

Diagrama de Sequência

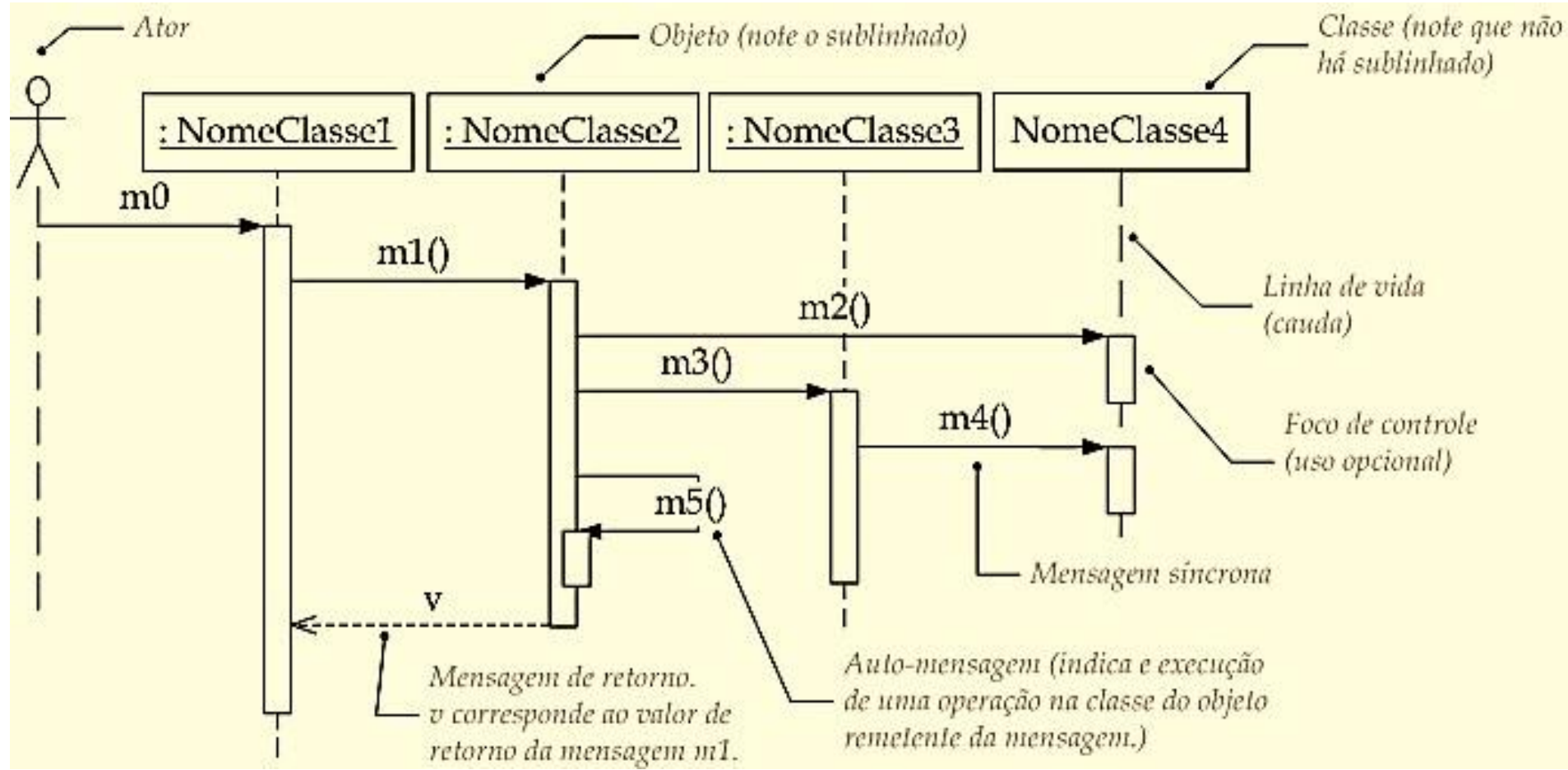


Introdução

- ❑ Procura determinar a sequência de eventos que ocorrem em um determinado processo.
- ❑ Identifica os métodos que devem ser disparados entre os atores e os objetos envolvidos.
- ❑ Baseia-se no Diagrama de Caso de Uso e, normalmente, há um diagrama de sequência para cada caso de uso.
- ❑ Também depende do Diagrama de Classe, uma vez que as classes dos objetos que aparecem no diagrama estão descritas no de Classe.
- ❑ Uma boa ferramenta para validar o Diagrama de Classe.

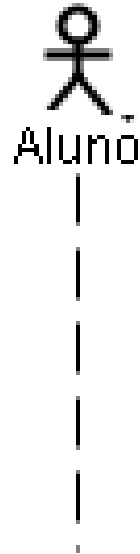
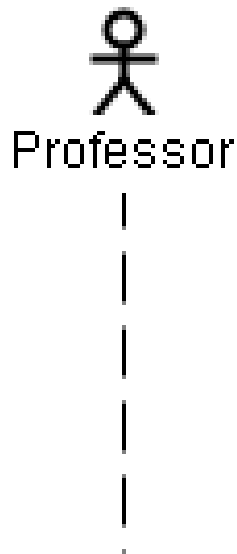


Introdução - modelo



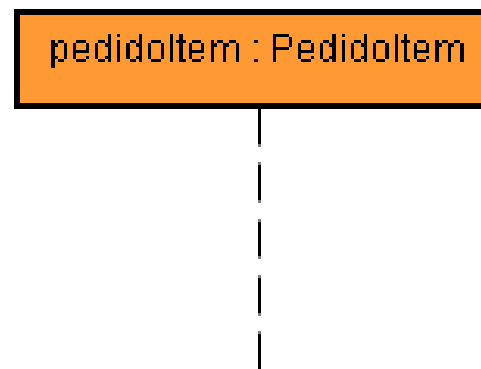
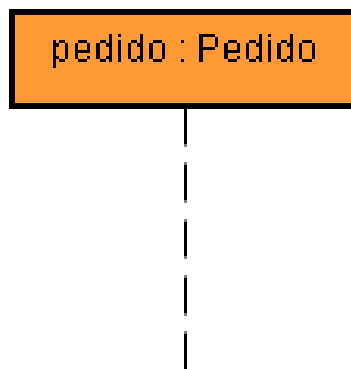
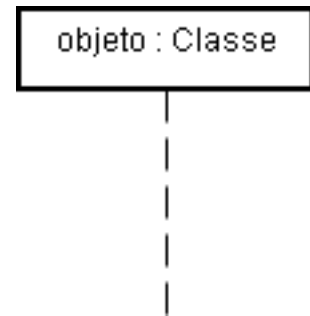
Atores

- ❑ São os mesmos do Diagrama de Caso de Uso e possuem a mesma representação, mas diferenciam-se por apresentarem uma “linha de vida”.
- ❑ Geralmente, um Diagrama de Sequência é iniciado por um evento externo, gerado por um ator.
- ❑ Exemplos:



Objetos

- ❑ Representam as instâncias das classes envolvidas no processo ilustrado pelo diagrama de sequência.
- ❑ Assim como os atores, os objetos possuem uma “linha de vida” vertical tracejada.
- ❑ Um objeto pode existir desde o início do processo ou ser criado durante a sua execução.
- ❑ Exemplos:



Linha de vida

- ❑ Representa o tempo em que um objeto existe durante um processo.
- ❑ Representada por uma linha vertical fina tracejada partindo do objeto.
- ❑ Quando o objeto é destruído, a linha de vida é interrompida com um "X".

Diferença nas chamadas

- ❑ Ação de Chamada (Call Action)
- ❑ O que é: Uma chamada de método (função) em outro objeto.
- ❑ Símbolo: Linha cheia com ponta de seta sólida.
- ❑ Semântica: Um objeto envia uma mensagem para outro pedindo que execute uma operação.
- ❑ Exemplo: `cliente.pagar(fatura)`

Diferença nas chamadas

- ❑ Ação de Envio (Send Action)
- ❑ O que é: Envio assíncrono de uma mensagem ou sinal.
- ❑ Símbolo: Linha cheia com ponta de seta aberta.
- ❑ Semântica: O objeto envia um sinal, mas não espera resposta imediata.
- ❑ Exemplo: Enviar um evento ou notificação, como um alarme.

Diferença nas chamadas

- ❑ Ação de Retorno (Return Action)
- ❑ O que é: Retorno de dados após uma chamada de método.
- ❑ Símbolo: Linha tracejada com seta aberta.
- ❑ Semântica: Representa o valor retornado por uma chamada anterior.
- ❑ Exemplo: `total = carrinho.calcularTotal()`

Diferença nas chamadas

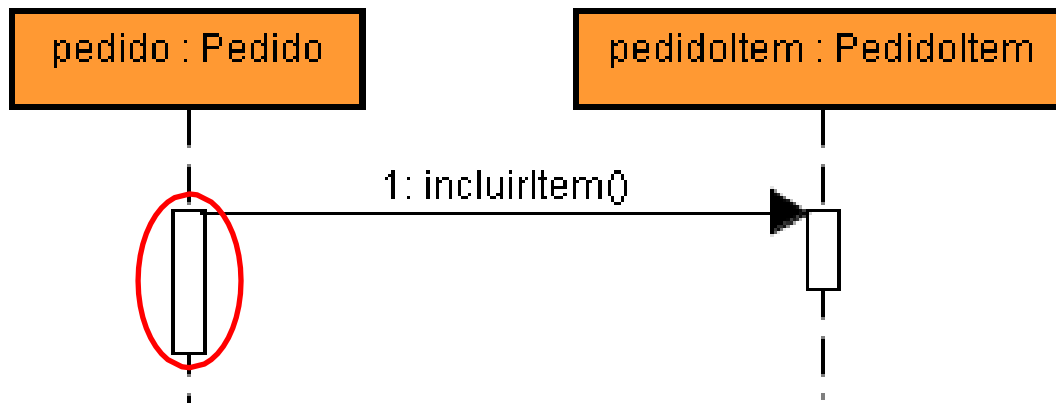
- ❑ Ação de Criação (Create Action)
- ❑ O que é: Um objeto cria outro durante a interação.
- ❑ Símbolo: Linha cheia com seta para a borda superior da lifeline do novo objeto (indica início da existência).
- ❑ Semântica: Um novo objeto é instanciado.
- ❑ Exemplo: `novoPedido = new Pedido()`

Diferença nas chamadas

- ❑ Ação de Destruição (Destroy Action)
- ❑ O que é: Um objeto é explicitamente destruído.
- ❑ Símbolo: Um X ao final da lifeline do objeto.
- ❑ Semântica: Marca o fim da existência de um objeto.
- ❑ Exemplo: delete(pedido)

Foco de Controle (ou Ativação)

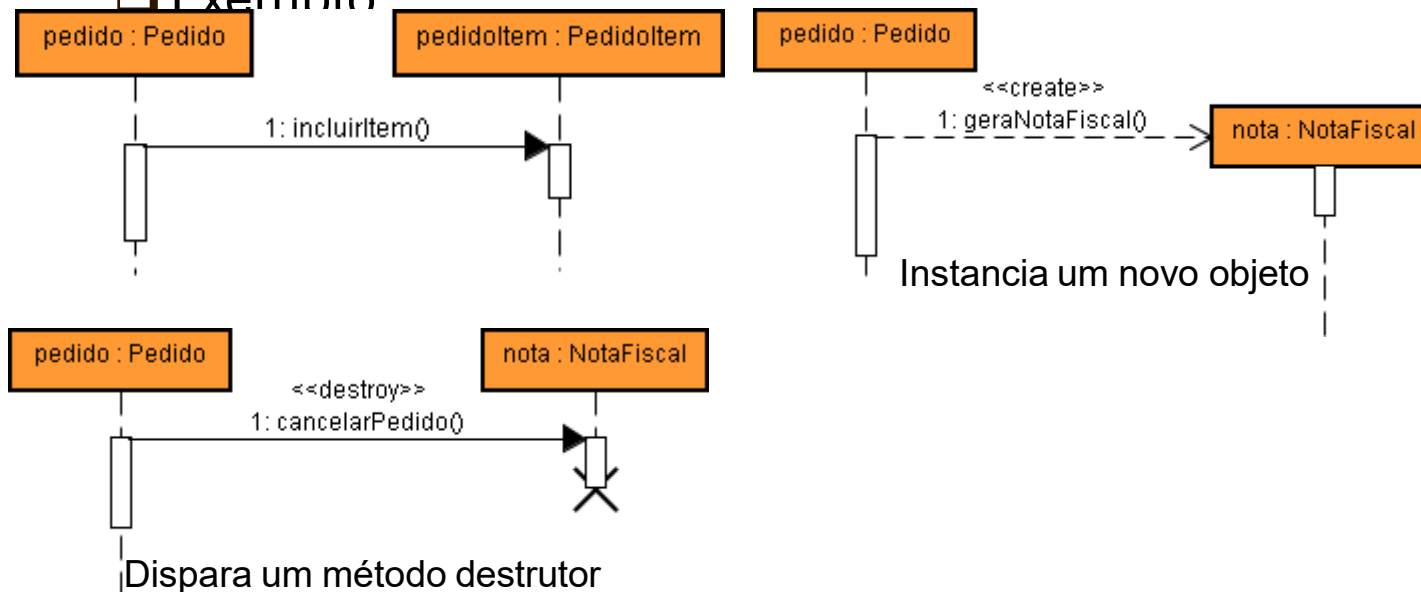
- ❑ Indica o período que o objeto está participando ativamente do processo.
- ❑ Os focos de controle são representados dentro da linha de vida, porém por uma linha mais grossa.
- ❑ Exemplo:



Mensagens ou Estímulos

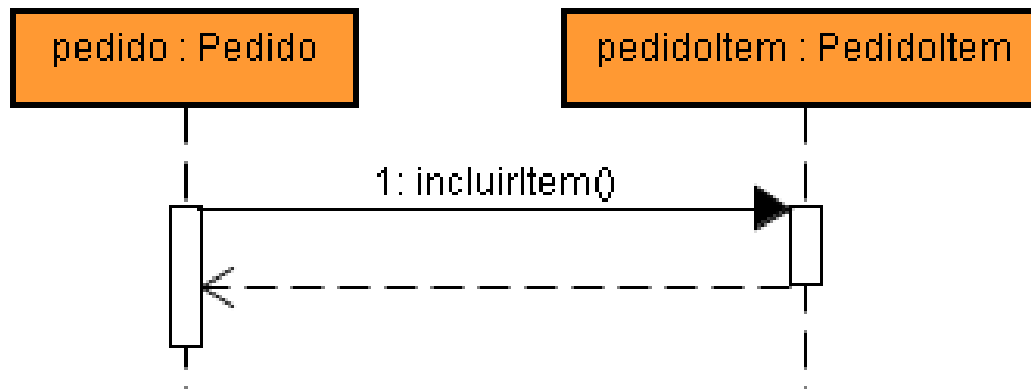
- ❑ Demonstram a ocorrência de eventos que, comumente, fazem chamadas a um método de algum objeto envolvido no processo.
- ❑ Representadas por uma seta e indica qual componente enviou a mensagem qual recebeu.

Exemplo:



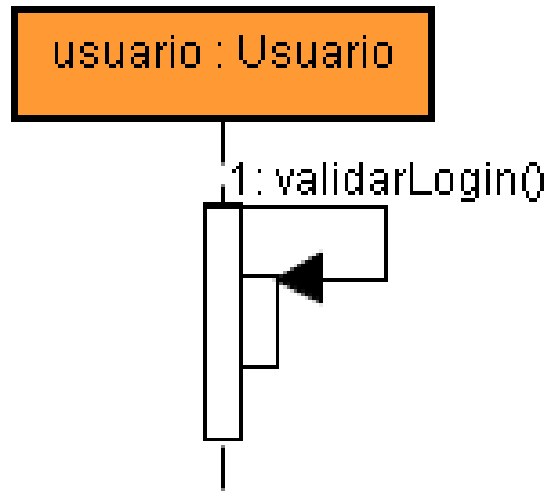
Mensagens de retorno

- ❑ Mensagem de resposta dada a um objeto que a chamou.
- ❑ Podem retornar informações específicas do método **chamado** ou, simplesmente, um "OK" ou "NOK", por exemplo.
- ❑ Representadas por uma seta tracejada contendo uma seta fina que aponta para o objeto ou ator que recebe o resultado do método chamado.
- ❑ Exemplo:

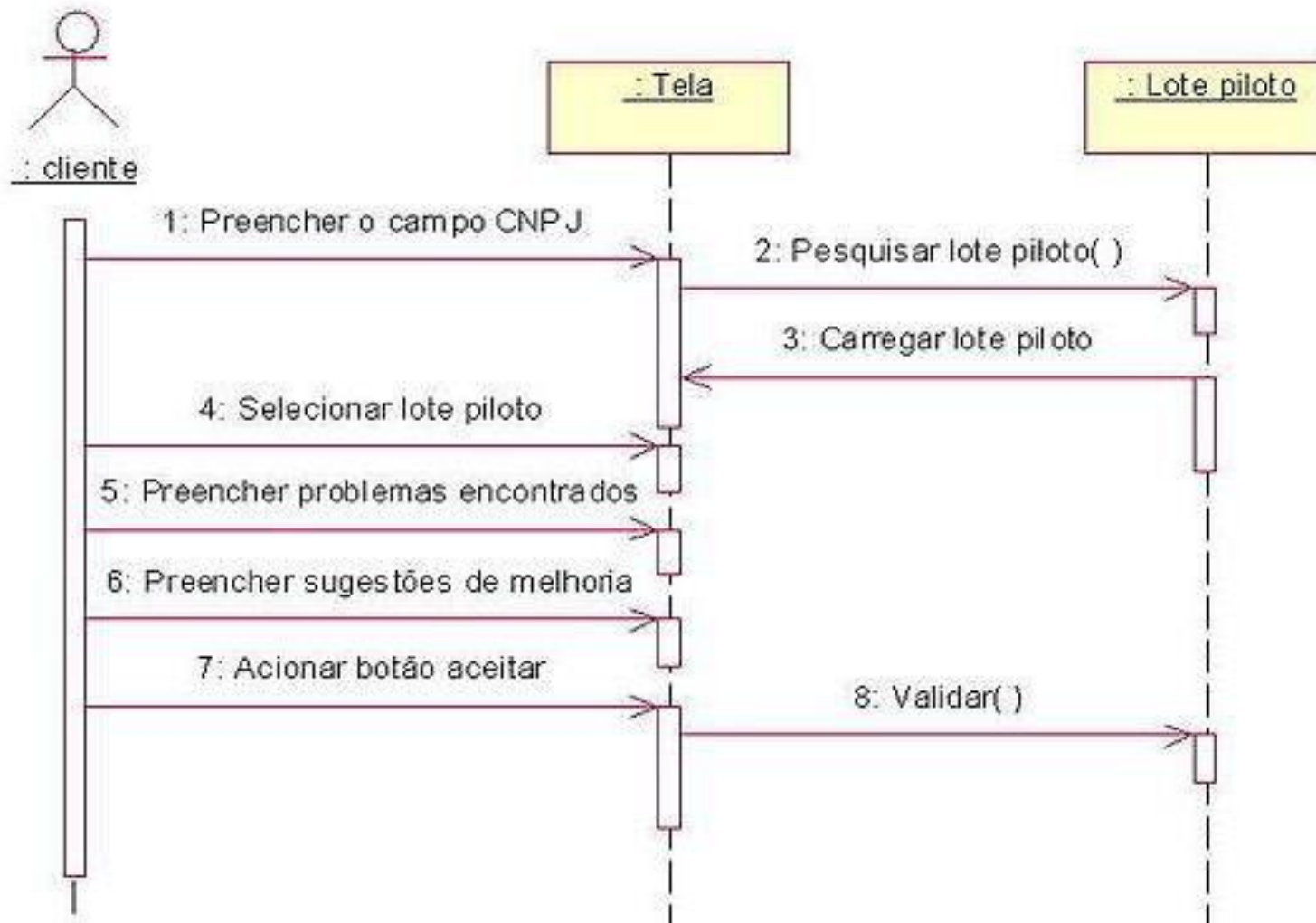


Auto-chamadas (ou Auto-delegações)

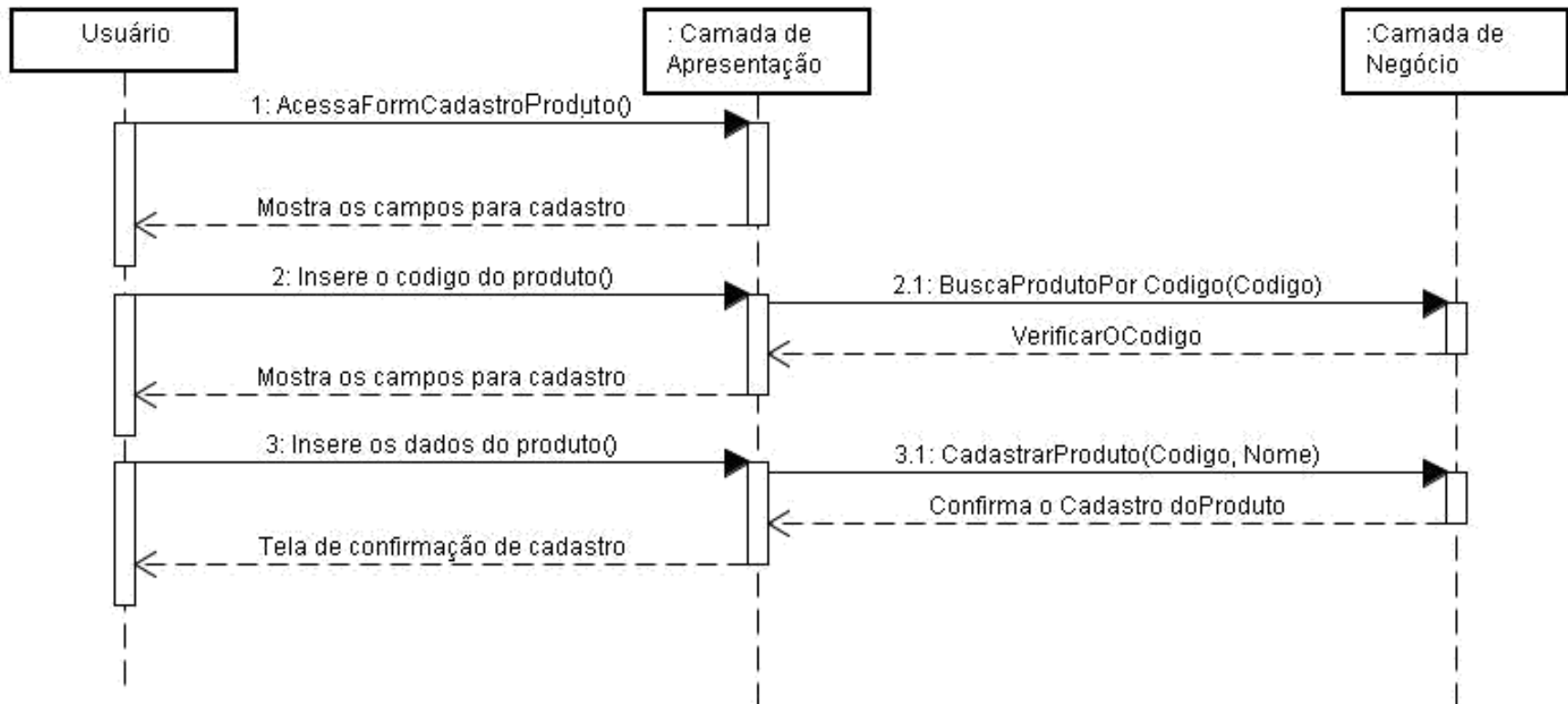
- ❑ Mensagens que partem da linha de vida de um objeto e atingem a linha de vida do próprio objeto.
- ❑ Exemplo:



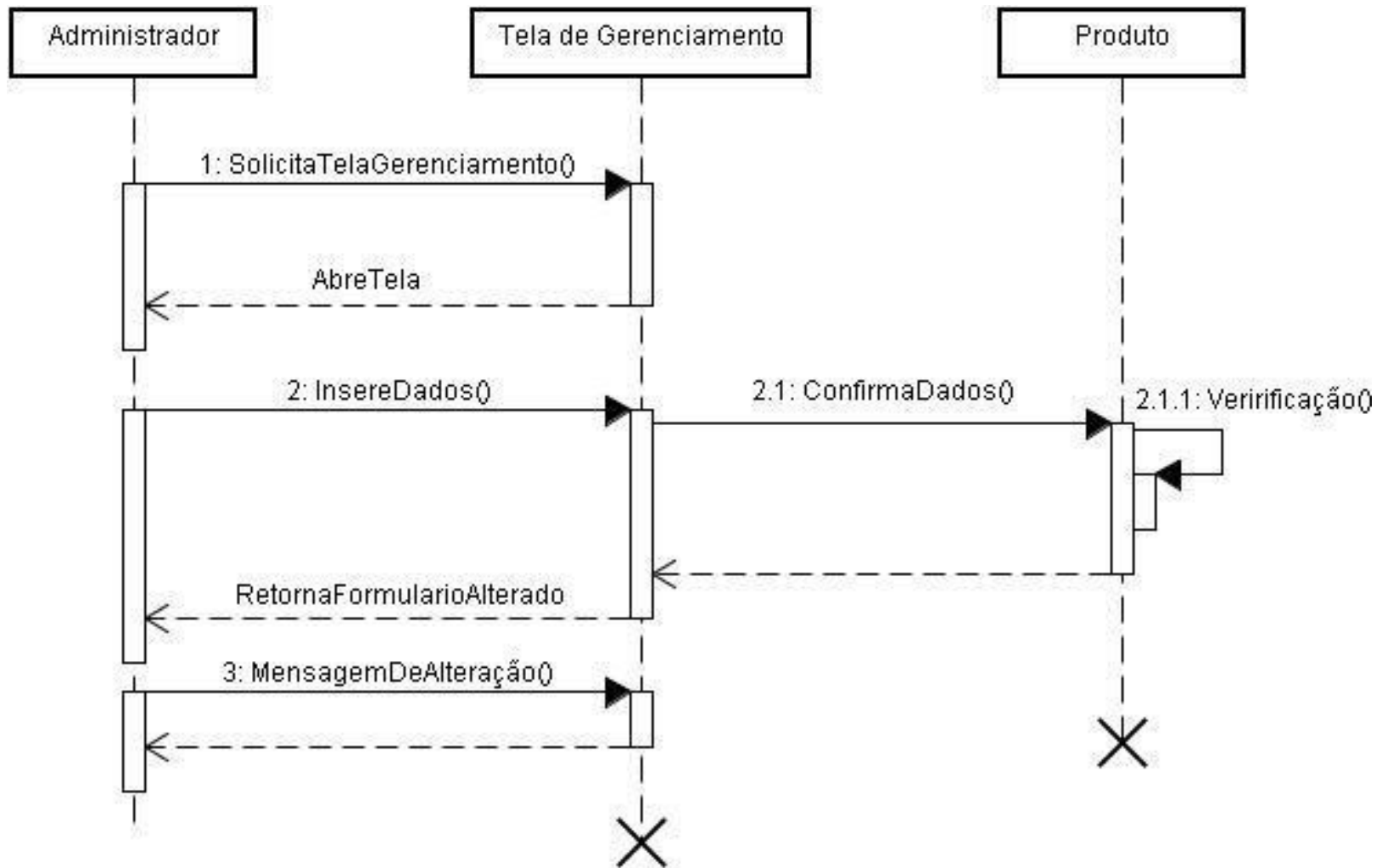
Exemplos



Exemplos



Exemplos



Exemplos

