Diagrama de Sequência

Instituto de Computação

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

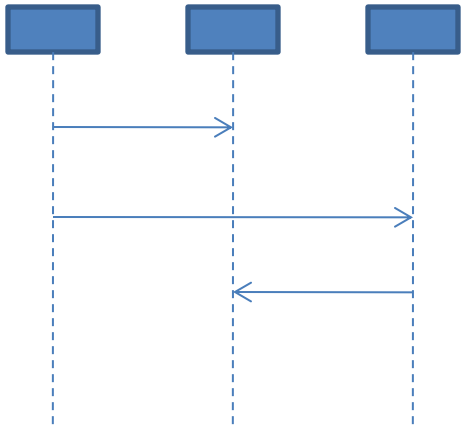
O que é?

Instituto de computação

- Diagrama criado para modelagem da interação entre objetos
  - Detalha como objetos colaboram para implementar um cenário de caso de uso
  - Útil para ajudar na identificação dos métodos das classes

A 1 km de distância...

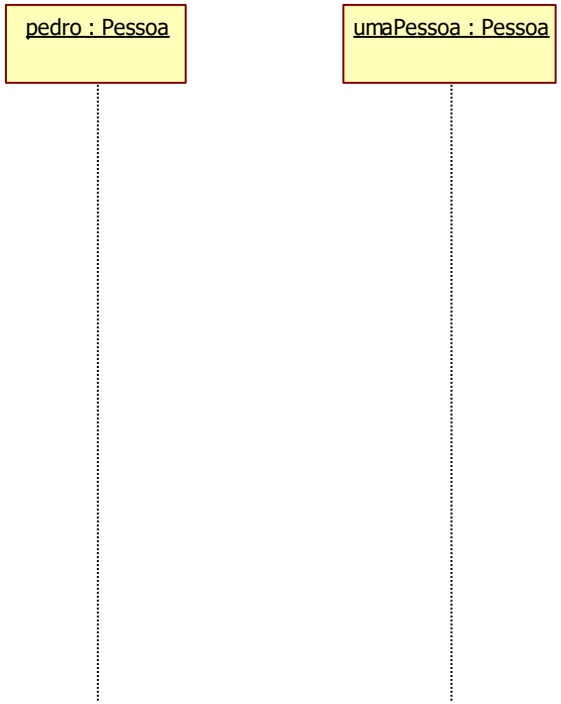
- Caixas representando objetos
- Linhas verticais representando a vida do objeto
- Linhas horizontais representando troca de mensagens (chamadas de métodos)



A 1 metro de distância... dos objetos

Instituto de computação

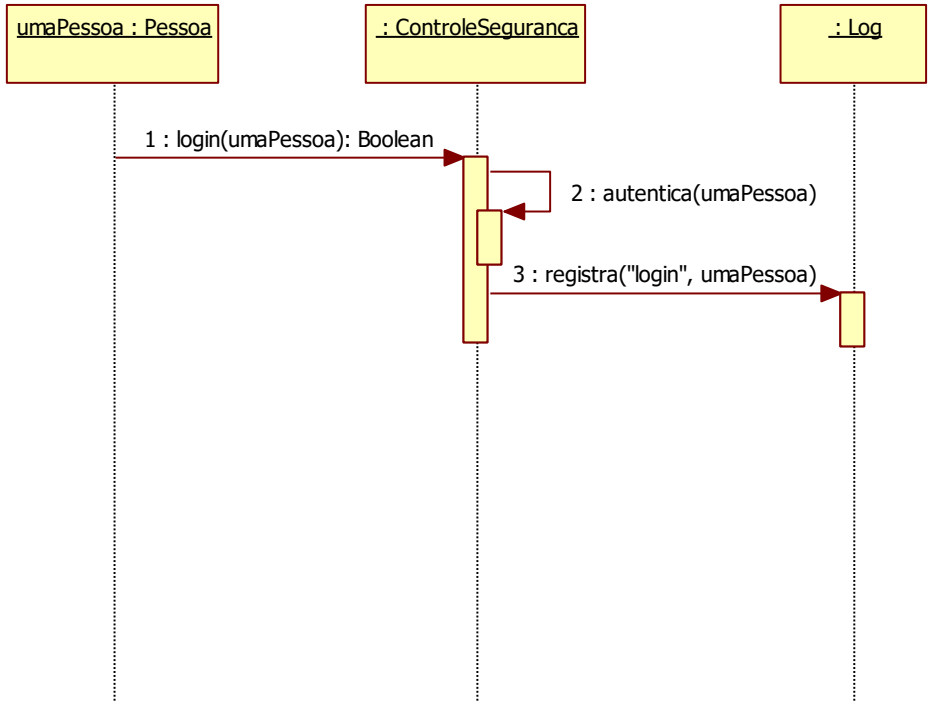
- Os objetos são instâncias de alguma classe definida no diagrama de classes
  - O nome de um objeto é da forma *nome : classe*
- Em situações onde um nome específico não pode ser identificado (ex.: pedro : Pessoa), utilize:
  - Um nome genérico (ex.: umaPessoa : Pessoa)
  - Um nome único (ex.: aPessoa : Pessoa)
  - Ou omita o nome (ex.: : Pessoa)
- Uma linha pontilhada sai do objeto representando o momento da sua criação em diante
  - Quanto mais para baixo, mais tempo passou



A 1 metro de distância...
Das mensagens

Instituto de computação

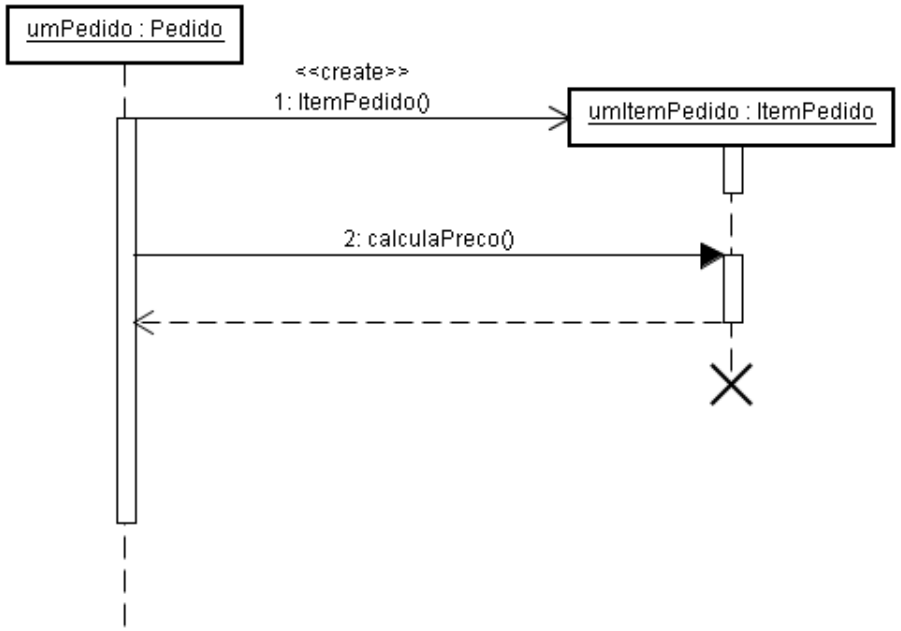
- A interação entre objetos é representada por troca de mensagens (chamadas de métodos)
  - Para outros objetos
  - Para o mesmo objeto (auto-mensagem)
- Uma mensagem contém a assinatura do método que está sendo chamado
- Uma barra de ativação indica o escopo de execução do método



A 1 metro de distância...
Das mensagens

Instituto de computação

- Mensagem de criação
  - Aponta diretamente para o objeto e é marcada com <<create>>
- Mensagem de retorno
  - Opcional, e normalmente é omitida
  - Usa seta tracejada
- Marca de destruição
  - Indica o término da vida de um objeto com um “X”



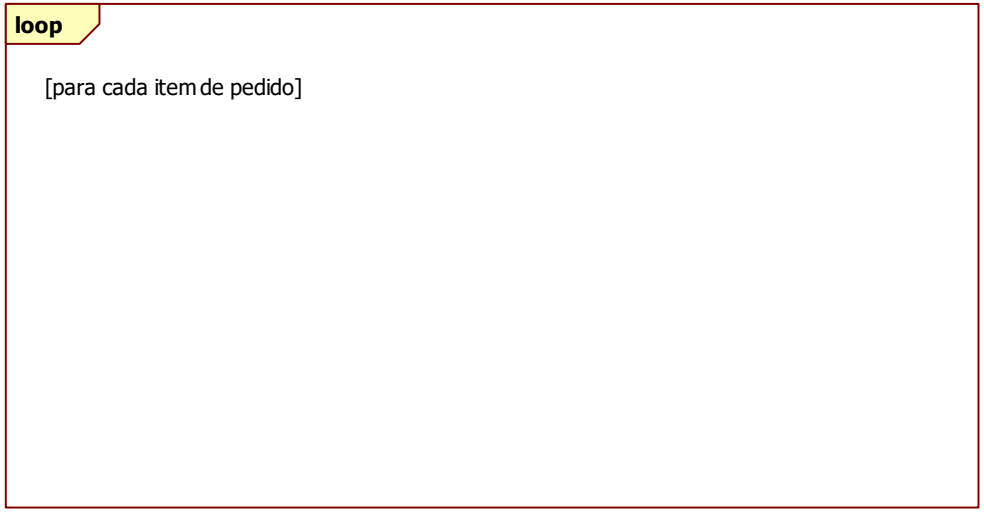
Mas como representar um algoritmo mais complexo?

- Exemplo:

```
Para cada item de produto
    Se o valor do produto for maior que 10000 então
        Despacha com cuidado
    Caso contrário
        Despacha normalmente
Se precisa de confirmação
    Envia confirmação
```

Repetições

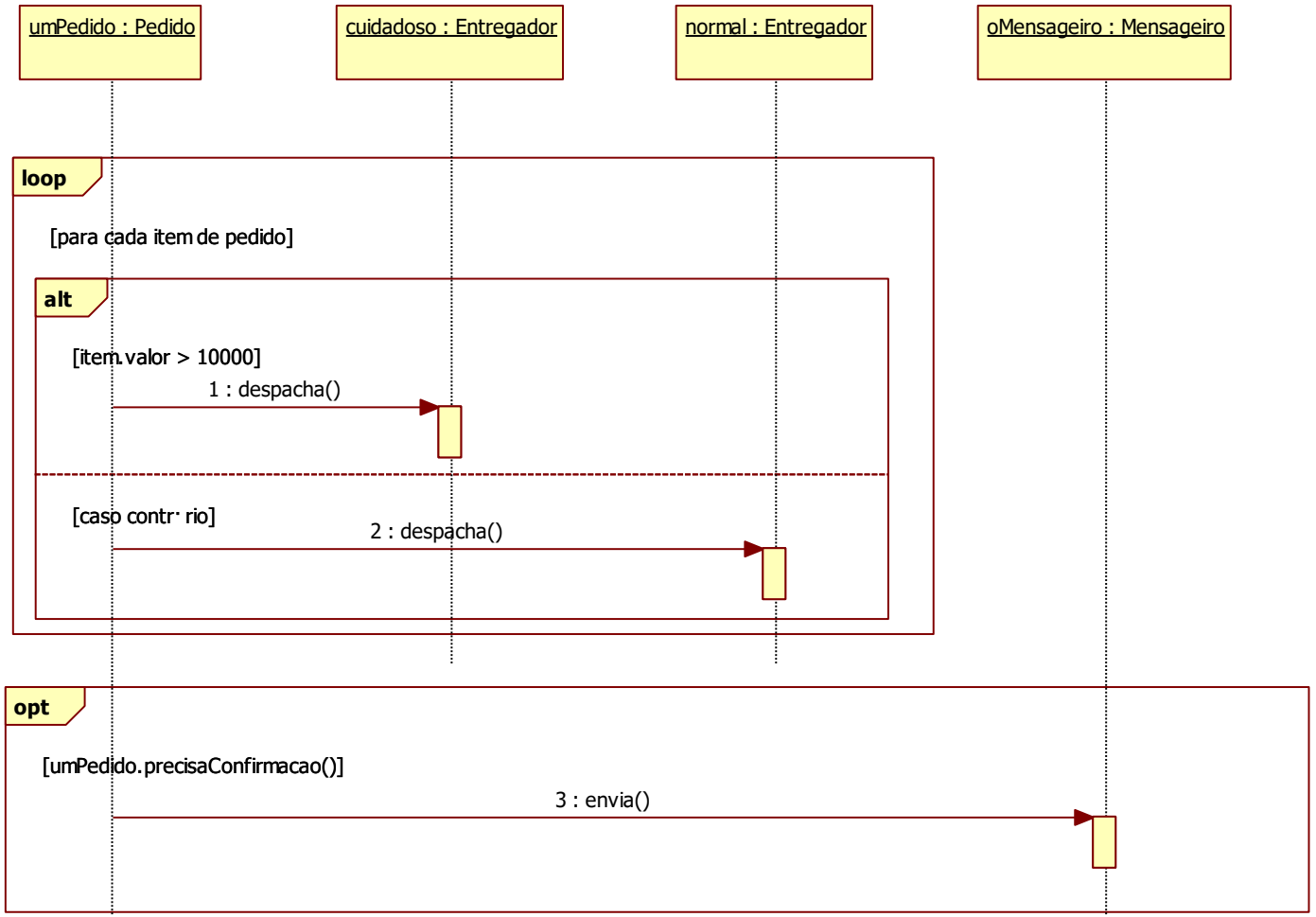
- O diagrama de sequência permite que repetições sejam feitas durante o fluxo
- Para isso são utilizados quadros (*frames*) do tipo *loop*



Decisões

- O diagrama de sequência permite que decisões sejam tomadas durante o fluxo
- Para isso são utilizados quadros (*frames*) do tipo *alt* ou *opt* com condições de guarda

Exemplo


umPedido : Pedido
cuidadoso : Entregador
normal : Entregador
oMensageiro : Mensageiro
loop
[para cada item de pedido]
alt
[item.valor > 10000]
1 : despacha()
[caso contrário]
2 : despacha()
opt
[umPedido.precisaConfirmacao()]
3 : envia()

Outros quadros disponíveis

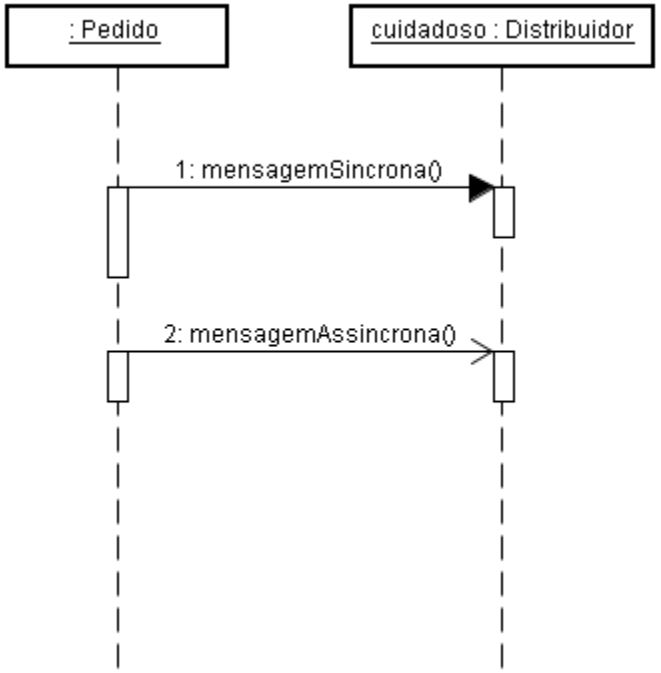
Instituto de computação

- Além dos quadros do tipo *loop*, *opt* e *alt*, existem outros tipos, entre eles:
  - *par*: Contém vários seguimentos e todos são executados em paralelo
  - *region*: Determina uma região crítica, que deve ter somente uma *thread* em execução em um dado momento

Chamada síncrona x assíncrona

Instituto de computação

- É possível utilizar dois tipos de chamada de métodos no diagrama de sequência:
  - 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)



Quando usar diagrama de sequência?

Instituto de computação

- Para representar a interação entre diferentes objetos visando atender a um caso de uso
- Para ajudar a encontrar os métodos do diagrama de classes
- Cuidado: não use diagrama de sequência...
  - Para métodos muito simples (ex.: get e set)
  - Para definição precisa de como será o código

Exercício

Instituto de computação

- Elabore um diagrama de sequência para o algoritmo *Quicksort* (versão ingênua)
  - Primeiro elemento da lista de entrada é o pivô
  - Cria outras duas listas com os elementos menores e maiores que o pivô
  - Ordena recursivamente as outras duas listas
  - Concatena a lista de menores ordenada, o pivô e a lista de maiores ordenada, criando a lista de saída ordenada

Bibliografia

- Fowler, Martin. 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd ed. Addison-Wesley Professional.

Diagrama de Sequência

Instituto de Computação

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br