

Diagramas de Interação

- Sequência, Comunicação, Interação Geral e Tempo -

Profa. Ana Paula Lemke

Interações

- A modelagem dos aspectos dinâmicos de um sistema é realizada através de **interações**.
- Uma interação é um comportamento que envolve um conjunto de mensagens trocadas entre objetos dentro de um determinado contexto, objetivando atingir um resultado específico.
- Interações acontecem em função da troca de mensagens entre objetos.

Comunicação entre os objetos - Mensagens

- Uma mensagem pode ser entendida como um pedido para execução de uma operação;
- Um objeto reage a uma mensagem executando a operação solicitada.
- Como decorrência da passagem de uma mensagem, repercute uma ação representada por um comando executável que é uma abstração de um procedimento computacional.
- Tipos de ações:
 - *call*: invoca uma operação sobre um objeto;
 - *return*: retorna um valor a um objeto emissor de mensagem;
 - *create*: criação de um objeto;
 - *destroy*: eliminação de um objeto.

Comunicação entre os objetos - Mensagens

Sintaxe para Mensagens:

return := **message** (**parameter**: **parameterType**):**returnType**

onde,

- **return*** é o nome do valor de retorno;
- **message** é o nome da mensagem;
- **parameter*** é o nome de um parâmetro da mensagem;
- **parameterType*** é o tipo do parâmetro;
- **returnType*** é o tipo do valor de retorno.

Observação: Os elementos com * são opcionais e dependem de como o método foi declarado na classe.

Diagramas de Interação

- Mostram a troca de mensagens em uma colaboração (um grupo de objetos que cooperam), para atingir um objetivo.
- São a base para a definição de operações nas classes. A direção de uma mensagem indica a classe que deve conter a operação que trata a mensagem correspondente.
- Um Diagrama de Interação é composto por:
 - Objetos
 - Ligações
 - Mensagens

Diagramas de Interação

- Permitem examinar:
 - A troca de mensagens entre os objetos sob o ponto de vista temporal;
 - As interações dos objetos dentro do contexto de suas relações estruturais, especificando as mensagens trocadas em função destas relações.
- Principais aplicações:
 - Visualização, especificação, construção e documentação da dinâmica de uma sociedade particular de objetos.
 - Podem ser usados para modelar o fluxo de controle de um caso de uso. No contexto de um caso de uso, uma interação representa um cenário.

Tipos de Diagramas de Interação

- Na UML 2.x, são definidos 4 tipos de diagramas de interação:
 - **Sequência:** enfatiza o ordenamento das mensagens trocadas entre os objetos.
 - **Comunicação:** enfatiza a organização estrutural dos objetos que trocam mensagens.
 - **Interação Geral:** definidos para visualizar o fluxo geral de controle, logo, não mostram em detalhes as mensagens trocadas pelos objetos.
 - **Tempo:** descreve as mudanças no estado ou condição de um objeto de uma classe durante um tempo.

Diagrama de Sequência

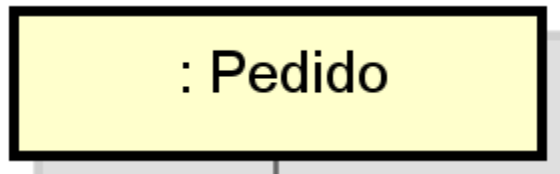
Diagrama de Sequência

- Diagrama de interação que enfatiza o ordenamento (sequência temporal) das mensagens trocadas entre os objetos.
- Representação em um diagrama de sequência:
 - Os objetos participantes da interação são colocados no topo do diagrama;
 - É recomendável colocar o objeto que inicia a interação mais a esquerda no diagrama;
 - As mensagens enviadas e recebidas pelos objetos participantes são colocadas ao longo de um eixo vertical, sendo que a ordenação temporal das mensagens deve ser feita de cima para baixo;
 - Mensagens partem do cliente ao servidor.

Notação do Diagrama de Sequência

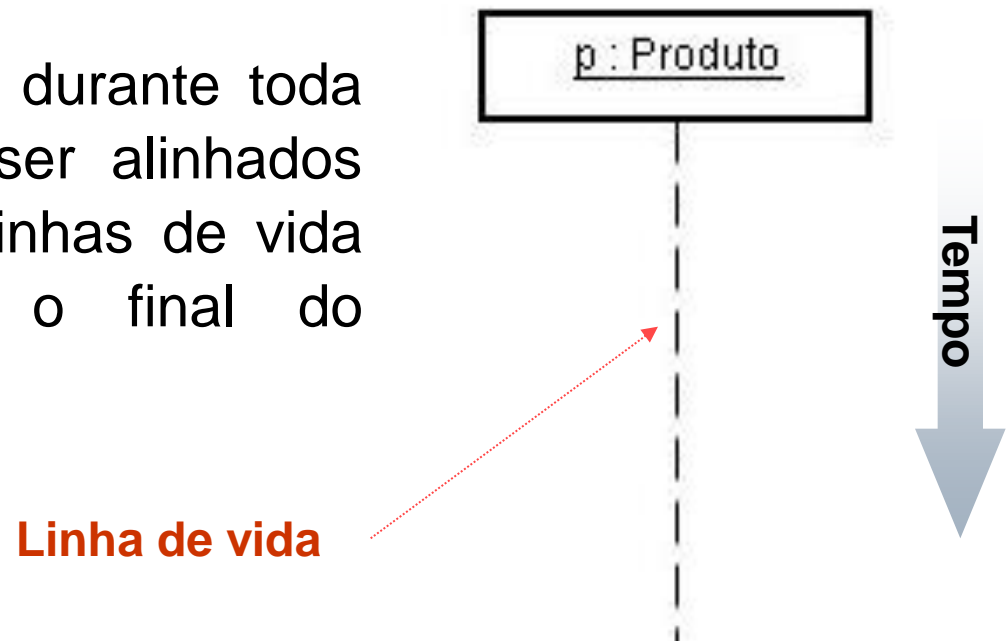
Objetos

- Os objetos do diagrama de sequência tem a mesma notação dos objetos do diagrama de objetos.
- Notação:



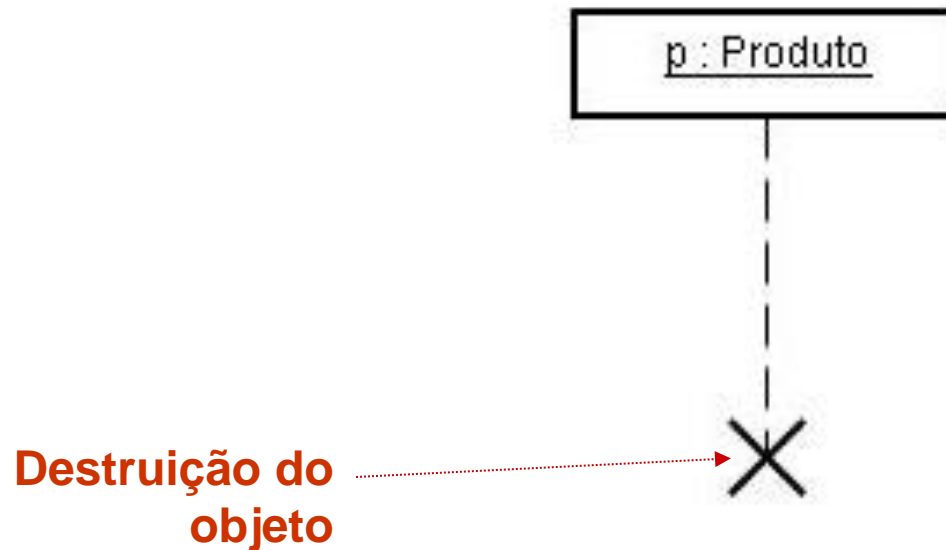
Linha de Vida

- Uma linha de vida de um objeto é representada por uma linha vertical tracejada, definindo a existência de um objeto ao longo de um período de tempo.
- Objetos que existem durante toda a interação devem ser alinhados no topo, com suas linhas de vida estendendo-se até o final do diagrama.



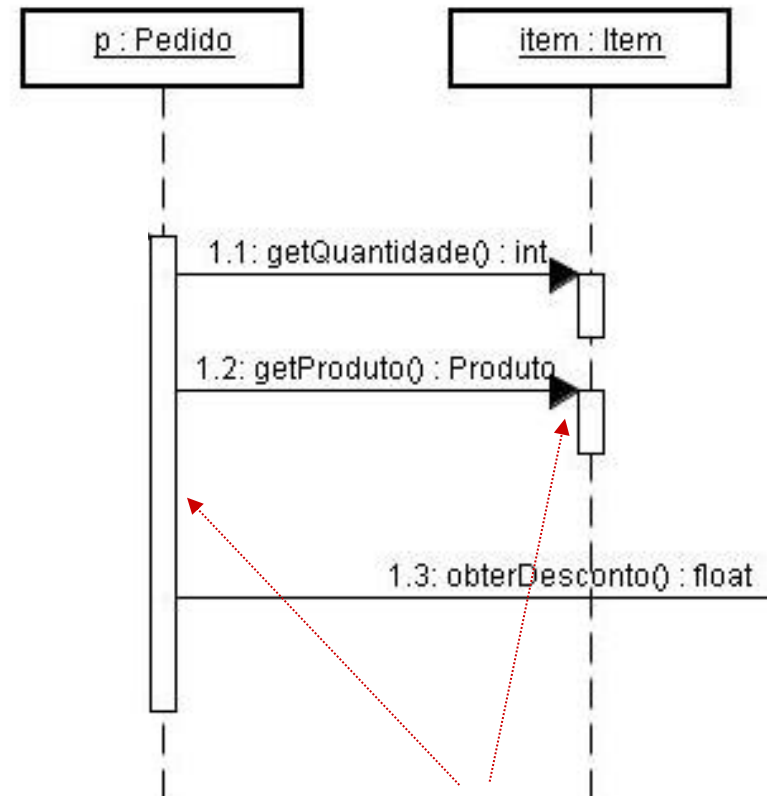
Destruição do objeto

- Objetos podem ser criados e destruídos durante uma interação. No caso de eliminação durante a interação, suas linhas de vida devem ser terminadas com um **X**.



Ocorrência de Execução (Especificação da Execução)

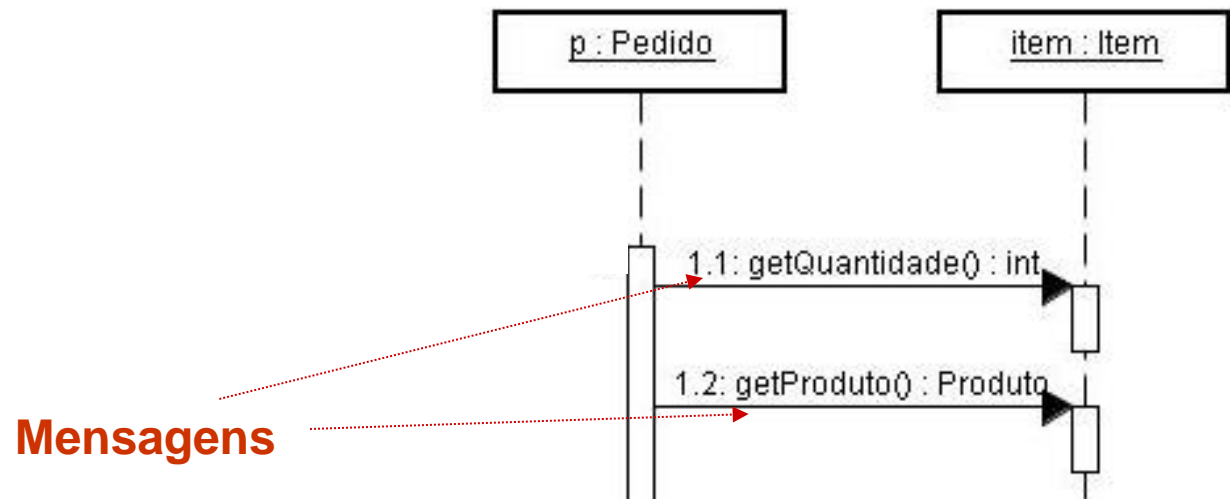
- Demonstra o período de tempo no qual o objeto executa uma ação e a relação de controle entre a ativação e o responsável pela sua invocação.
- Uma ocorrência de execução é ilustrada através de um retângulo cujo topo é alinhado com o início da ação e cuja base representa o seu término.
- Observação: já chamado de **foco de controle**.



Ocorrência de Execução

Mensagens

- Objetos colaboram enviando mensagens uns para os outros.
- Quando um objeto envia uma mensagem a outro objeto, ele está solicitando a execução de um método no outro objeto.
- A seta aponta sempre para onde a operação será executada.

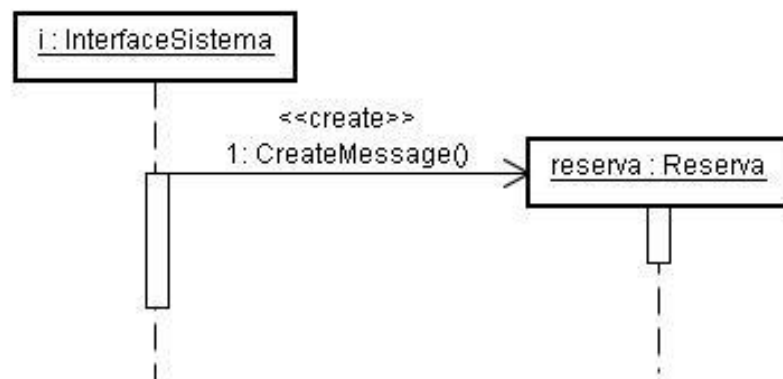


Mensagens Síncronas e Assíncronas

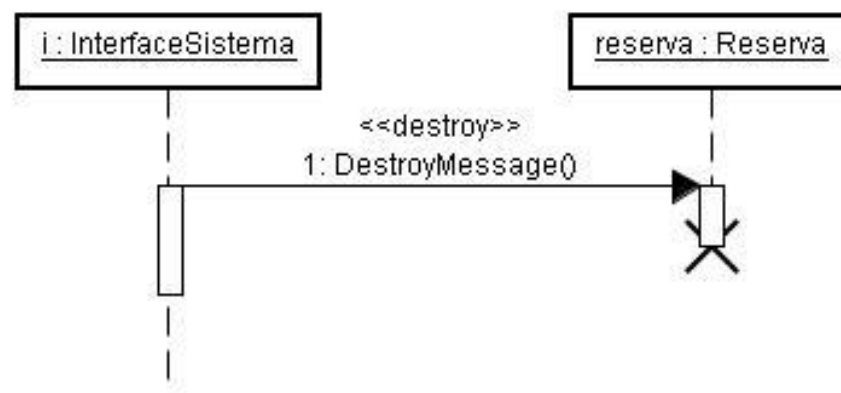
- **Mensagem síncrona:**
 - Indica que o objeto que enviou a mensagem irá aguardar a conclusão do processamento da mensagem (entendida como um sinal de sincronismo) feito pelo objeto destino, para só então prosseguir seu fluxo de execução.
 - Notação: —————▶
- **Mensagem assíncrona**
 - O objeto de origem envia a mensagem e prossegue seu processamento independentemente do tratamento da mensagem feita no objeto destino.
 - Notação: —————>

Tipos Especiais de Mensagens

Mensagem de criação



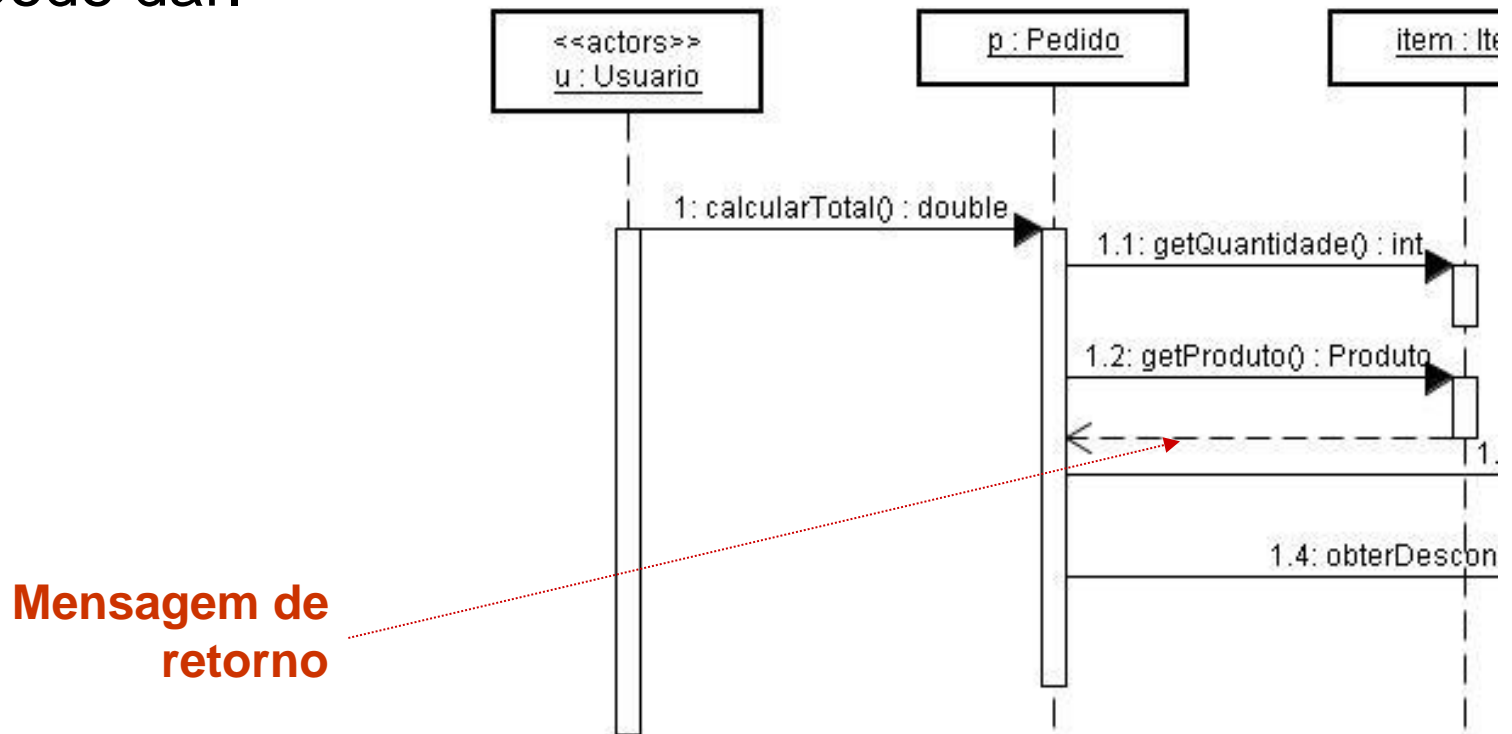
Mensagem de destruição



Tipos Especiais de Mensagens

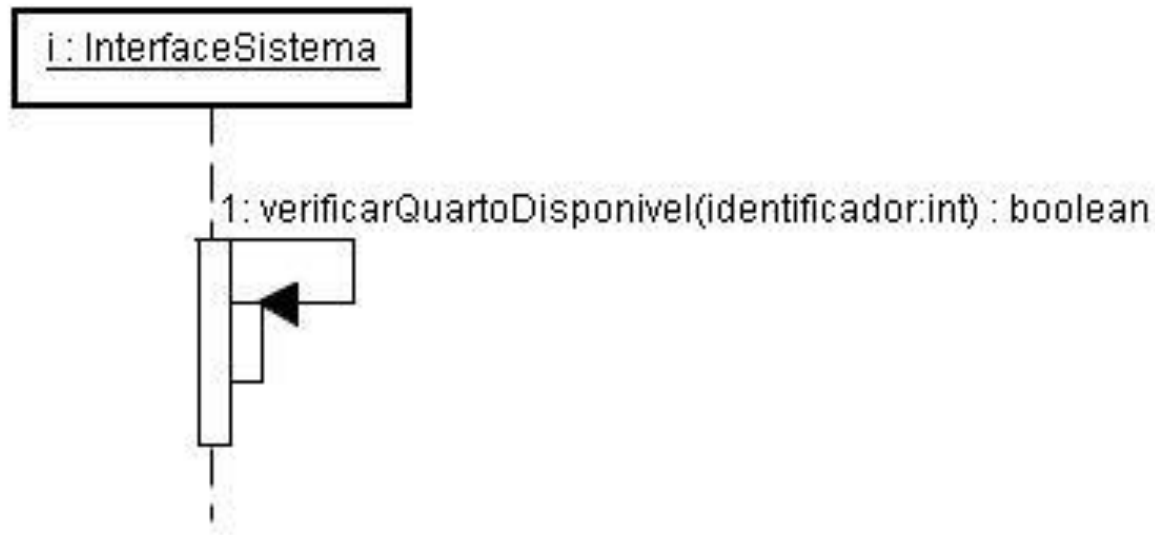
Mensagem de Retorno

- A seta tracejada indica o retorno que uma mensagem pode dar.



Auto delegação

- Uma mensagem enviada de um objeto para ele próprio chama-se auto delegação.
- Mensagens de auto delegação podem ser síncronas ou assíncronas.



Molduras *(Combined Fragments)*

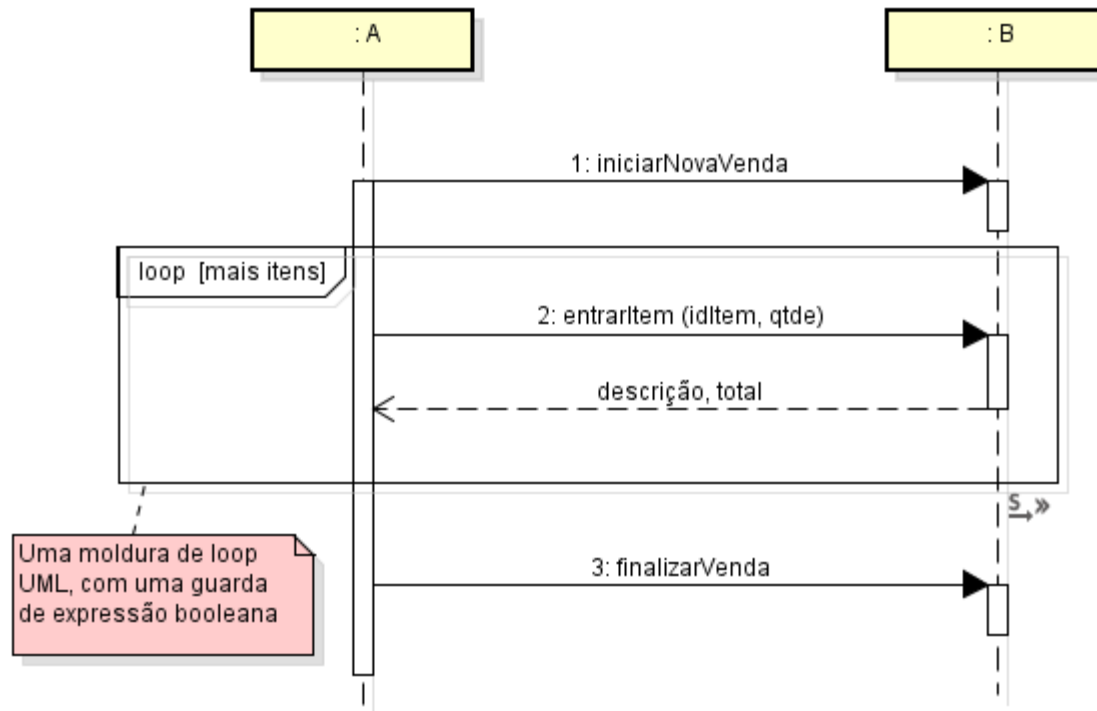
- São regiões ou fragmentos dos diagramas de sequência que apresentam um **operador** ou rótulo e **uma condição de guarda** (cláusula condicional).

| Operador da Moldura | Significado |
|---------------------|---|
| alt | Fragmento alternativo para lógica condicional de exclusão mútua, expresso em guardas. |
| loop | Fragmento de loop, enquanto a guarda é verdadeira. Pode-se também escrever <i>loop(n)</i> para indicar a iteração <i>n</i> vezes. Existe discussão de que a especificação será aperfeiçoada para definir um loop FOR. |
| opt | Fragmento opcional executado se a guarda é verdadeira. |
| par | Fragmentos paralelos executados em paralelo. |
| região | Região crítica dentro da qual apenas uma linha (thread) de execução pode ser executada. |

Fonte: livro “Utilizando UML e Padrões”, de Craig Larman (pág. 251).

Molduras (Combined Fragments)

- Exemplo de uso de moldura do tipo **loop**:



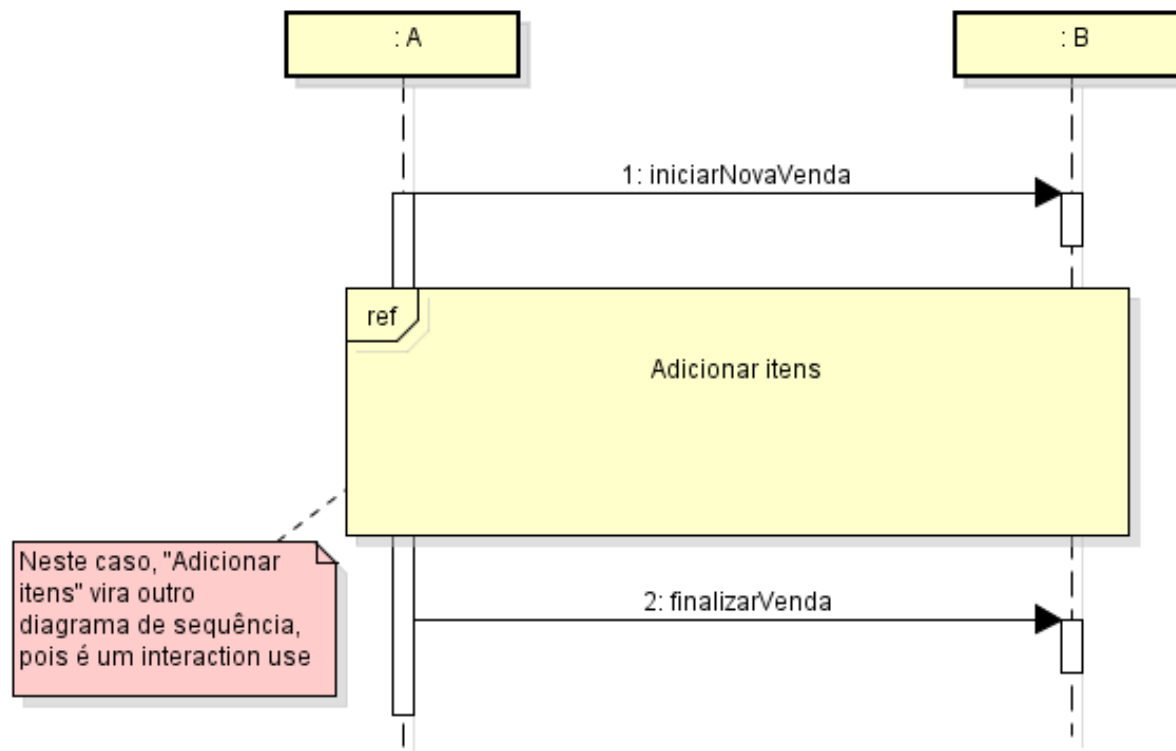
Uso de Interação *(interaction use)*

- Permite usar (ou chamar) outra interação.
 - Similar às sub-atividades do diagrama de atividades (apenas muda o que é encapsulado).
- Útil em diagramas grandes ou complexos.
- Pode ser utilizado para indicar relacionamentos de dependência do tipo inclusão ou extensão entre casos de uso.

Fonte: <https://www.uml-diagrams.org/sequence-diagrams.html#interaction-use>

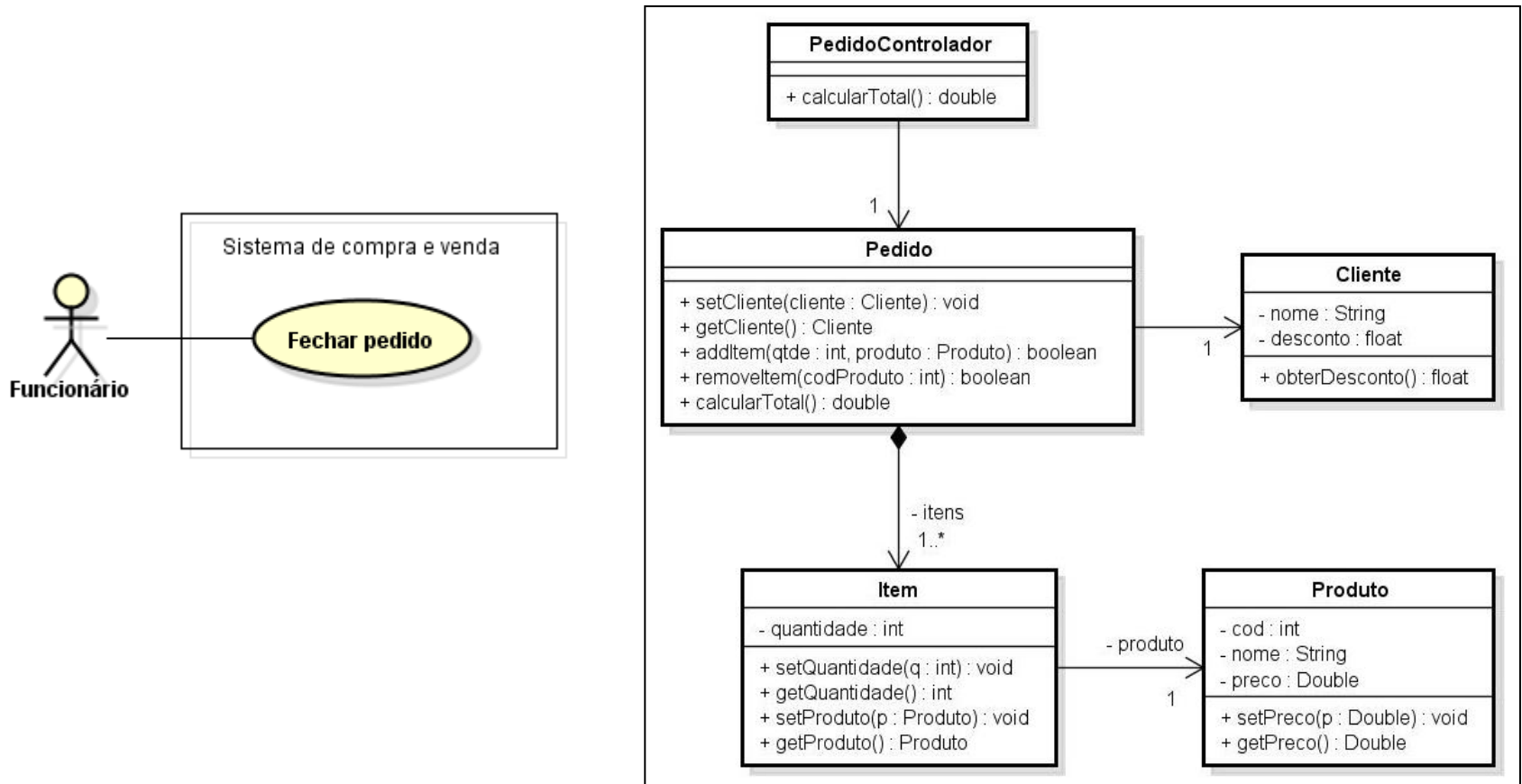
Uso de Interação *(interaction use)*

- Exemplo de uso:



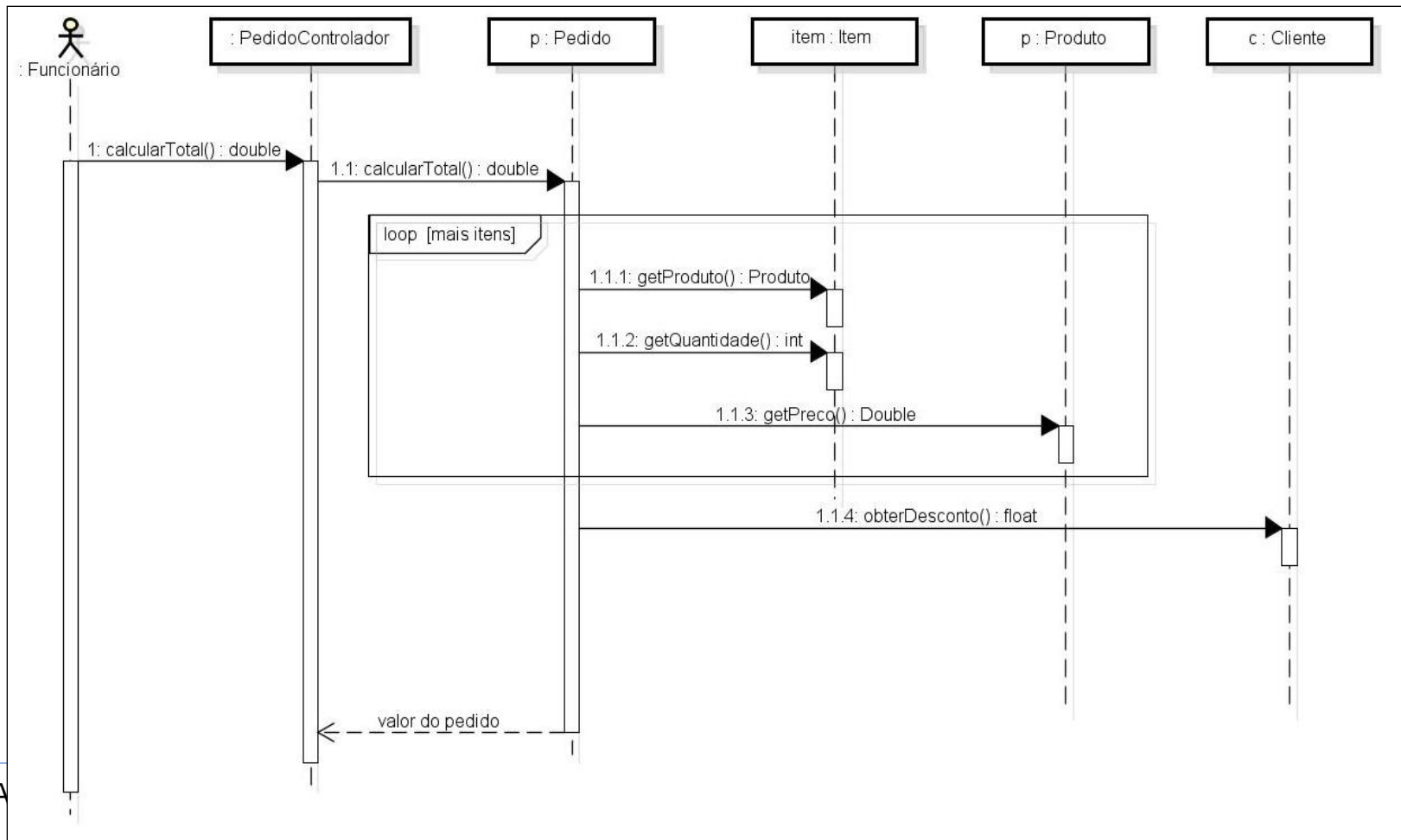
Exemplo – Sistema de Compra e Venda

- Fragmento de diagramas de um sistema de compra e venda (considerando o caso de uso “Fechar pedido”)

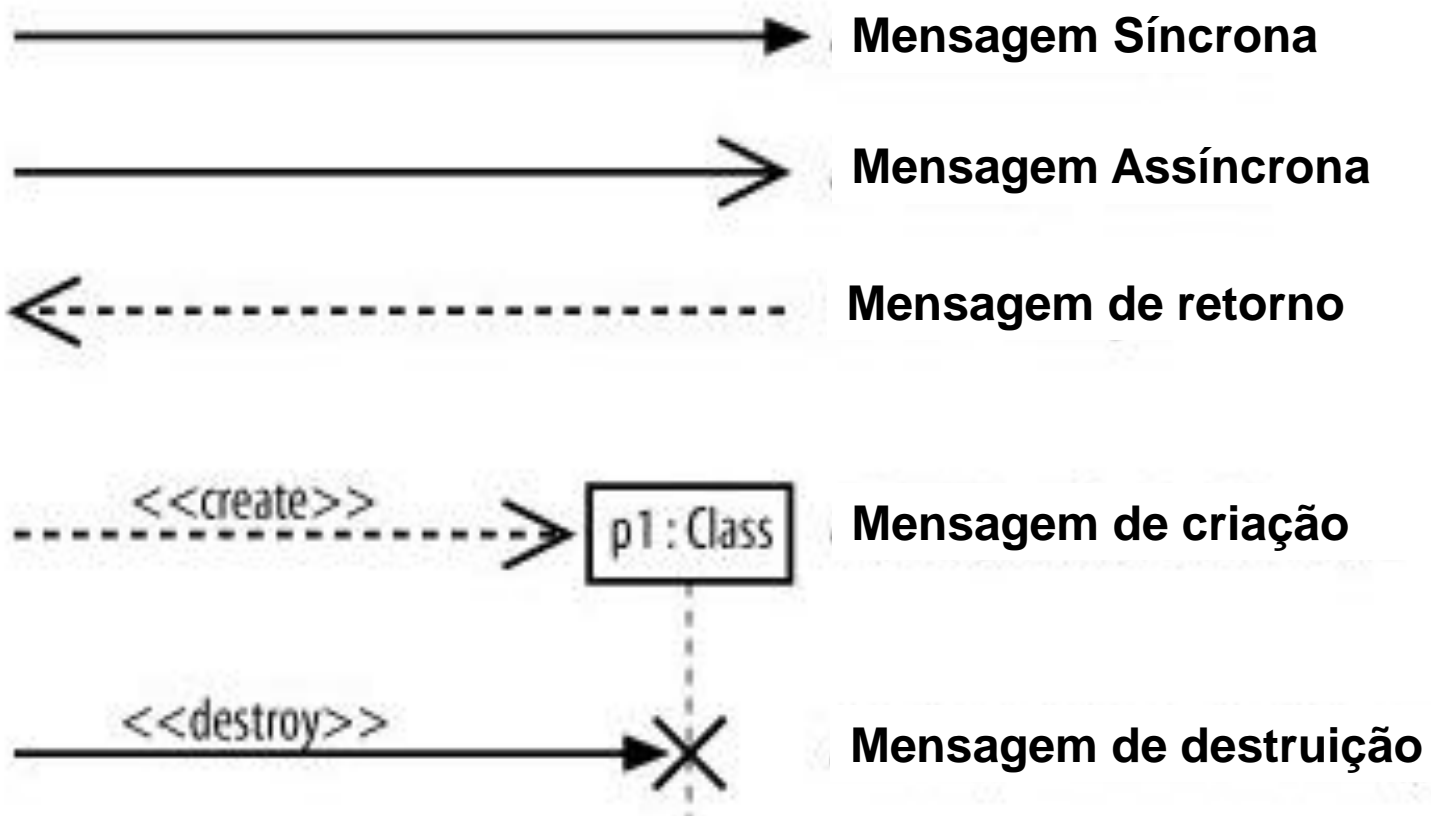


Exemplo – Sistema de Compra e Venda

- Diagrama de Sequência para o caso de uso “Fechar pedido”.



Resumo da notação das mensagens



Técnicas de Modelagem

- Definir o contexto a ser modelado (sistema, subsistema ou caso de uso).
- Definir os objetos participantes:
 - Nomear os objetos (e/ou identificar as classes correspondentes).
 - Distribuir os objetos da esquerda para a direita (o mais à esquerda é o objeto ou ator iniciador).
- Definir linhas de vida:
 - Objetos persistentes são alinhados no topo e a linha é contínua.
 - Objetos temporários são criados e excluídos durante a execução.
- Mostrar a troca de mensagens entre objetos:
 - A passagem de tempo é de cima para baixo.

Exercício Resolvido

- **Sistema de gestão de ocupação de quartos de um hotel:** para acessar o sistema, todo funcionário deverá se autenticar com login (que será um e-mail) e senha próprios.
- Documentação requerida:
 - Modelo de casos de uso (diagrama + especificação textual);
 - Diagrama de classes de projeto;
 - Diagrama de sequência do caso de uso identificado.

Tarefa 1

Considere a seguinte funcionalidade de um **sistema de gestão de ocupação de quartos de um hotel**:

- “O sistema deve permitir que um funcionário adicione reserva para um cliente em determinado período”.

Com base na funcionalidade descrita:

- Atualize o diagrama de casos de uso;
- Faça a especificação textual do novo caso de uso identificado;
- Atualize o diagrama de classes;
- Faça um novo diagrama de sequência para o novo caso de uso identificado.