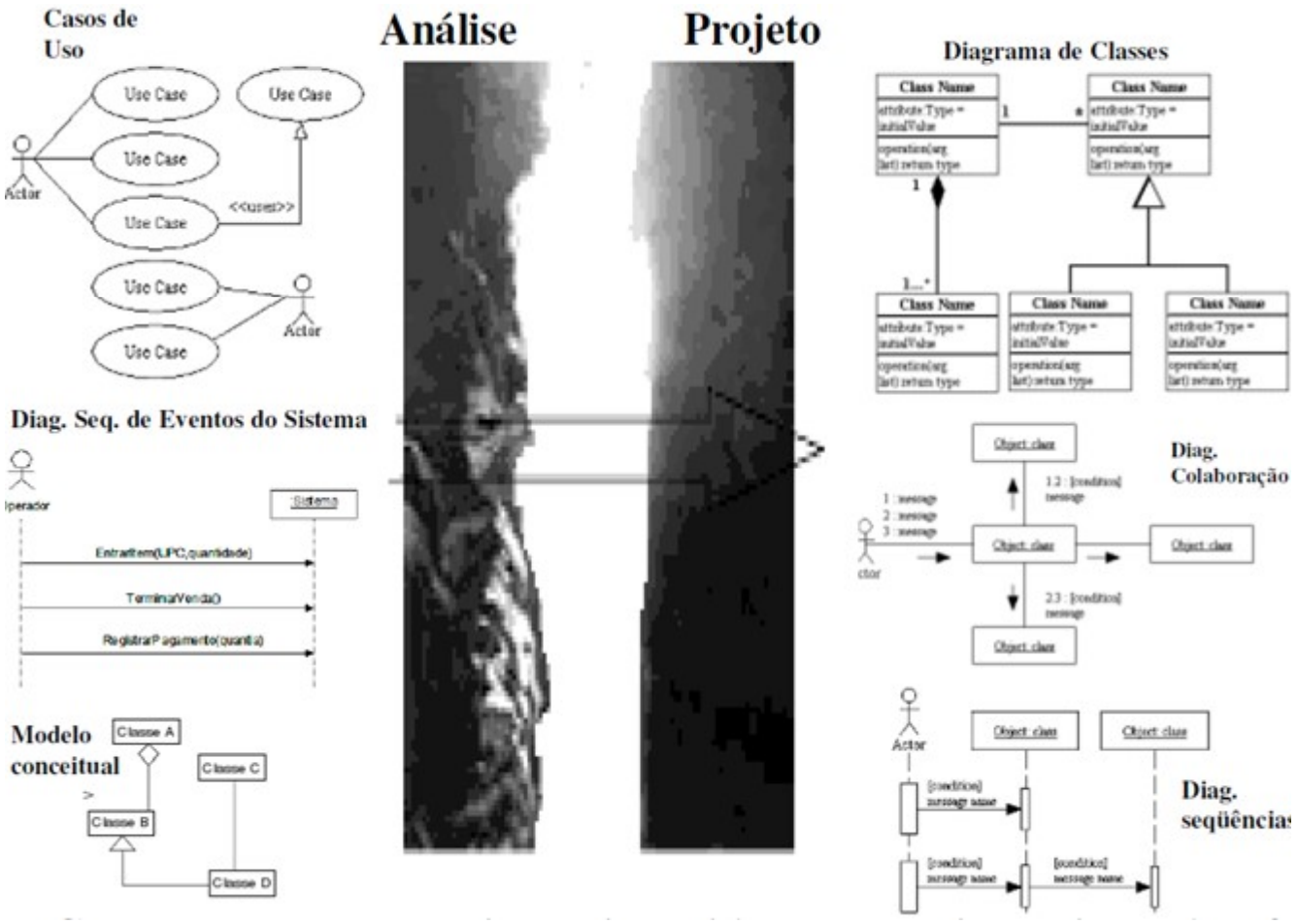
Tradução da especificação do sistema em termos da arquitetura a partir do conhecimento adquirido com requisitos (funcionais e não funcionais

Projeto detalhado

Refinamento progressivo e adição de detalhes à arquitetura visando a codificação → *Modelos de projeto*

Modelos de análise → Modelos de projeto

Capturam resultados do processo de investigação do domínio do problema



Capturam resultados do processo de investigação do domínio da solução

Modelos de análise → Modelos de projeto

Artefatos da fase de análise servem como insumo para desenvolvimento de uma solução lógica

> *"Identificados os requisitos e o modelo conceitual, acrescente os métodos às classes de software e defina as mensagens/troca de mensagens para atender aos requisitos"*

O que isso quer dizer? Com que artefatos vamos trabalhar?

- Diagramas de classes e Diagramas de interação!
- Nossa estratégia
  - Primeiro: Diagramas de interação (Sequência e Colaboração)
  - Em seguida: Diagramas de classes (de projeto) → Detalhes na próxima aula!

Diagramas de interação

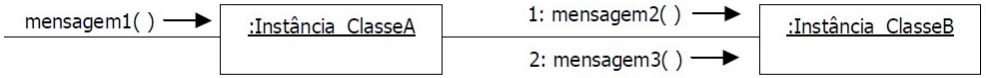
Ilustram como objetos interagem através de mensagens para cumprir suas tarefas (aspecto dinâmico do sistema)

Permite validar classes, responsabilidades e colaboradores identificados anteriormente, além de refinar o diagrama de classes

A partir deles raciocinamos sobre detalhes concretos de quais mensagens enviar, para quem e em que ordem

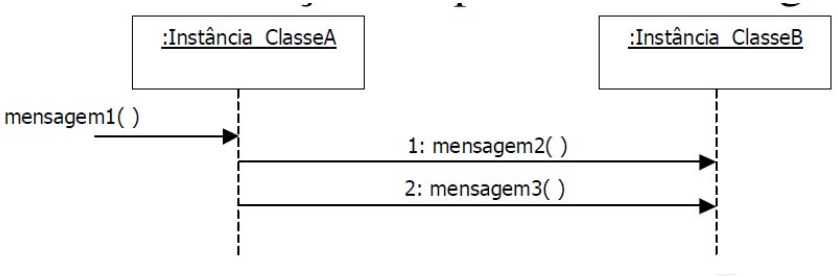
Diagramas de colaboração

Ênfase na organização estrutural dos objetos que enviam e recebem mensagens



Diagramas de sequência

Ênfase na ordenação temporal das mensagens



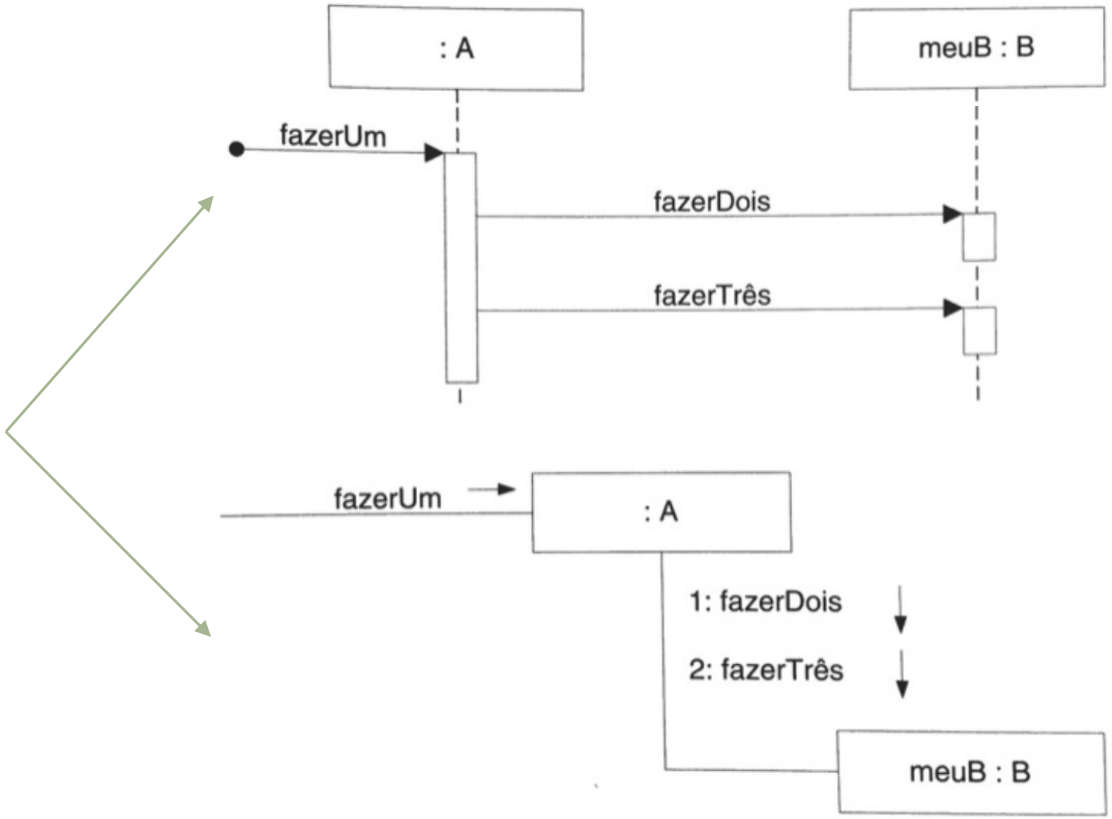
Diagramas de interação

Diagramas são equivalentes

Diagrama de sequência ←→ Diagrama de comunicação

```
public class A
{
private B meuB = new B();

public void fazerUm()
{
    meuB.fazerDois();
    meuB.fazerTres();
}
//...
}
```



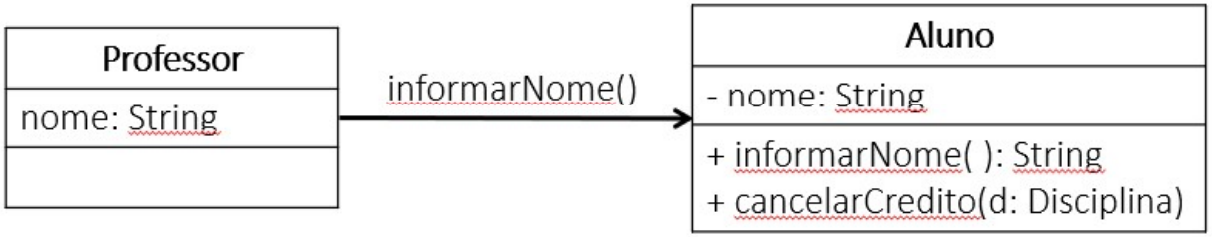
Diagramas de interação

Tipo	Pontos fortes	Pontos fracos
Sequência	- Mostra com clareza a sequência ou ordem temporal das mensagens - Amplo conjunto de opções detalhadas	Deve ser estendido para a direita quando são acrescidos novos modelos, o que consome espaço
Comunicação	Economia de espaço (flexibilidade de adicionar novos objetos em duas dimensões)	- É mais difícil de ver a sequência das mensagens - Menos opções de notação

Mensagens

Interação entre objetos ocorre através de mensagens

Solicitação de execução de uma operação em outro objeto



Sempre do objeto remetente para objeto receptor!

Sintaxe

retorno = mensagem(parametro: tipoDeParametro): tipoDeRetorno

Ex.: d = obterDescProduto(id: IdItem) : DescricaoDoProduto

Mensagens

Tipos

Simples 

Utilizada quando natureza da mensagem não é relevante

Síncrona 

Objeto remetente espera que objeto receptor processe a mensagem antes de recomeçar seu processamento

Assíncrona 

Objeto remetente não espera resposta da mensagem para prosseguir seu processamento

Retorno 

Indica o término de uma operação. Não é uma mensagem, mas o retorno de uma mensagem. Desbloqueia uma mensagem síncrona

Diagrama de sequência

- Notação básica
  - Caixas de linha de vida
  - Foco de controle e barras de especificação de execução
  - Ilustração de respostas/retornos
  - Mensagem para “self” ou “this”
  - Criação de instância
  - Destruição de objeto
  - Molduras de diagramas
    - *Mensagem condicional*
    - *Iteração*
    - *Aninhamento*
  - Organização de diagramas complexos

Diagrama de sequência

Caixas de linha de vida

Mostram participantes em interações

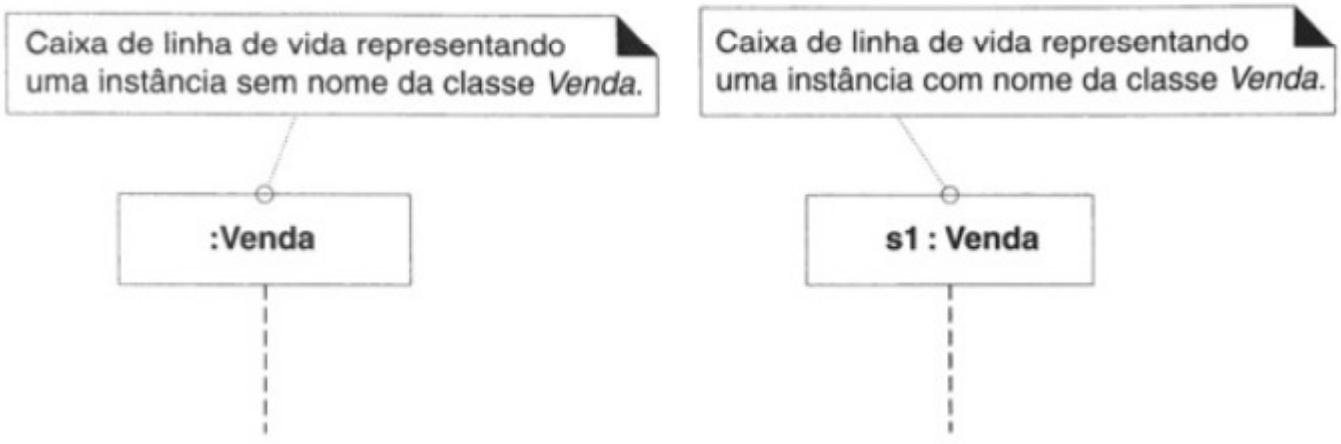


Diagrama de sequência

Mensagens e foco de controle com barras de especificação de execução

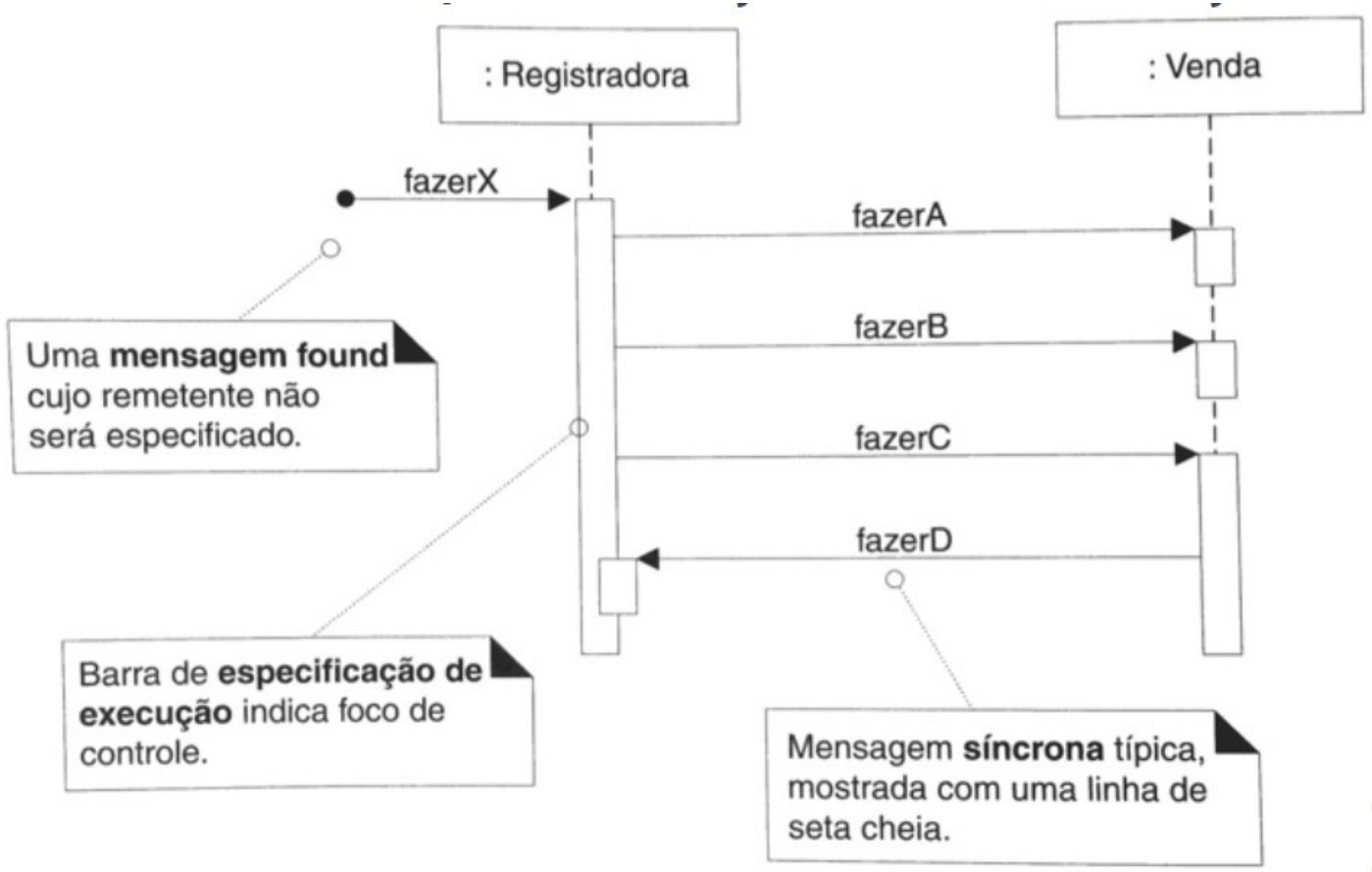
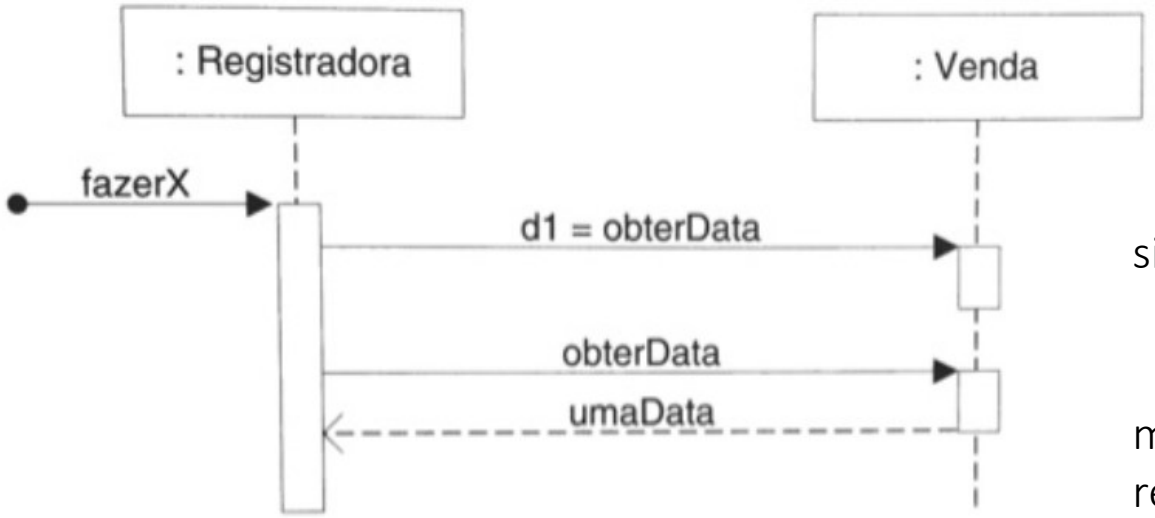


Diagrama de sequência

Ilustração de resposta/retornos (dois modos)



Modo 1
sintaxe de mensagem

Modo 2
mensagem de retorno no final da barra de ativação

Diagrama de sequência

Mensagem para “this”

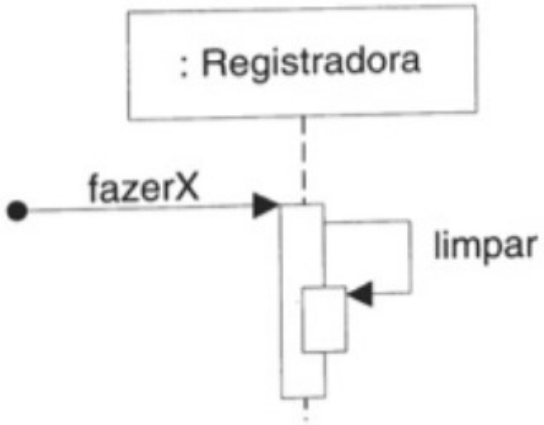


Diagrama de sequência

Criação de instância

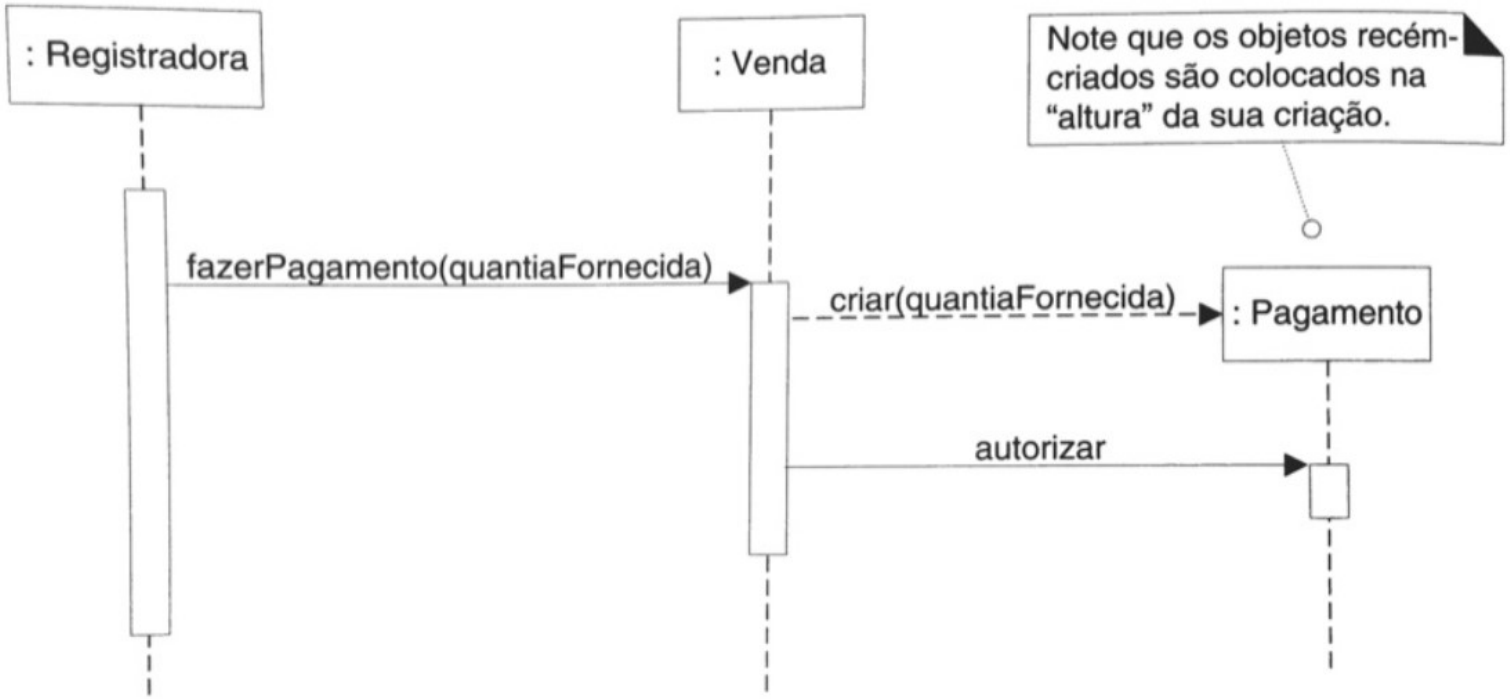


Diagrama de sequência

Destruição de objeto

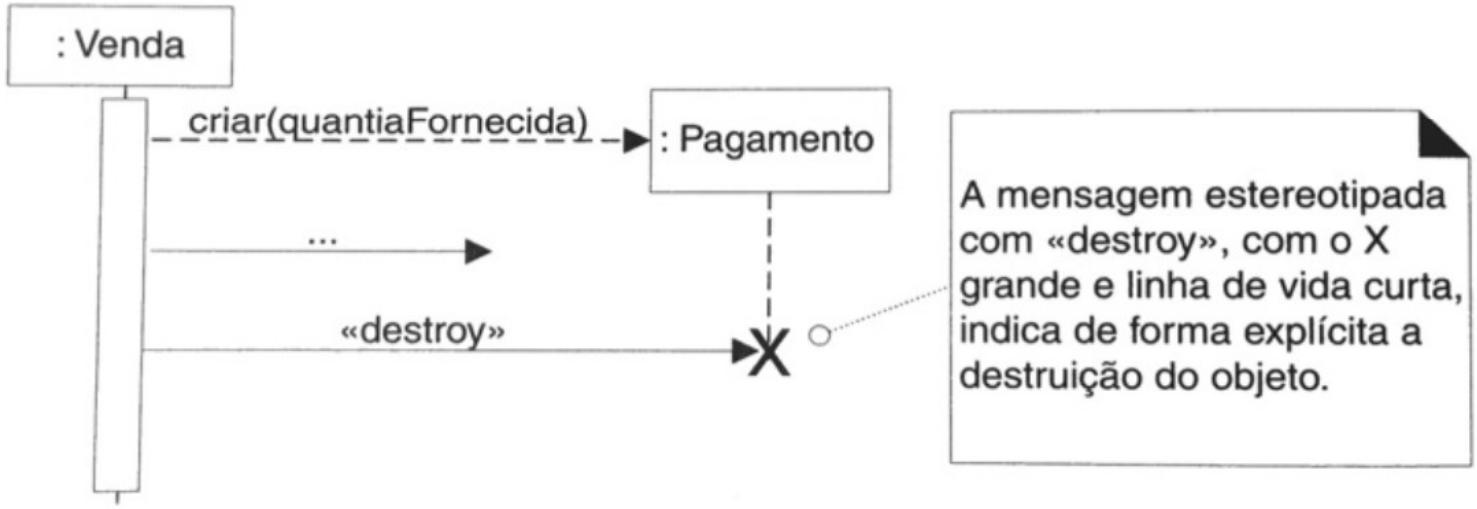


Diagrama de sequência

Molduras

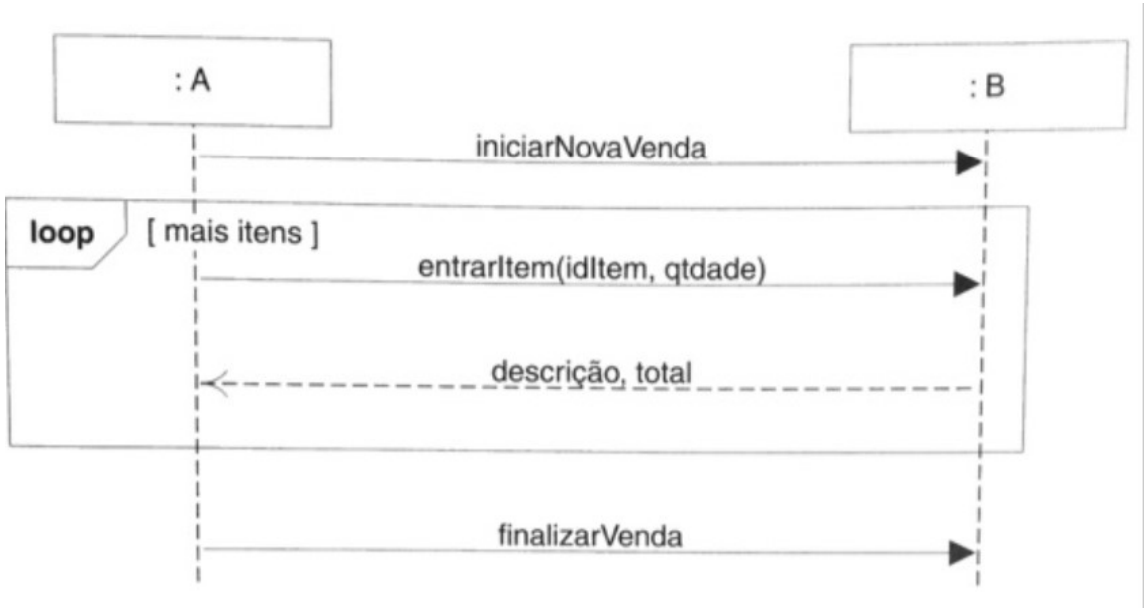


Diagrama de sequência

Molduras

Operador da moldura	Significado
opt	Fragmento opcional executado se a guarda é verdadeira.
alt	Fragmento alternativo para lógica condicional de exclusão mútua, expresso em guardas.
loop	Fragmento de loop, enquanto a guarda é verdadeira.
par	Fragmentos executados em paralelo

Diagrama de sequência

Molduras

Mensagem condicional

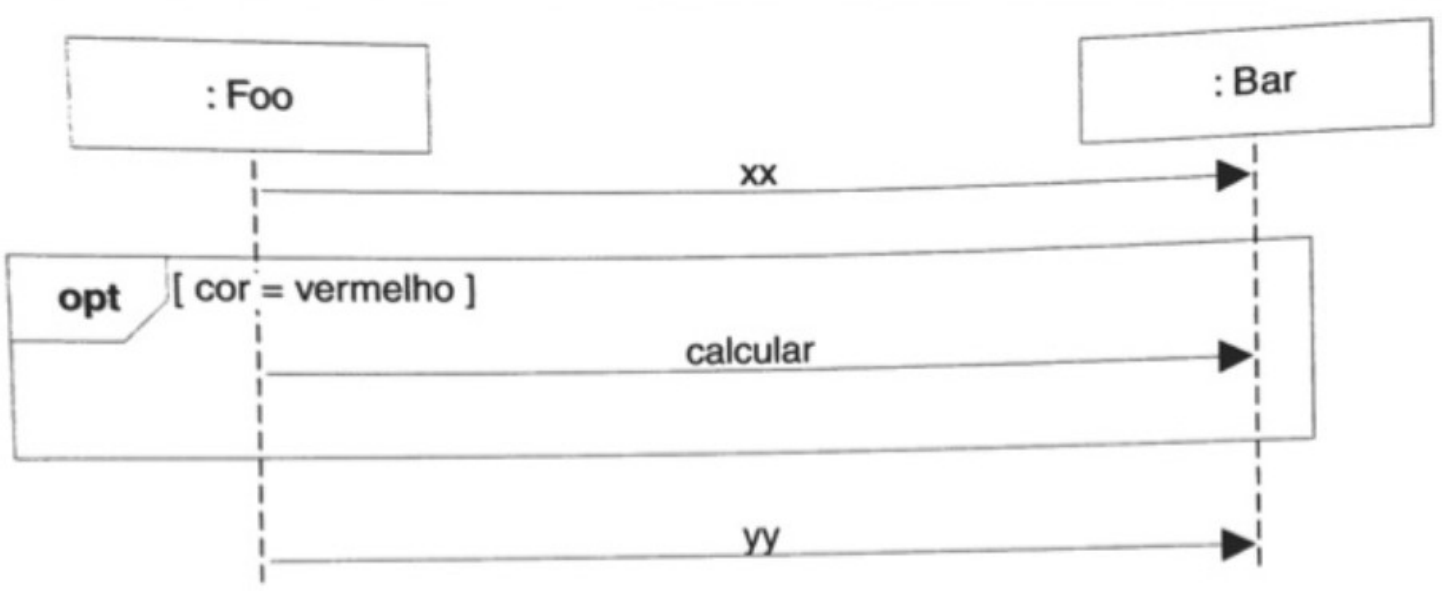


Diagrama de sequência

Molduras

Mensagem condicional mutuamente excludente

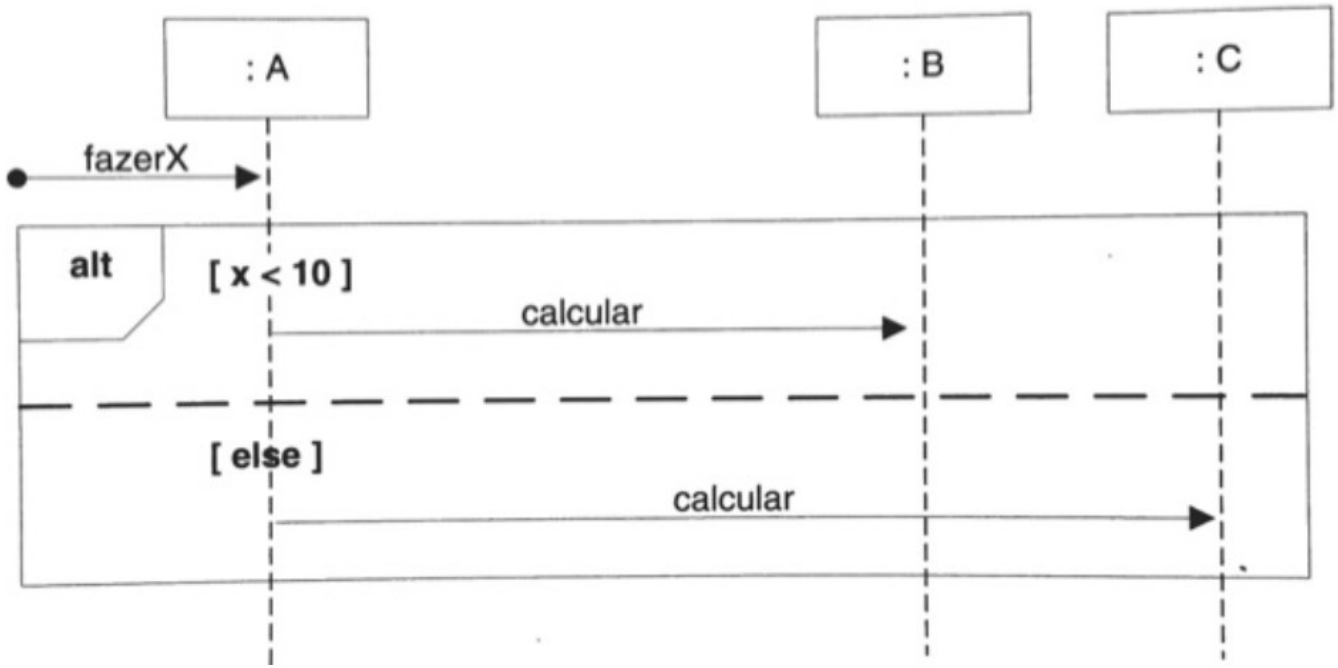


Diagrama de sequência

Molduras

Iteração (notação explícita)

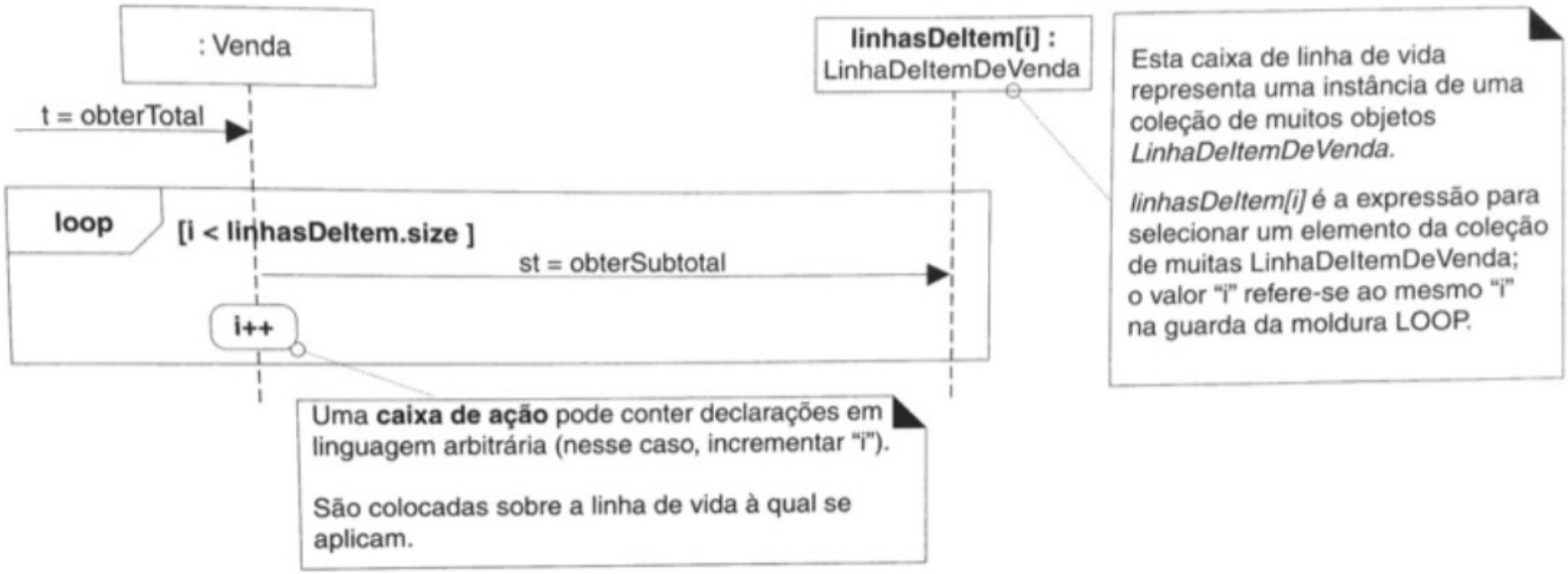


Diagrama de sequência

Molduras

Iteração (notação implícita)

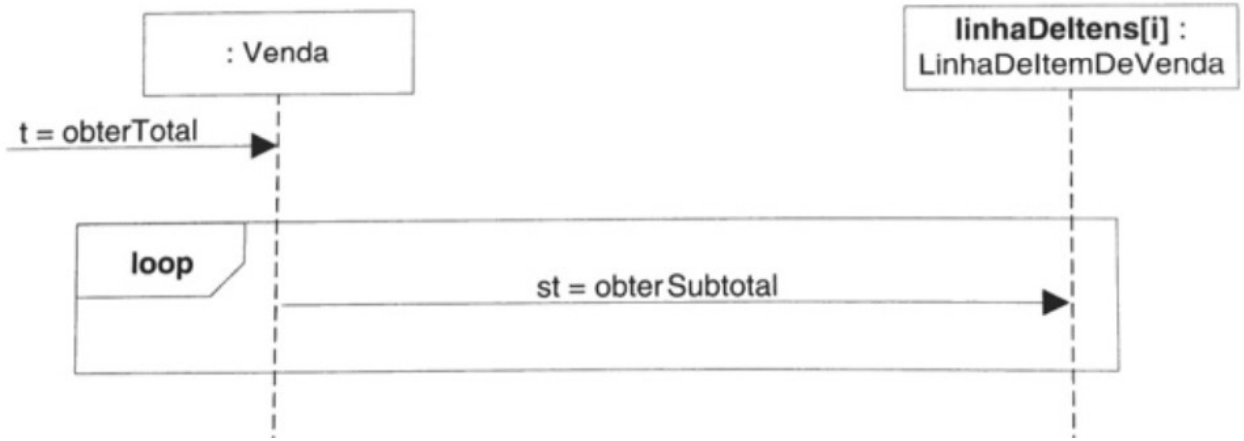


Diagrama de sequência

Molduras

Aninhamento

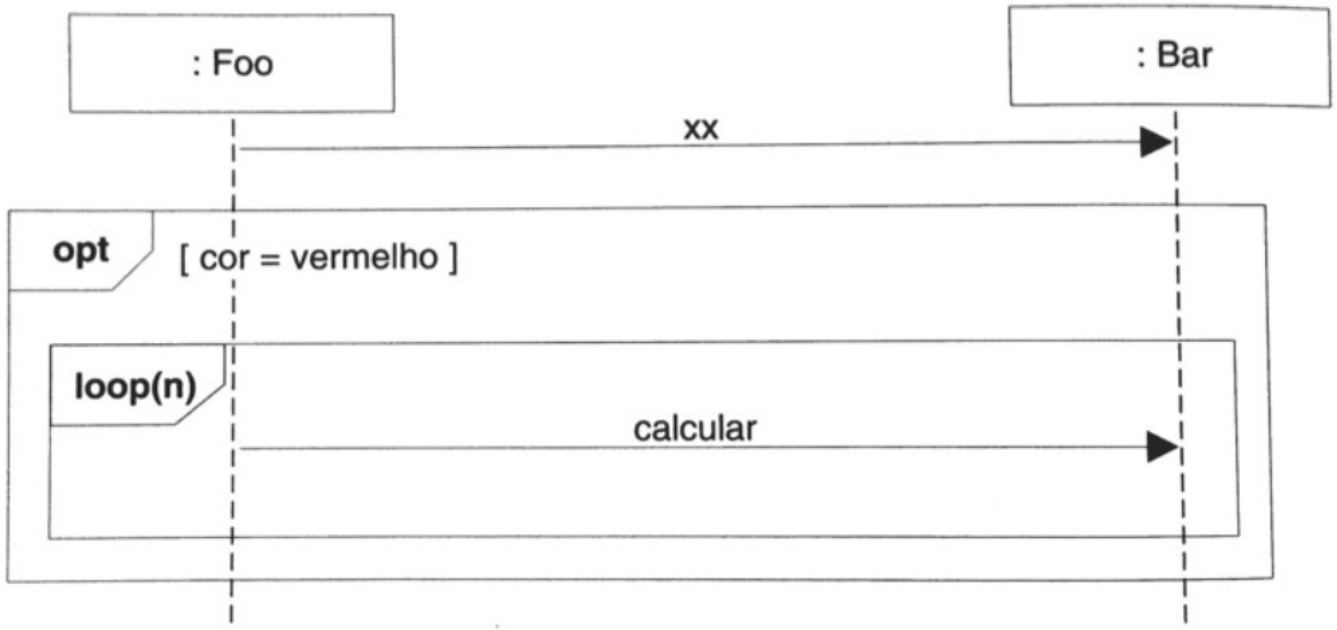


Diagrama de sequência

Organizando diagramas complexos

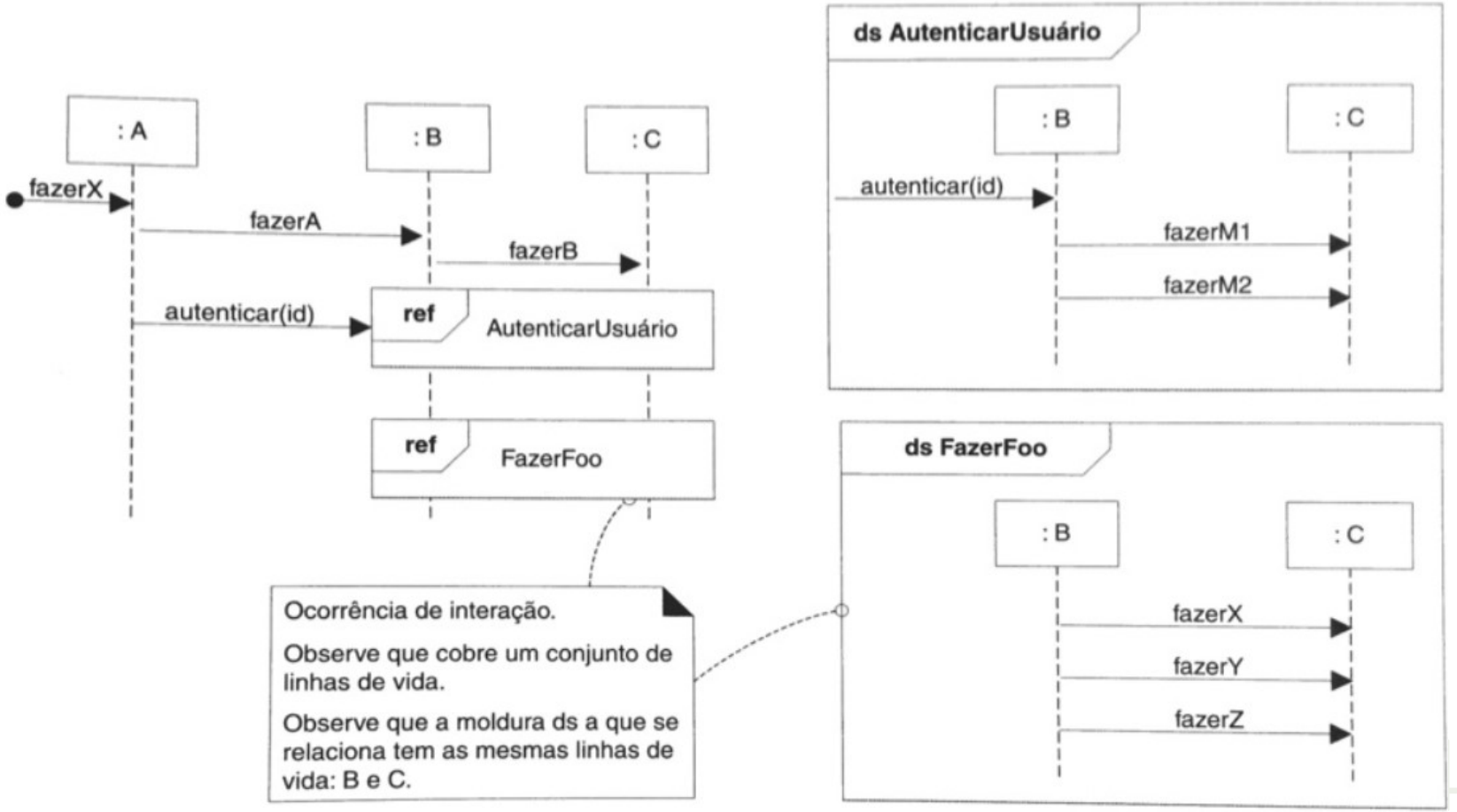


Diagrama de comunicação

Notação básica
- Linhas de ligação
- Mensagens
- Mensagem para “self” ou “this”
- Criação de instância
- Ordem de mensagens
- Mensagem condicional
- Iteração

Diagrama de comunicação

Linhas de ligação

Nota: Mensagens múltiplas e mensagens nos dois sentidos fluem ao longo da mesma ligação!

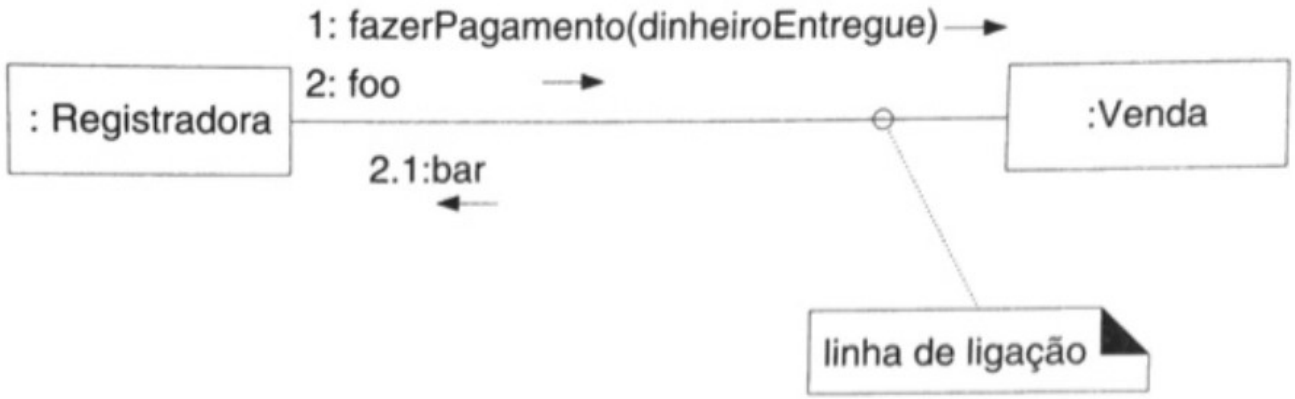


Diagrama de comunicação

Mensagens

Dica: Não numere a mensagem inicial. Simplifica a numeração geral.

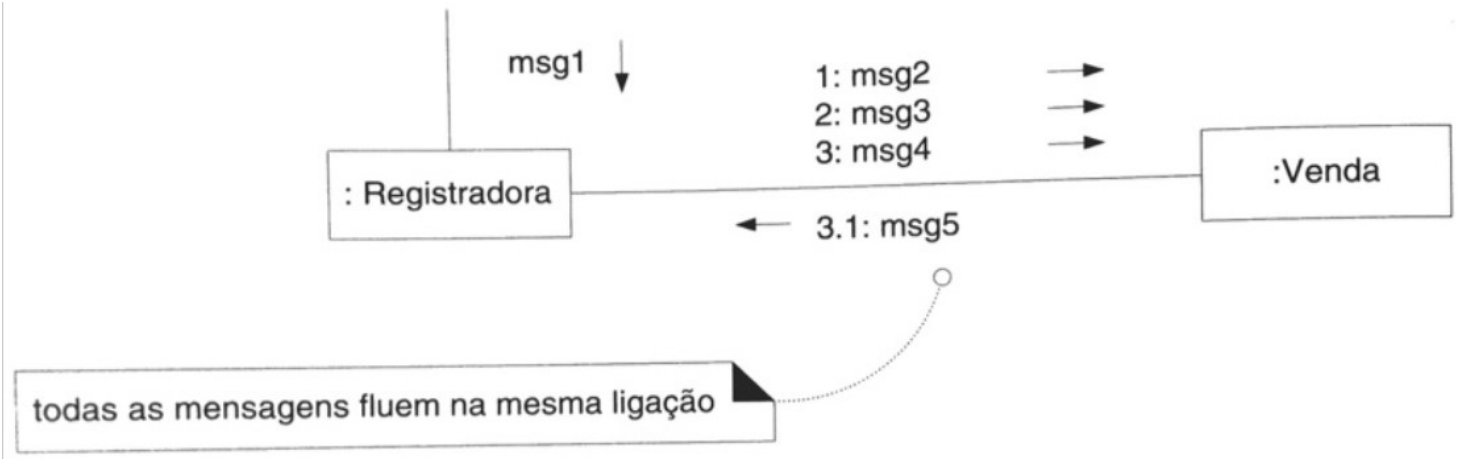


Diagrama de comunicação

Mensagens para “this”

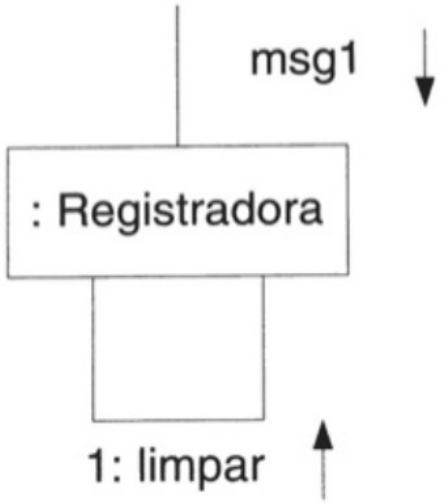


Diagrama de comunicação

Criação de instância

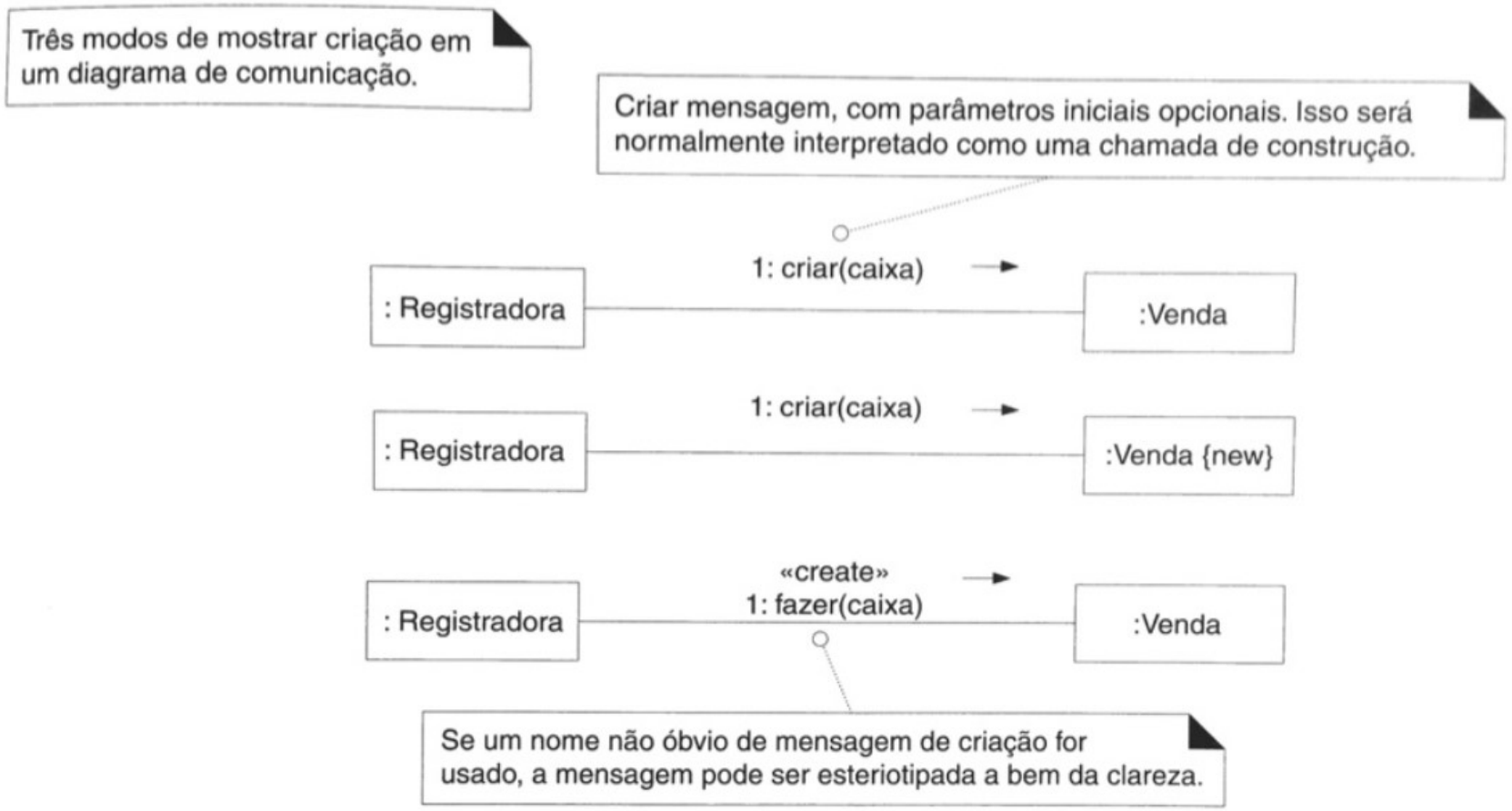


Diagrama de comunicação

Ordem de mensagens

Possível detalhar em vários “subpassos”

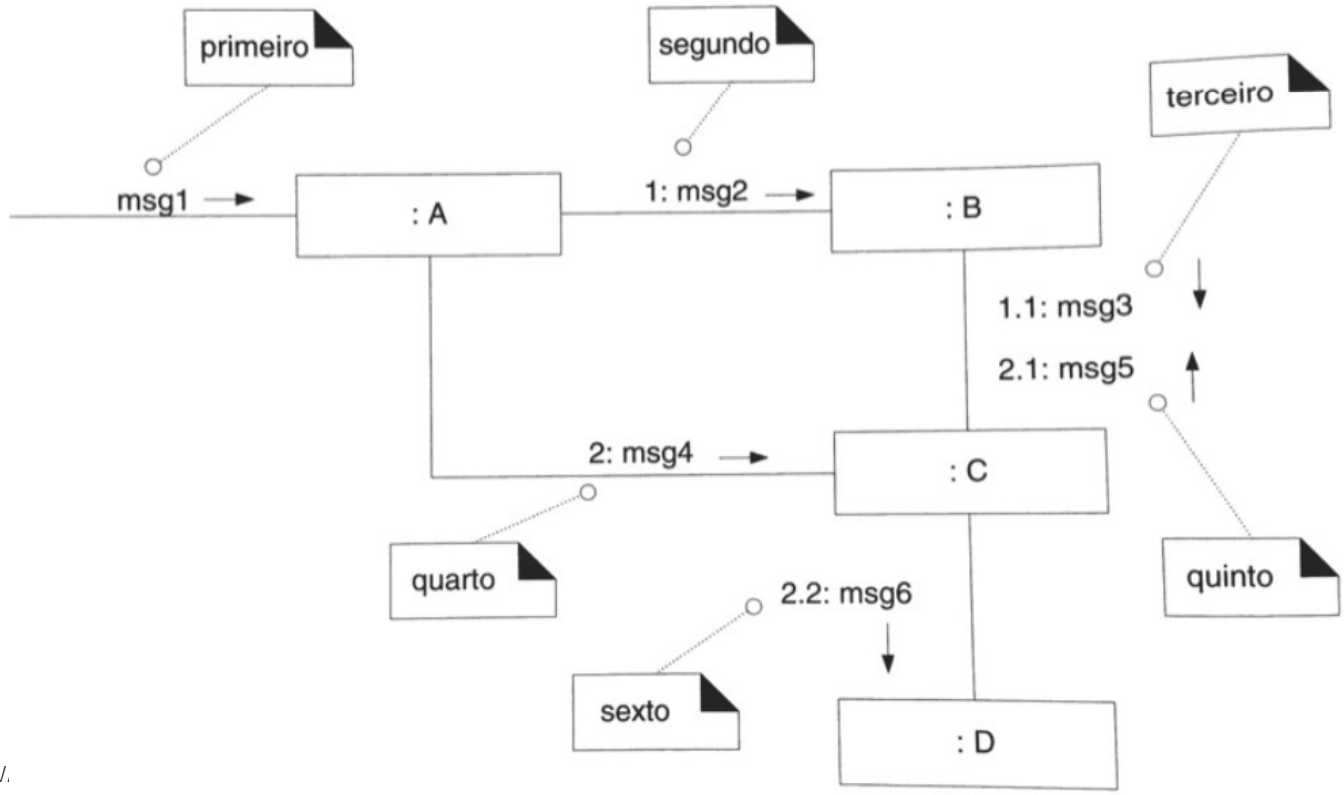


Diagrama de comunicação

Mensagem condicional

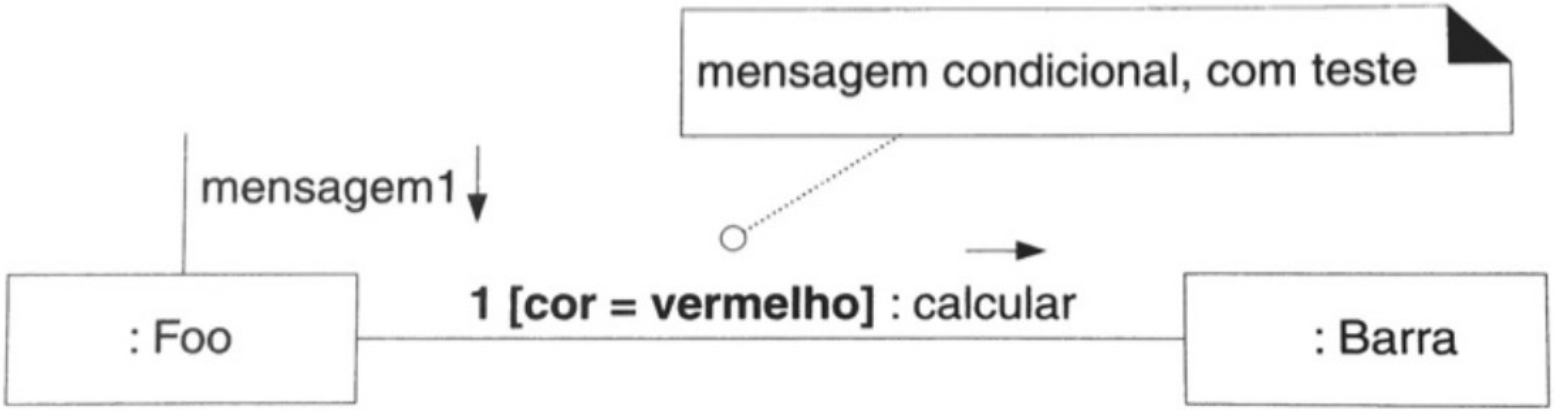


Diagrama de comunicação

Mensagem condicional mutuamente excludente

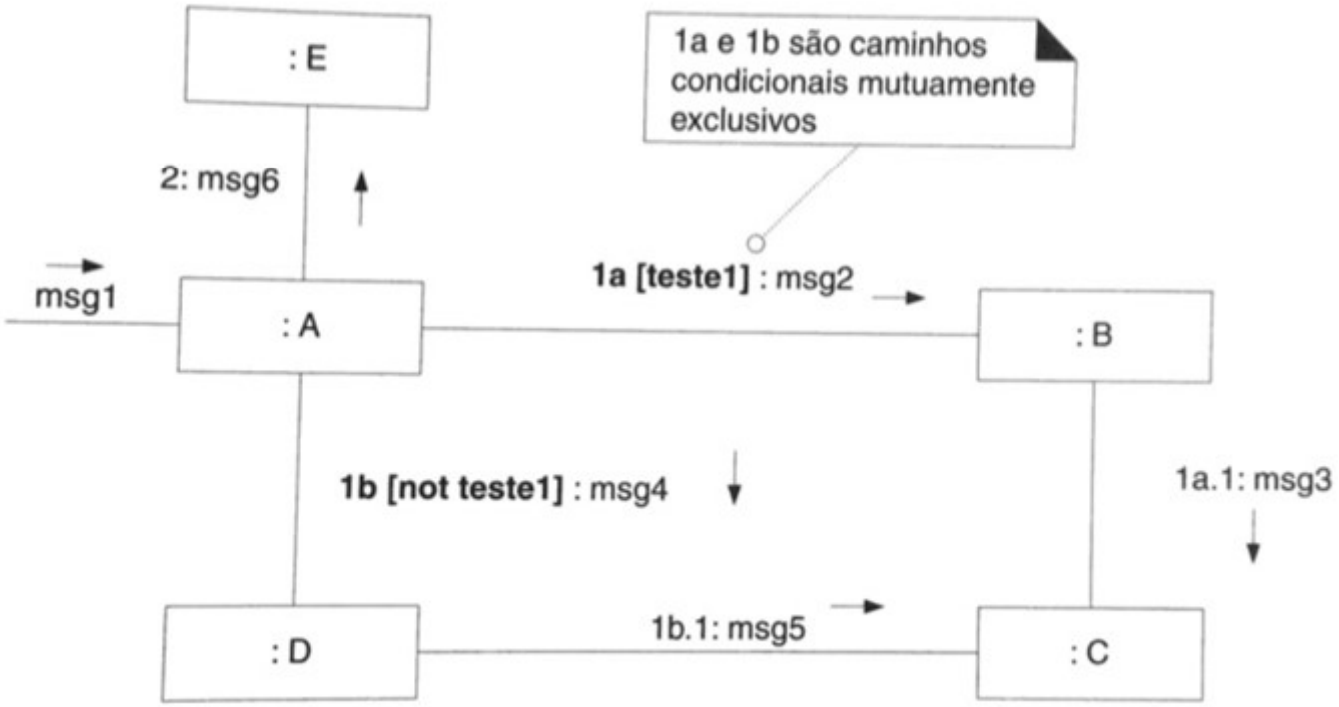


Diagrama de comunicação

Iteração (notação explícita)

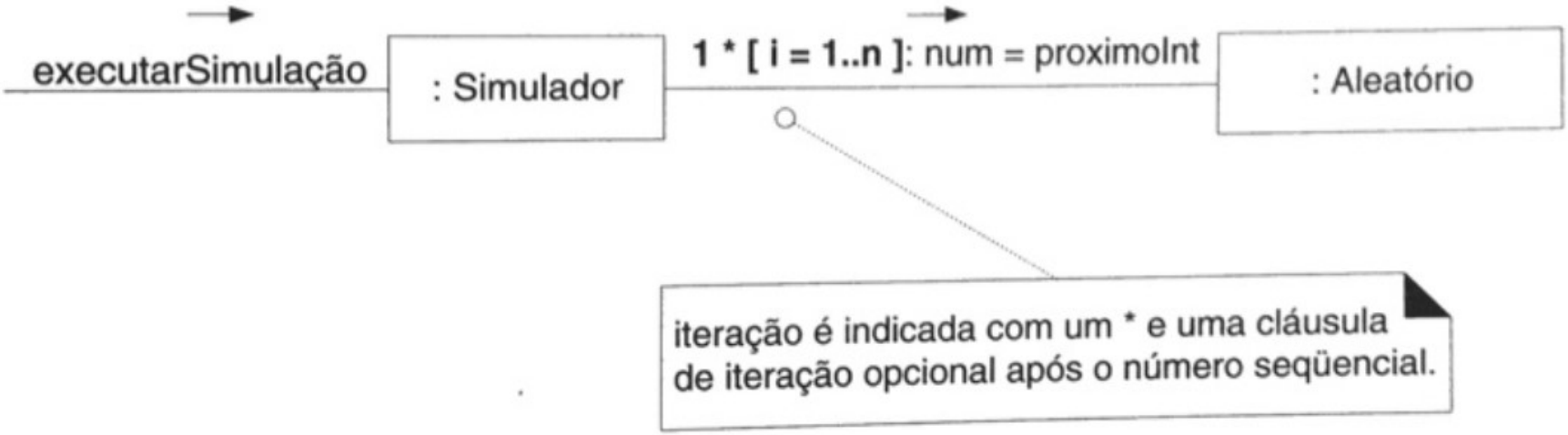
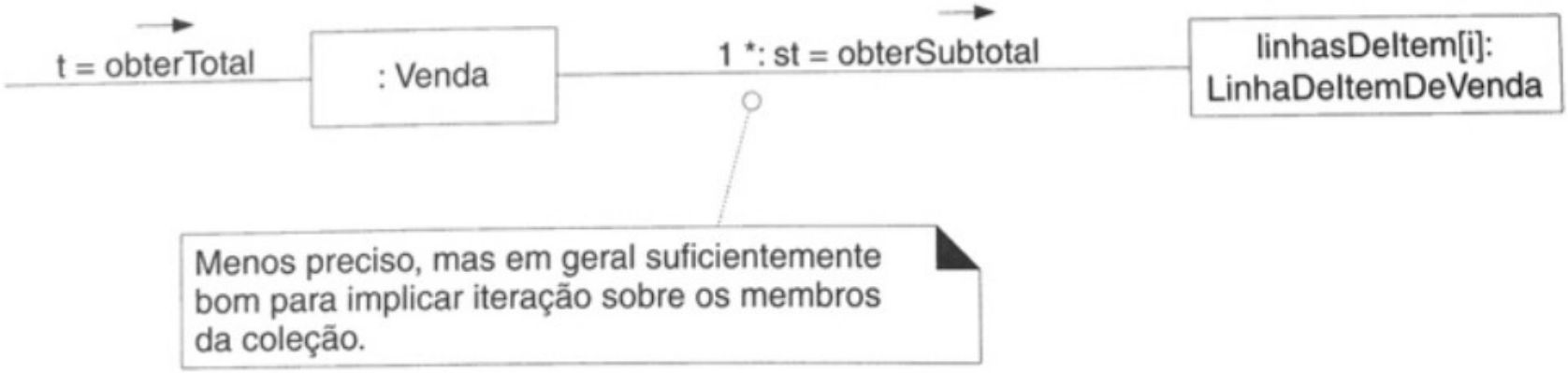


Diagrama de comunicação

Iteração (notação implícita)



Projeto detalhado
Diagramas de interação

Bruna Diirr

brunadiirr@ic.uff.br