

Aula 04 - Diagrama de Sequência

Disciplina: Engenharia de Software

Prof. Adriano Nakamura





Objetivo

- Apresentar o diagrama 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

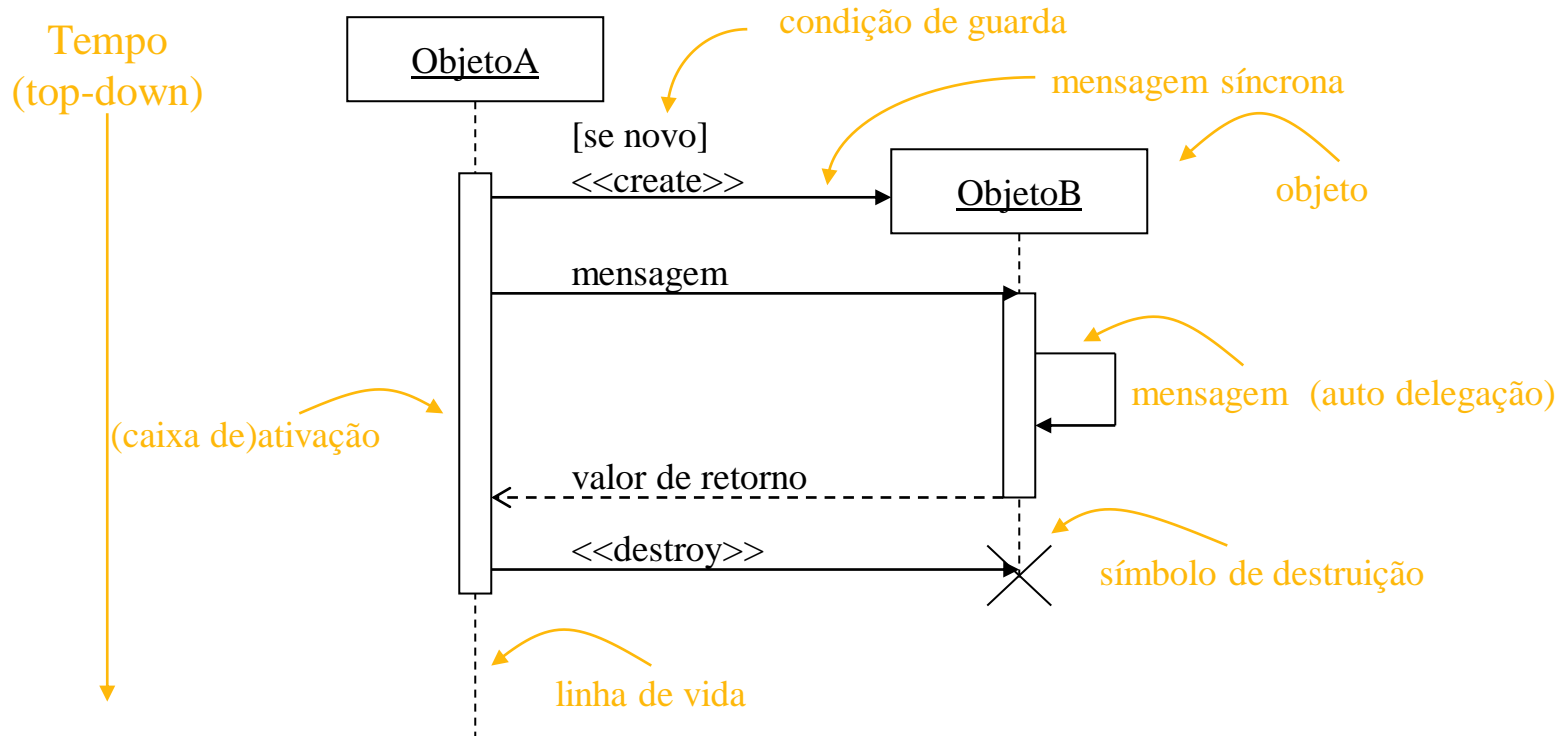
Diagramas de Interação

- 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 quatro tipos de diagramas que enfatizam interações entre objetos
 - **Diagrama de Seqüência**
 - **Diagrama de Colaboração/Comunicação**
 - **Vista Geral de Interação**
 - **Temporal ou *Timing***

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 Comunicação
 - Interação enfatizando o **relacionamento** entre os objetos

Diagrama de seqüência



Termos e conceitos

- Objetos
- Linhas de vida
- Mensagens
- Focos de controle

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 classe Floricultor não identificado)

objI: (um objeto objI sem classe definida)

Objetos


joao:Dentist

jose
Floricultor

central
CentralFloricultu

floricultorPetrop
Floricultor

1: enviarFlores("Rosas","Maria","Petropolis","Rua x, 9"):boolean

1.1: atendeCidade("Petropolis"):boolean

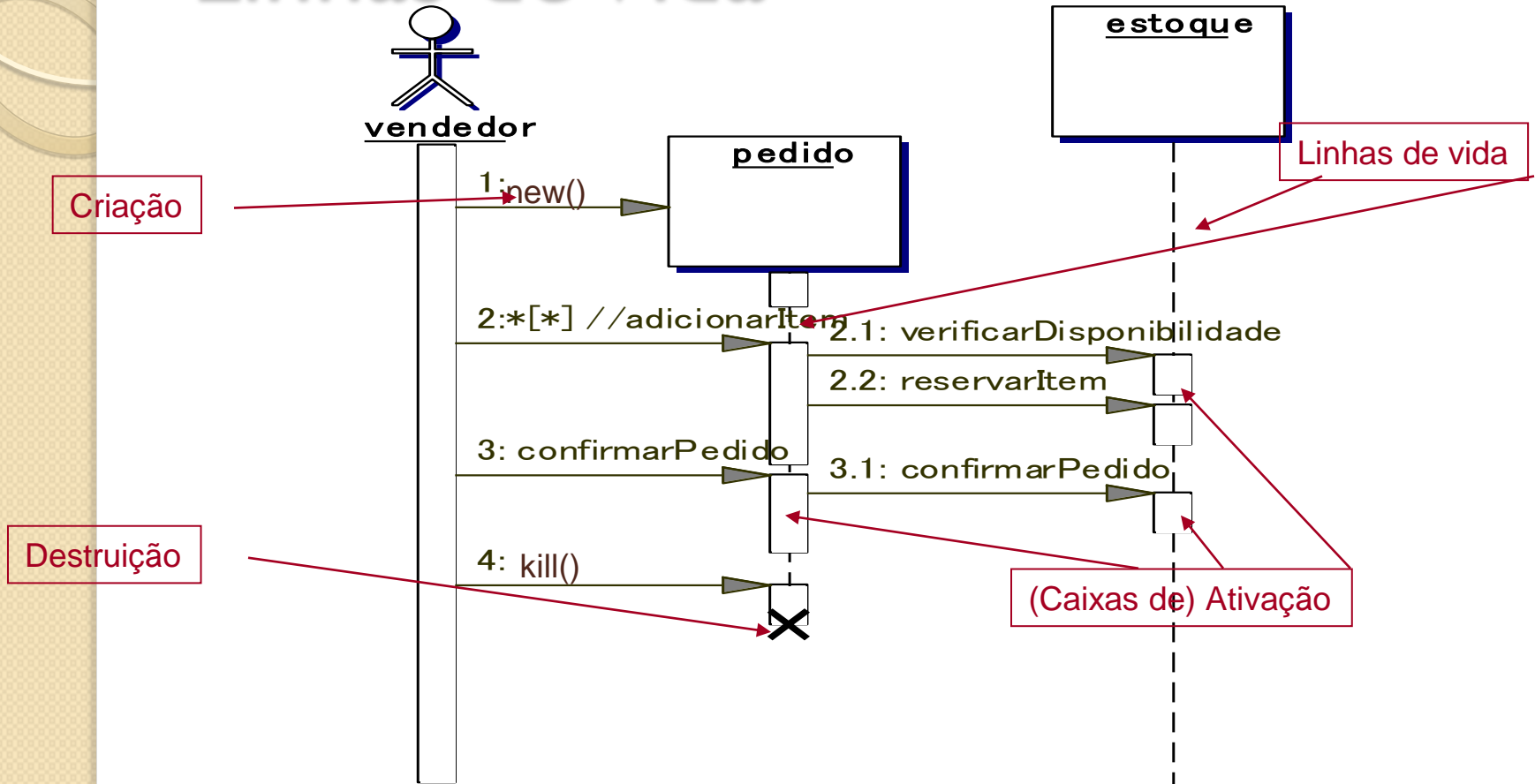
1.2:[se nao na cid...] getFloricultorNaCidade("Petropolis"):Floricultor

1.3: aceitaEncomenda("Rosas","Rua X,9"):boolean

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

Linhas de Vida



Mensagens

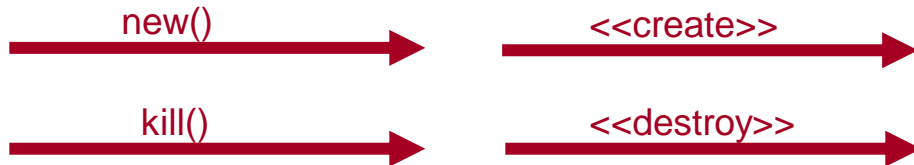
- Objetos interagem através da troca de mensagens
 - Setas sólidas que vão do objeto solicitante para o solicitado
 - Para o próprio objeto: auto-delegação
 - Rotulados com os nomes dos estímulos mais os argumentos (ou valores dos argumentos) do estímulo
- Sintaxe

```
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

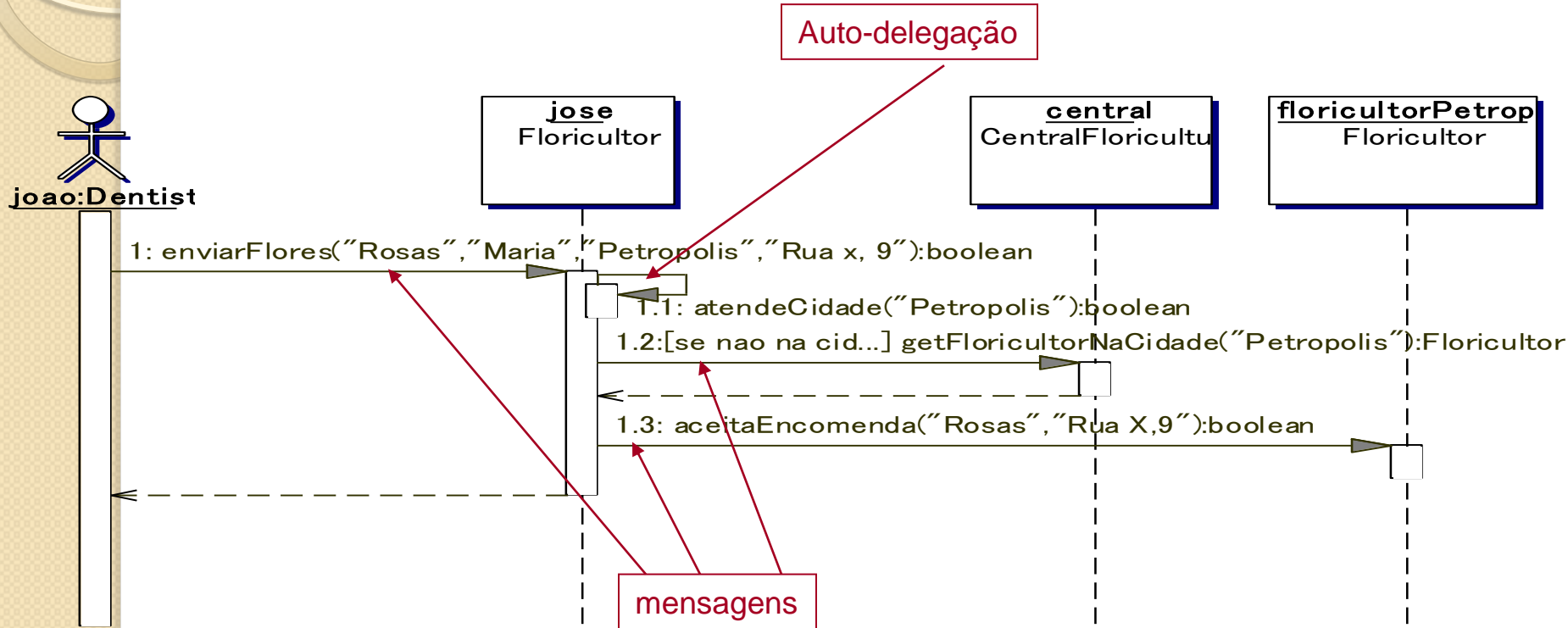
- Tipos de ação que uma mensagem pode representar
 - call
 - Invoca uma operação sobre um objeto
 - Objeto pode mandar uma chamada para si próprio
 - Resultando na execução local de uma operação
 - return
 - Representa o retorno de um valor para o objeto que chamou a operação
 - Opcional
 - create
 - Criação de um objeto
 - destroy
 - Eliminação de um objeto



Mensagens - Representações

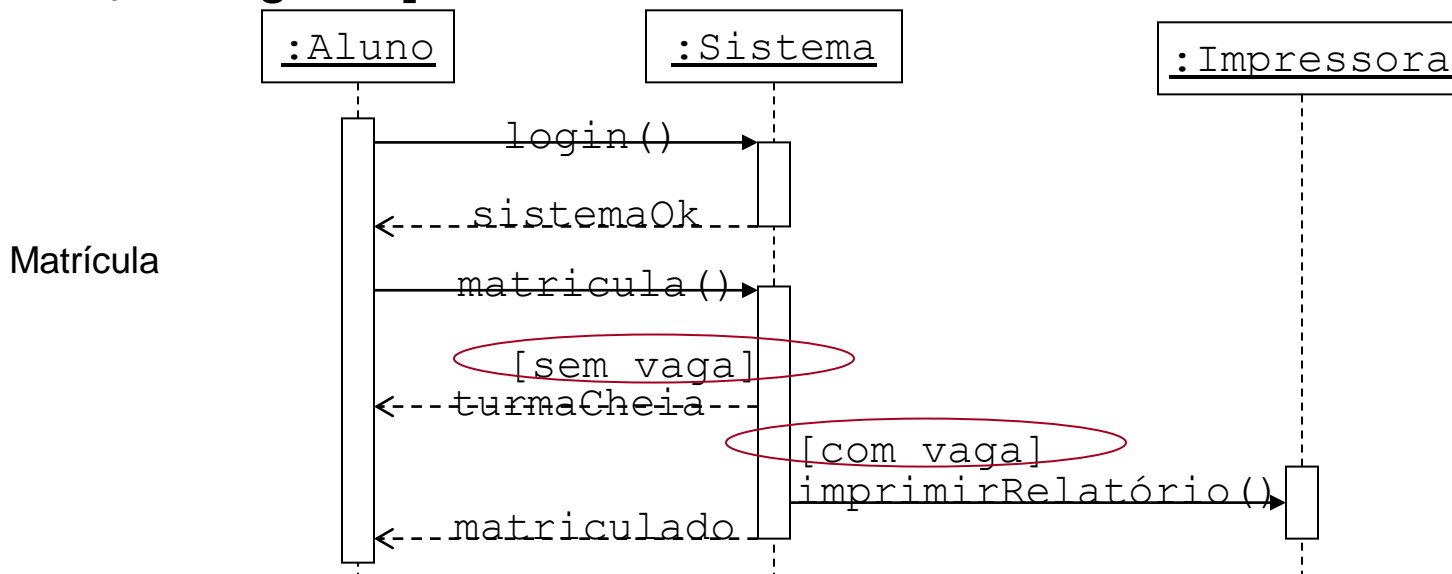
| Símbolo | Significado |
|---|--------------------------------|
|  | Mensagem síncrona |
|  | Mensagem assíncrona |
|  | Mensagem de retorno (opcional) |

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(...)

Mensagens - Iteração

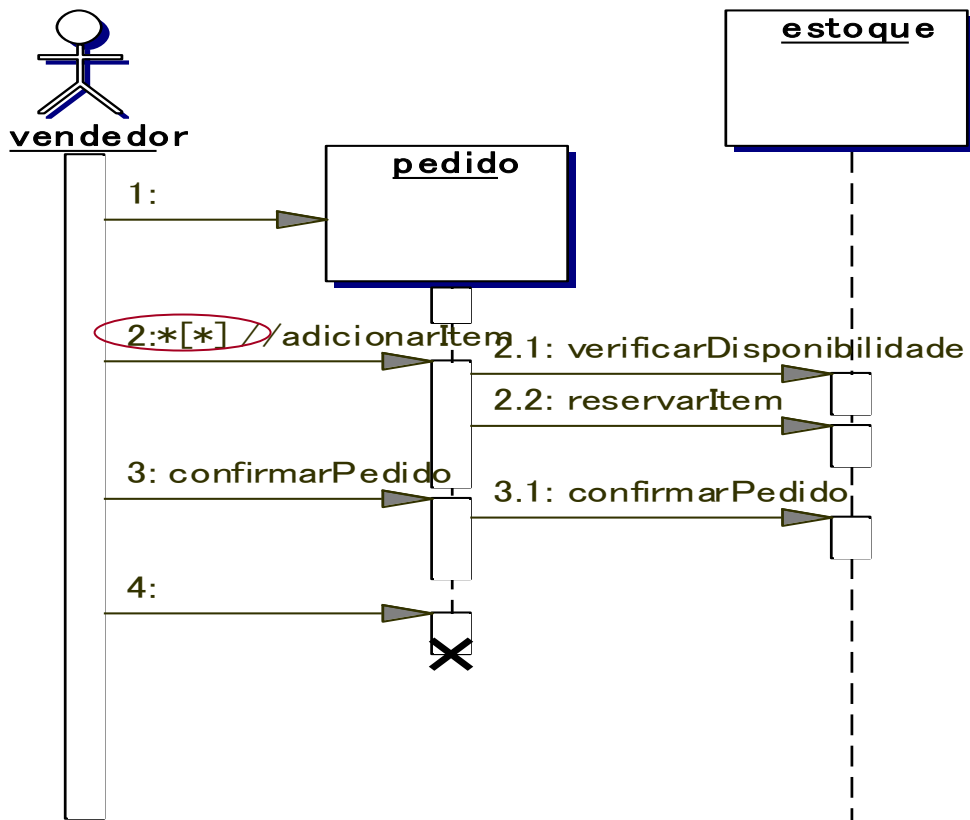
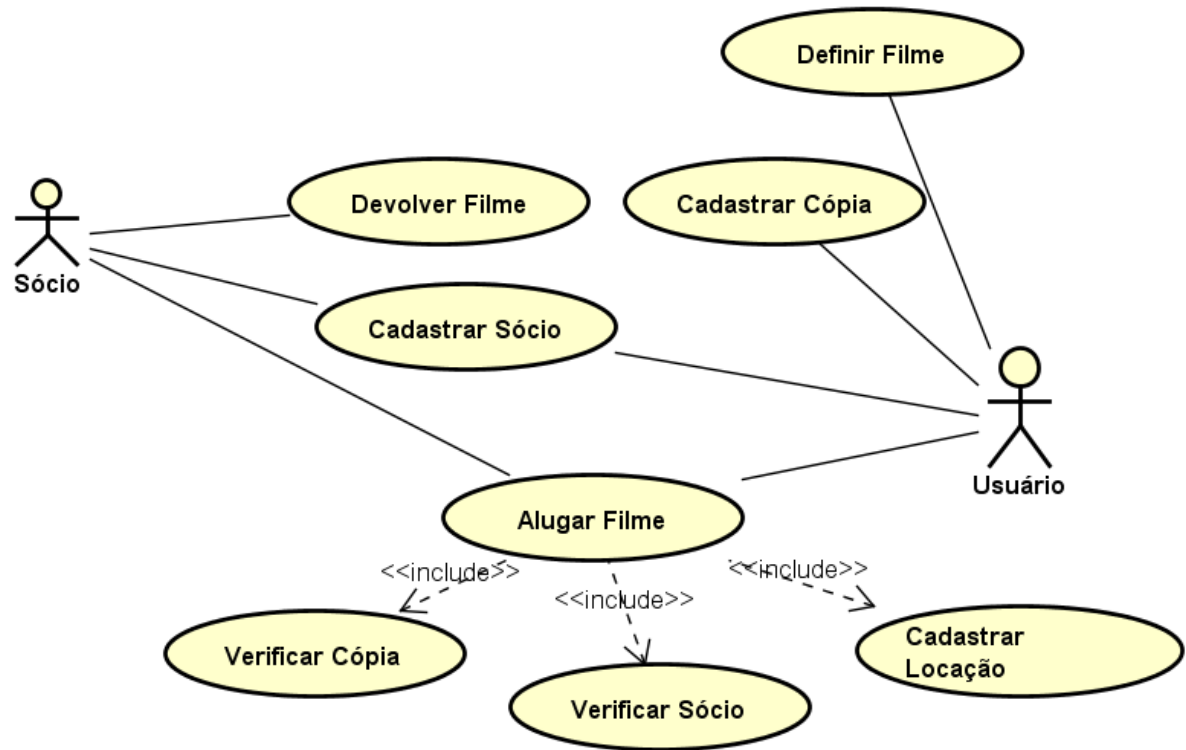


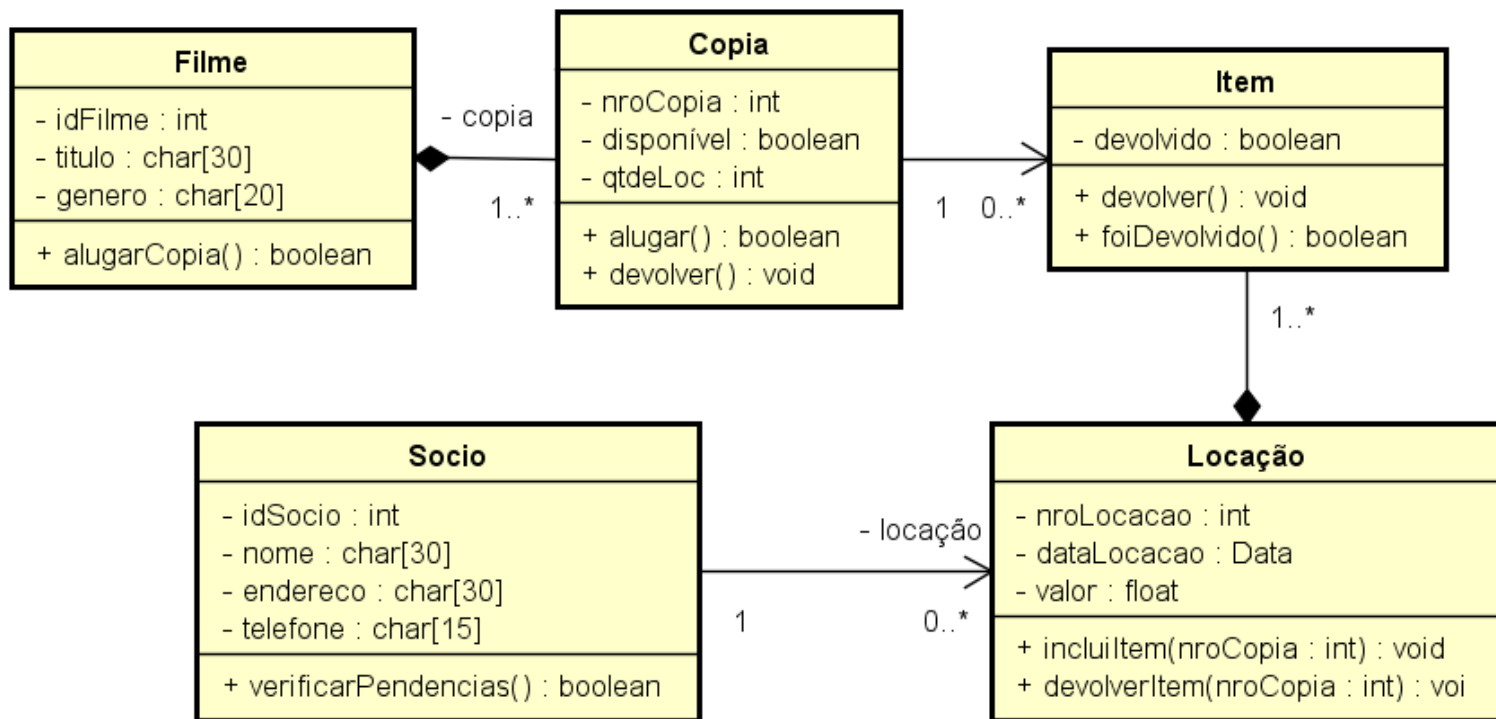
Diagrama de Seqüência Construção

- 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

Exemplo I - Locadora



Exemplo I - Locadora



Exemplo I - Locadora

- Caso de Uso considerado: Alugar Filme
 - Durante o processo de locação de fitas, deve-se verificar se o sócio se encontra devidamente registrado e se não há locações pendentes
 - Para verificar as pendências é necessário averiguar se itens já locados foram todos devolvidos
 - Caso não haja pendências, deve-se iniciar o registro da nova locação, verificando as cópias disponíveis e lançando cada item locado
 - A cópia referente ao item locado deve ser atualizado como não disponível

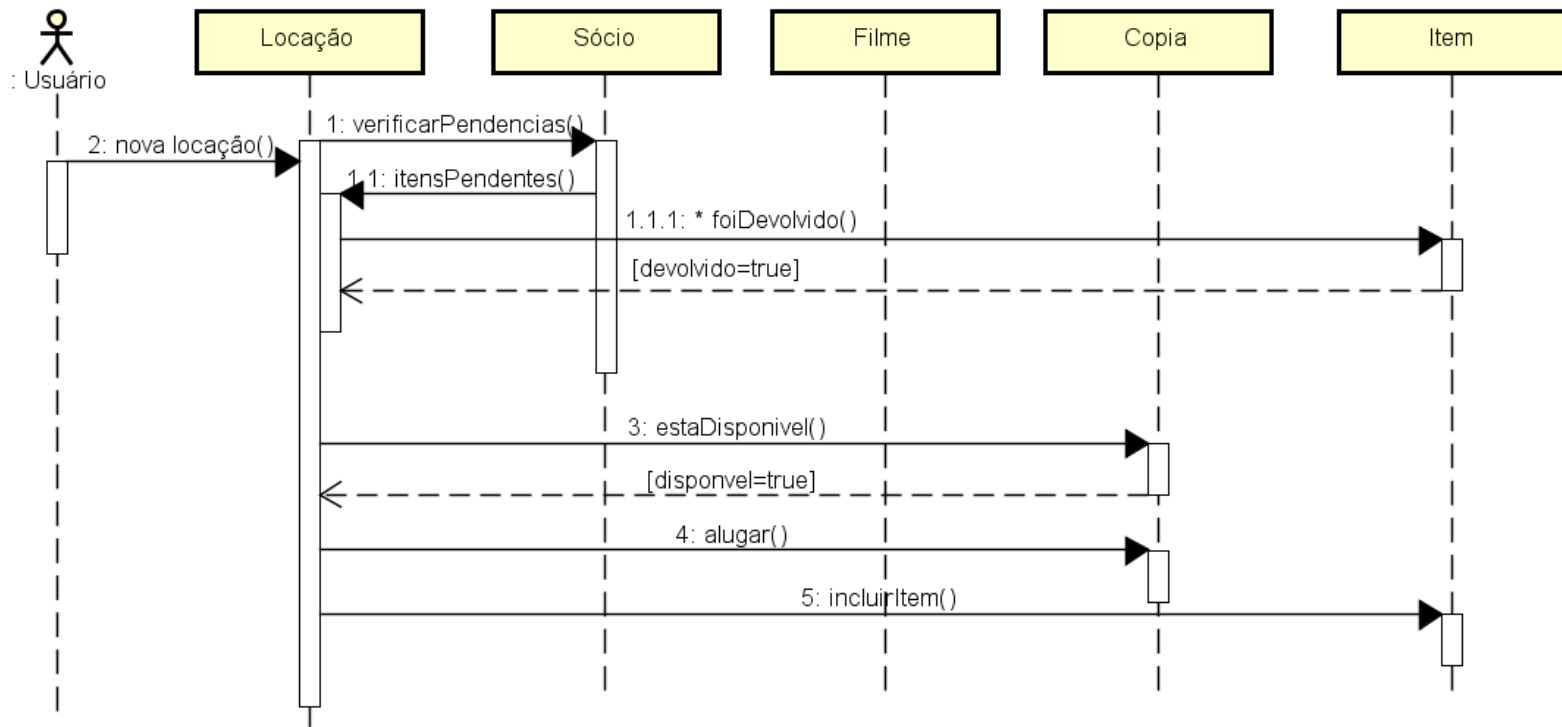
Exemplo I - Locadora

- Objetos que interagem:
 - Sócio
 - Filme
 - Cópia
 - Item
 - Locação

Exemplo I - Locadora

- Objeto que começa a interação:
 - Locação
- Identificar as **mensagens** trocadas entre os objetos
- Identificar a **sequência** destas mensagens

Exemplo I - Locadora



Exemplo I - Locadora

- Caso de Uso considerado: Devolver Filme
 - A devolução das cópias parte da locação feita pelo sócio.
 - Para cada item a ser devolvido deve ser alterado sua situação (devolvido = true) e também alterado a situação da cópia para disponível (disponível = true)

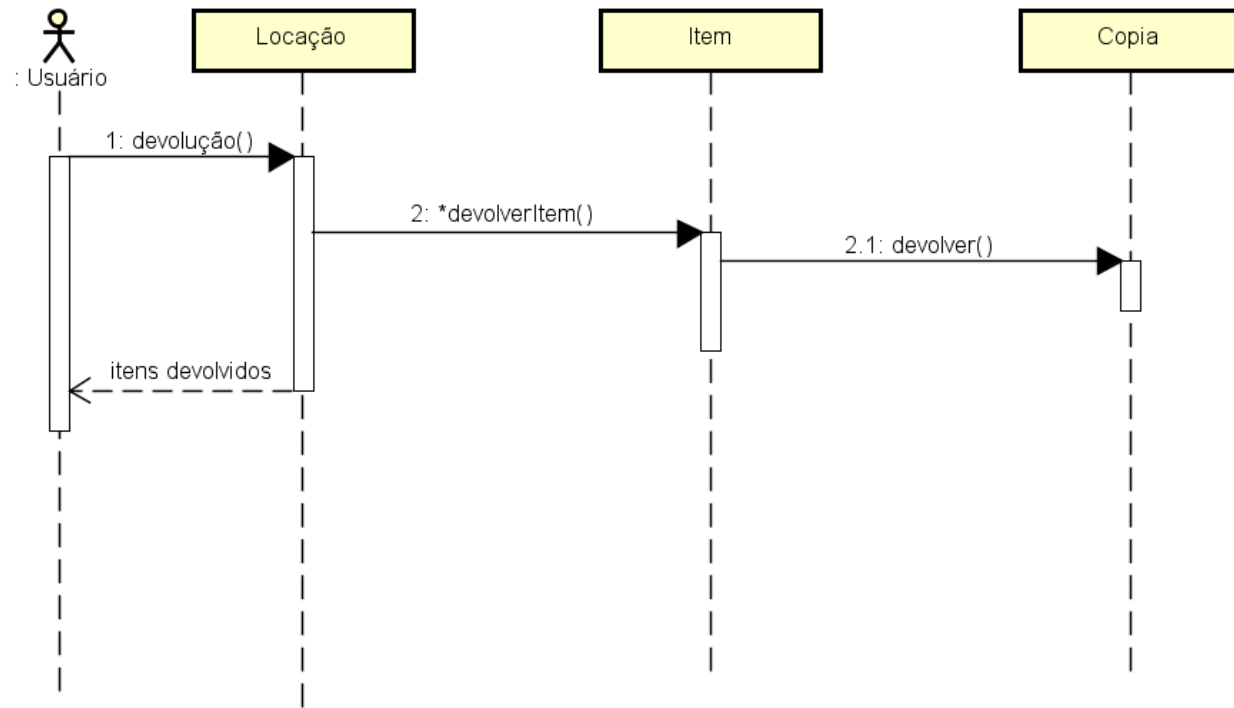
Exemplo I - Locadora

- Objetos que interagem:
 - Cópia
 - Item
 - Locação

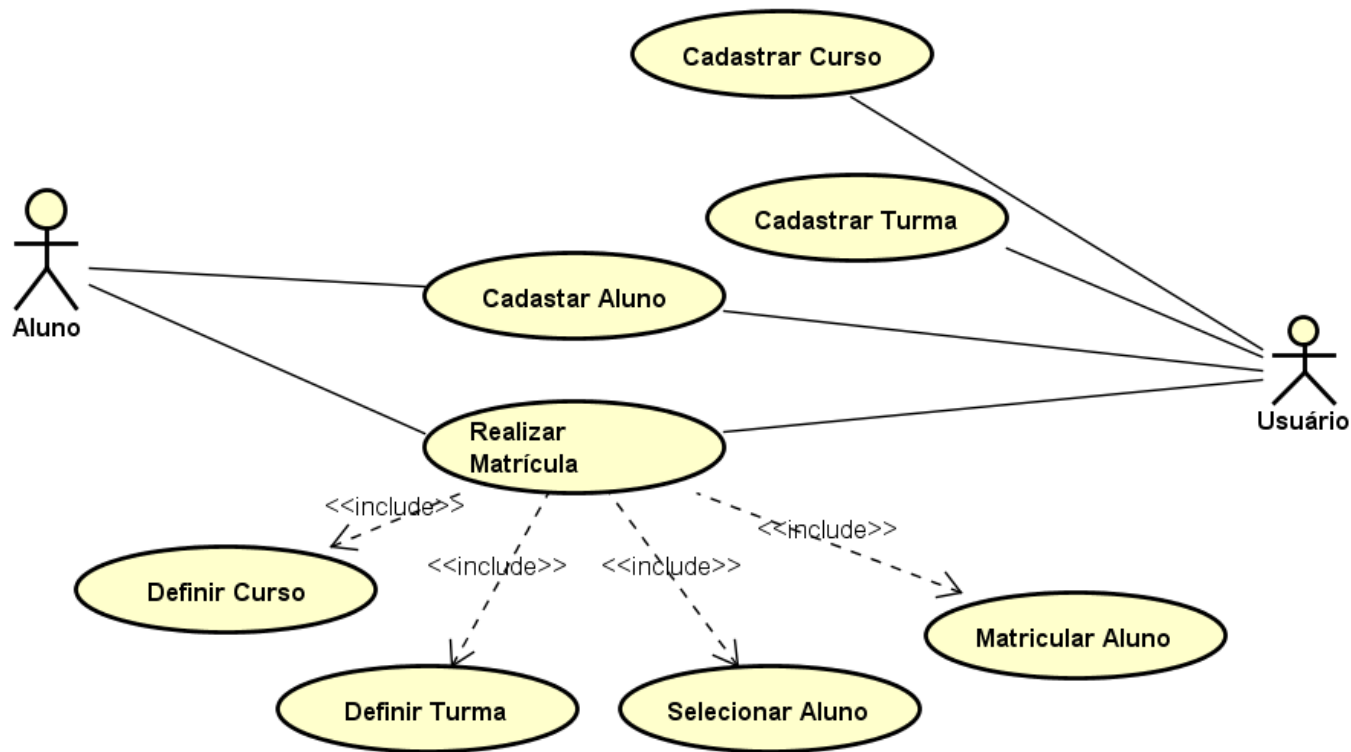
Exemplo I - Locadora

- Objeto que começa a interação:
 - Locação
- Identificar as **mensagens** trocadas entre os objetos
- Identificar a **sequência** destas mensagens

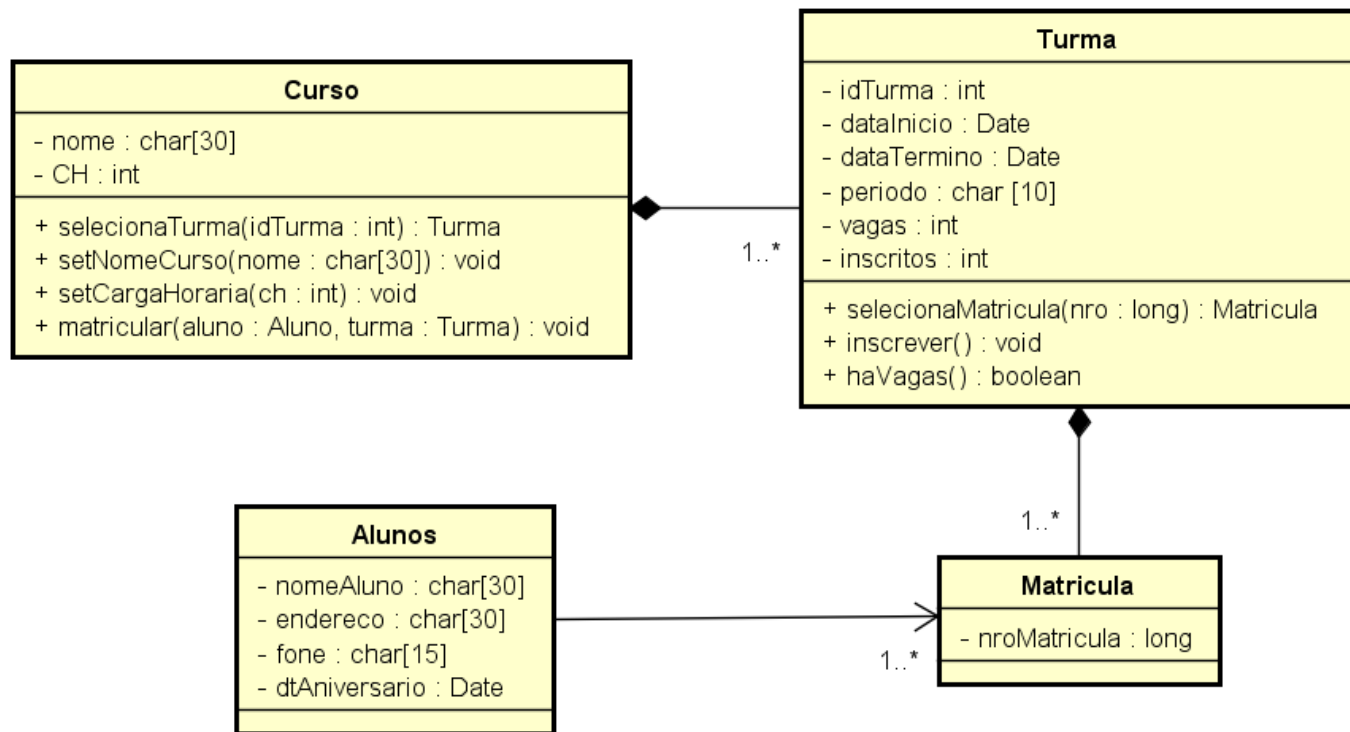
Exemplo I - Locadora



Exemplo 2 – Controle de Cursos



Exemplo 2 – Controle de Cursos



Exemplo 2 – Controle de Cursos

- Caso de uso considerado: Realizar Matrícula
 - O usuário deve selecionar o curso no qual o aluno deseja se matricular.
 - Na sequência deve selecionar a turma e verificar se ainda há vagas disponíveis
 - Verificar se o aluno está devidamente cadastrado
 - Efetivar sua matrícula

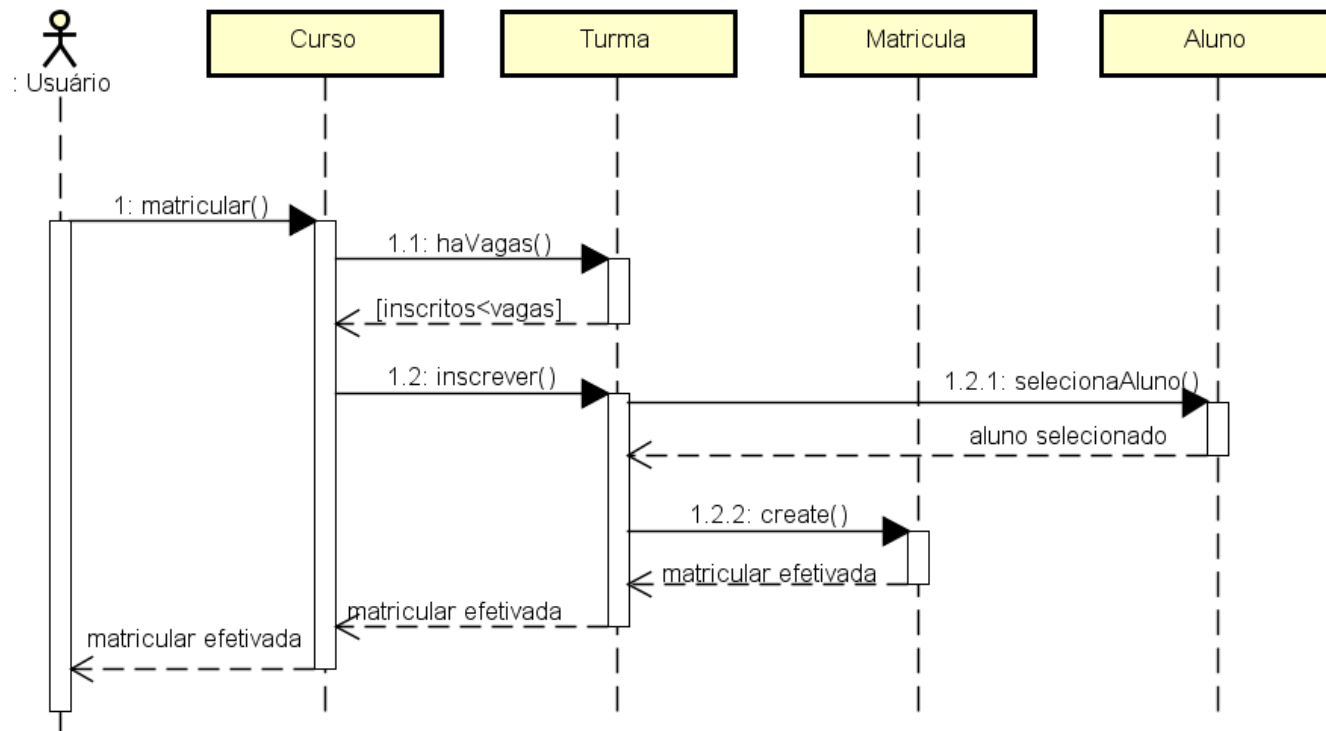
Exemplo 2 – Controle de Cursos

- Objetos que interagem:
 - Aluno
 - Curso
 - Turma
 - Matrícula

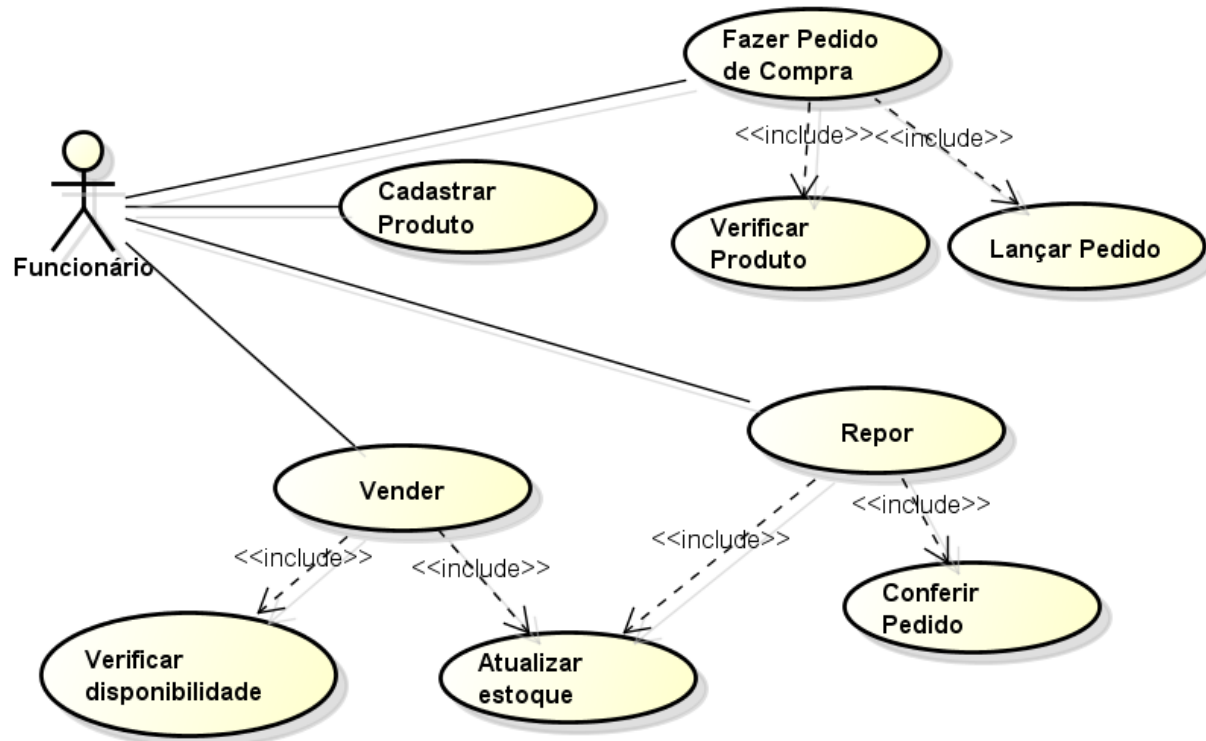
Exemplo 2 – Controle de Cursos

- Objeto que começa a interação:
 - Curso
- Identificar as **mensagens** trocadas entre os objetos
- Identificar a **sequência** destas mensagens

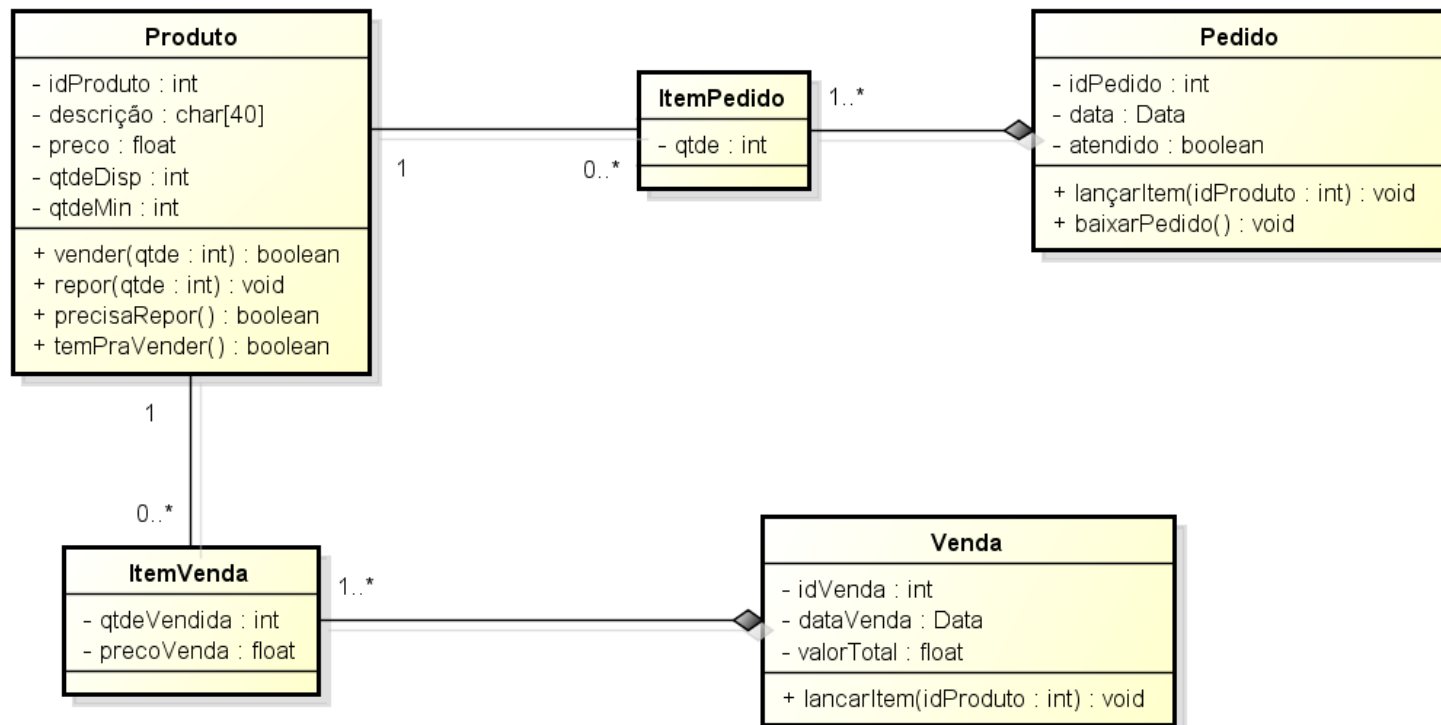
Exemplo 2 – Controle de Cursos



Exercício – Controle de Estoque



Exercício – Controle de Estoque



Exercício – Controle de Estoque

- I. Considere o caso de uso “Vender”
 - Para realizar uma venda o funcionário precisa identificar os produtos com suas respectivas quantidades
 - Para cada produto é necessário verificar se há quantidade disponível, em caso afirmativo é dada baixa na quantidade vendida e feito o lançamento do item de venda

Exercício – Controle de Estoque

- 2. Considere o caso de uso “Fazer Pedido”
 - Ao final do expediente o funcionário verifica a necessidade de fazer pedidos para repor o estoque.
 - Para realizar um pedido os produtos são consultados para verificar se precisa repor, em caso afirmativo, para cada produto é lançado um item com sua respectiva quantidade

Exercício – Controle de Estoque

- 3. Considere o caso de uso “Repor”
 - Quando um pedido é atendido o funcionário precisa dar entrada no estoque.
 - O pedido recebido é conferido com o pedido feito (cadastrado) item por item.
 - Uma vez conferido o item, é dada a entrada do produto no estoque.
 - Uma vez finalizados todos os itens o pedido é marcado como atendido.