

CC6522

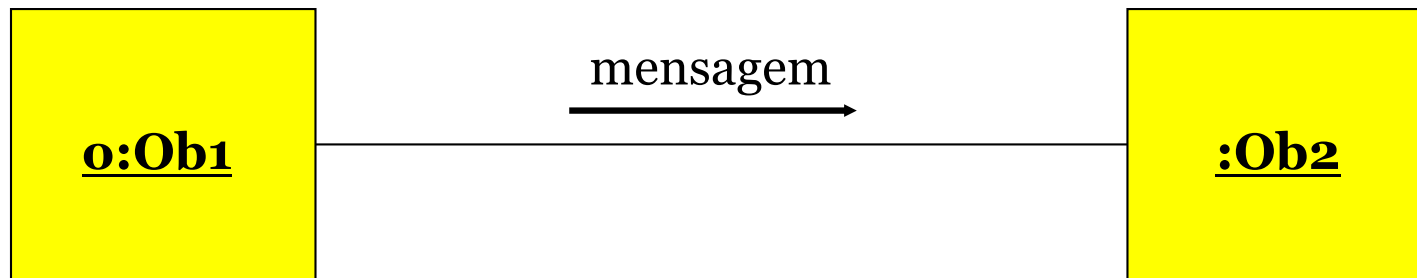
Modelagem de Software Orientado a Objetos

Diagramas de Sequência

Interações

- Comportamento que
 - Envolve conjunto de mensagens trocadas entre objetos dentro de um determinado contexto
 - Objetiva atingir resultado específico
- Acontecem em função da troca de mensagens entre objetos
- Usadas para a modelagem dos aspectos dinâmicos de um sistema

Comunicação entre Objetos



Mensagem =

Ident. Objeto

Ident. Operação

Parâmetros

- **Mensagem**

Recepção de mensagem por um objeto

- Considerado instância de evento

Decorrência da passagem de uma mensagem

- Repercute ação representada por um comando executável
- Comando Executável: abstração de procedimento computacional

Diagrama de Seqüência de Sistema

- Serve para sistematizar os fluxos dos casos de uso expandidos.
- Deve-se construir pelo menos um para o fluxo principal de cada caso de uso.
- Os fluxos alternativos podem ter um diagrama de seqüência próprio ou serem embutidos no diagrama do fluxo principal, se não forem complexos.

Objetivo

- Identificar **operações** e **consultas** que o sistema implementará para realizar as transações previstas nos casos de uso.
- Um diagrama de sequência mostra uma interação, isto é, uma sequência de mensagens trocadas entre vários objetos num determinado contexto (caso de uso, operação, etc.)
- Enfatiza a comunicação e passagem de controle entre objetos ao longo do tempo
- Útil para descrever uma sequência particular de funcionamento, mas não muitas sequências alternativas e ciclos nem ações realizadas por um objeto que não envolvem comunicação com outros objetos

Tipos de transação

- Identificadas no caso de uso e transportadas para o primeiro nível do diagrama de seqüência (dos atores para a aplicação/interface):

Evento e Resposta de sistema

- Colocadas no segundo nível (da aplicação interface para o controle/domínio), são *métodos* que efetivamente realizam a funcionalidade do sistema:

Operação e Consulta de sistema

Combinações

- Tanto o evento quanto a resposta de sistema podem ter uma seqüência de operações e consultas de sistema associadas, não necessariamente uma só.

Componentes - Objetos

- Apresentados na **dimensão horizontal** do diagrama
- **Ordem** dos objetos não é considerada
Dispô-los de forma a tornar o diagrama “mais legível”
- Objetos tem nomes
obj:Classe

Ex.: joão:Dentista

:Floricultor (um objeto floricultor não identificado)

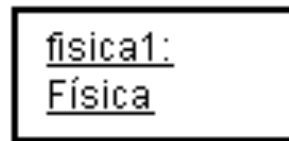
obj1: (um objeto obj1 sem classe definida)

Componentes - Objetos

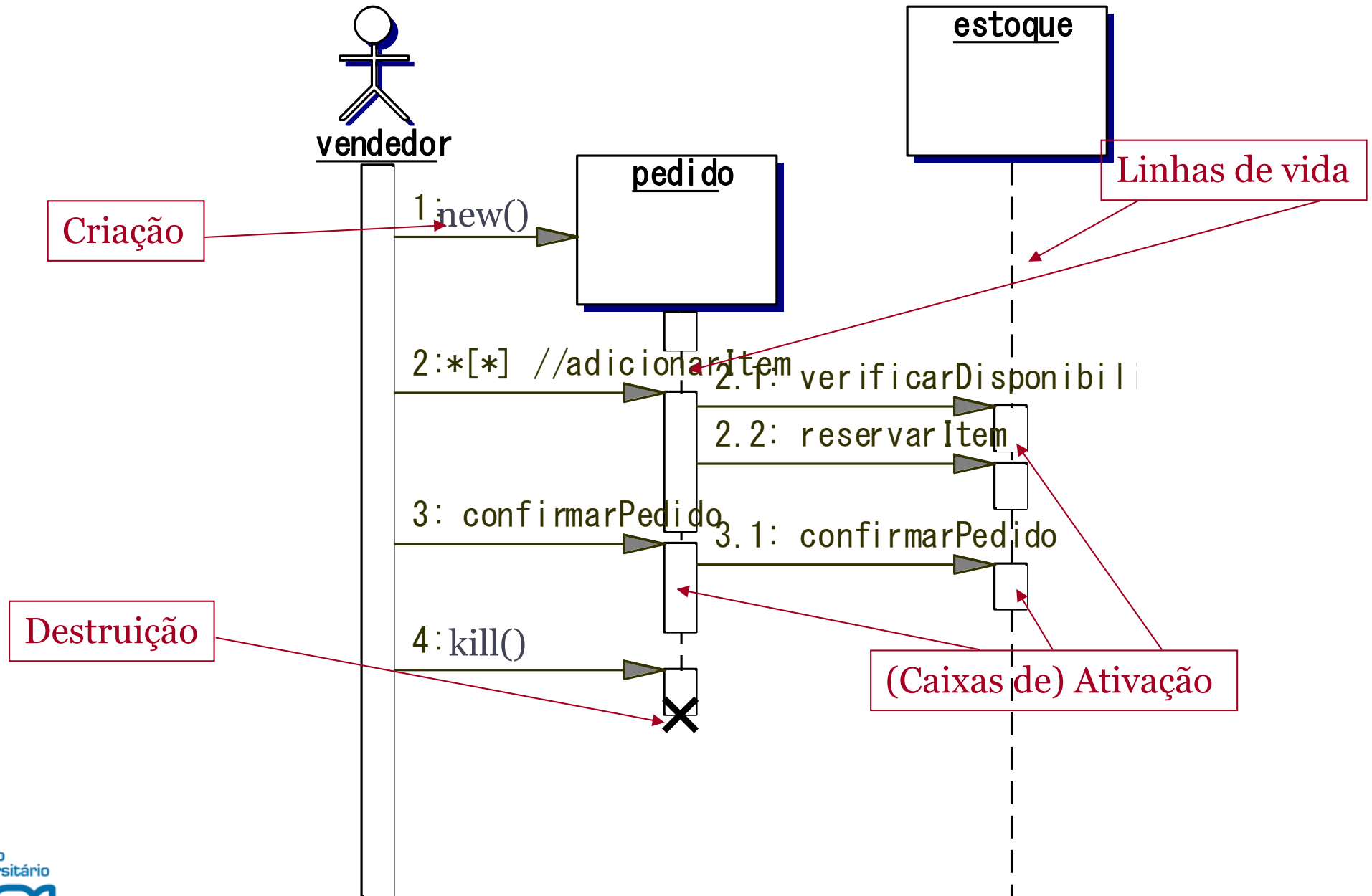
- Representam as instâncias das classes
- Retângulos contendo um texto
 - Primeira parte, em minúsculo, o nome do objeto
 - Segunda parte, em letras iniciais maiúsculas, o nome da classe
 - Informações separadas por dois pontos (:)
- Linha de vida
 - Linha vertical tracejada

Componentes - Objetos

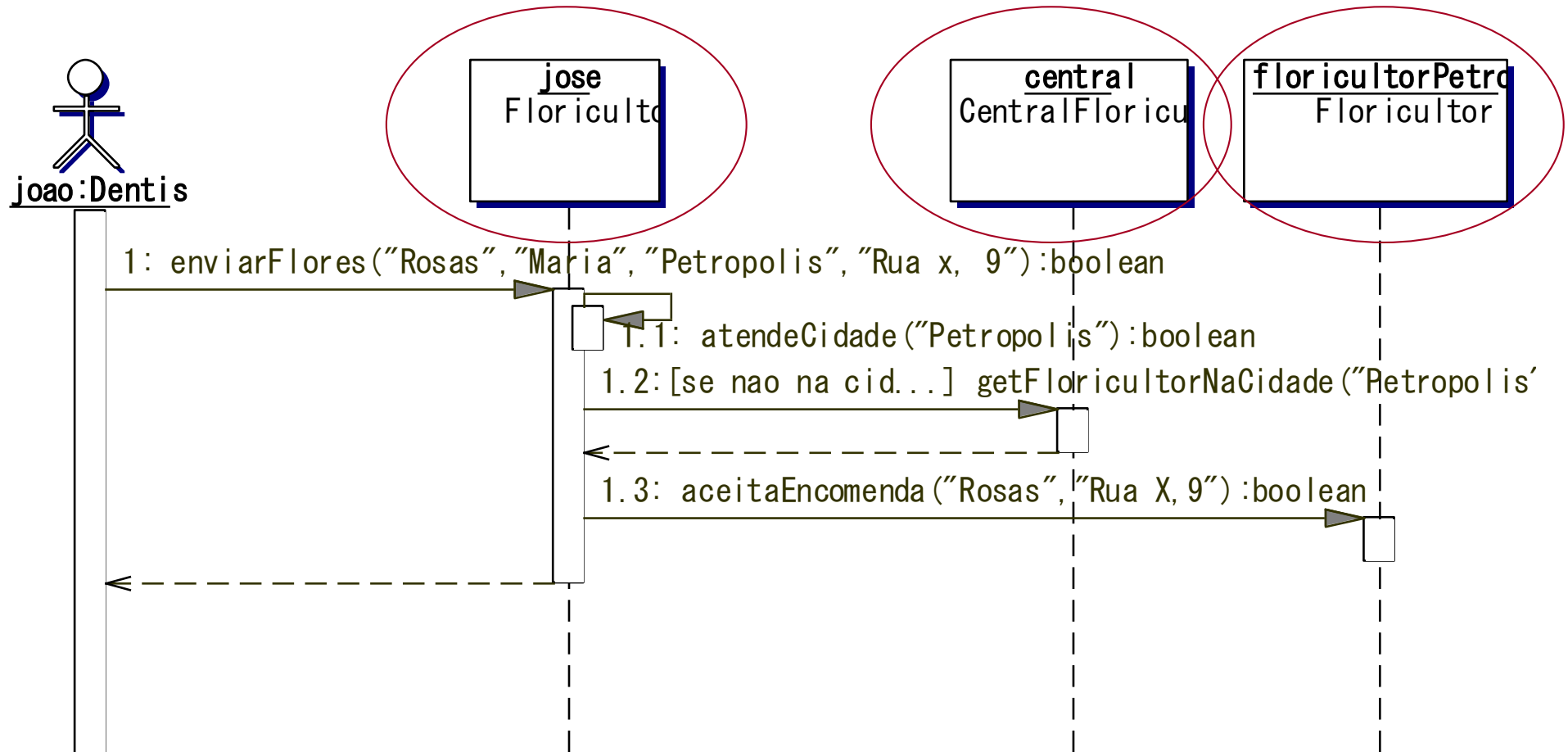
- **Exemplo**



Linhas de Vida



Componentes - Objetos



Componentes - Linhas de Vida

- **Dimensão vertical** do diagrama
- Apresentam o **tempo de vida** dos objetos
- Pode apresentar a **ativação** ou a **desativação** dos objetos

Indicam que os objetos estão executando algo

- Foco de controle

Caixas de ativação podem ser empilhadas

- Indica chamada de método do próprio objeto
- Objeto jose no slide anterior
- Podem representar a **criação** e a **destruição** de objetos

Componentes - Linhas de Vida

- Representa o tempo que um objeto existiu durante um processo
- Linhas finas verticais tracejadas

Iniciam no retângulo que representa o objeto

Interrompida por um “X” quando o objeto é destruído



Mensagens - Tipos

- Tipos de ação que uma mensagem pode representar
call

- Invoca uma operação sobre um objeto
 - Objeto pode mandar uma chamada para si próprio
Resultando na execução local de uma operação

return

- Representa o retorno de um valor para o objeto que chamou a operação
- Opcional

create

- Criação de um objeto



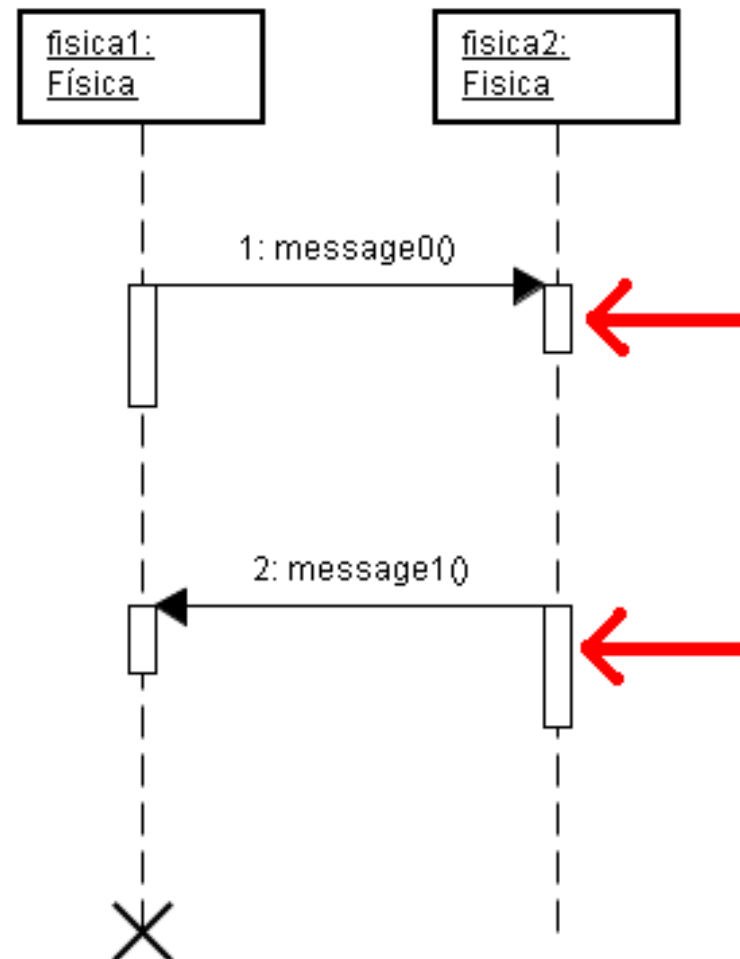
destroy

- Eliminação de um objeto

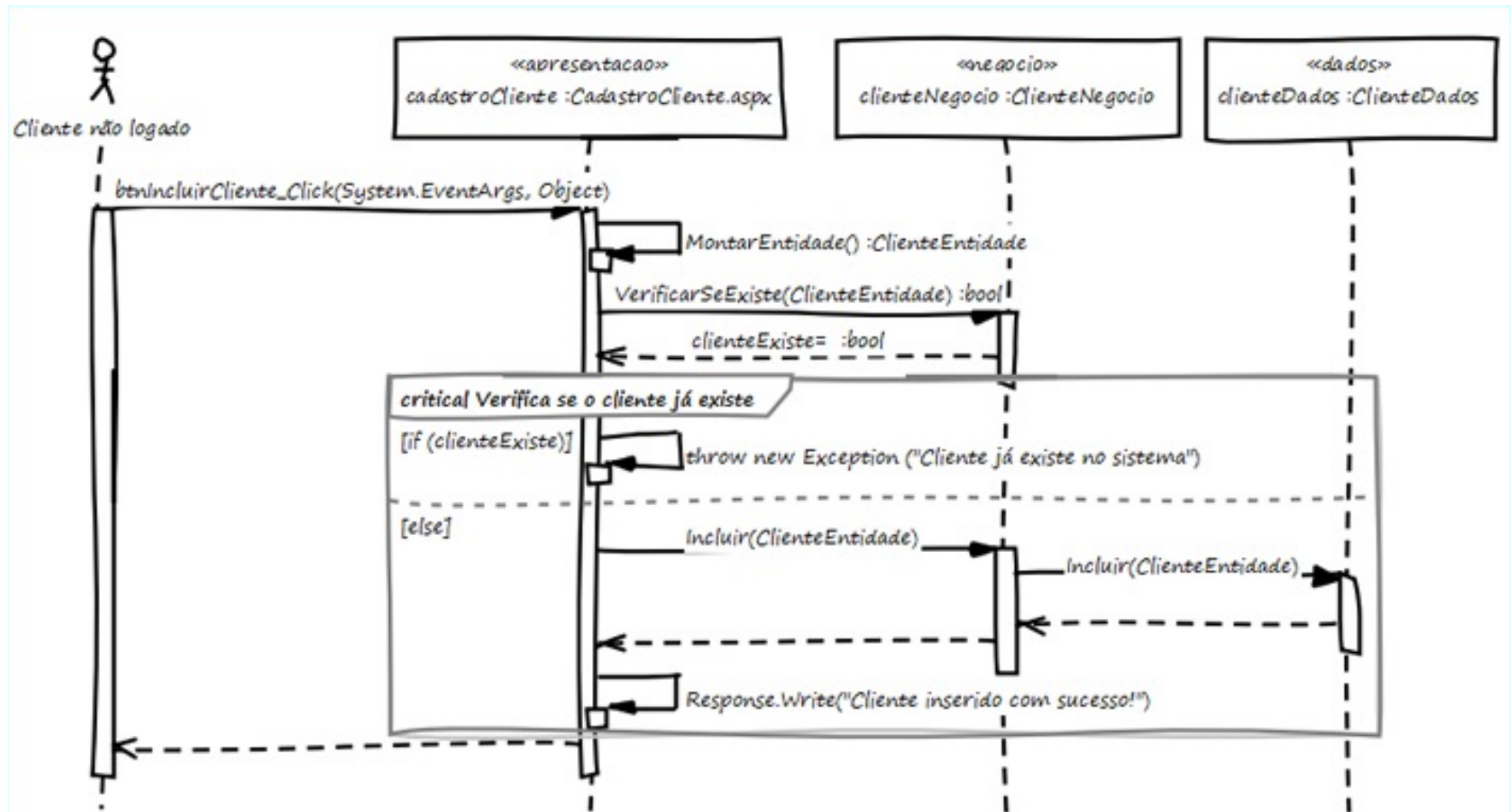


Componentes - FOCO DE CONTROLE/ATIVAÇÃO

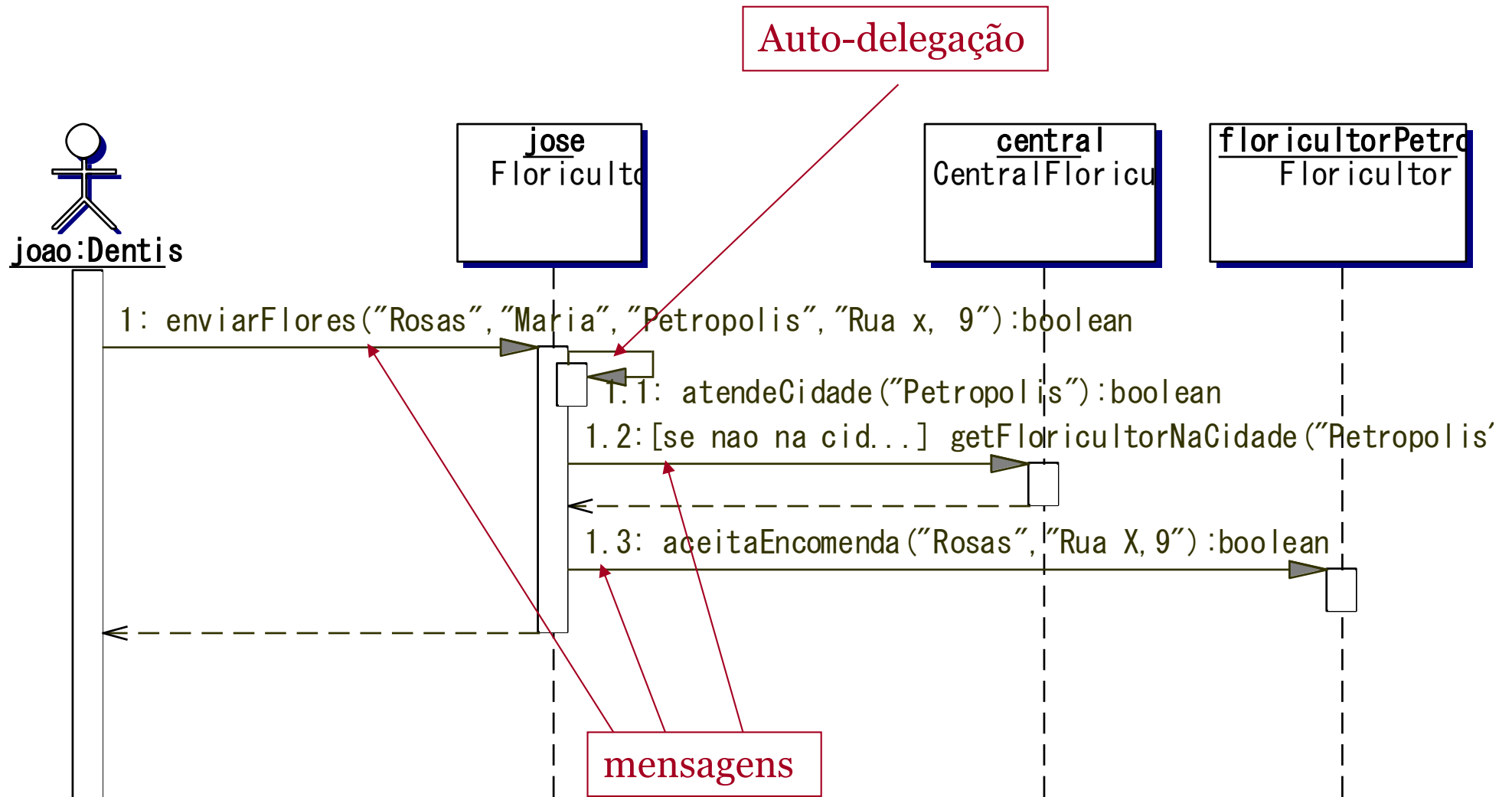
- **Exemplo**



Componentes - Mensagens



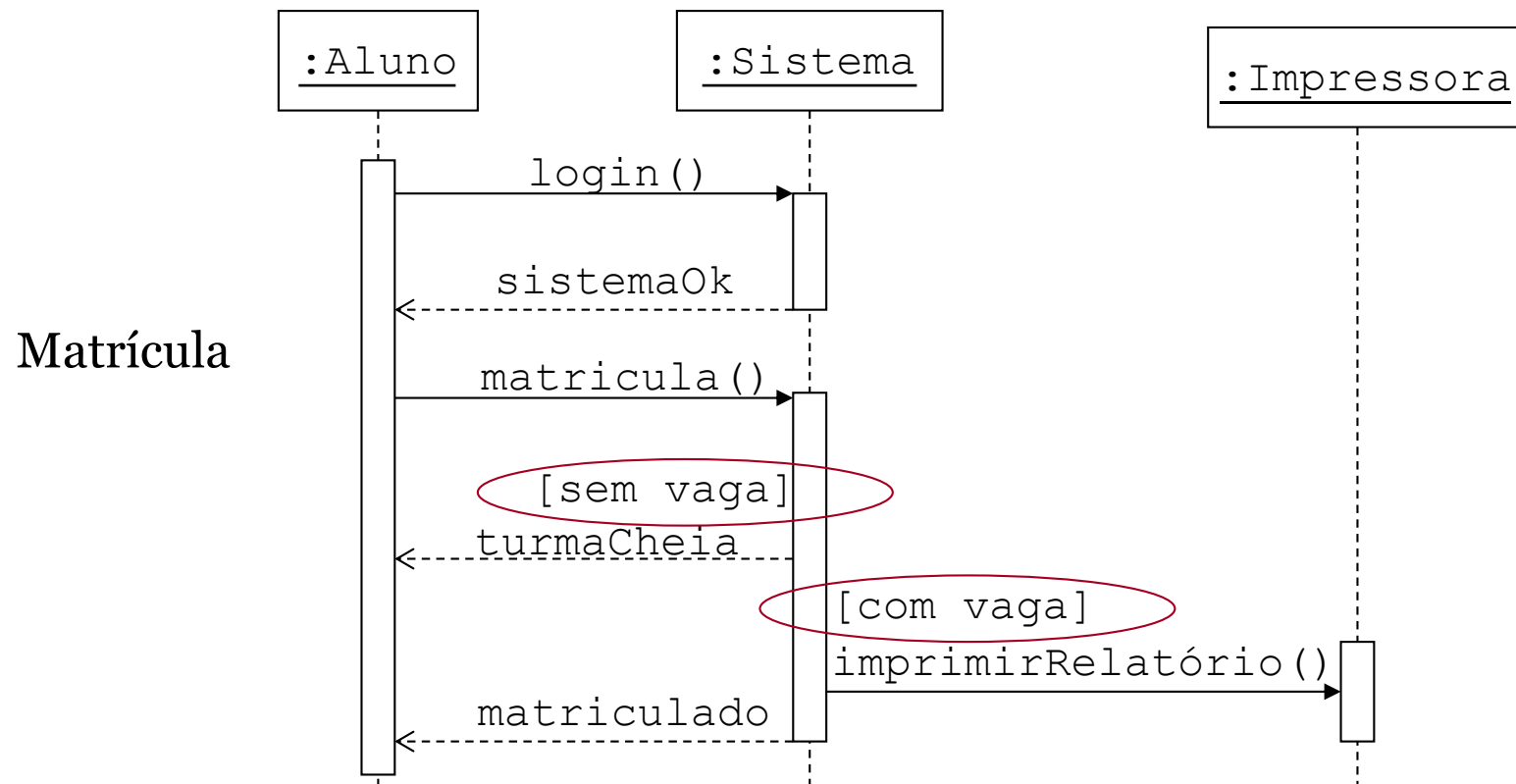
Mensagens



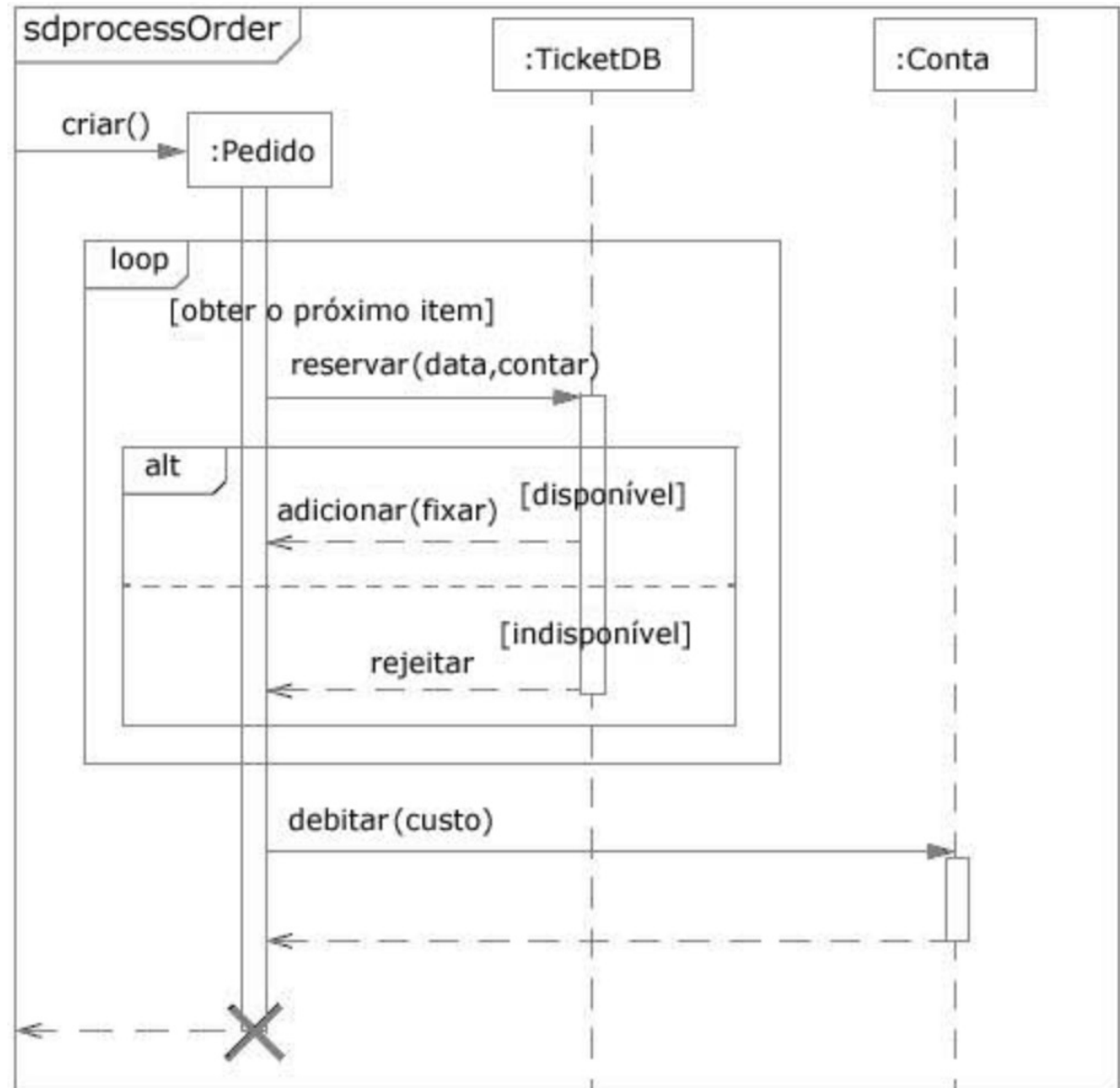
Mensagens

Condições de Guarda

- Mensagens podem apresentar condições de guarda
condições em que a mensagem é enviada
[condição de guarda]



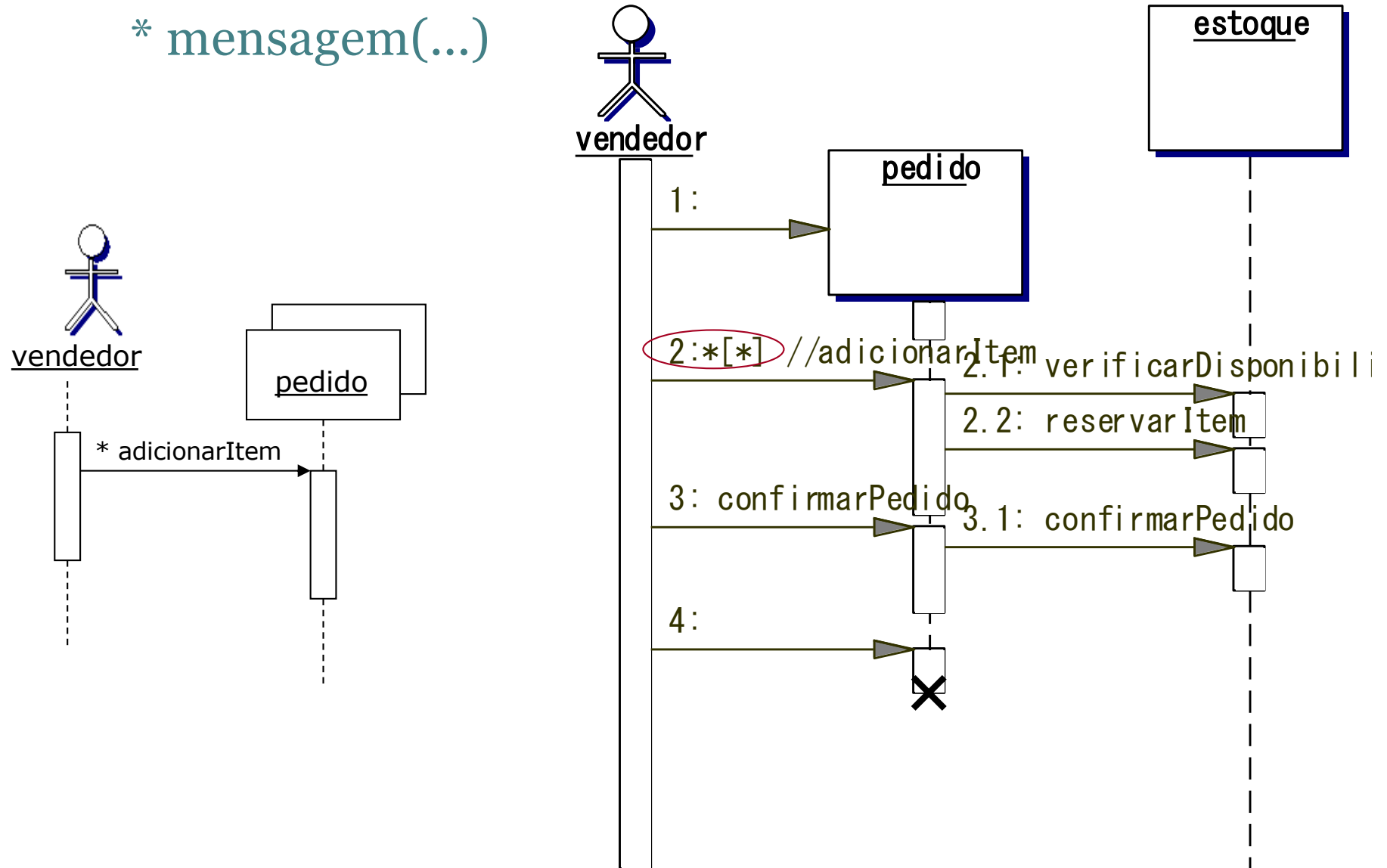
Mensagens 2.0



Mensagens - Iteração

- Uma mensagem pode ser enviada repetidas vezes

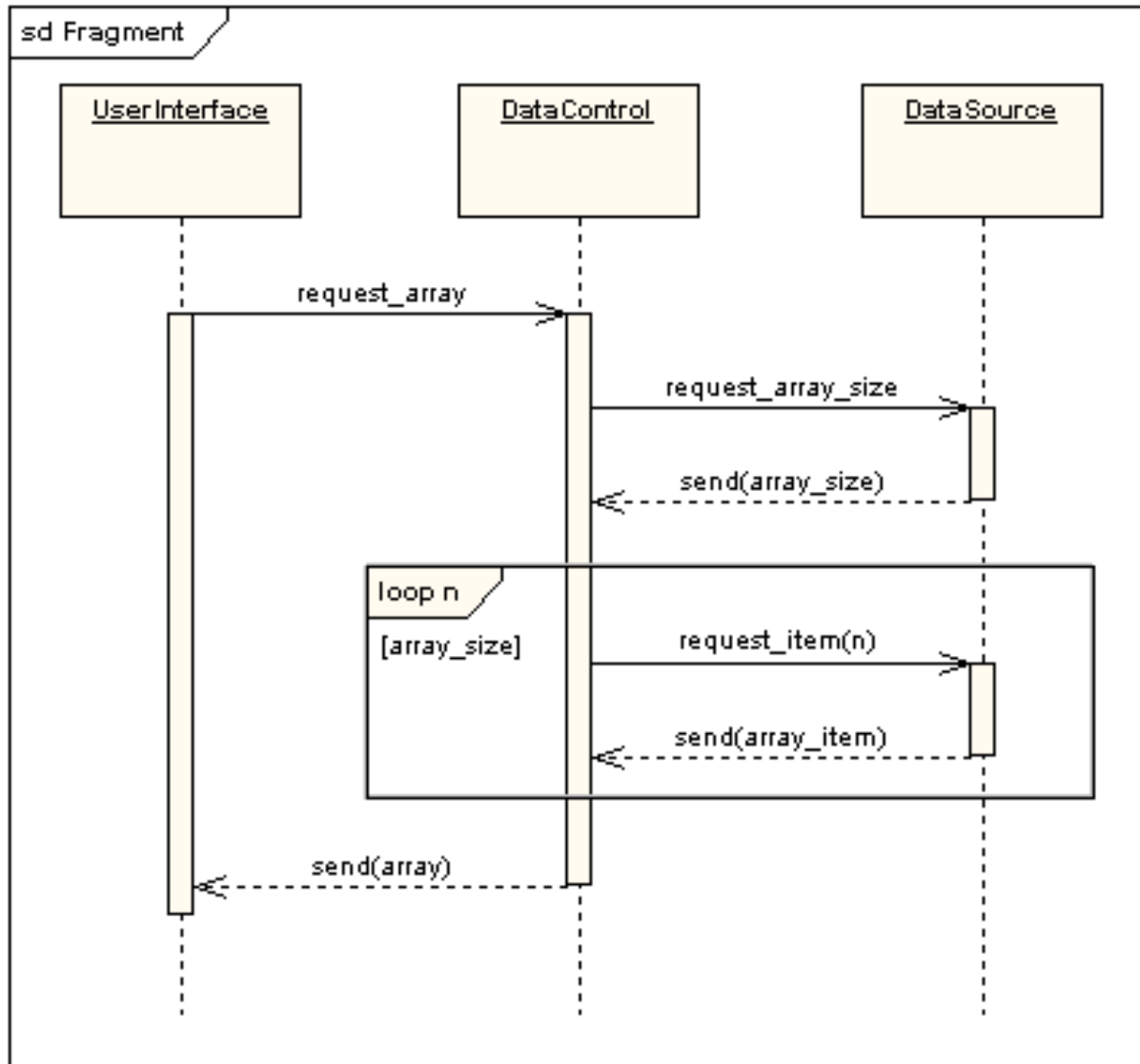
* mensagem(...)



Foco de Controle

- Período de tempo que o objeto executa uma ação
- Relação de controle entre ativação e o responsável pela sua invocação

Fragmentos



- Loop fragment encloses a series of messages which are repeated.
- **Alternative fragment** (denoted “alt”) models if...then...else constructs.
- **Option fragment** (denoted “opt”) models switch constructs.
- **Parallel fragment** (denoted “par”) models concurrent processing.
- **Weak sequencing fragment** (denoted “seq”) encloses a number of sequences for which all the messages must be processed in a preceding segment before the following segment can start, but which does not impose any sequencing within a segment on messages that don't share a lifeline.
- **Assertion fragment** (denoted “assert”) designates that any sequence not shown as an operand of the assertion is invalid.

Sub diagrams

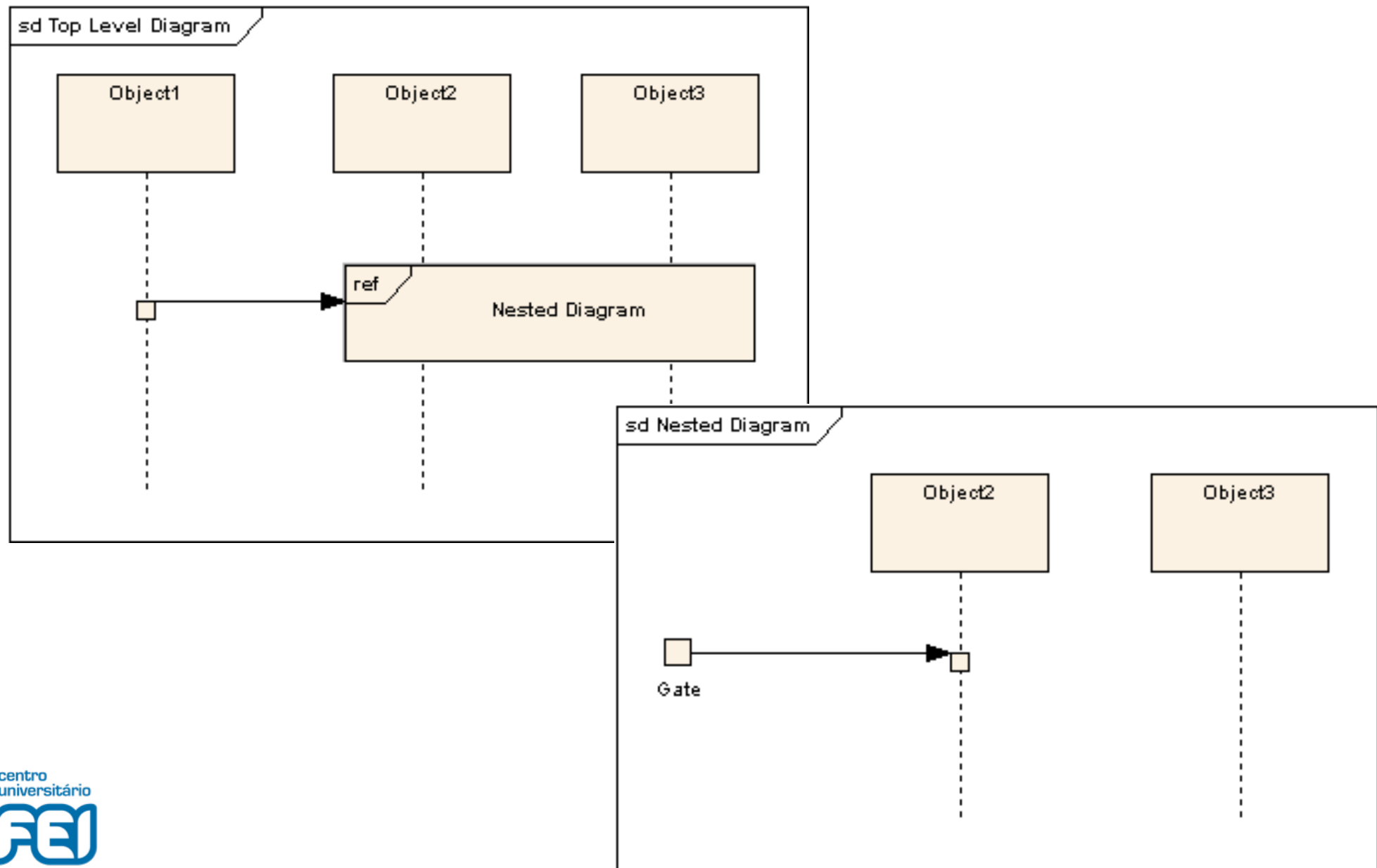


Diagrama de Seqüência - Construção

- Escolher um **caso de uso**
- Identificar os **objetos** que fazem parte da **interação**
- Identificar o objeto que **começa** a interação
- Identificar as **mensagens** trocadas entre os objetos
- Identificar a **sequência** destas mensagens

Dicas finais

- Preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos de determinado processo;
- Baseado em um caso de uso definido;
- Apóia-se no diagrama de classes;
- Todos os atores mapeados no Diagrama de Casos de Uso devem estar representados nos Diagramas de Sequencia
- Não devem ser construídos diagramas apenas com “interface”. Todos deve mostrar a troca de mensagens entre as classes do seu projeto.

Lembra o problema da locadora?? Então...

