

T13-14 - Modelação comportamental

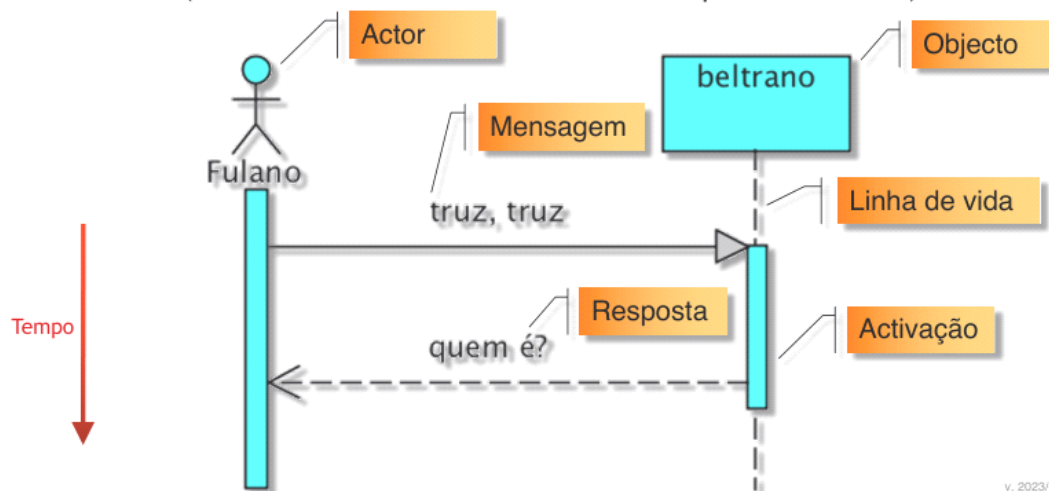
2 de dezembro de 2023 21:14

MODELAÇÃO COMPORTAMENTAL (Diagramas de sequência)

↳ Foco no ordenamento temporal das trocas de mensagens

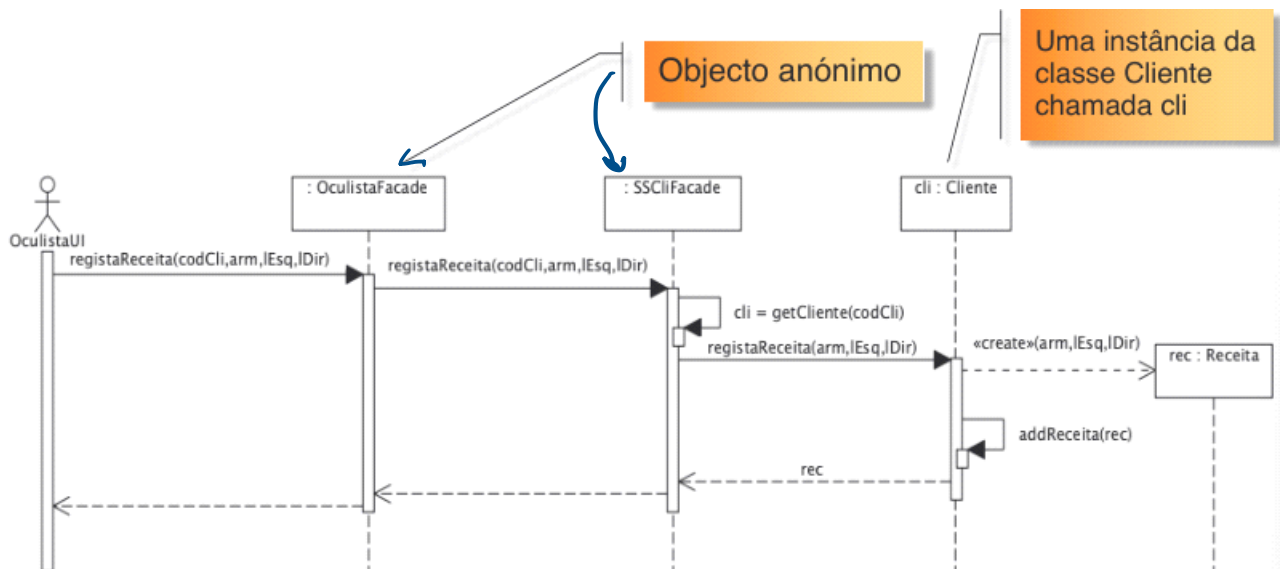
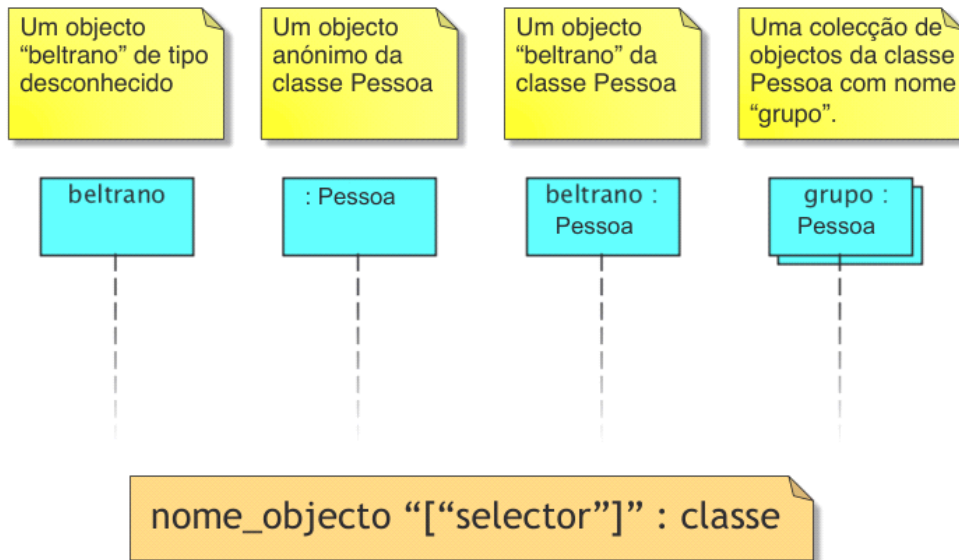
Notação essencial

- representam as interações entre objectos através das mensagens que são trocadas entre eles
- a ênfase é colocada na ordenação temporal das mensagens
- permitem analisar a distribuição de “responsabilidade” pelas diferentes entidades (analisar onde está a ser efectuado o processamento)



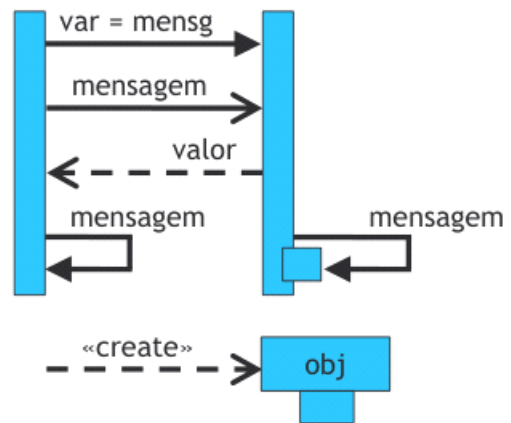
v. 2023/21

Objectos

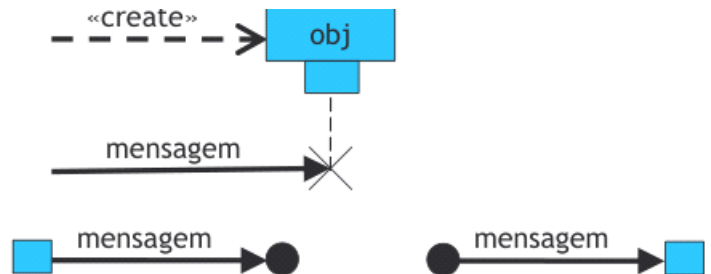


Mensagens

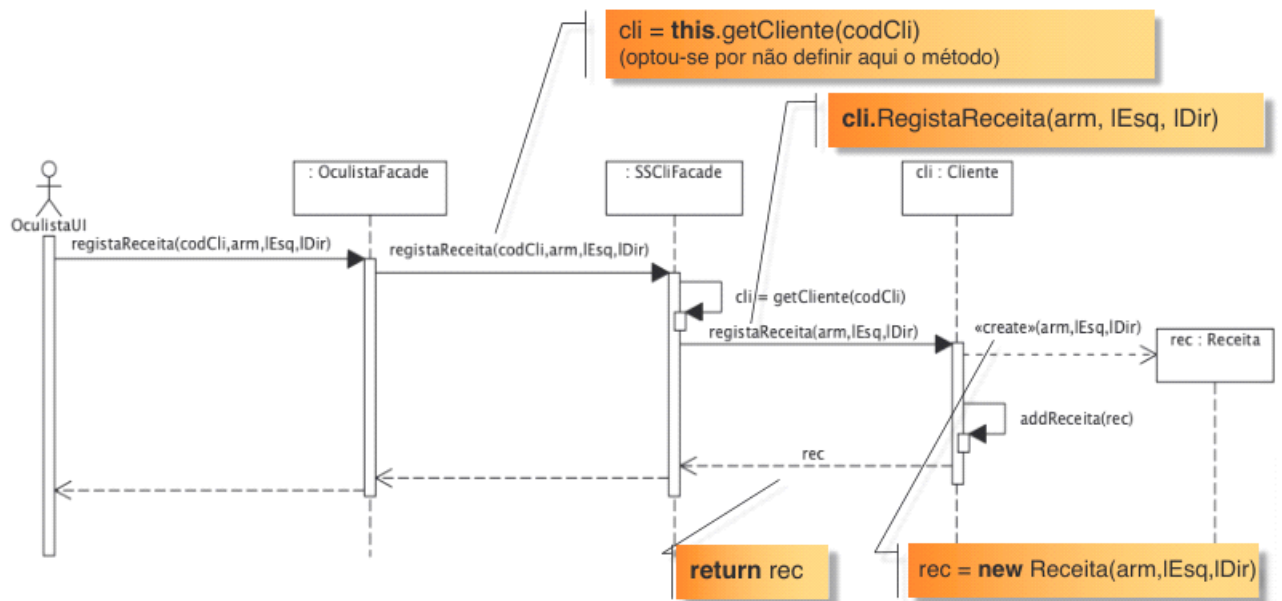
- invocação síncrona
- invocação assíncrona
- return/resultado
- self messages
- criar objectos



- criar objectos
- destruir objectos
- lost/found messages



[atributo '='] nome_da_operação_sinal [argumentos] [':' tipo_resultado]

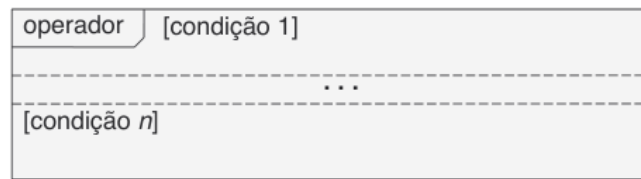


- Todas as invocações são síncronas.
- Dois dos métodos não estão aqui definidos (+ construtor).
- **Atenção!** O objecto que envia a mensagem tem que "conhecer" o objecto a quem a envia.

v. 2023/24

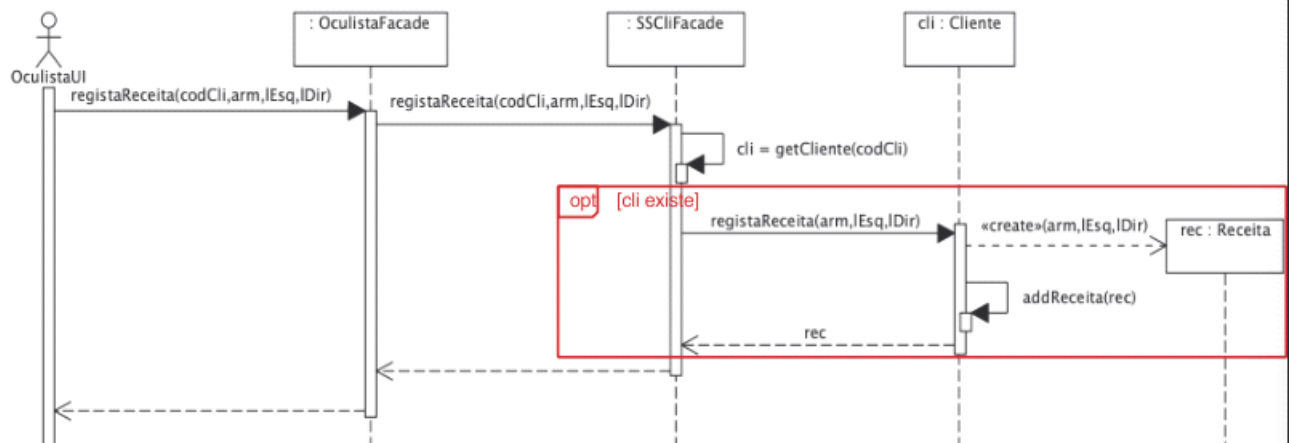
Fragmentos Combinados

- Um fragmento combinado agrupa conjuntos de mensagens
- Permitem expressar fluxos condicionais e estruturar os modelos



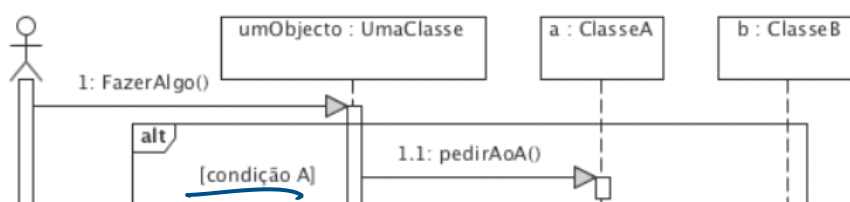
- Operadores mais comuns
 - **alt** - define fragmentos alternativos (mutuamente exclusivos)
 - **loop** / **loop(n)** - fragmento é repetido enquanto a guarda for verdadeira / **n** vezes
 - **opt** - fragmento opcional (ocorre se a guarda for verdadeira)
 - **par** - fragmentos ocorrem em paralelo
 - **break** - termina o fluxo
 - **ref** - referência a outro diagrama

⇒ Operador *opt*

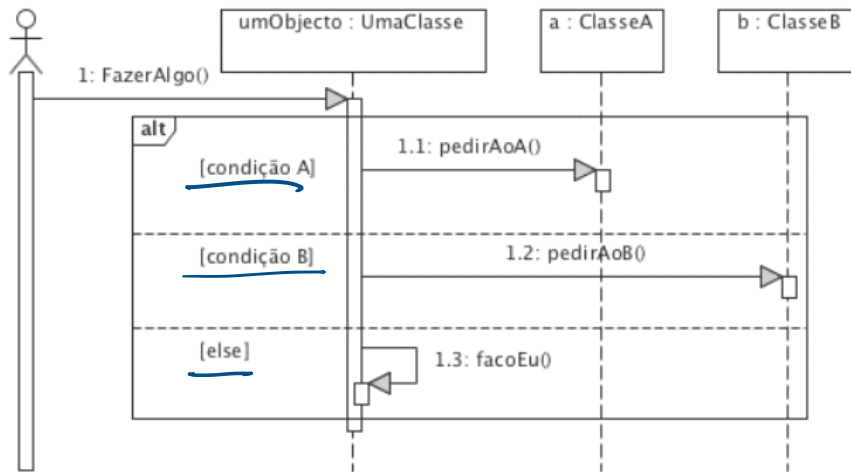


Registo só é efectuado se cliente existe.

⇒ Operador *alt*



→ Operador **alt**

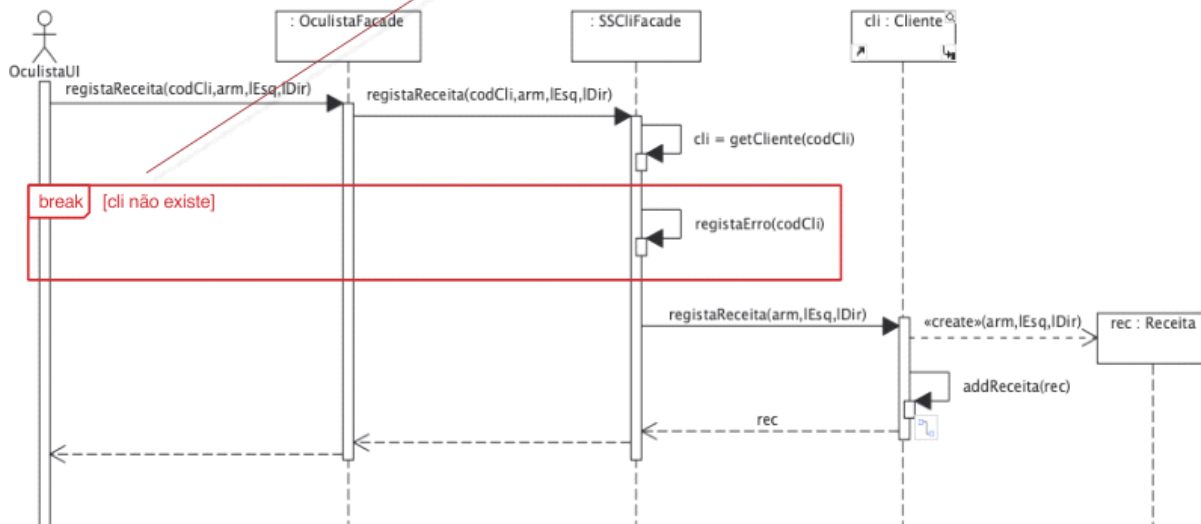


- Os fluxos possíveis são mutuamente exclusivos, pelo que apenas um deles será seguido.
- Se mais que uma condição se verificar, não está definido qual acontece.

v. 2023/20

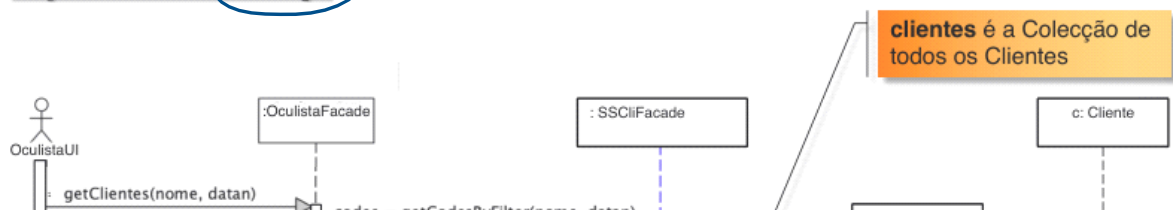
Operador **break**

Se cliente não existe, regista erro e termina.

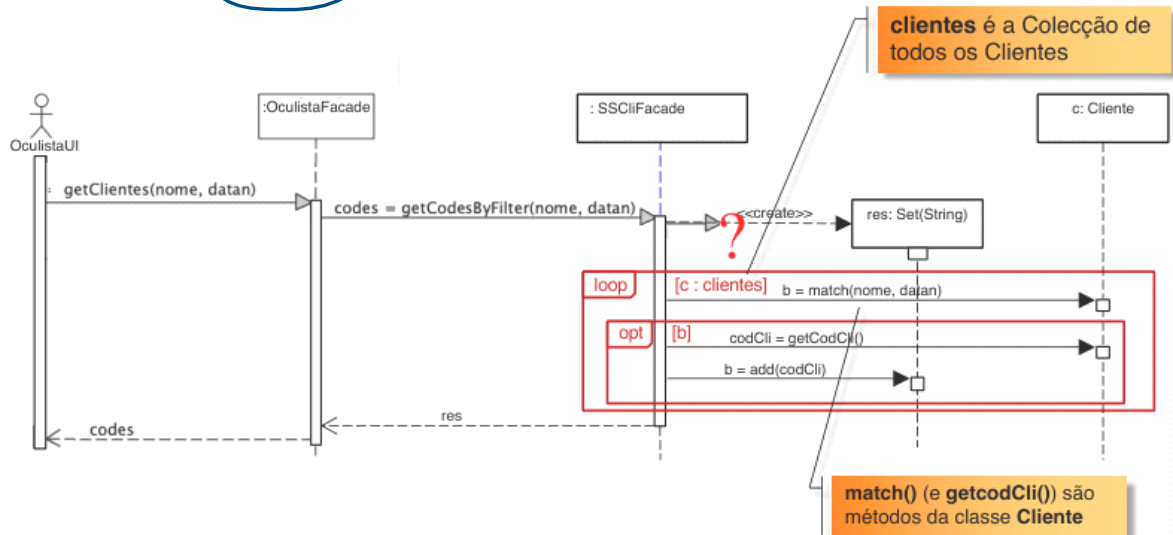


Registo só é efectuado se cliente existe.

Operador **loop**

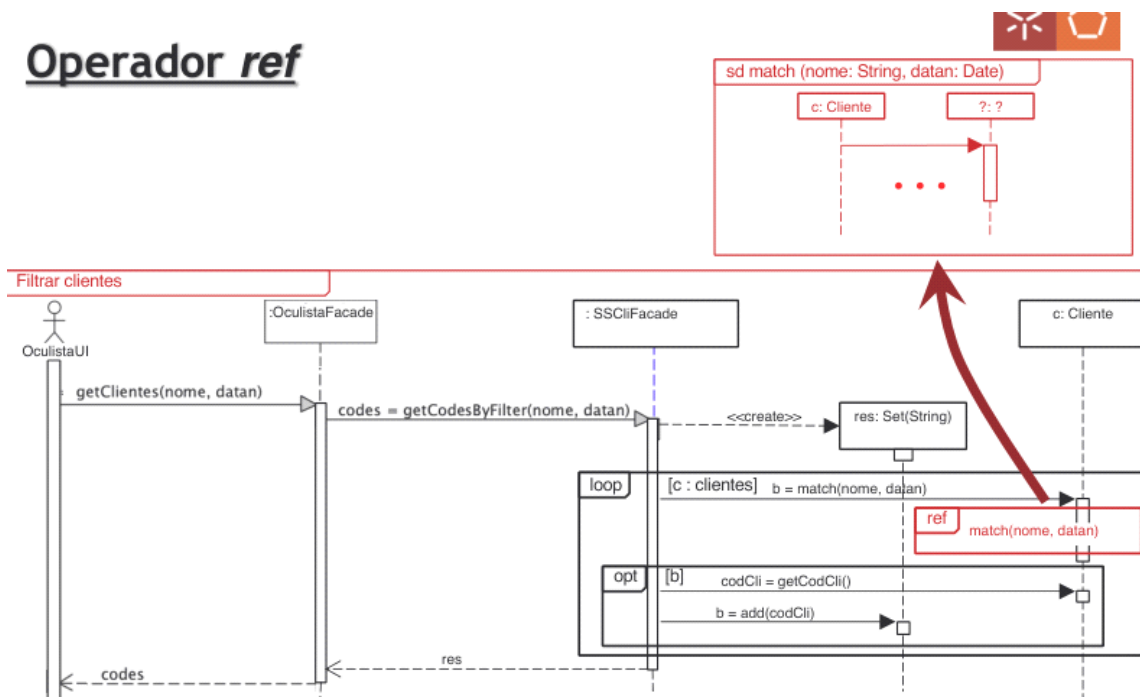


Operador *loop*



codes é o conjunto dos códigos dos clientes que satisfazem o critério.

Operador *ref*



- Todos os diagramas devem ter um nome
- Um SD pode reutilizar outros SD referenciando-os num fragmento com o operador **ref** – permite estruturar os modelos

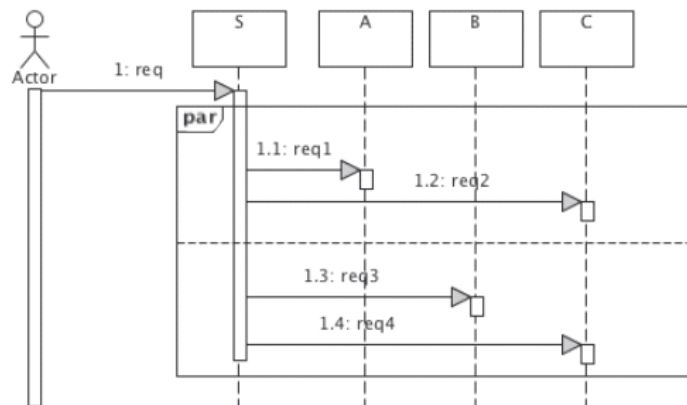
v. 2023/24

Outros operadores

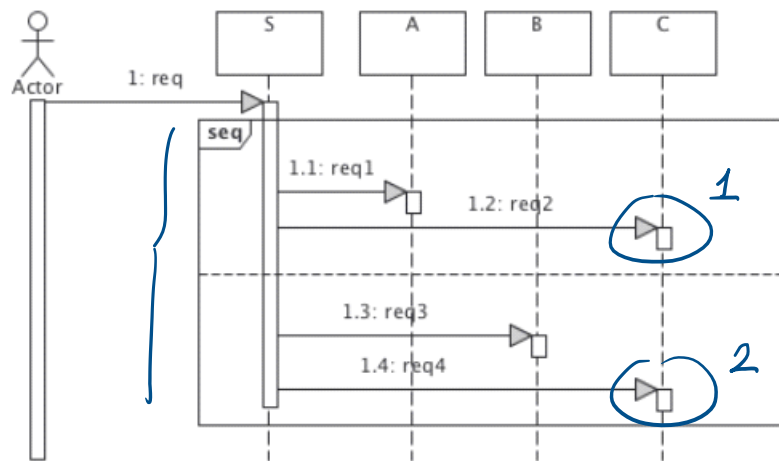


- **critical** - o operando executa de forma atômica
- **par** - os operandos executam em paralelo
- **seq** (sequenciação fraca) - todos os operandos executam em paralelo, mas eventos enviados a uma mesma linha de vida acontecem na mesma sequência dos operandos
- **strict** - os operandos executam em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

Operador *par*

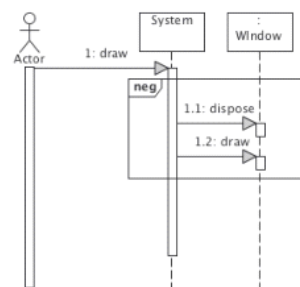


Eventos *req1* e *req2* podem acontecer em paralelo com eventos *req3* e *req4*. Nenhuma ordem é imposta.



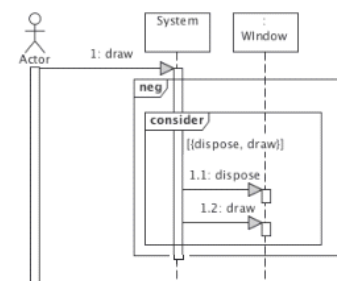
Eventos *req1* e *req3* podem acontecer em paralelo. Evento *req2* acontece antes de evento *req4* (porque ambos vão para C).

Operador *neg*



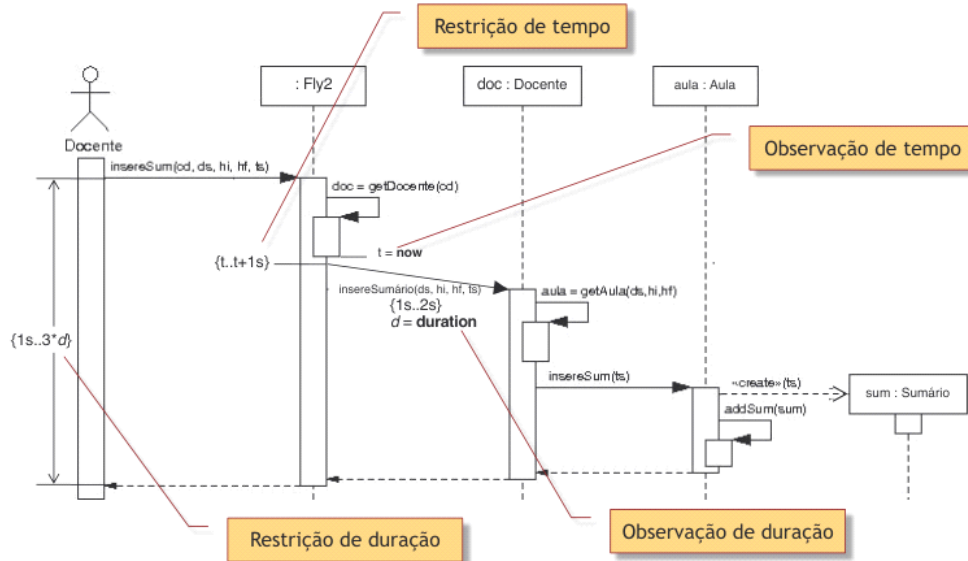
Não é válido desenhar numa janela depois de ela ter sido removida.

Operador *consider*



Porque podem existir outros eventos pelo meio...

Restrições de tempo / duração



Distribuição de responsabilidades

