

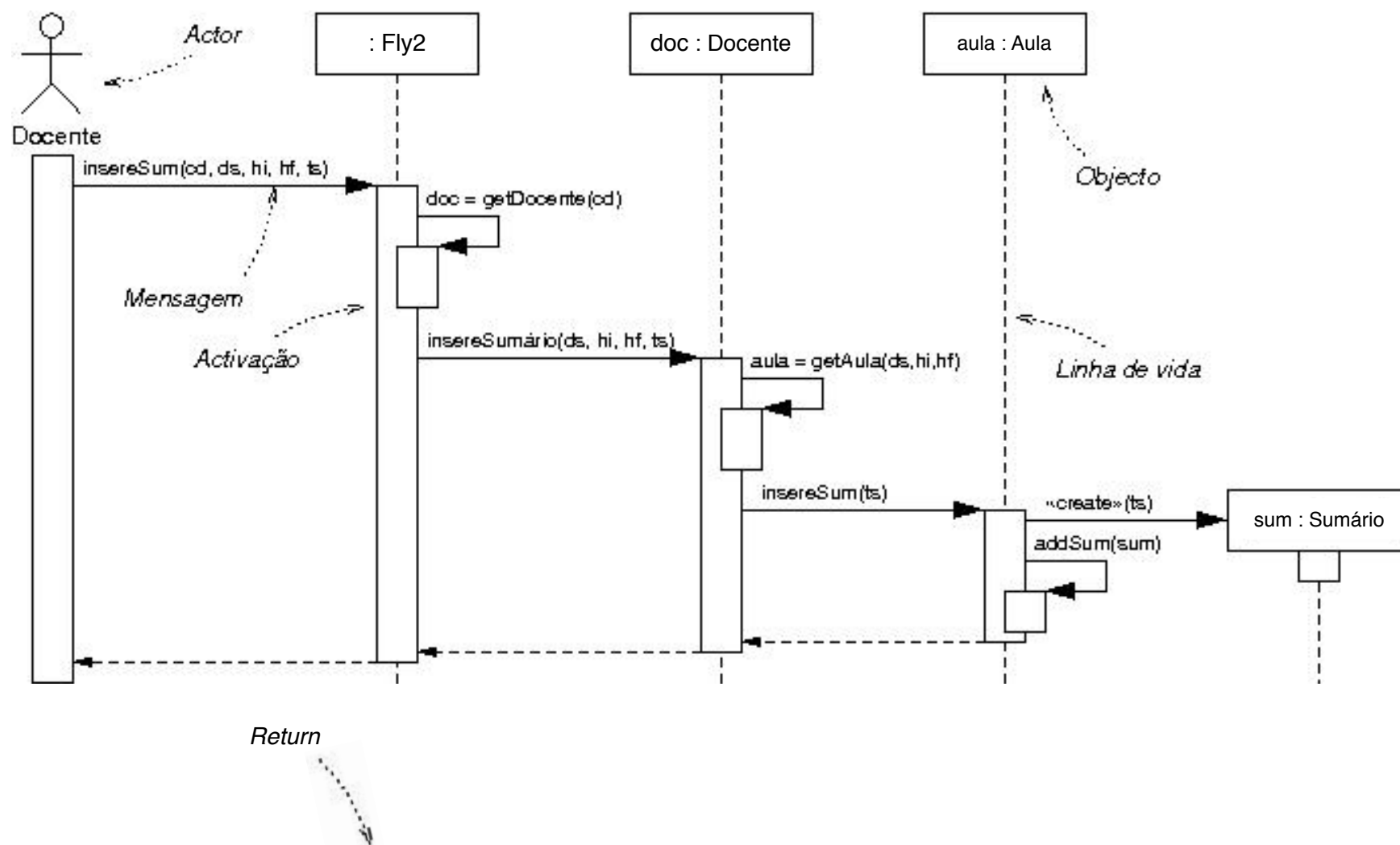


Desenvolvimento de Sistemas Software

Aula Teórica 12: Diagramas de Sequência II



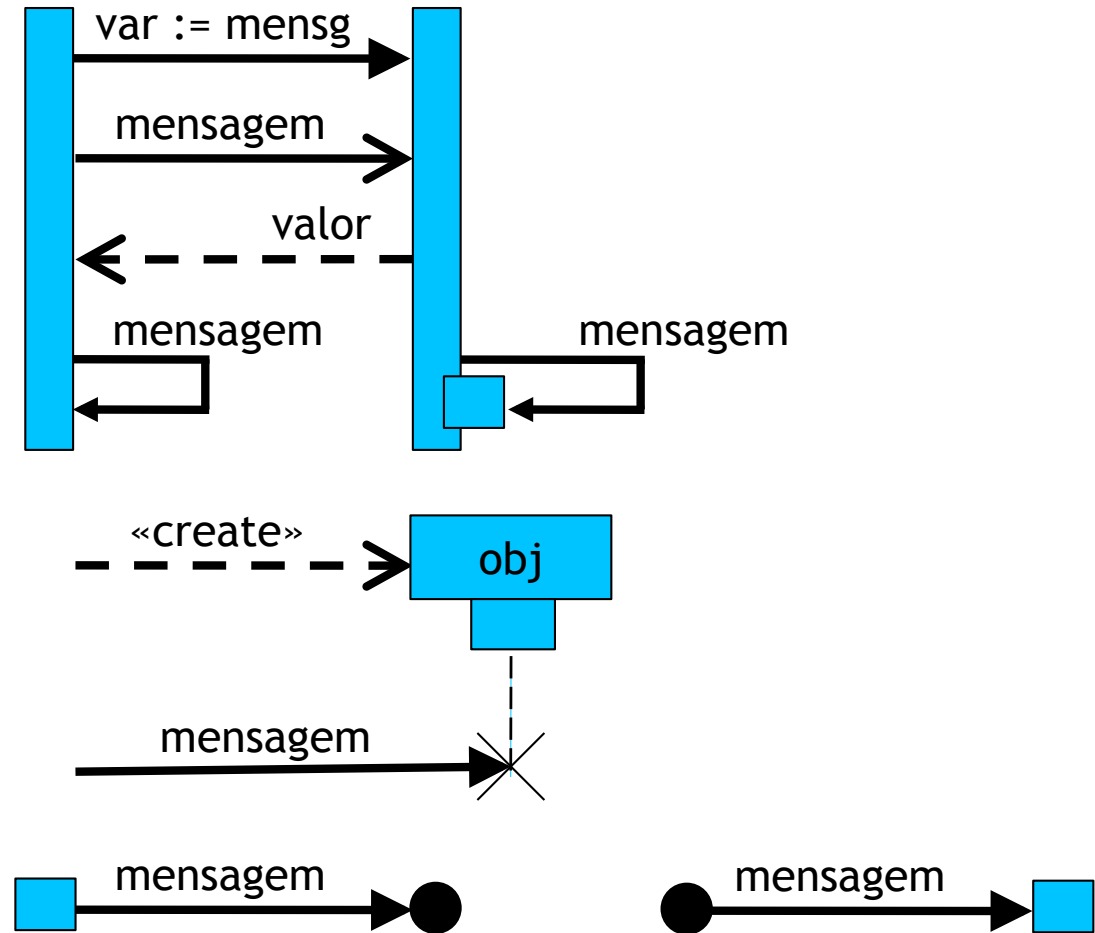
Resumo da aula anterior...





Mensagens

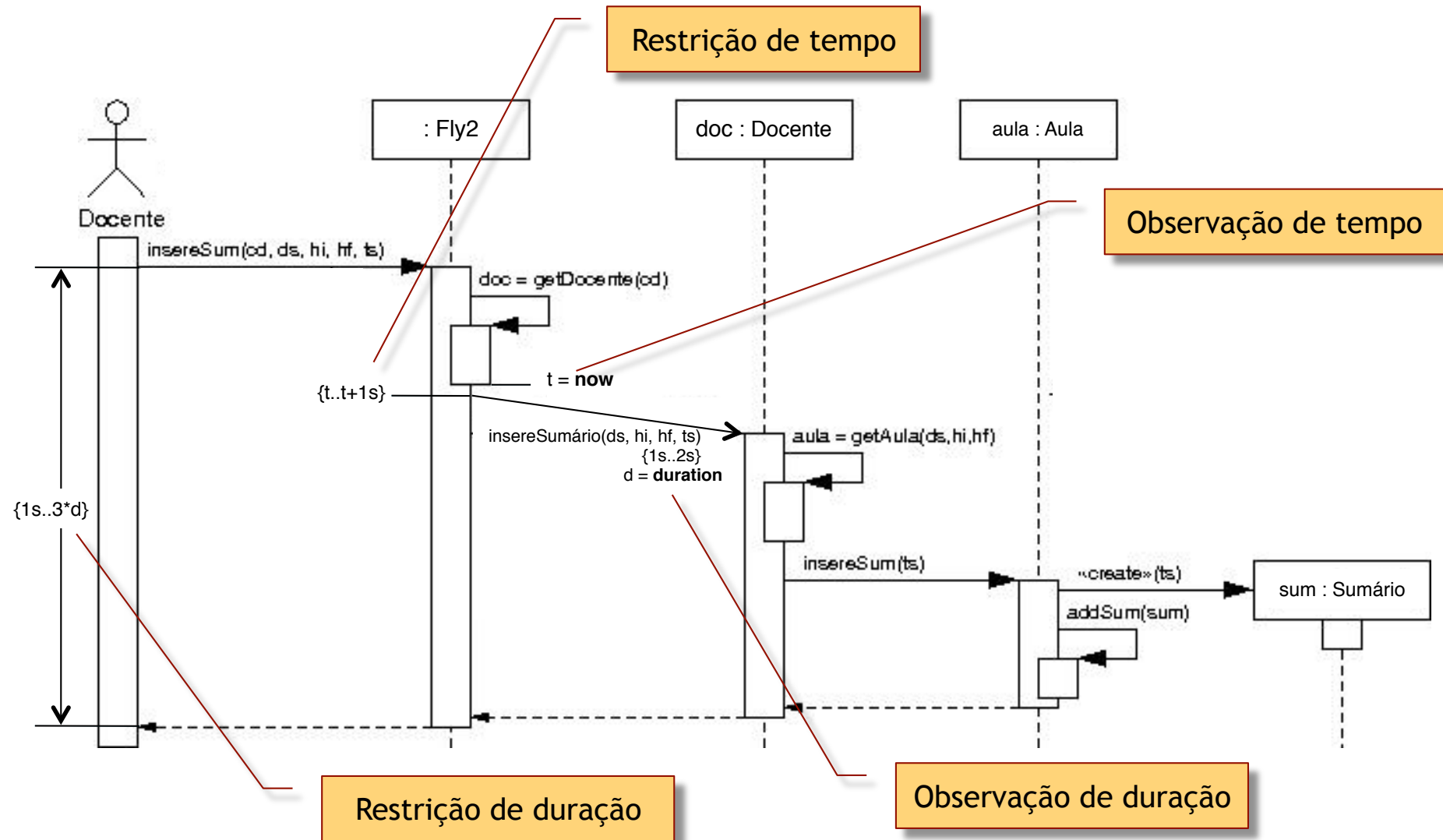
- invocação síncrona
- invocação assíncrona
- return/resultado
- self messages
- criar objectos
- destruir objectos
- lost/found messages



[attribute '='] signal-or-operation-name [arguments] [':' return-value]

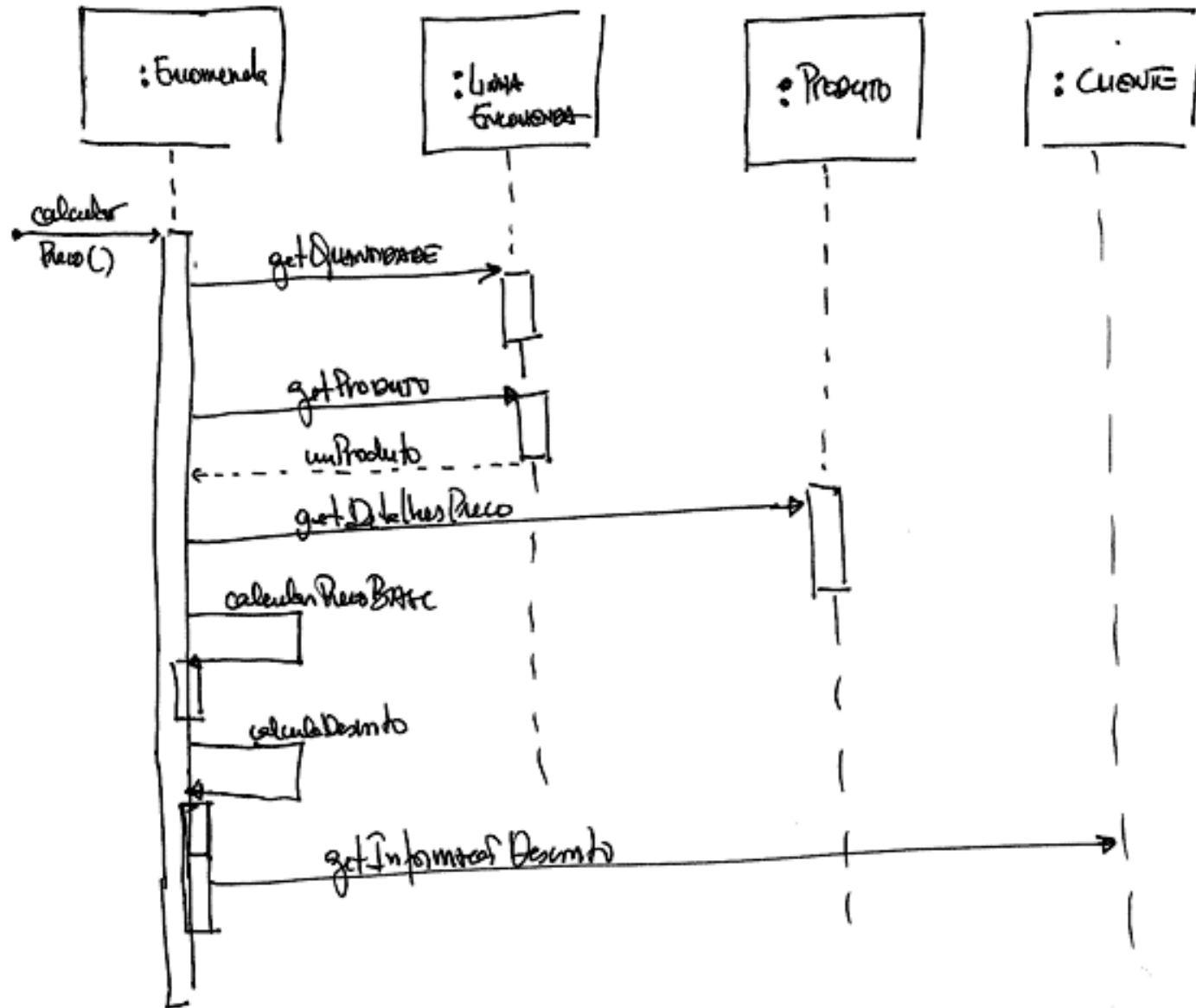


Restrições de tempo / duração



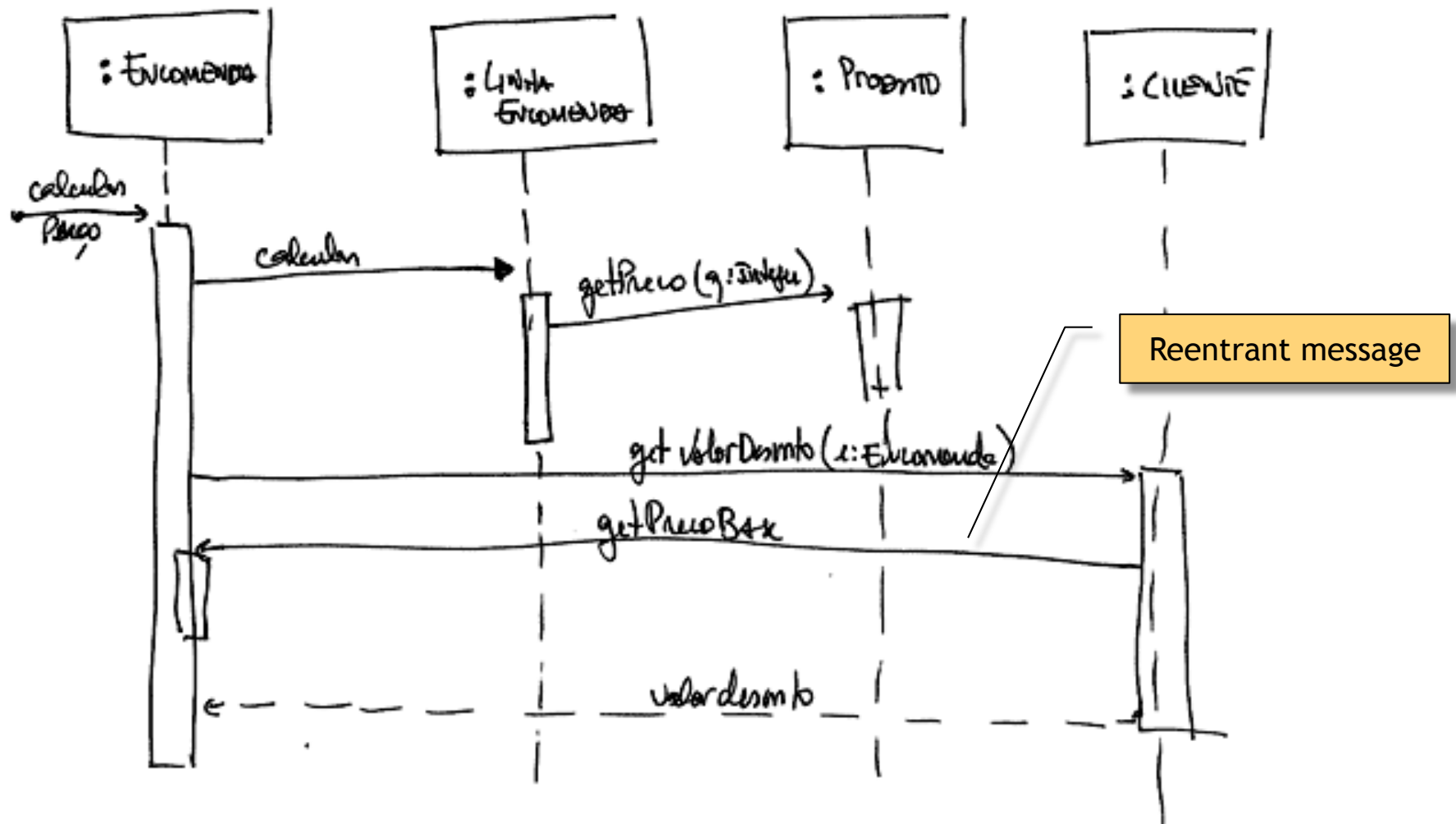
Distribuição de responsabilidades

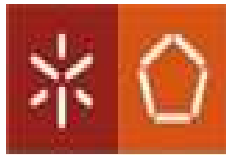
- Cálculo do preço de uma encomenda



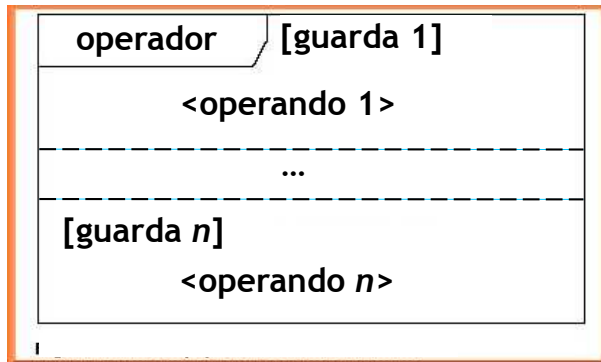
Distribuição de responsabilidades

- Cálculo do preço de uma encomenda
 - diagrama alternativo com controlo descentralizado





Fragmentos combinados



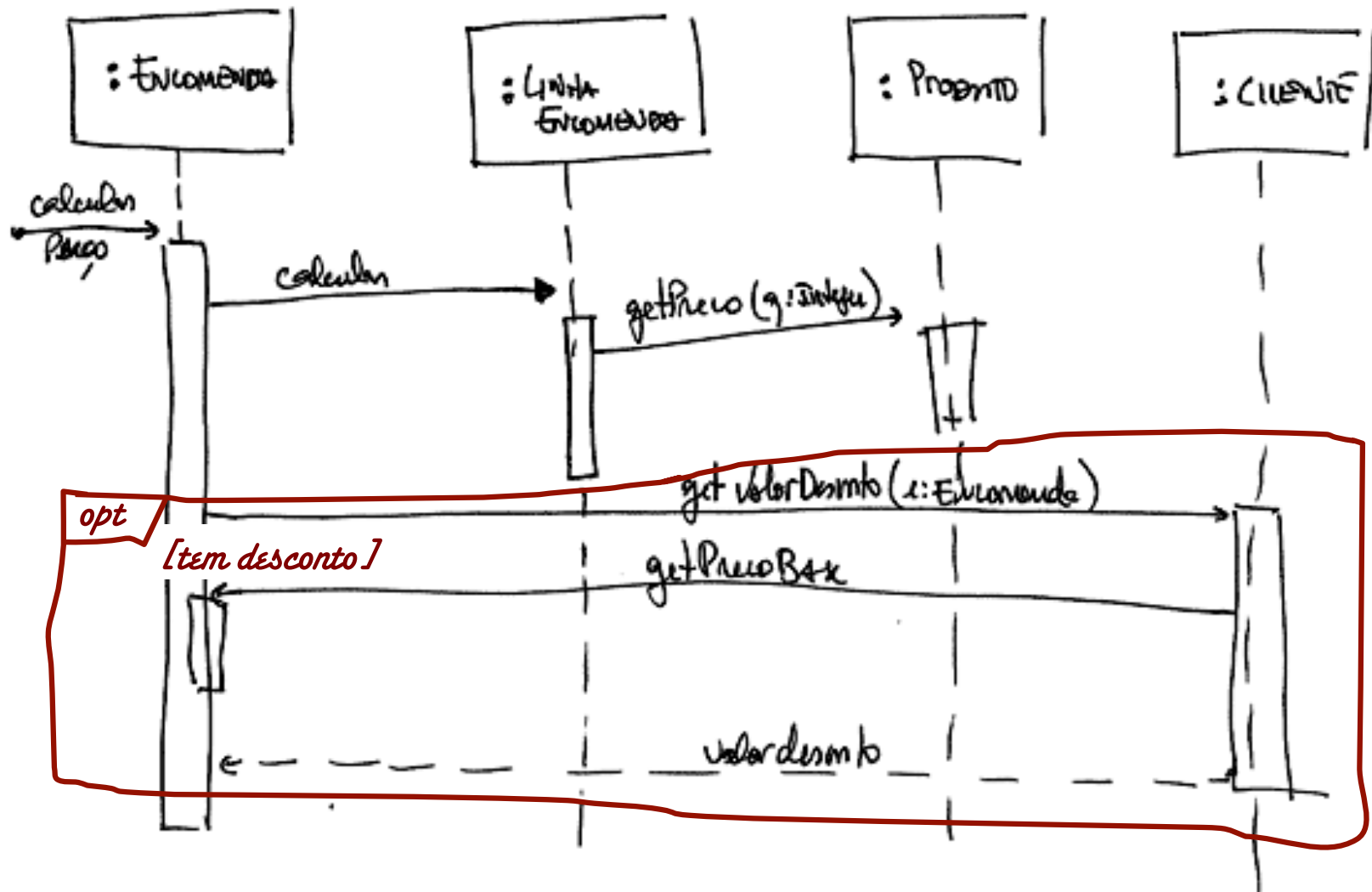
Fragmentos combinados são regiões ou fragmentos dos diagramas. Servem para assinalar fragmentos condicionais, loops, regiões críticas ou até outros diagramas.

As condições chamam-se **guardas**.

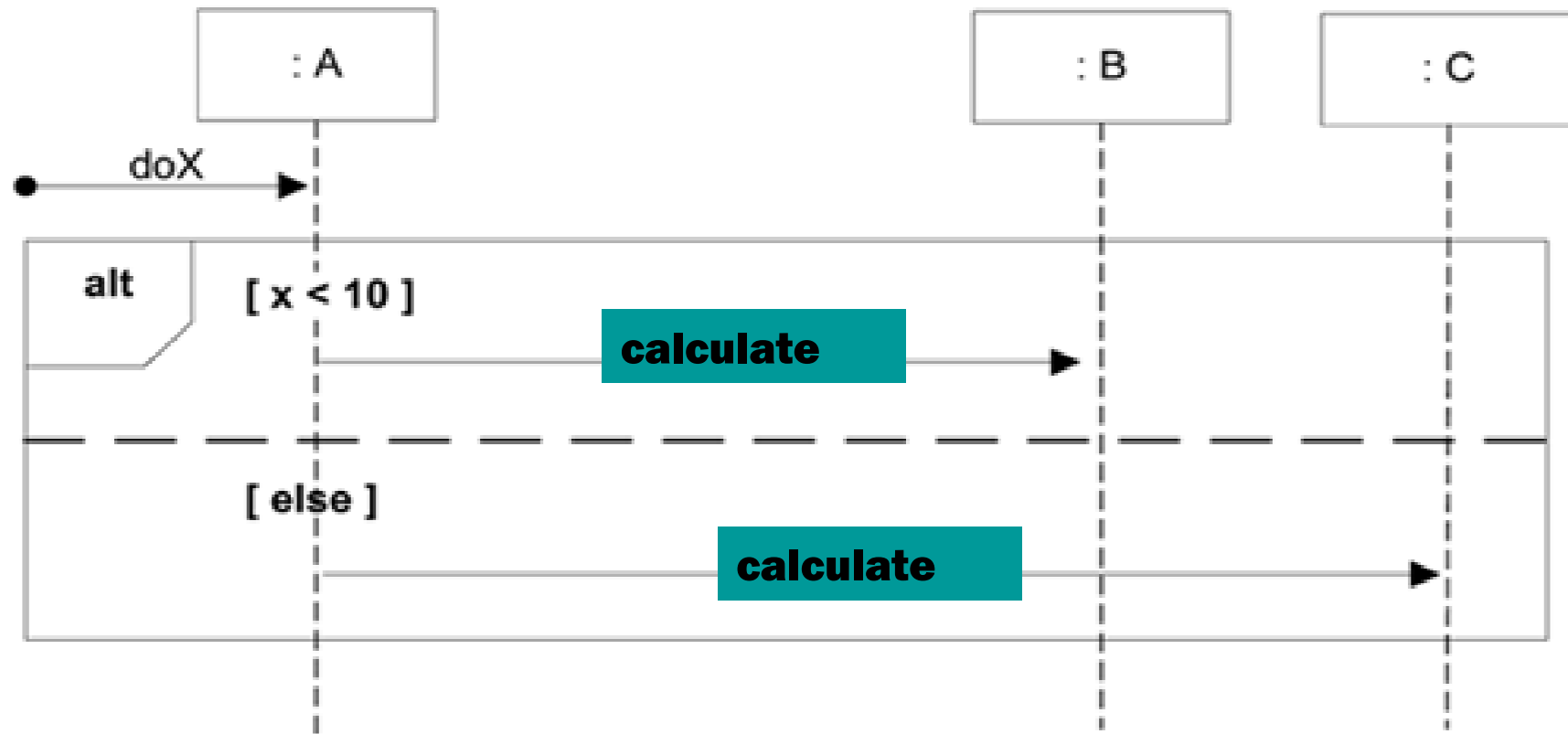
**Operadores
usados com
mais
frequência em
DS, como
instrumentos de
modularidade e
estruturação**

Operador	Semântica
alt	Define dois fragmentos mutuamente exclusivos cuja condição lógica de exclusão (ou não) é expressa numa guarda escrita entre [...]
loop loop(n)	Fragmento cíclico que vai ser repetido enquanto a guarda for verdadeira.
opt	Fragmento opcional que é executado apenas se a guarda for verdadeira
par	Fragmentos que são executados em paralelo
break	A sua execução termina sem retorno ao diagrama exterior
ref	Referencia a um outro diagrama (ou até método)

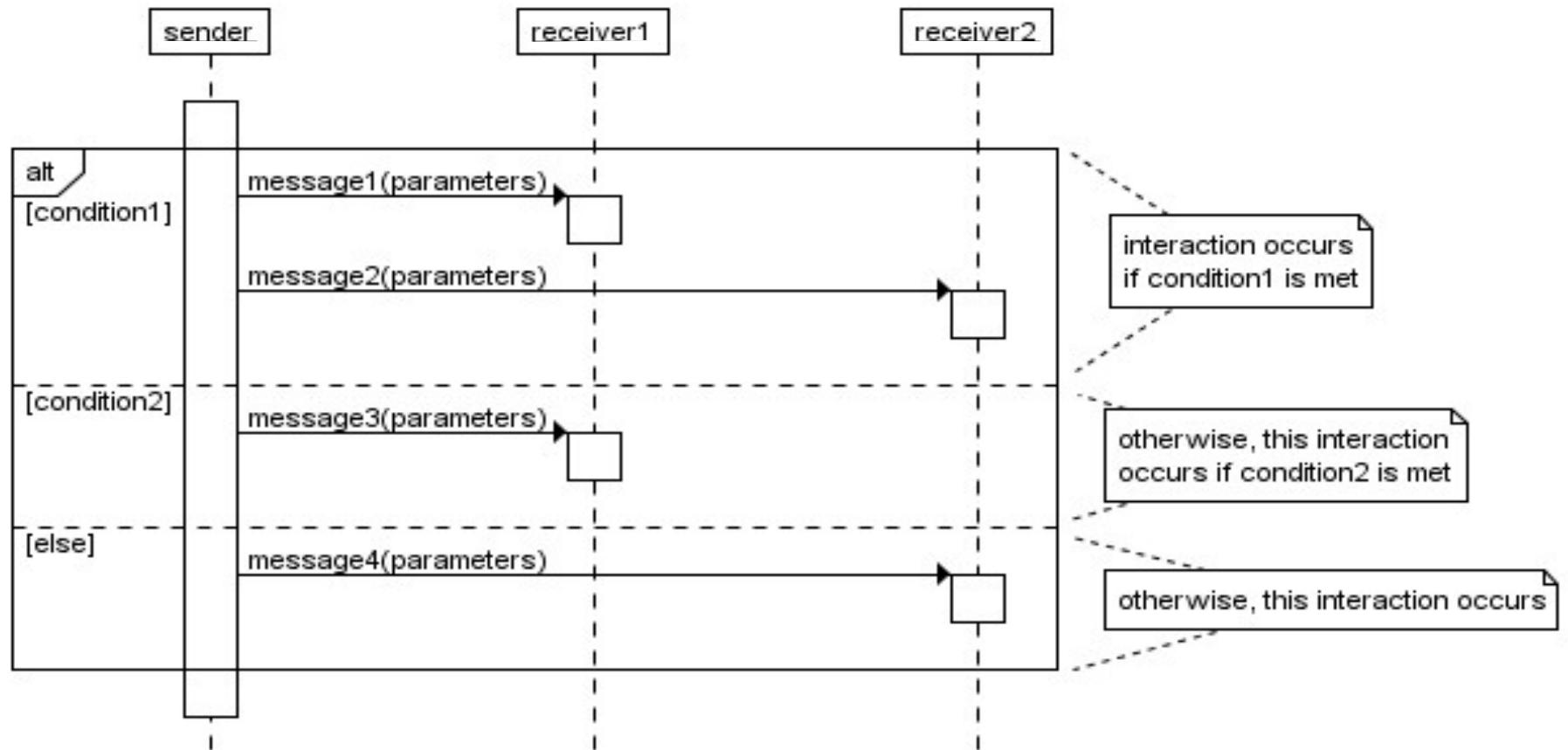
Operador *opt*

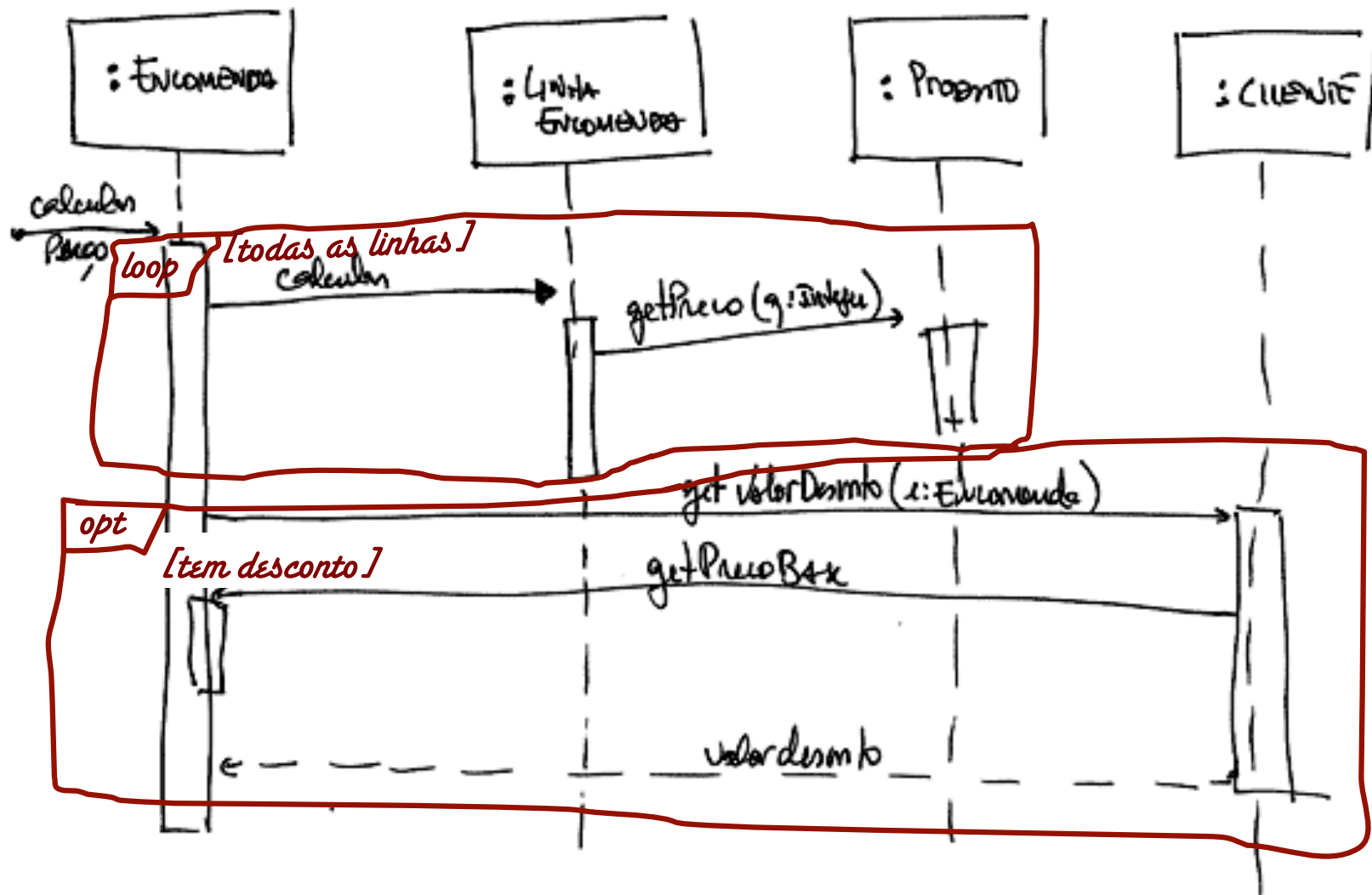


Cálculo do desconto só é efectuado se a aguarda **tem desconto** se verificar.



Os 2 fluxos possíveis (conforme o valor da guarda) são mutuamente exclusivos, pelo que apenas um deles será seguido.



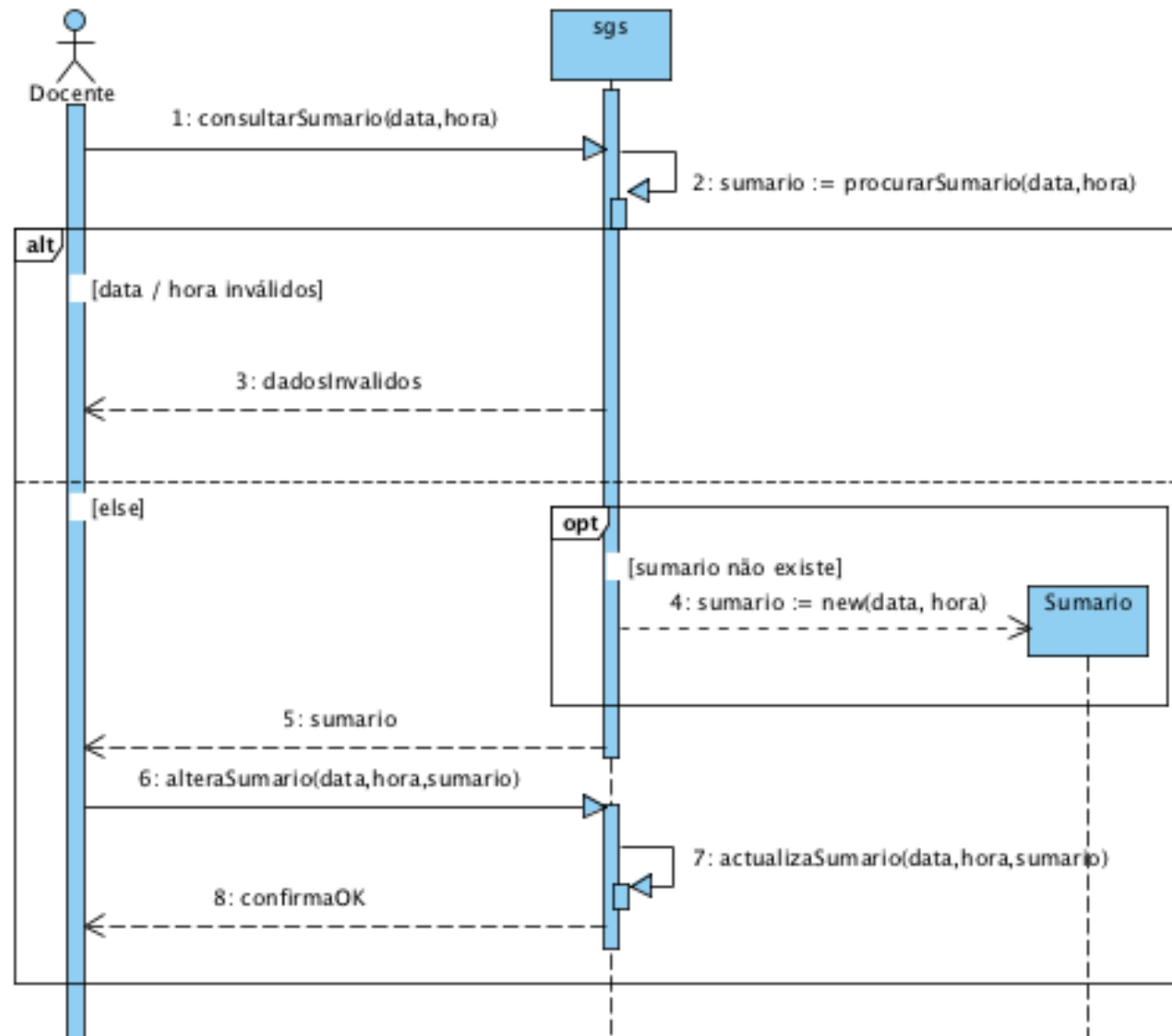


Cálculo do preço é efectuado para todas as linhas da encomenda.



Um exemplo melhor formatado...

Este diagrama de sequência representa um Use case!

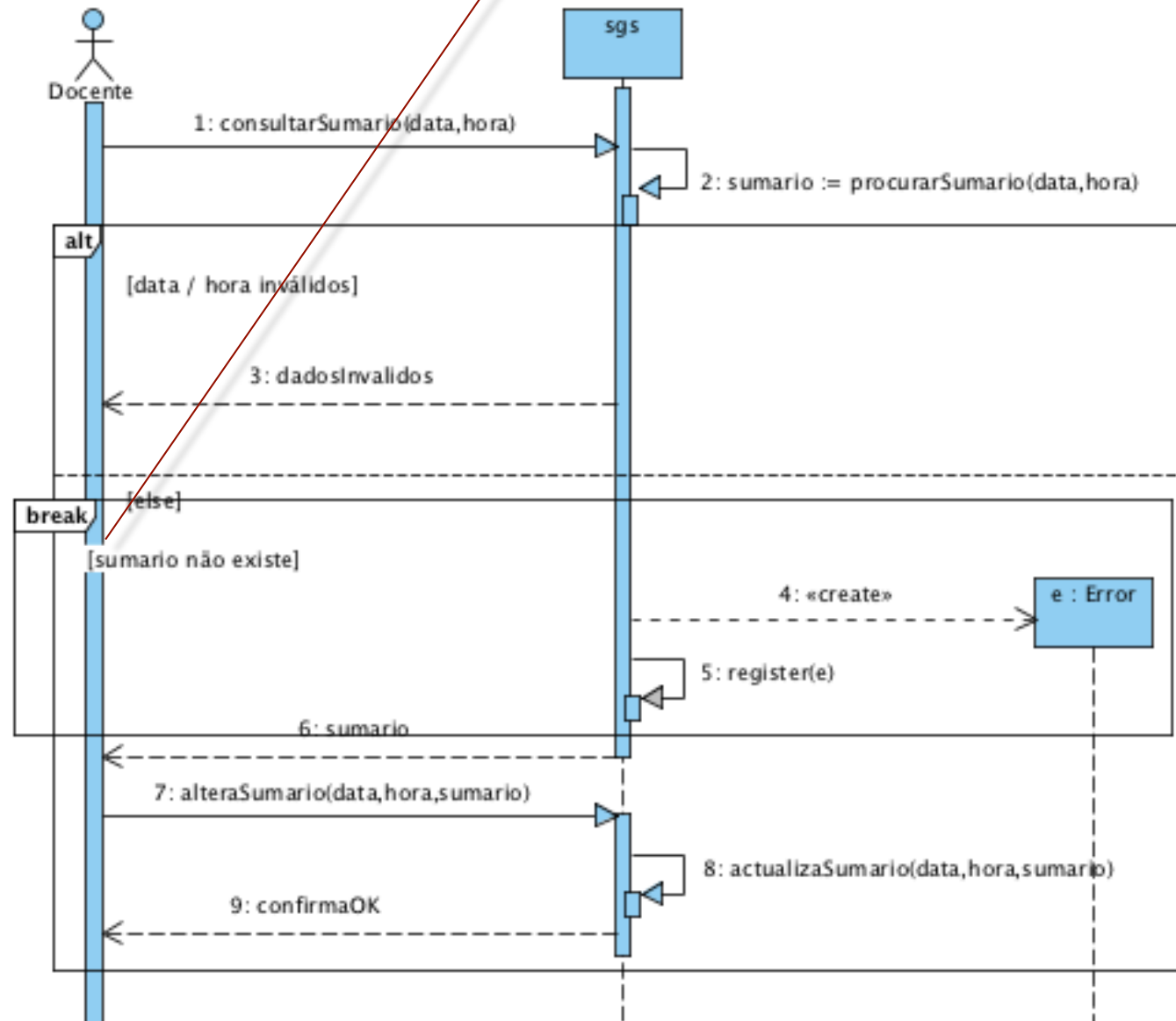


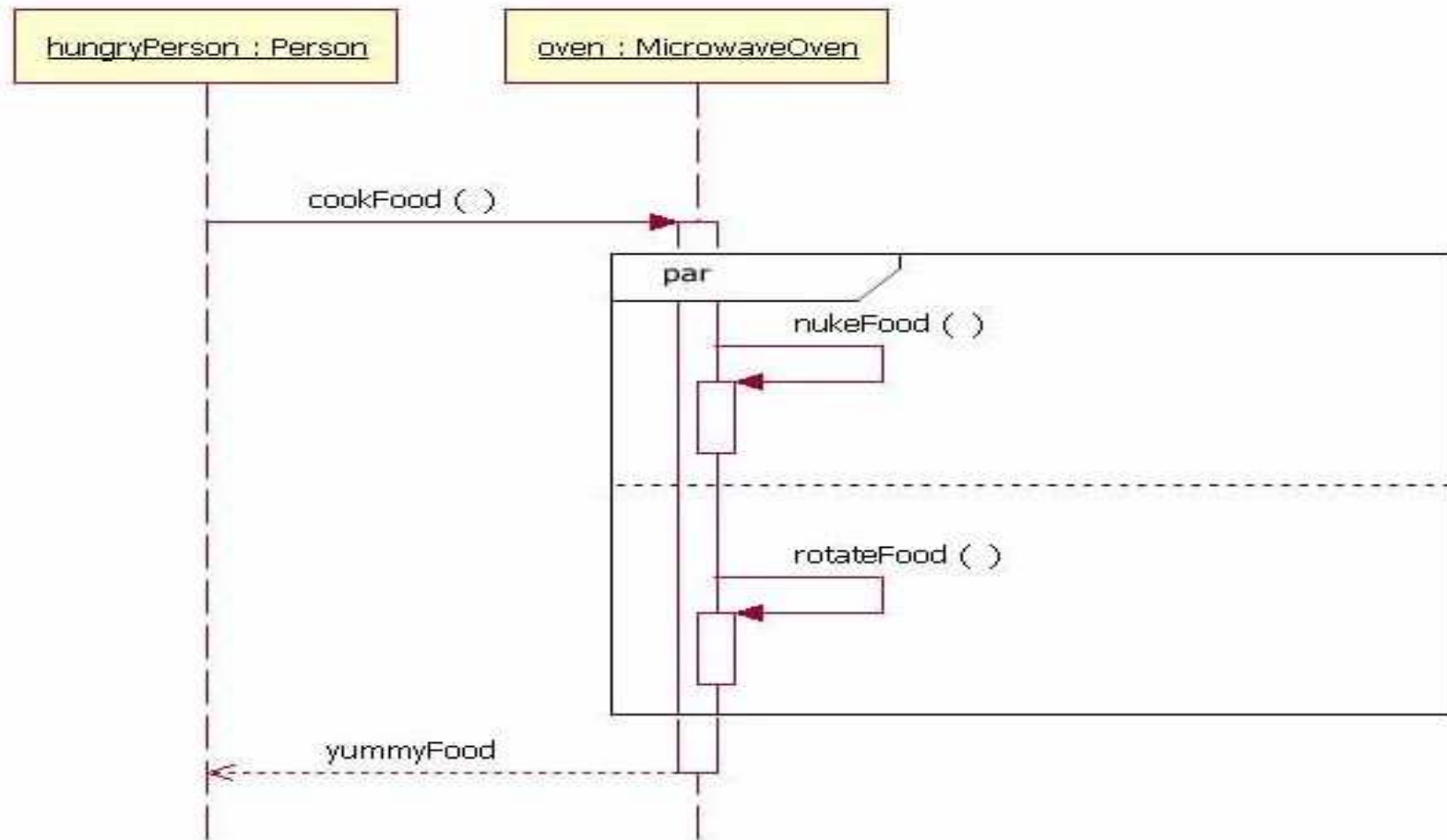


275

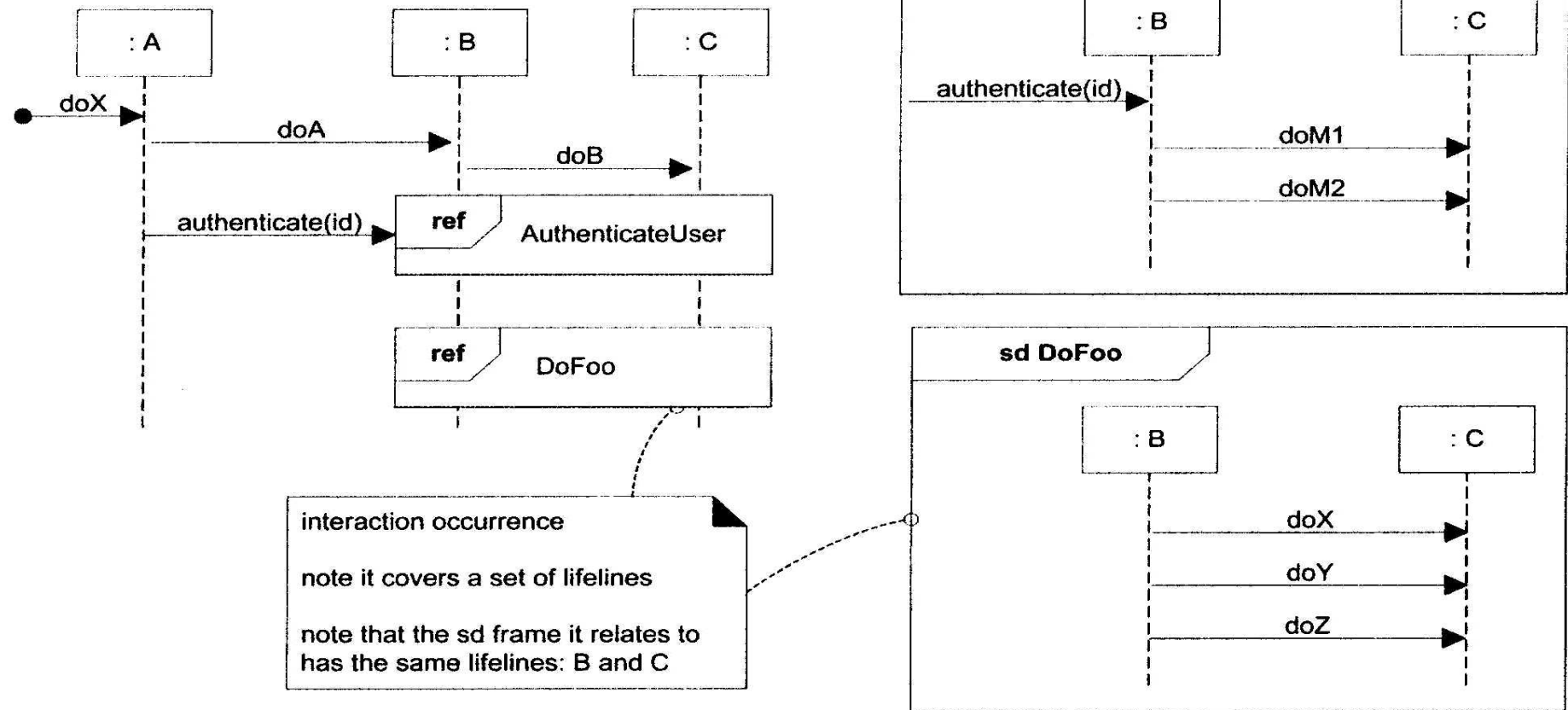
Operador *break*

Se não existe, regista erro e termina.





Uma pessoa esfomeada envia a um micro-ondas uma mensagem para cozinhar uma refeição. O micro-ondas envia a si próprio duas mensagens, uma para “bombardear” e outra para “rodar” a comida, tarefas que são realizadas em paralelo. Quando ambas estiverem concluídas, a esfomeada pessoa recebe como resultado comida “nham-nham” (ie. deliciosa).



A estruturação de diagramas pode ser feita identificando diagramas de sequência usando **sd nome** e referenciando-os usando o frame **ref**.



DIAGRAMAS DE SEQUÊNCIA

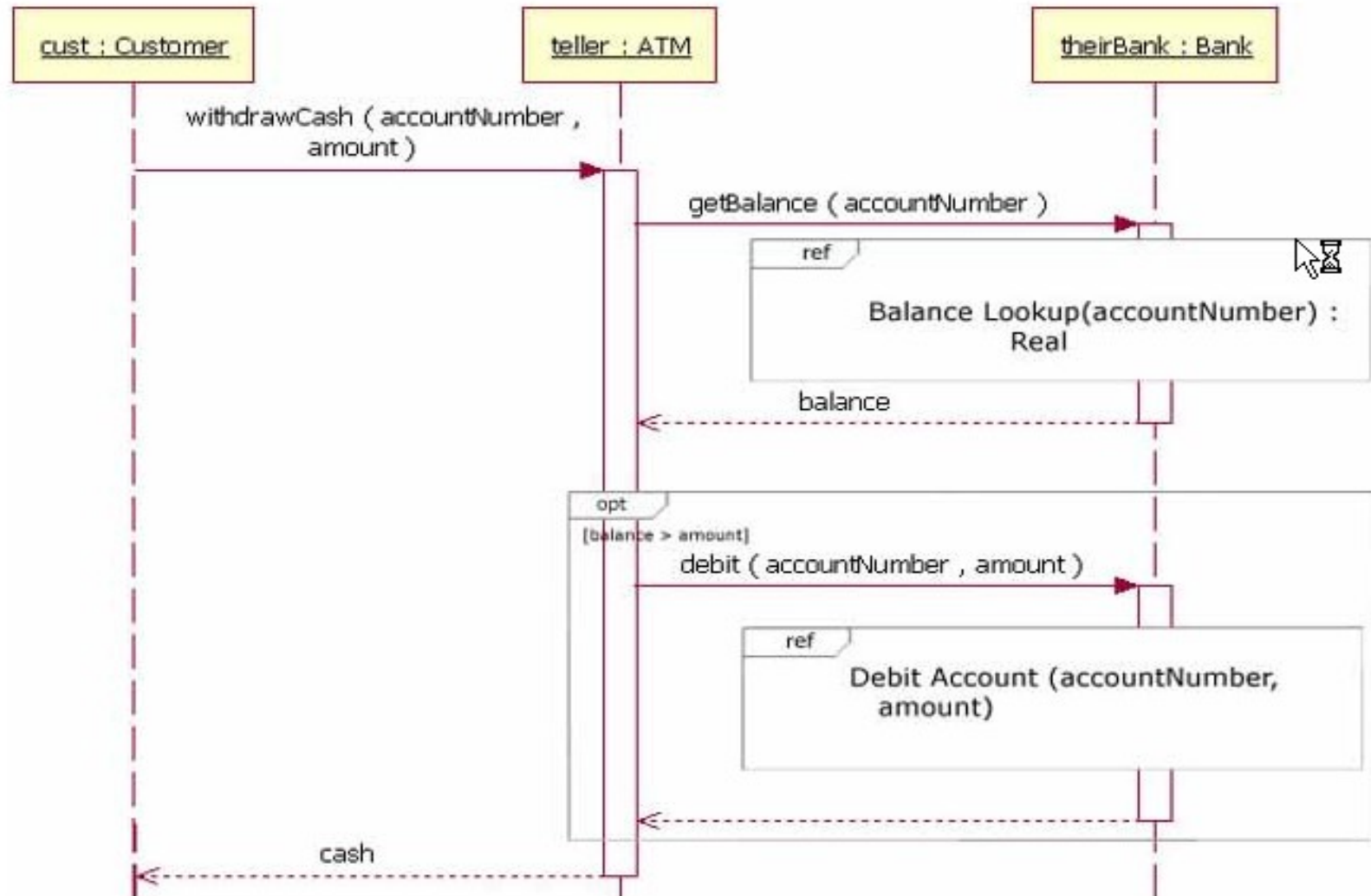


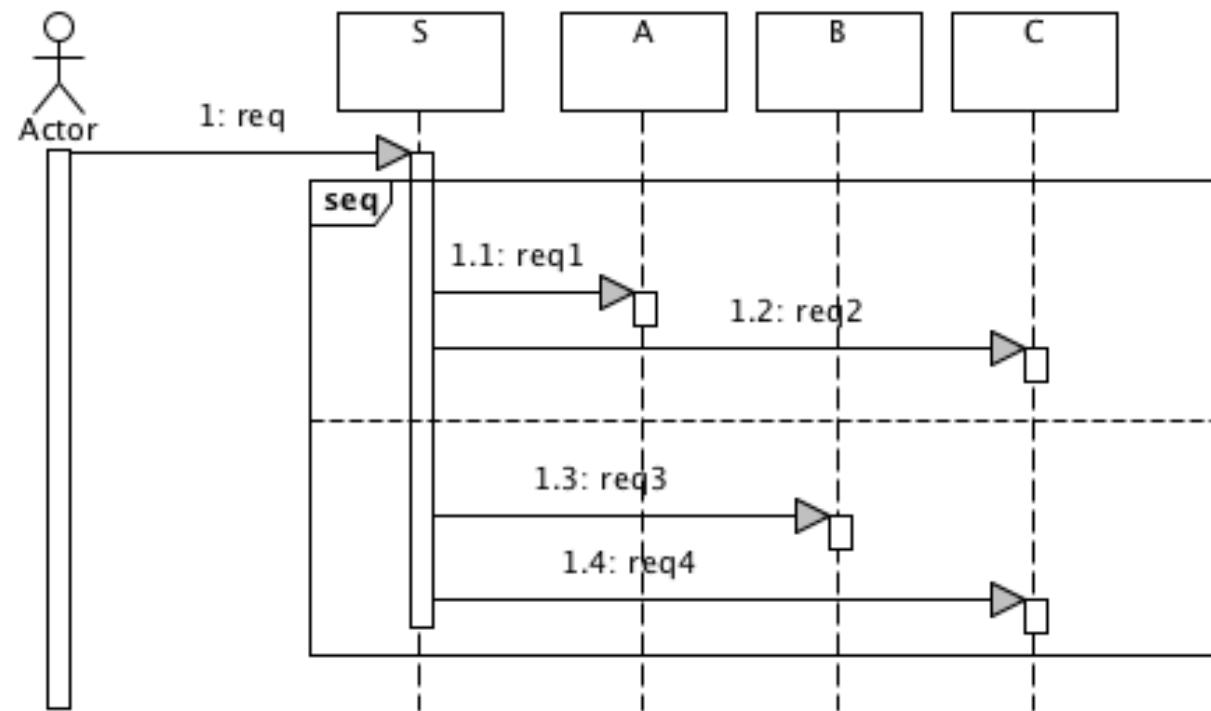
Diagrama de Sequência que referencia dois métodos



Outros operadores

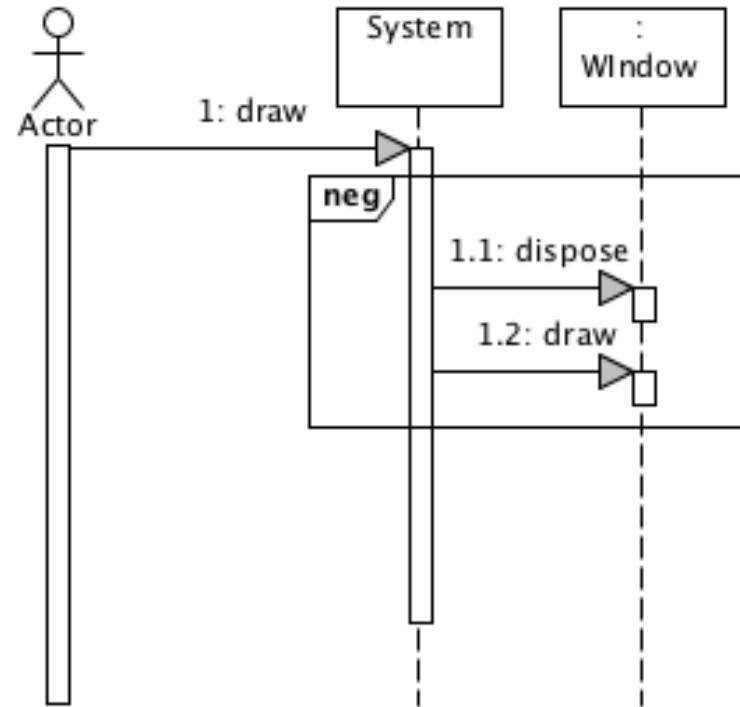
- **critical** - o operando executa de forma atómica
- **seq** (sequenciação fraca) - todos os operandos executam em paralelo, mas eventos enviados a uma mesma linha de vida acontecem na mesma sequência dos operandos
- **strict** - os operandos executam em sequência
- **neg** - negação, o operando mostra uma interacção inválida
- **assert** - mostra o único comportamento válido naquele ponto
- **ignore** - indica mensagens intencionalmente omitidas da interacção (ignore {m1, m2, ...})
- **consider** - indica mensagens intencionalmente incluídas na interacção (dual de ignore)

Operador *seq*



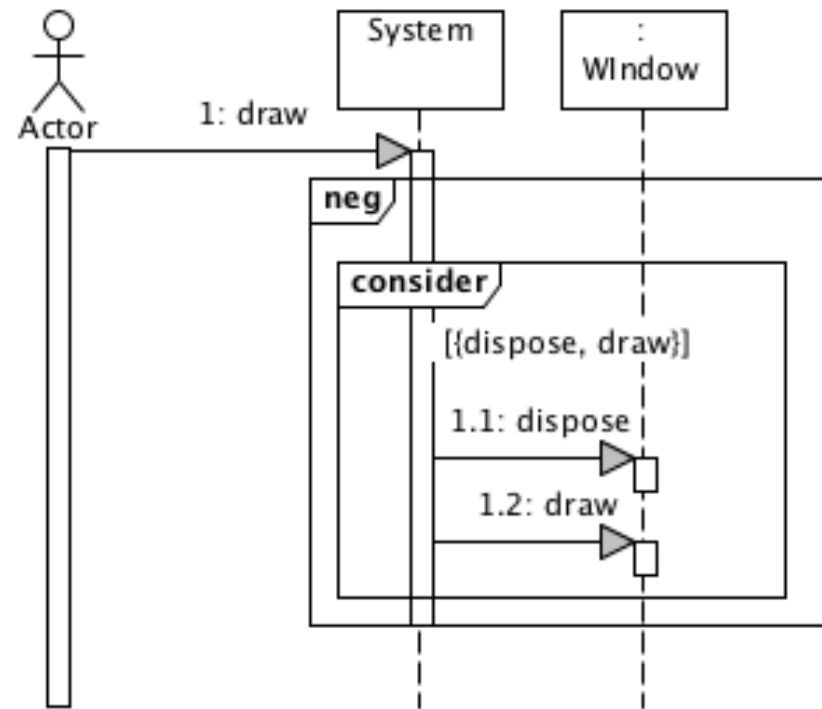
Eventos *req1* e *req3* podem acontecer em paralelo. Evento *req2* acontece antes de evento *req4* (porque ambos vão para C).

Operadro *neg*

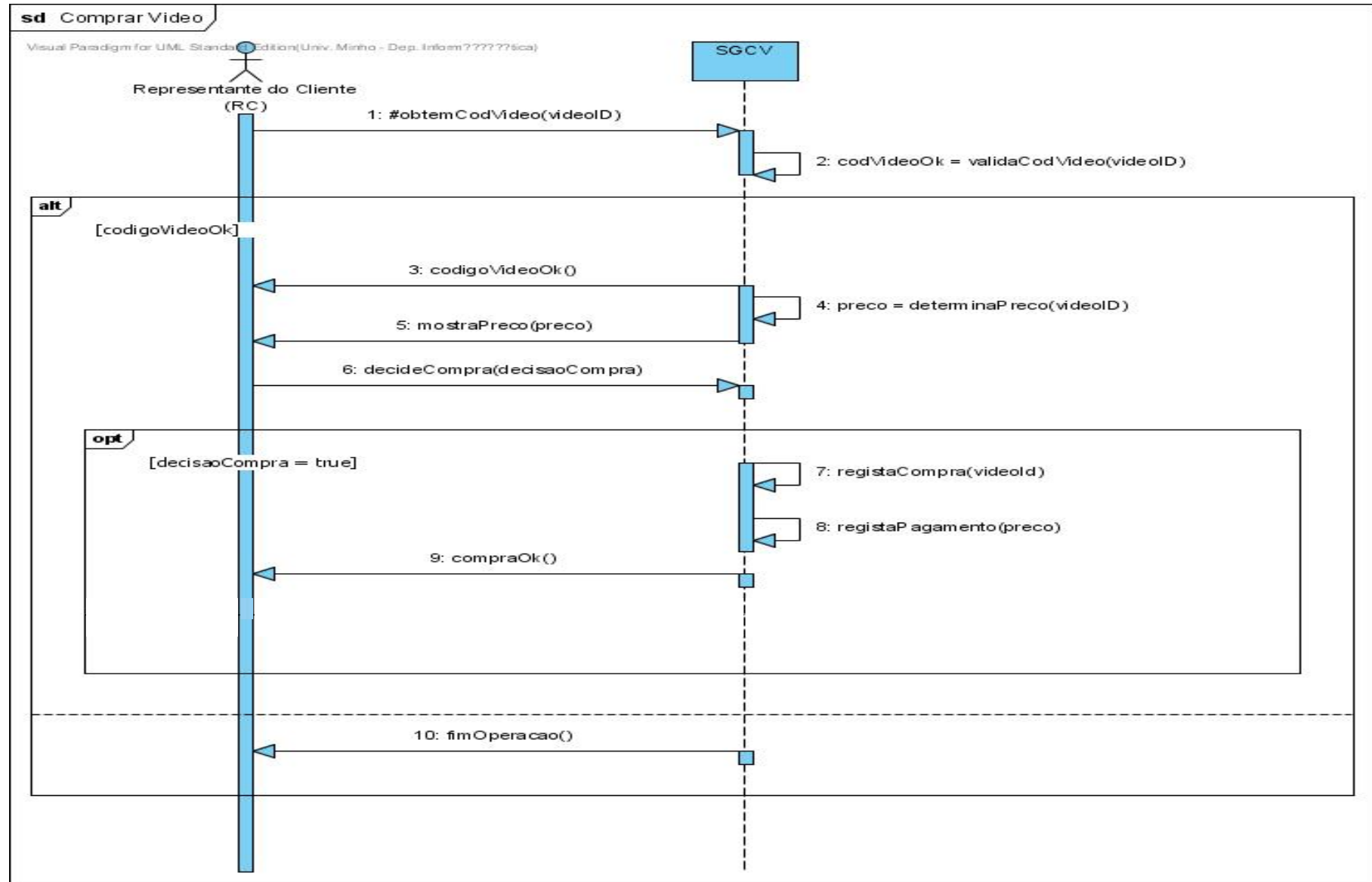


Não é válido desenhar numa janela depois de ela ter sido removida.

Operador *consider*



Podem existir outros eventos pelo meio...
(possível problema?)

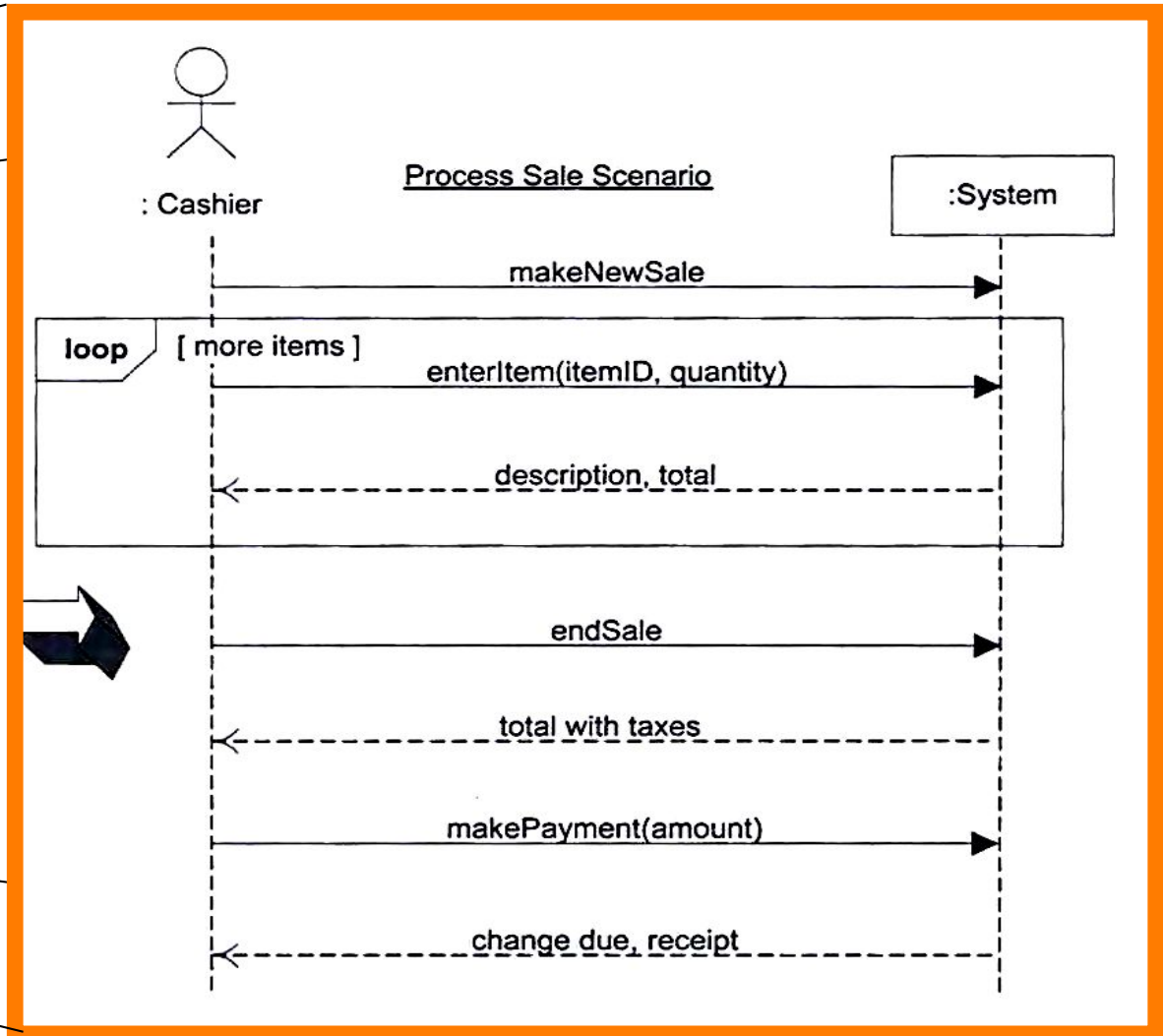




REPRESENTAÇÃO DE USE CASE EM DIAGRAMAS DE SEQUÊNCIA

Simple cash-only Process Sale scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
- ...





Diagramas de Sequência

Sumário

- Diagramas de Sequência (continuação)
- Representação de restrições de tempo / duração
- Diagramas de sequência e distribuição de responsabilidades
- Fragmentos combinados: operadores, oprenados e guardas
- Operadores mais comuns: opt, alt, loop, break, par, ref. Exemplos
- Outros operadores: critical, seq, strict, neg, assert, ignore, consider
- Especificação de Use Cases com diagramas de sequência: primeira abordagem.