



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

UML – Diagrama de sequência

Eduardo Kinder Almentero
ekalmentero@gmail.com

Termos e conceitos

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

Atores

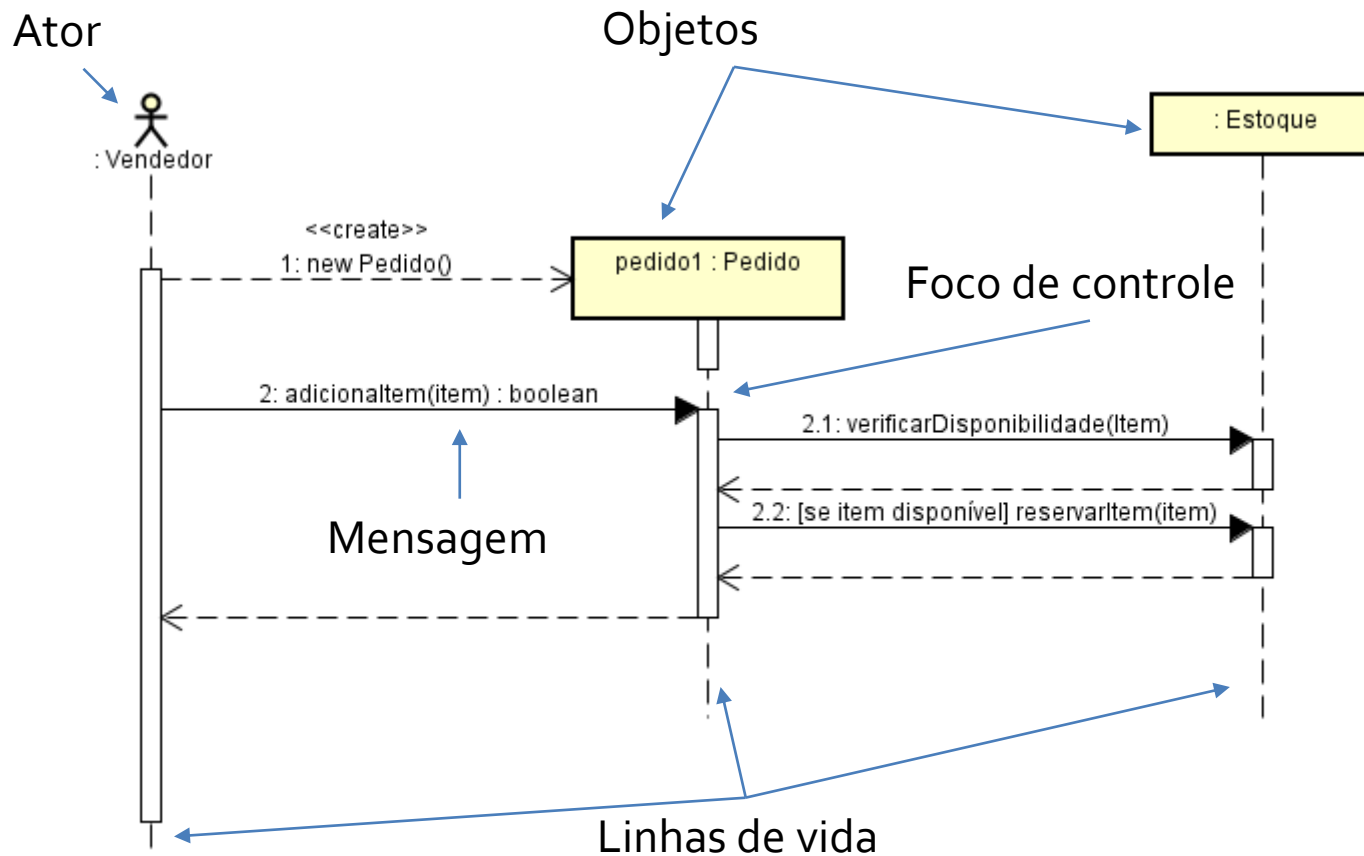
- São os mesmos do Diagrama de Caso de Uso e possuem a mesma representação, mas diferenciam-se por apresentarem uma “linha de vida”.



Objetos

- Apresentados na dimensão horizontal do diagrama.
- Ordem dos objetos não é considerada.
- É recomendado organizá-los de forma a tornar o diagrama “mais legível”
- Objetos tem nomes e são instâncias de classes
 - nomeObj:Classe
- Ex.: pedido1:Pedido
 - :Vendedor(um objeto cliente não identificado)
 - obj1: (um objeto obj1 sem classe definida)

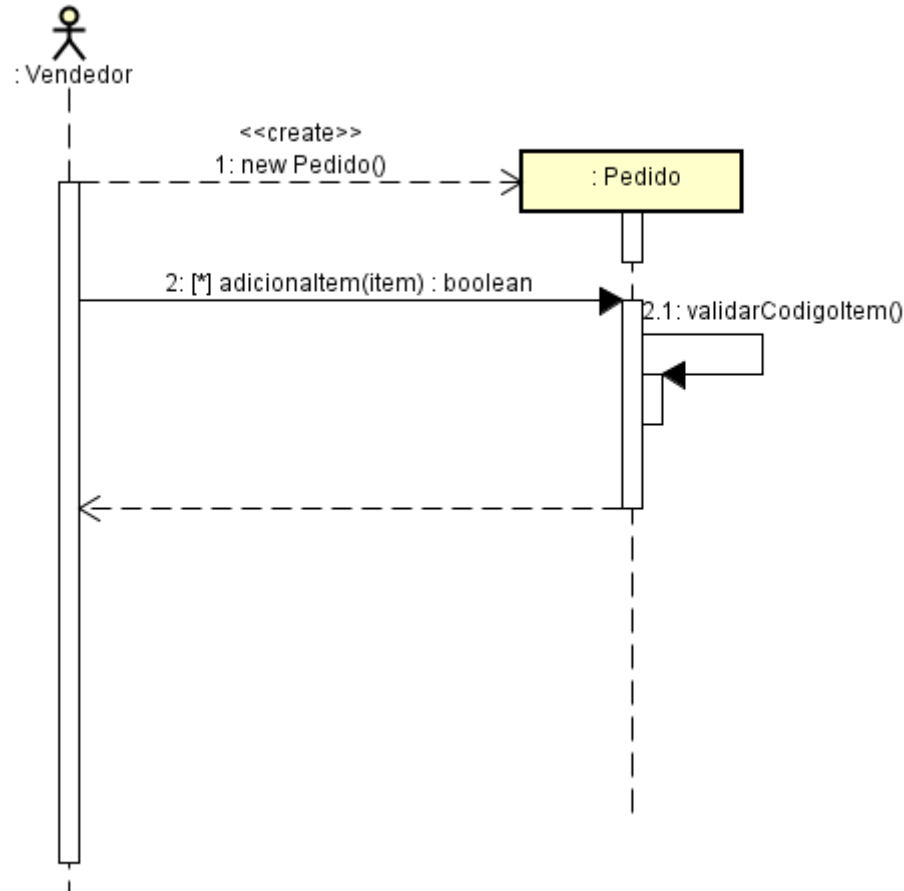
Principais Elementos do Diagrama



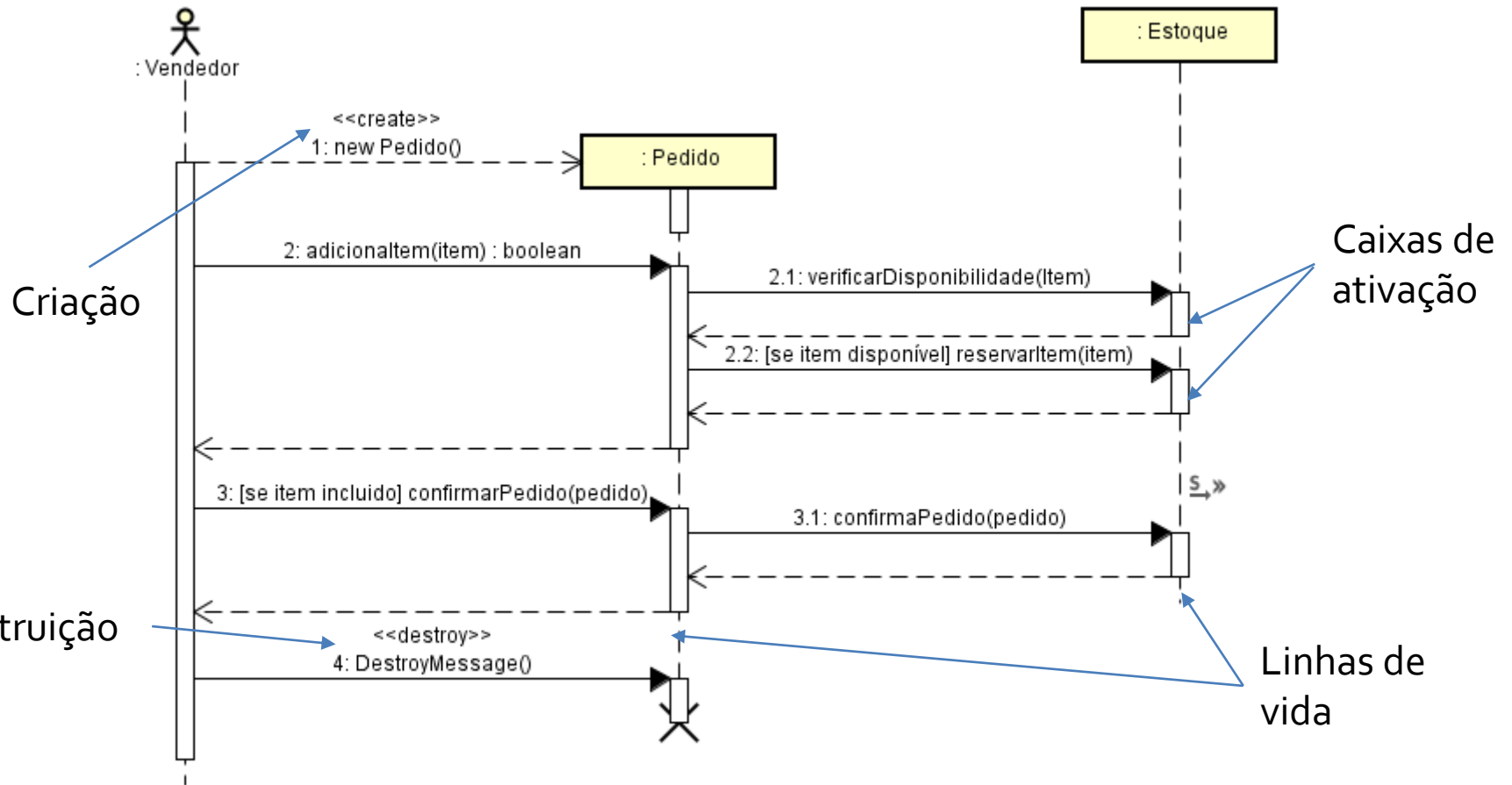
Linhas de vida

- 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
- Podem representar a criação e a destruição de objetos.

Chamada de método do próprio objeto



Linhas de vida



Mensagens

- Objetos interagem através da troca de mensagens.
 - Setas sólidas (contínua) 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

Tipos de mensagens

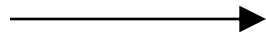
- Tipos de ação que uma mensagem pode representar
 - call
 - Invoca uma operação sobre um objeto
 - Objeto pode mandar uma chamada para si próprio, resultando na execução local de uma operação
- return
 - Representa o retorno de um valor para o objeto que chamou a operação
- create
 - Criação de um objeto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

`new()` → `<<create>>`

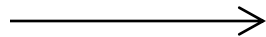
`kill()` → `<<destroy>>`

Representações de mensagens

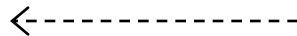
- Mensagem síncrona - a execução fica bloqueada até o retorno do método.



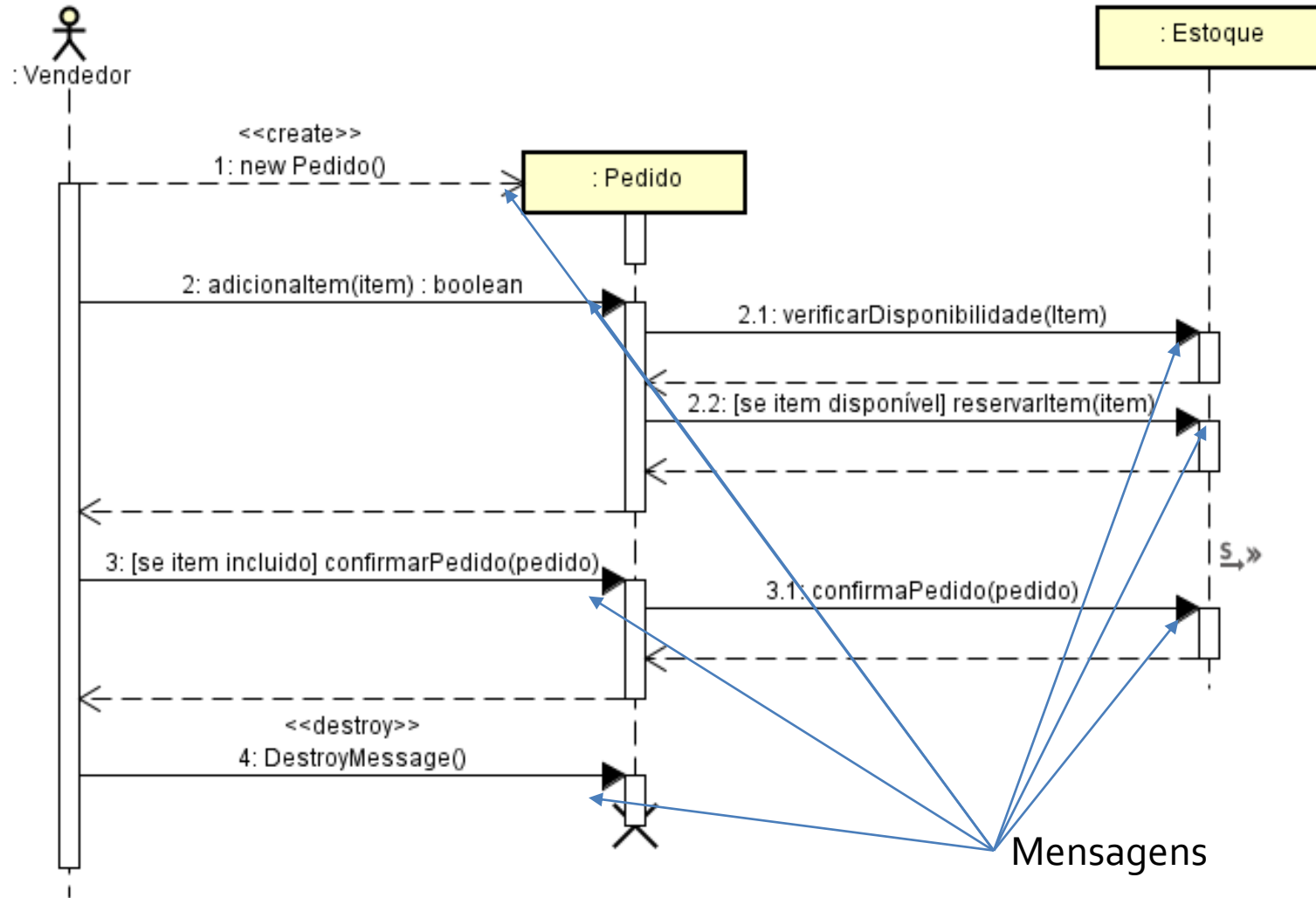
- Mensagem assíncrona - a execução continua em paralelo ao método que foi chamado (fork implícito)



- Mensagem de retorno



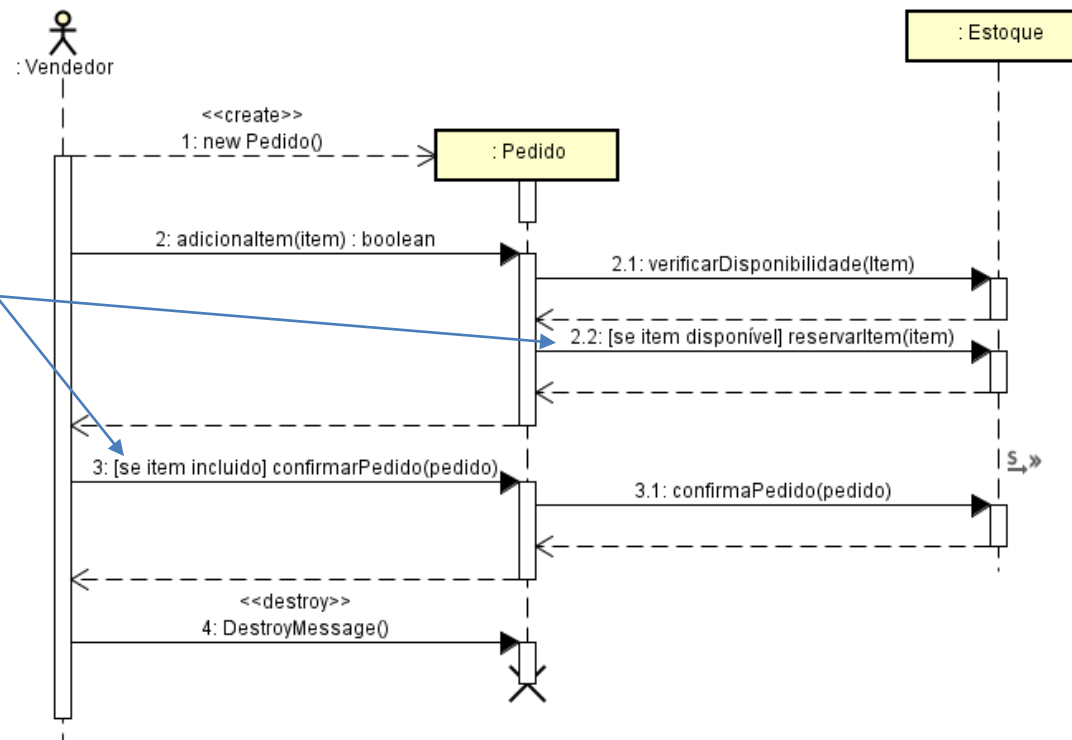
Mensagens



Condições de guarda de mensagens

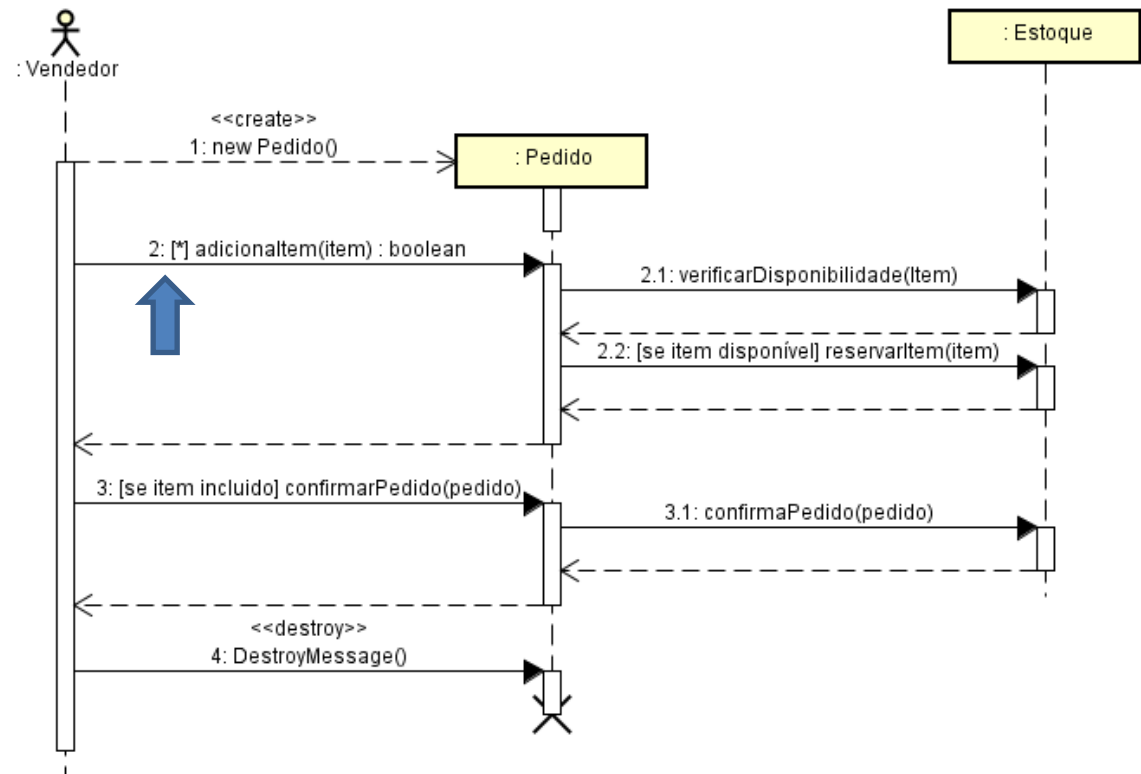
- Mensagens podem apresentar condições de guarda
 - condições em que a mensagem é enviada
 - Sintaxe: [condição de guarda]

Condições 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.
- 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 gross

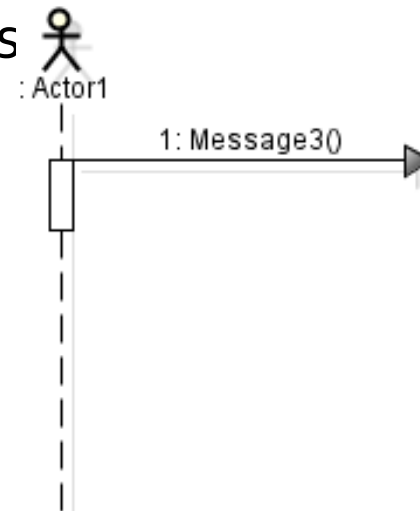
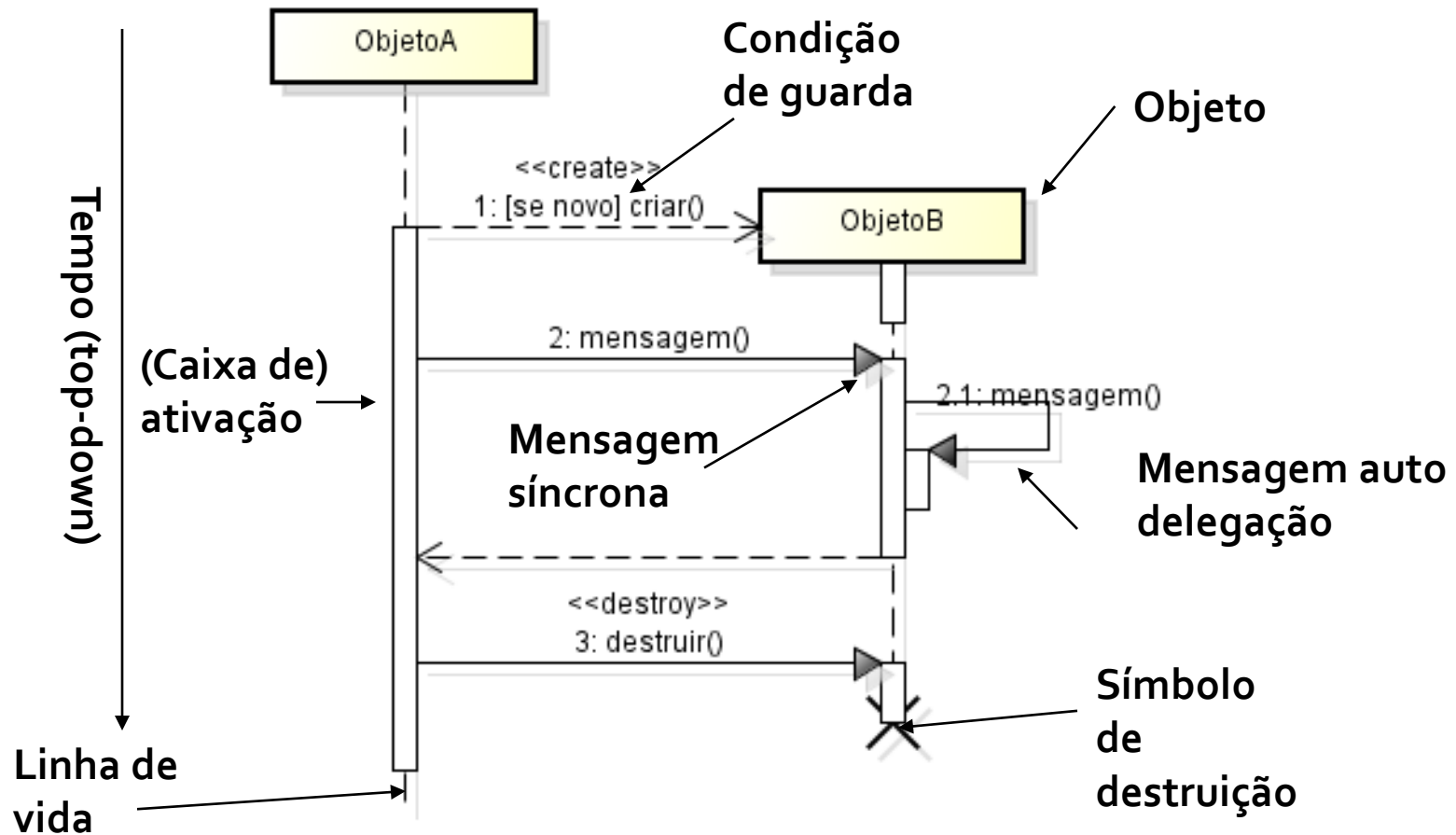


Diagrama de sequência



Observações

- O diagrama deve **refletir a arquitetura** (padrão) utilizado e, portanto, é comum que **além de atores e objetos** (classes), **sejam representados também**:
 - Camadas (ex.: camada de controle do padrão MVC);
 - Dispositivos físicos (sensores e dispositivos de entrada e saída);
 - Componentes de interface (como páginas html ou jsp por exemplo);
 - Componentes externos ao sistema (como SGDB, por exemplo).

Outros Elementos do Diagrama

- Objeto de fronteira (*boundary object*)
 - Estereótipo de objeto que representa uma fronteira do sistema, normalmente uma interface com o usuário.
- Objeto de controle (*control object*)
 - Estereótipo de objeto que faz a mediação entre os limites (boundaries) e as entidades (dados), implementando a lógica de controle para gerenciar os diversos elementos e interações;
 - Os controles podem ser implementados através de componentes mais simples do que um objeto.
- Objeto entidade (*entity object*)
 - Estereótipo de objeto que representa um mecanismo de armazenamento ou persistência que captura a informação ou conhecimento.

Notação



Boundary
object

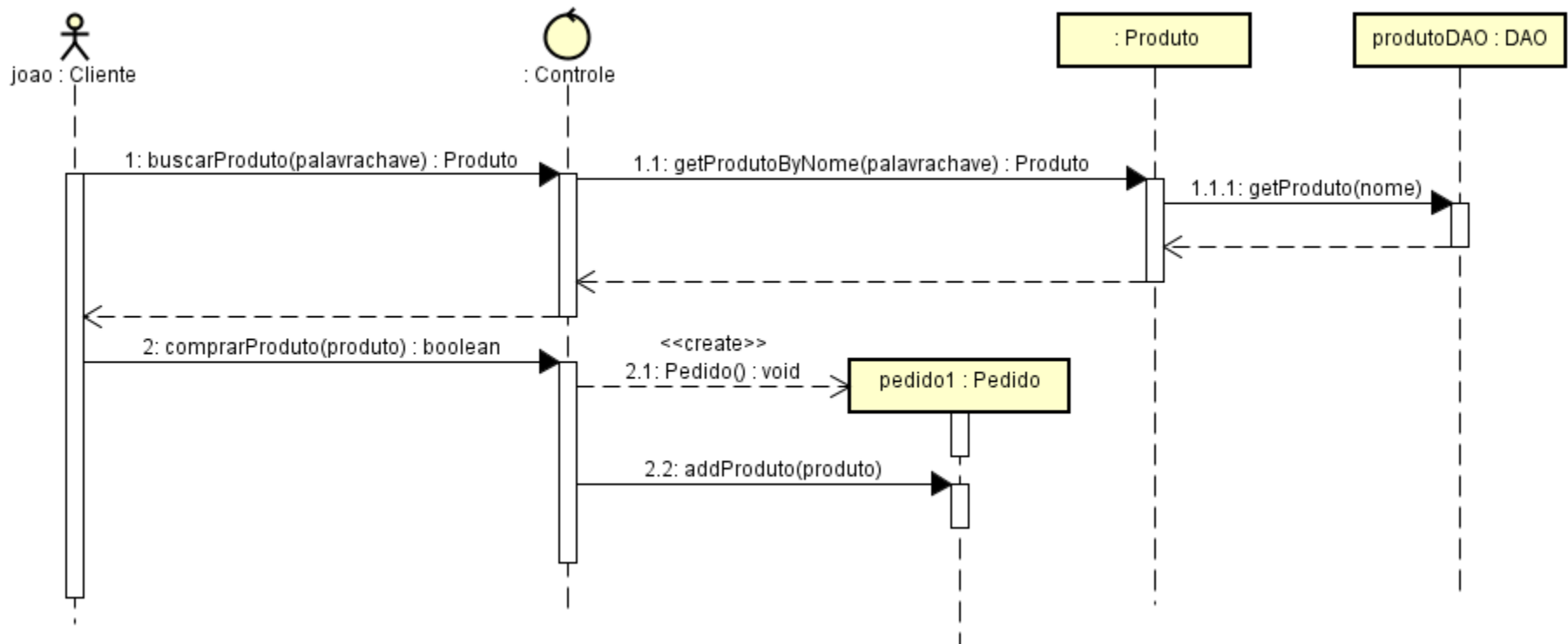


Entity
object



Control
object

Arquitetura Refletida no Diagrama de Sequência - MVC

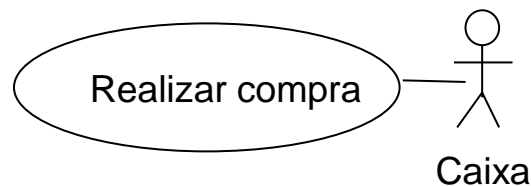


Passos para criação de um diagrama de sequência

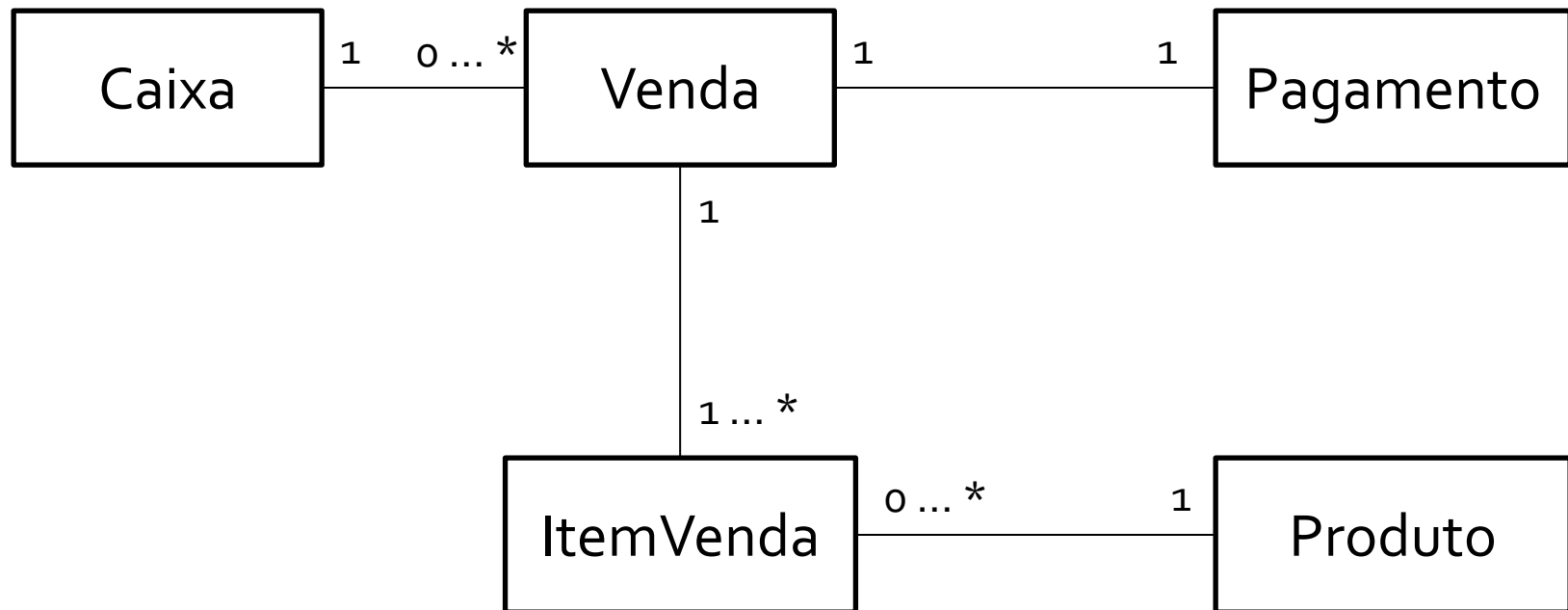
- Escolher um caso de uso.
- Identificar os objetos e outros componentes que fazem parte da interação.
- Identificar quem inicia a interação.
- Identificar as mensagens trocadas entre os objetos.
- Identificar a sequência destas mensagens.

Exemplo Prático

- Caso de Uso resumido
 - Nome: Realizar compra
 - Objetivo: Permitir a aquisição de produtos pelo cliente.
 - Atores: Cliente e caixa
 - Fluxo principal:
 1. O cliente chega ao caixa com os itens escolhidos.
 2. O caixa registra os itens comprados.
 3. O sistema totaliza a venda.
 4. O caixa fecha a venda e recebe o pagamento.



Exemplo Prático



Ferramentas de Modelagem

- StarUML (avaliação por tempo indefinido)
 - <https://staruml.io/>
- Draw.io
 - <https://app.diagrams.net/>
 - Web

Referências

- Boock, G. and Rumbaugh, J. The Unified Modeling Language User Guide . Addison-Wesley, 1999
- Arlow, J. and Neustadt, I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Rumbaugh, J.; Jacobson, I. and Booch , G. The Unified Modeling Language Reference Manual, 2nd Edition, The Addison-Wesley Object Technology Series, 2004.
- Boock, G.; Rumbaugh, J. and Jacobson, I; Unified Modeling Language User Guide, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Jacobson, I; Boock, G. and Rumbaugh, J., Unified Software Development Process, Addison-Wesley, Janeiro 1999.
- Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design Prentice-Hall, New Jersey - USA, 1997



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

Perguntas?