

CC6521
Modelagem de Sistemas

Diagramas de Seqüência



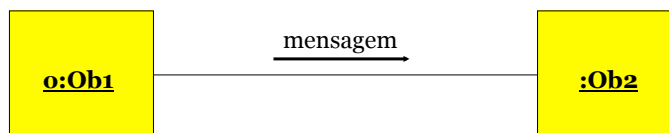
Prof. Ricardo Destro
1º Semestre de 2015

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



Diagramas de Interação

- Deseja-se **representar o comportamento** de vários objetos
 - Dentro de um único caso de uso
 - A partir das **mensagens** que são passadas entre eles
- Objetivo
 - Definir um **contexto** de caso de uso
 - Estabelecer os **objetos** que interagem e seus **relacionamentos**
- Termo genérico que se aplica a dois tipos de diagramas que enfatizam interações entre objetos
 - **Diagrama de Seqüência**
 - **Diagrama de Colaboração**



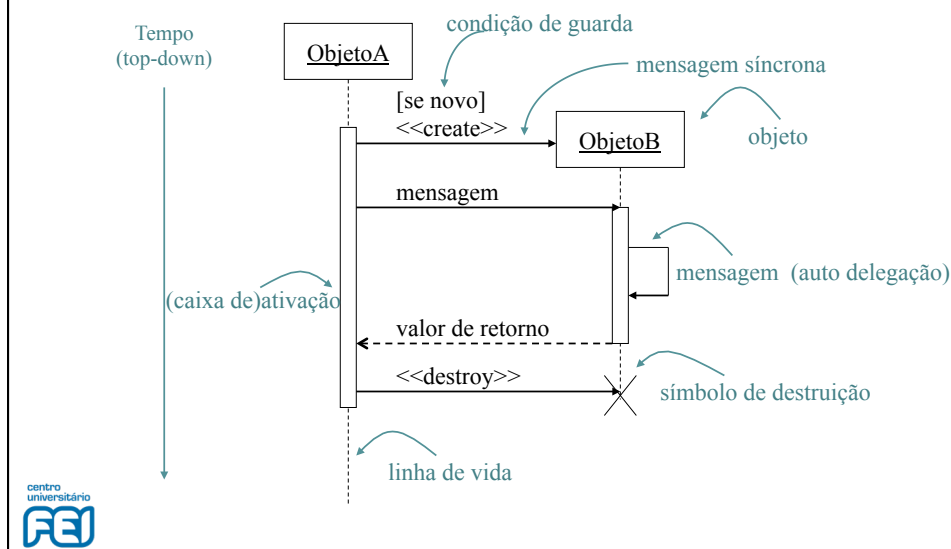
Duas formas de representação

- Informações bastante similares mas de maneira diferente
 - Diagrama de Seqüência
 - Interação enfatizando o **tempo de seqüência**
 - Mostra objetos participando em interações de acordo com suas linhas de vida e as mensagens que trocam
 - Diagrama de Colaboração
 - Interação enfatizando o **relacionamento** entre os objetos

Diagrama de Seqüência de Sistema

- Serve para sistematizar os fluxos dos casos de uso expandidos.
- Deve-se construir 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.

Diagrama de sequência



Objetivo

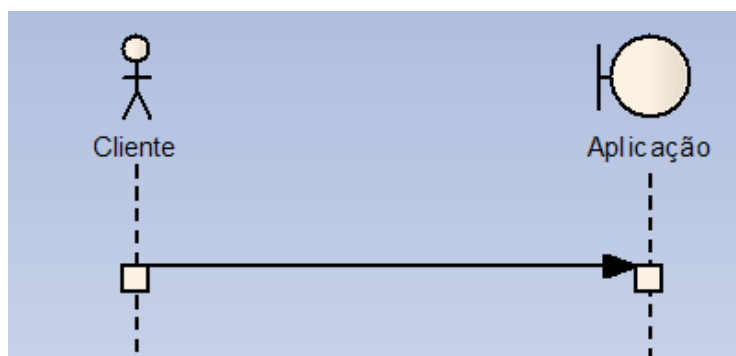
- Identificar **operações** e **consultas** que o sistema implementará para realizar as transações previstas nos casos de uso.
- Um diagrama de sequência mostra uma interação, isto é, uma sequência de mensagens trocadas entre vários objetos num determinado contexto (caso de uso, operação, etc.)
- Enfatiza 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

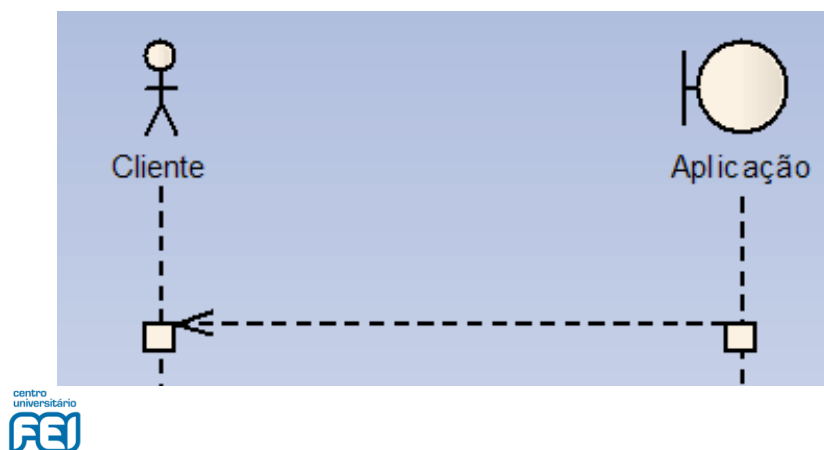
Evento de Sistema

- Informação flui de um dos atores para a aplicação



Resposta de Sistema

- Informação flui da aplicação para os atores

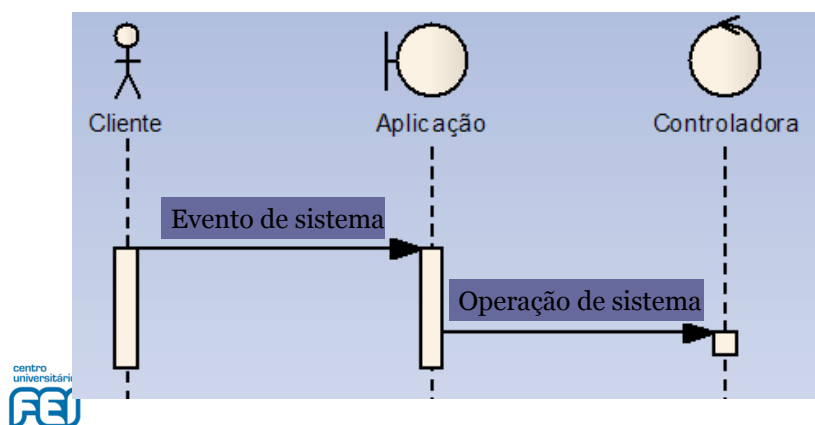


Consequências

- Eventos e respostas do sistema têm consequências na camada de aplicação de um sistema.
- Essas consequências são procedimentos (métodos) de entrada e saída que terão que ser analisadas, projetadas e implementadas para realizar os casos de uso.

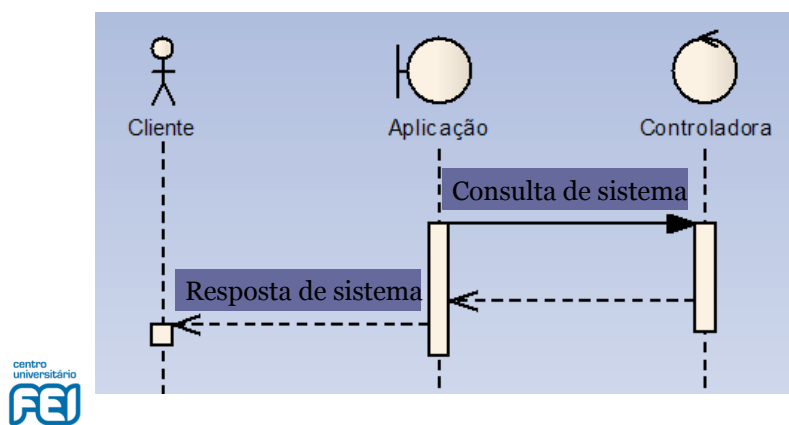
Conseqüências de um evento de sistema

- **Operação de Sistema**, quando se tratar de entrada de dados ou controle:



Conseqüências de uma resposta de sistema

- **Consulta de Sistema**: produz as informações necessárias



Combinações

- Tanto o evento quanto a resposta de sistema podem ter uma sequência 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)

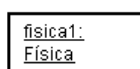
Componentes - Objetos

- 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

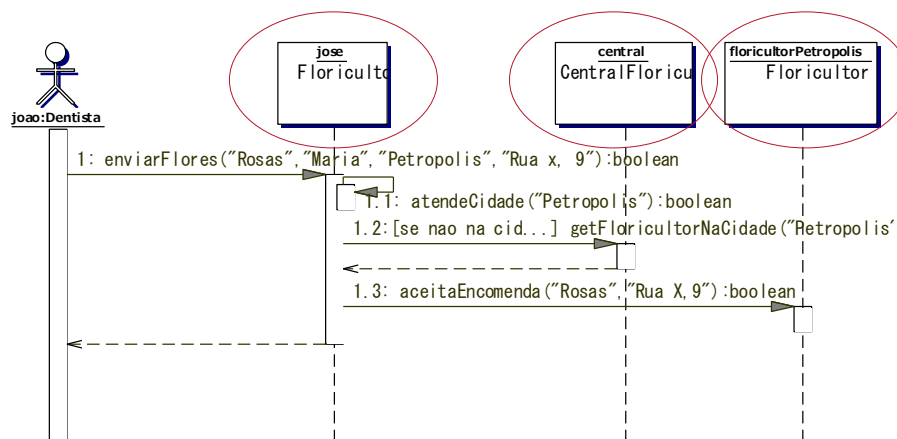


Componentes - Objetos

- Exemplo



Componentes - Objetos



Componentes - Linhas de Vida

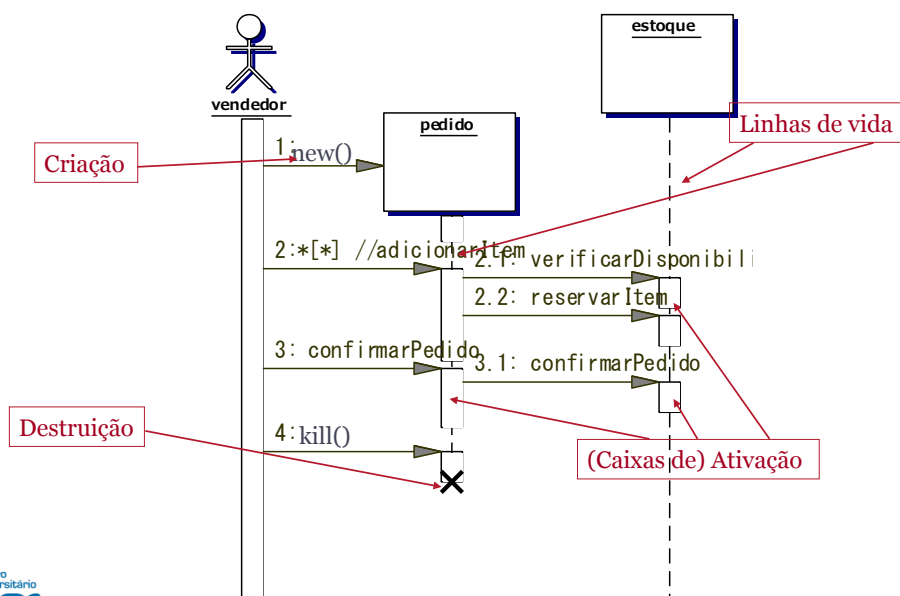
- **Dimensão vertical** do diagrama
- Apresentam o **tempo de vida** dos objetos
- Pode apresentar a **ativação** ou a **desativação** dos objetos
 - Indicam que os objetos estão executando algo
 - Foco de controle
 - Caixas de ativação podem ser empilhadas
 - Indica chamada de método do próprio objeto
 - 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



Linhas de Vida



Mensagens

- Objetos interagem através da troca de mensagens
 - Setas sólidas que vão do objeto solicitante para o solicitado
 - Para o próprio objeto: auto-delegação
 - Rotulados com os nomes dos estímulos mais os argumentos (ou valores dos argumentos) do estímulo

Mensagens

- Sintaxe


```
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 nome do tipo desse parâmetro
 - **returnType** é o tipo do valor de retorno

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



Mensagens - Representações

Símbolo	Significado
	Mensagem síncrona
	Mensagem assíncrona
	Mensagem de retorno (opcional)

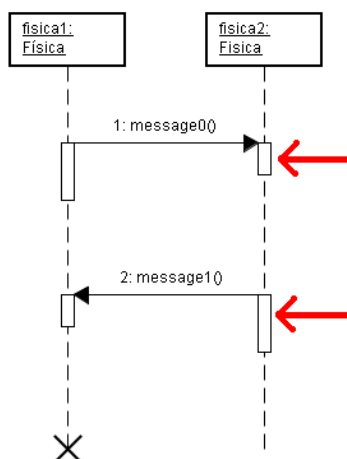
Componentes - FOCO DE CONTROLE/ATIVAÇÃO

- Indica os períodos em que um determinado objeto está participando ativamente do processo
 - Executando um ou mais métodos do processo
- Representados por extensões mais grossas/largas da Linha de Vida



Componentes - FOCO DE CONTROLE/ATIVAÇÃO

- **Exemplo**

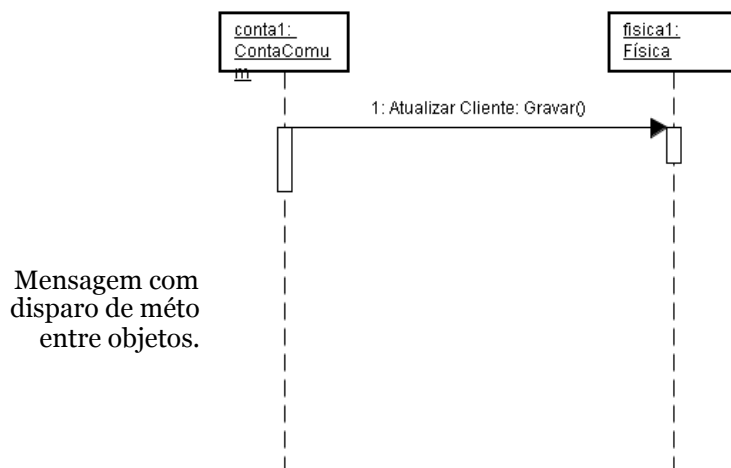


Componentes - MENSAGENS/ ESTÍMULOS

- Demonstram a ocorrência de eventos que normalmente forçam a chamada de um método em algum dos objetos envolvidos no processo
- Mensagens entre:
 - Ator e Ator
 - Ator e Objeto
 - Objeto e Objeto
 - Objeto e Ator

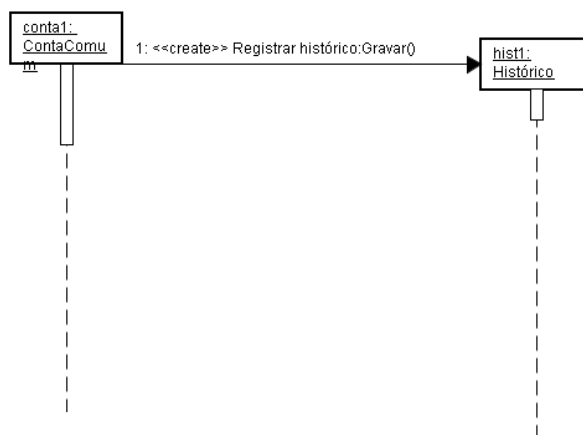


Componentes - MENSAGENS/ ESTÍMULOS



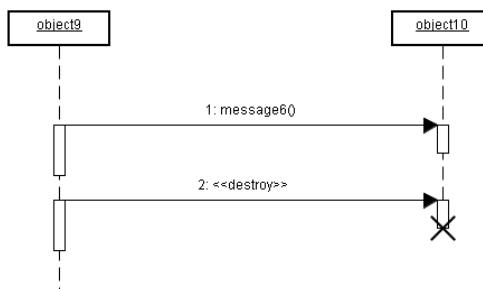
Componentes - Mensagens

Mensagem
que
instancia
um novo
objeto



Componentes - Mensagens

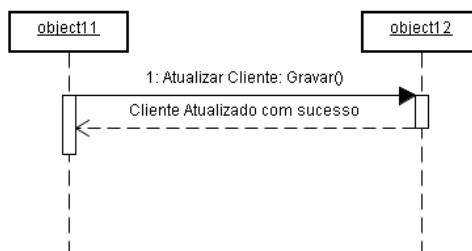
Mensagem que
dispara um
método
destrutor –
elimina um
objeto não mais
necessário.



Componentes - Mensagens

Mensagem de Retorno ... Linha tracejada.

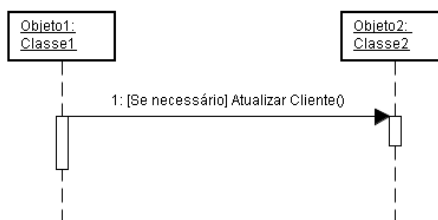
Podem retornar valores ou status...



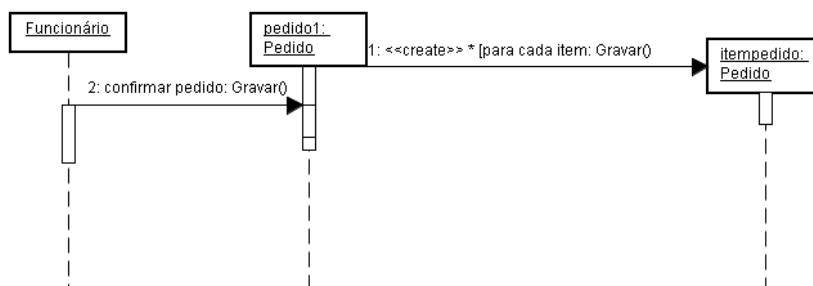
Componentes - Mensagens

Mensagem com Condição de Guarda

Entre colchetes []



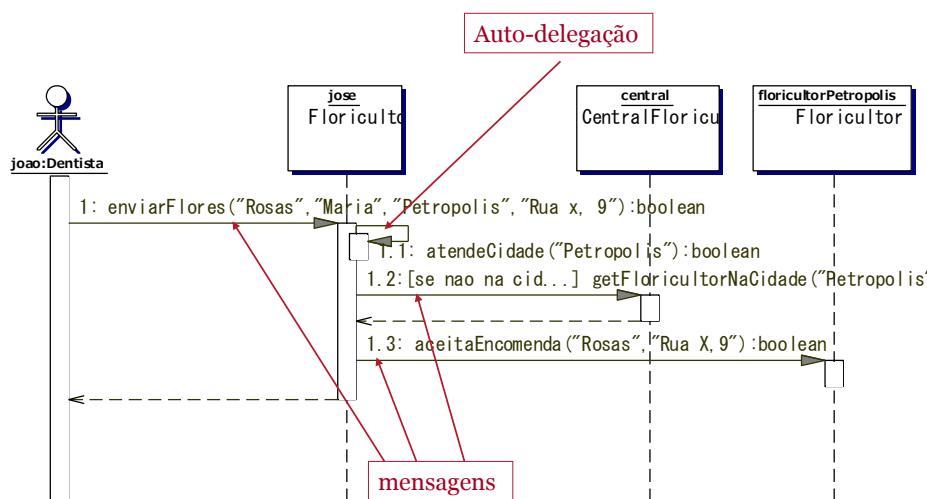
Componentes - Mensagens



Mensagem com Condição de Guarda

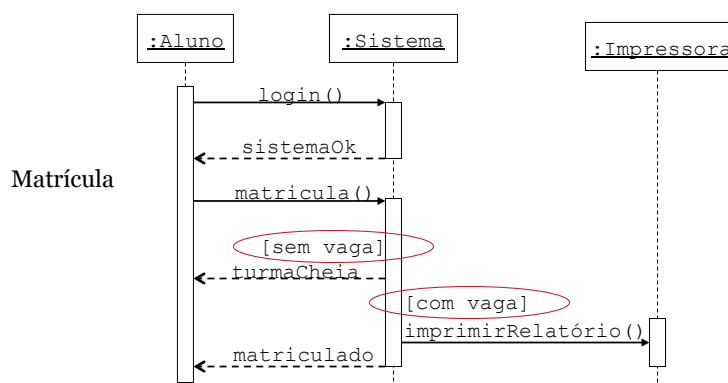
Disparo de mensagem a vários objetos (*)

Mensagens



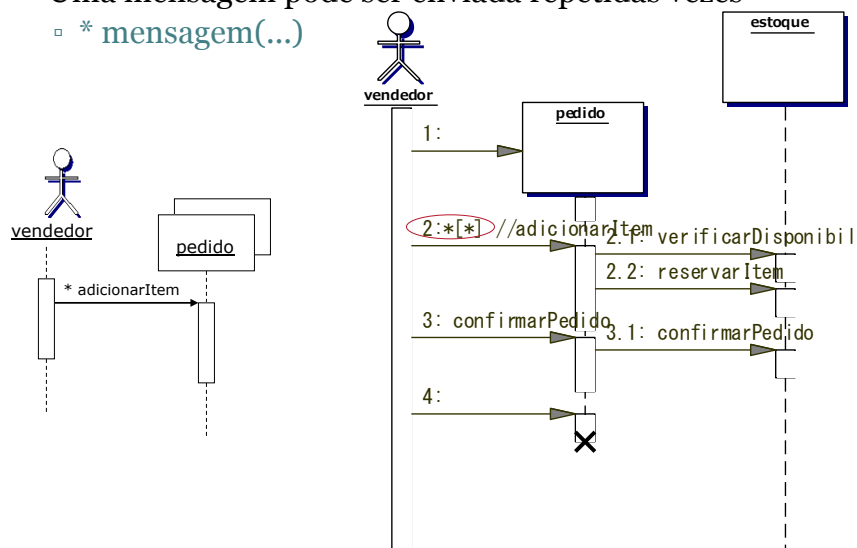
Mensagens Condições de Guarda

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



Mensagens - Iteração

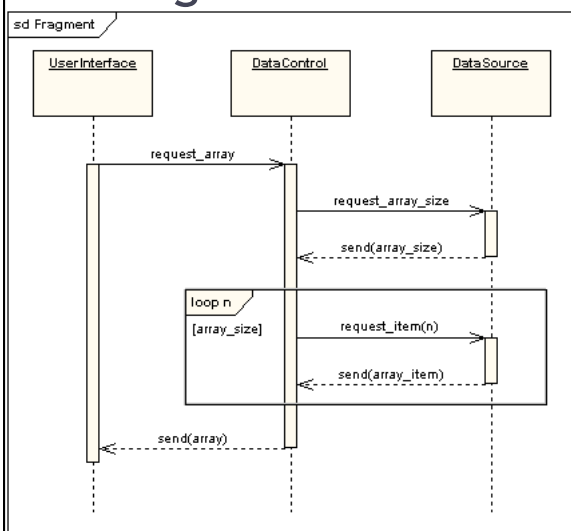
- Uma mensagem pode ser enviada repetidas vezes
 - * mensagem(...)



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.

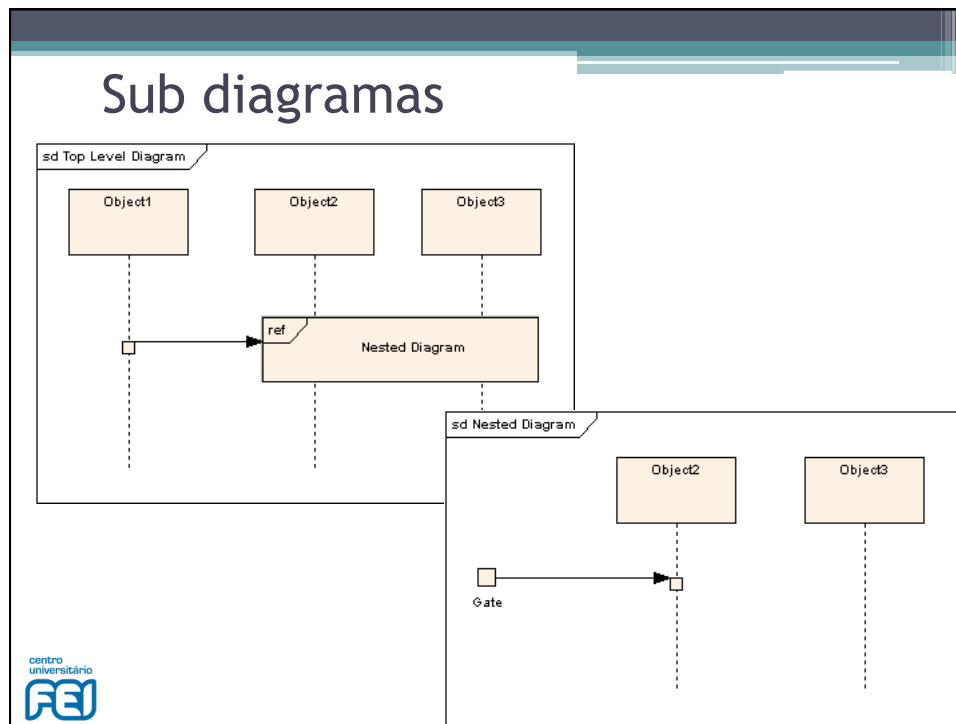


Diagrama de Seqüência - Construção

- 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 **seqüência** 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;