



## Aula 09

---

# Análise de Sistemas Orientado à Objetos

## Prof. Me. Joseffe Barroso de Oliveira

UNIP - Universidade Paulista  
Análise e Desenvolvimento de Sistemas

# Diagrama de Sequência

- \* Preocupa-se com a ordem temporal em que as mensagens são trocadas
- \* Pode ser usado para detalhar um Caso de Uso
- \* Identifica:
  - \* O evento gerador da funcionalidade modelada (ator responsável pelo evento)
  - \* Os objetos envolvidos na ação

# Diagrama de Sequência Estrutura/Composição

- \* O diagrama de Atividades é representado por:

- \* Atores
- \* Objetos
- \* Linhas de Vida
- \* Ativações do Objeto
- \* Mensagens

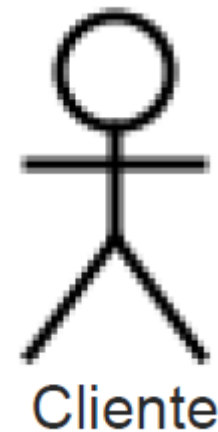
# Diagrama de Sequência

## “O que representam?”

- \* Mostram a sequência em que os eventos ocorrem em um determinado processo
  - \* Quais condições devem ser satisfeitas
  - \* Quais métodos devem ser disparados
  - \* Em que ordem os métodos são disparados
- \* Diagrama de Sequência **não** representa atributos

# Diagrama de Sequência “Atores”

- \* Exatamente os mesmos descritos no Diagrama de Casos de Uso
- \* Entidade externas que:
  - \* Interagem com o sistema
  - \* Solicitam serviços



# Diagrama de Sequência “Objetos”

- \* Indicam instâncias de uma classe envolvidas no processo
  - \* As classes são mostradas em Diagramas de Classes
- \* Representados por retângulos
  - \* Nome do objeto (inicial minúsculo)
  - \* Nome da classe (inicial maiúscula)
  - \* Separação por dois pontos (:)



minhaConta: Conta

A rectangular box containing the text 'minhaConta: Conta'. The text is split into two parts by a colon. 'minhaConta' is in a dark blue font, and 'Conta' is in a black font.

# Diagrama de Sequência

## “Linha de Vida”

- \* Linha vertical tracejada abaixo do objeto
  - \* Partem do retângulo que representa o objeto
- \* Representa o tempo em que um objeto existe durante o processo

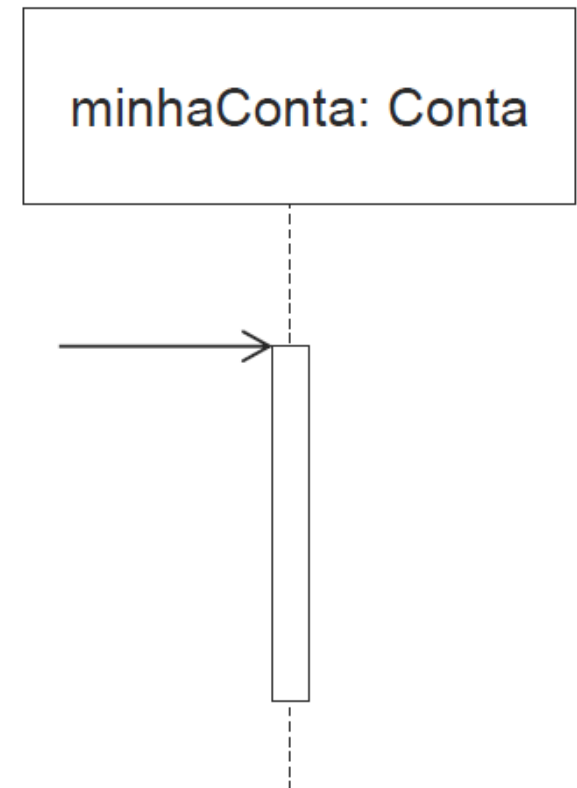


```
classDiagram
    class minhaConta["minhaConta: Conta"]
    minhaConta --> 
```

minhaConta: Conta

# Diagrama de Sequência “Ativação do Objeto”

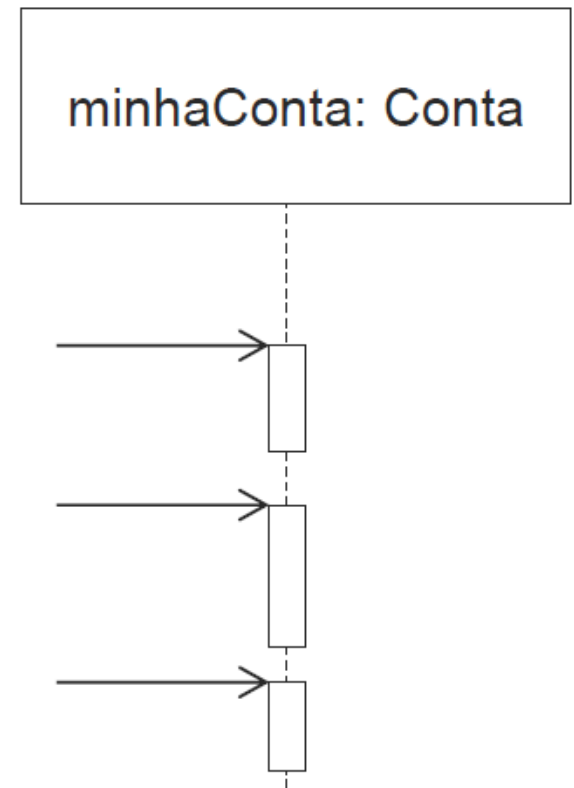
- \* O objeto é ativado quando este recebe um estímulo
  - \* Exemplo: recebimento de uma mensagem
- \* Um retângulo magro indica o período em que o objeto está participando ativamente do processo





# Diagrama de Sequência “Ativação do Objeto”

- \* Podem haver vários períodos em que o objeto se encontra ativo.



# Diagrama de Sequência “Mensagens”

- \* Representam a comunicação entre objetos e/ou atores do Diagrama de Sequência
- \* Exemplos de mensagens
  - \* Chamadas de um método de um objeto por outro objeto
  - \* Comunicação entre dois atores

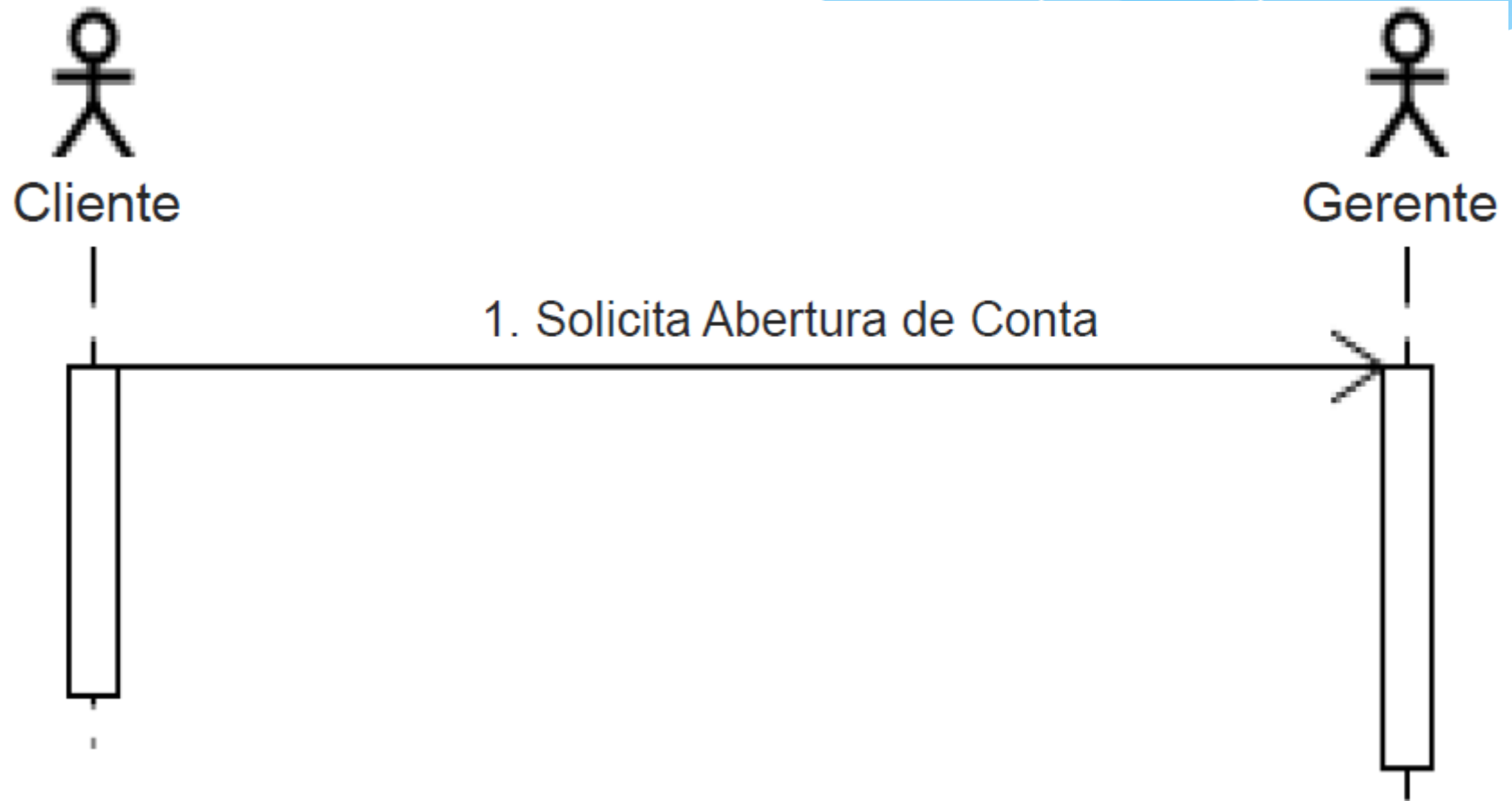
# Diagrama de Sequência

## “Tipos de Mensagens”

- \* Ator para Ator
- \* Ator para Objeto
- \* Objeto para Objeto
- \* Objeto para Ator

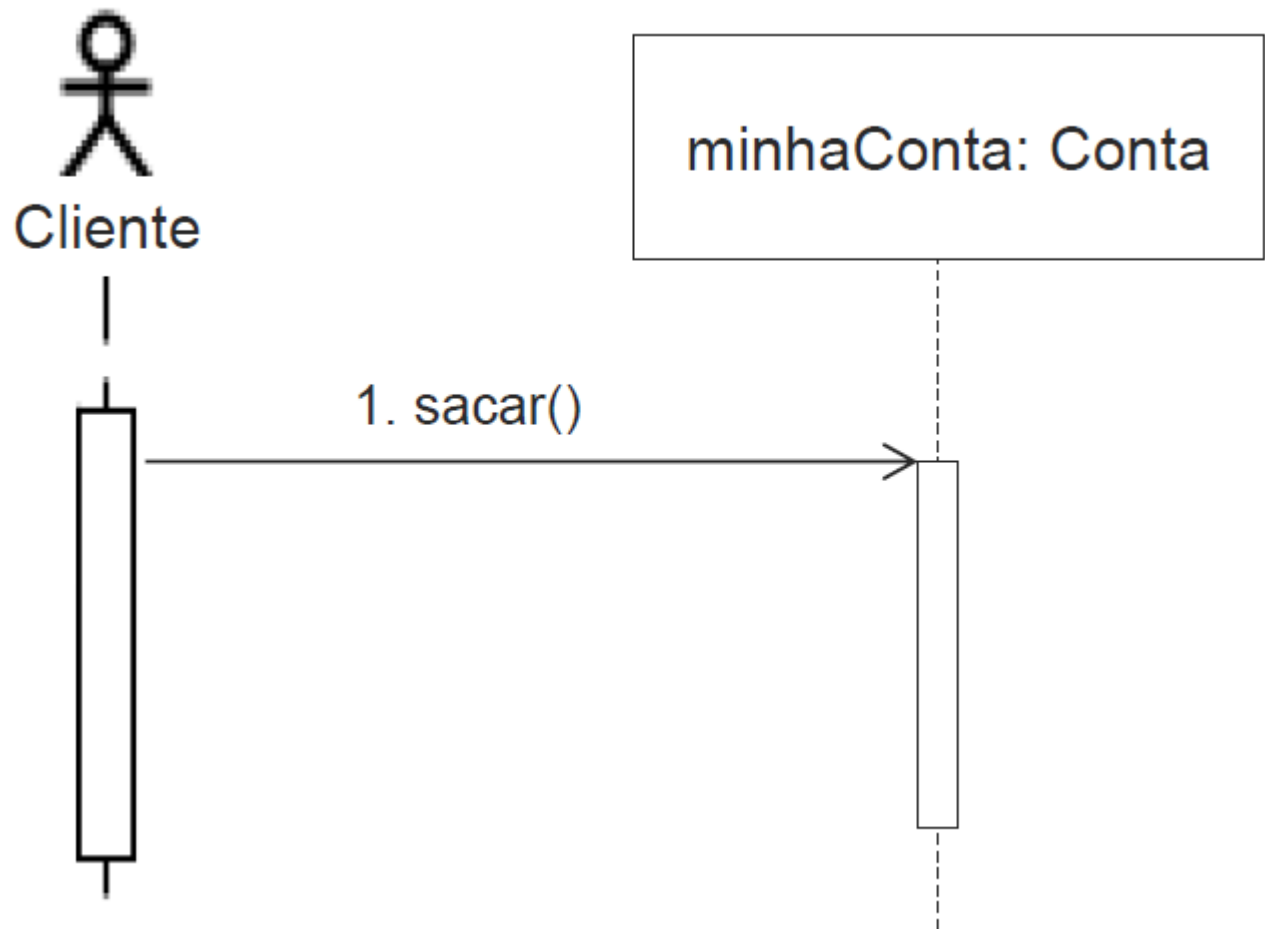
# Diagrama de Sequência

## “Mensagens: Ator para Ator”



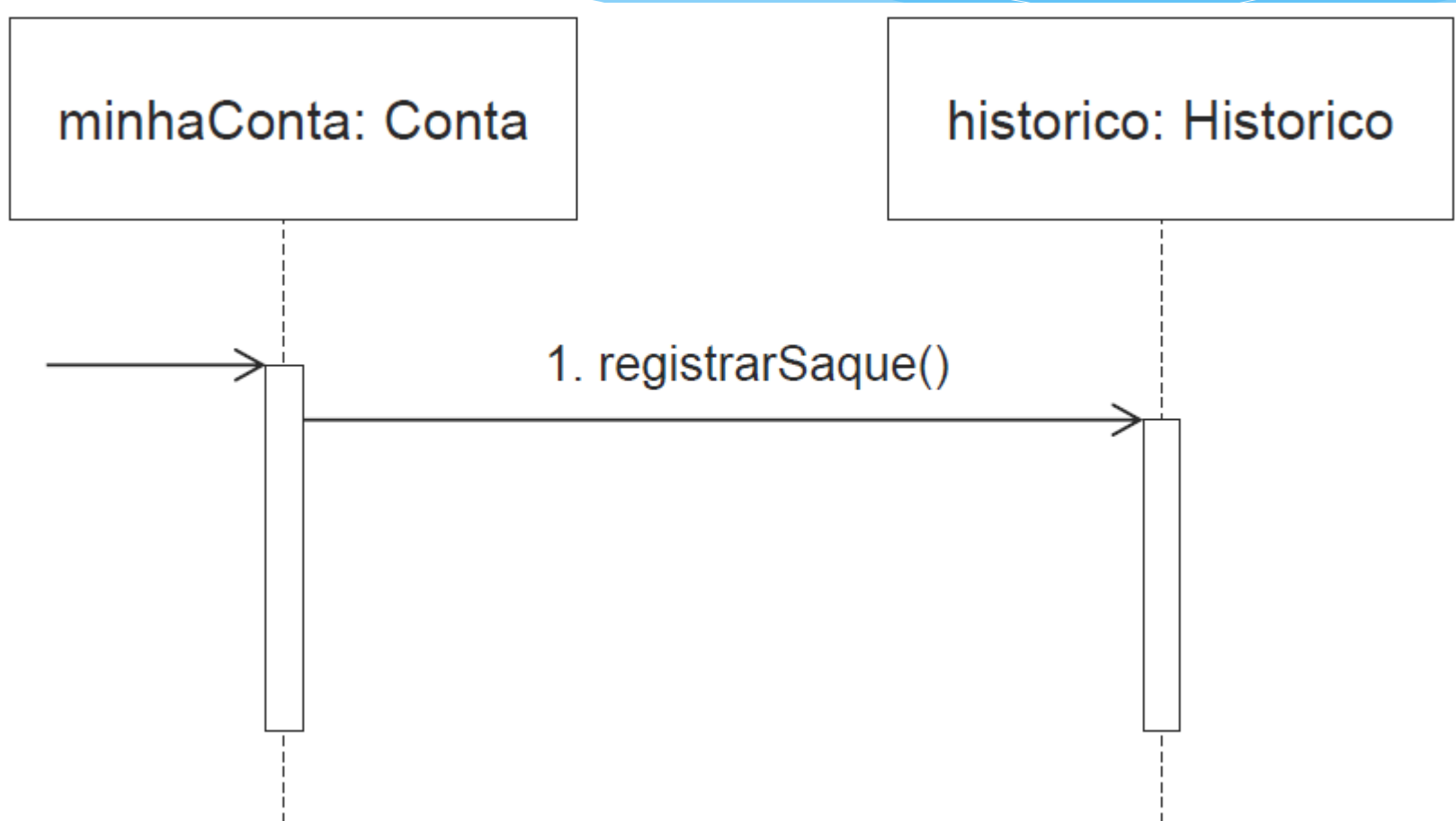
# Diagrama de Sequência

## “Mensagens: Ator para Objeto”



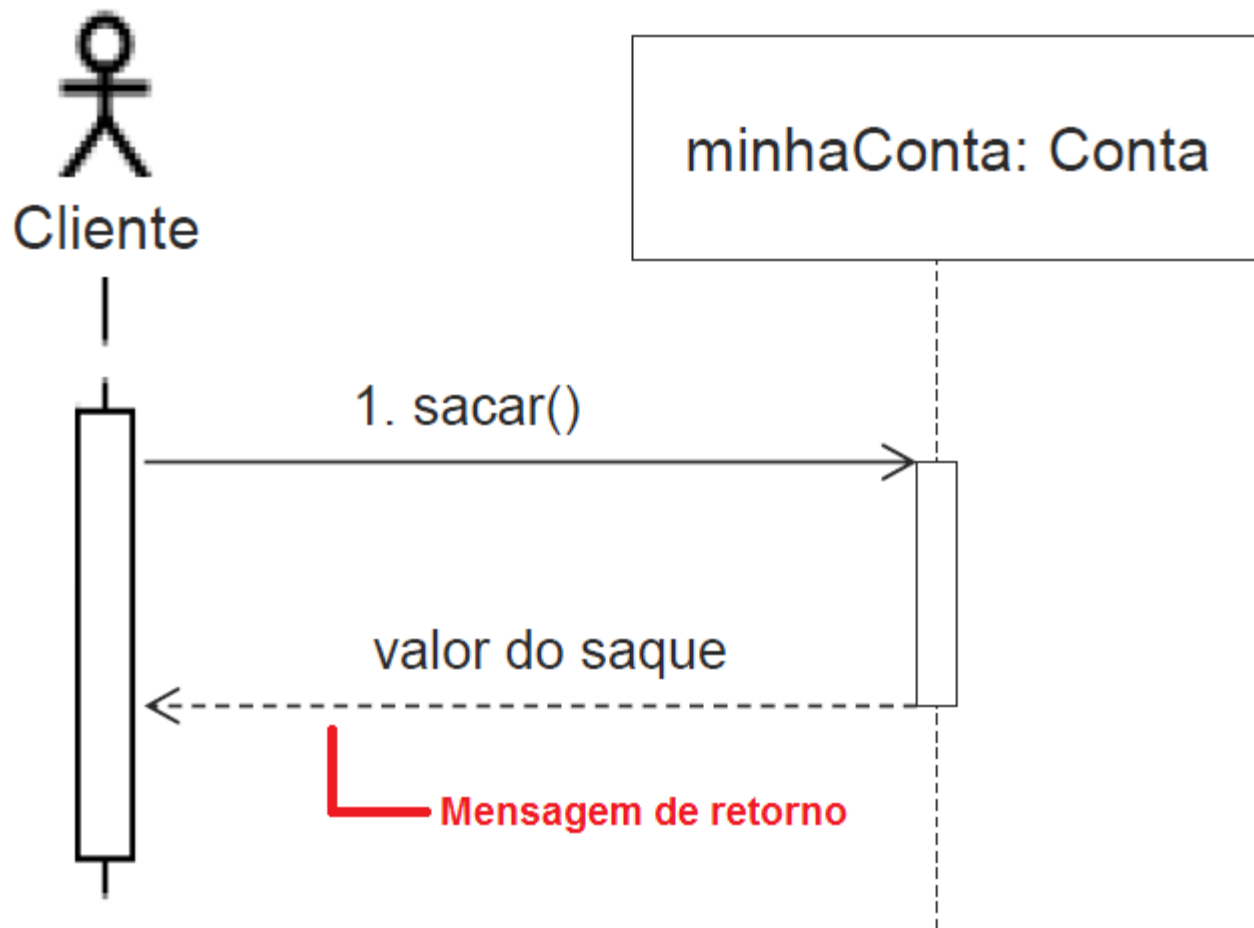
# Diagrama de Sequência

## “Mensagens: Objeto para Objeto”



# Diagrama de Sequência

## “Mensagens: Objeto para Ator”



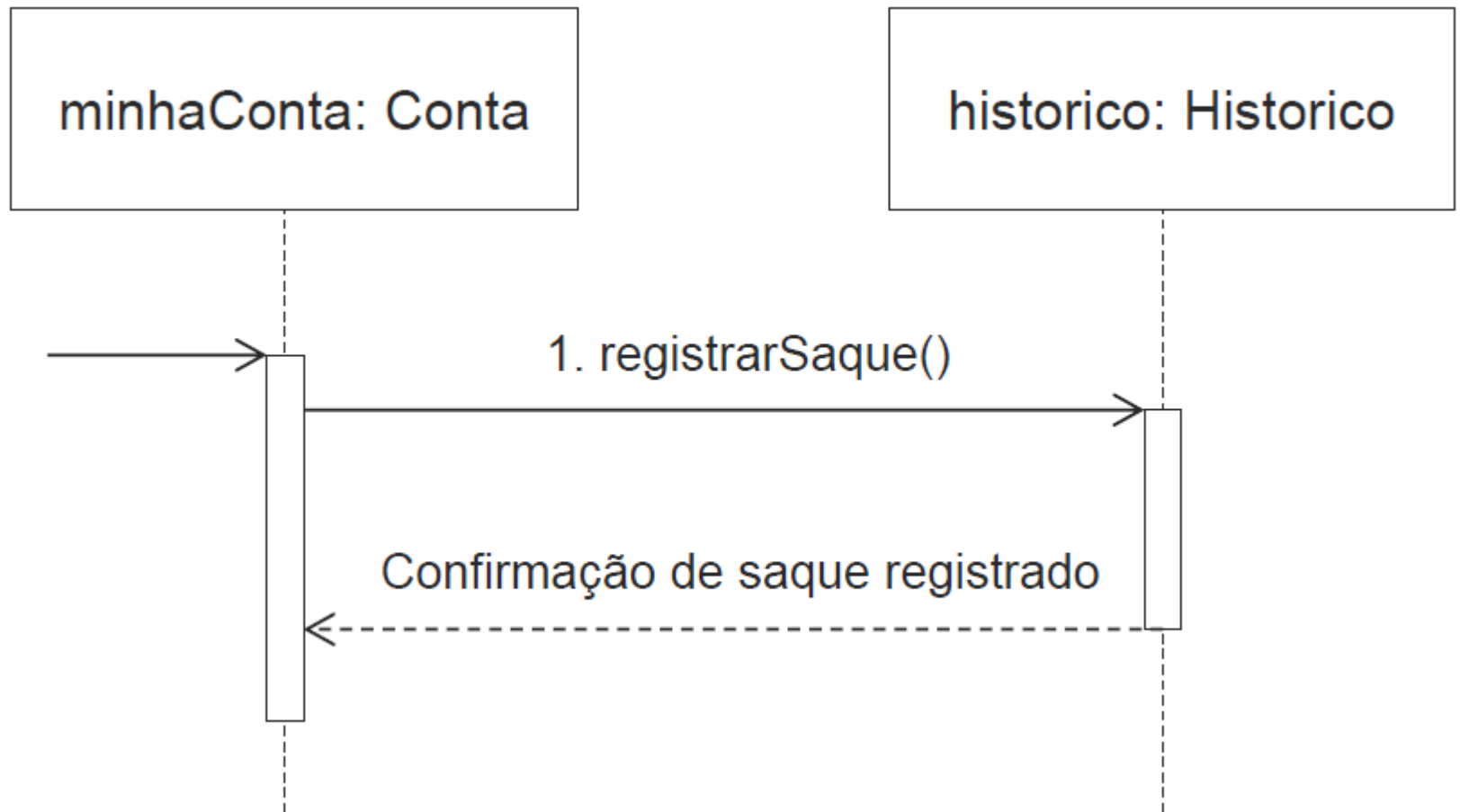
# Diagrama de Sequência

## “Mensagens de Retorno”

- \* Além de resposta ao ator, mensagens de retorno podem indicar respostas para objetos
  - \* Pode retornar informações específicas do método chamado
- \* Mensagens de retorno são opcionais em Diagramas de Sequência



# Diagrama de Sequência “Mensagens de Retorno”

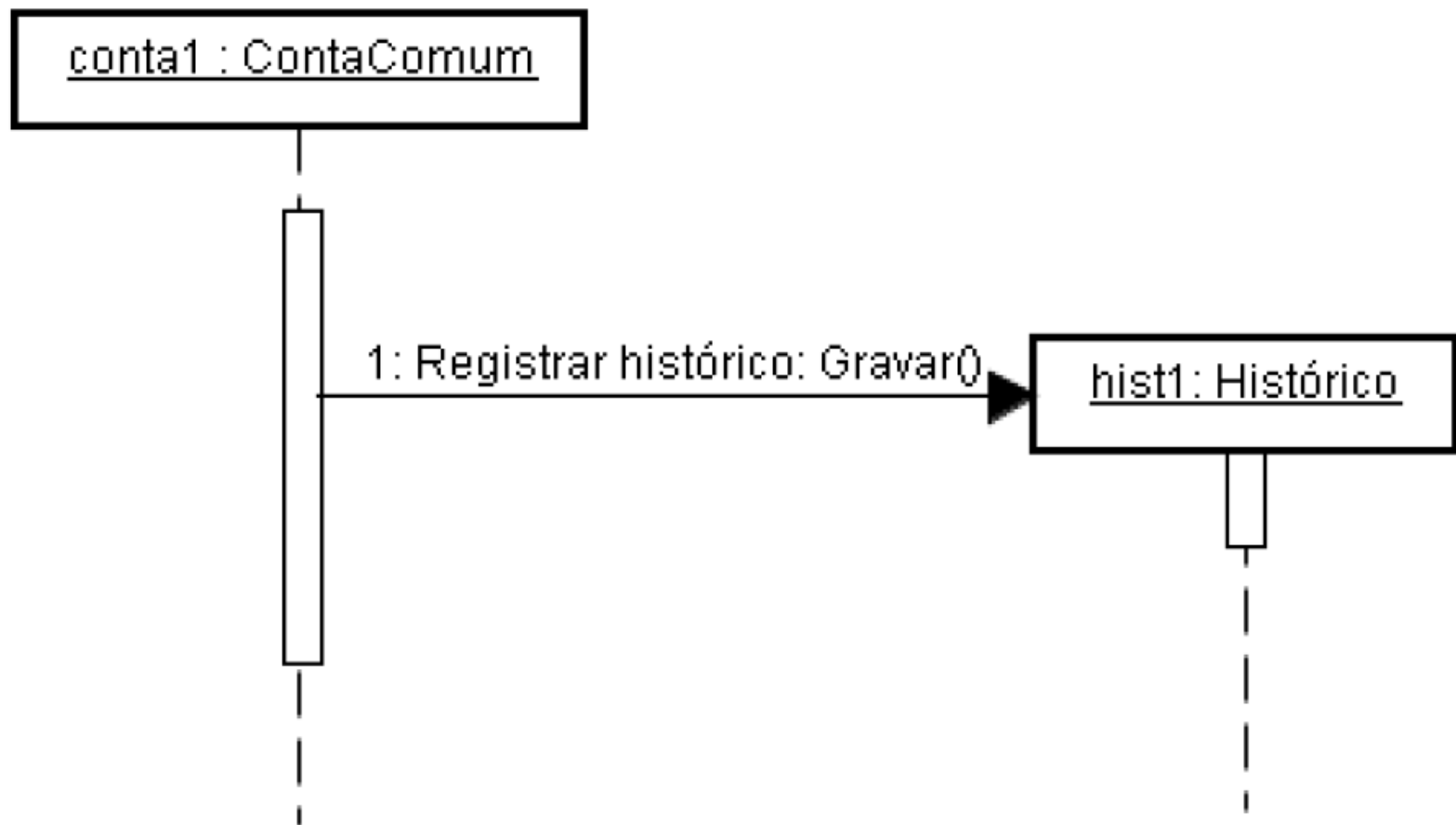


# Diagrama de Sequência

## “Instanciação (Criação) de um Objeto”

- \* A seta atinge o retângulo que representa o objeto
  - \* O objeto passa a existir a partir daquele momento
- \* A mensagem representa a chamada do método construtor

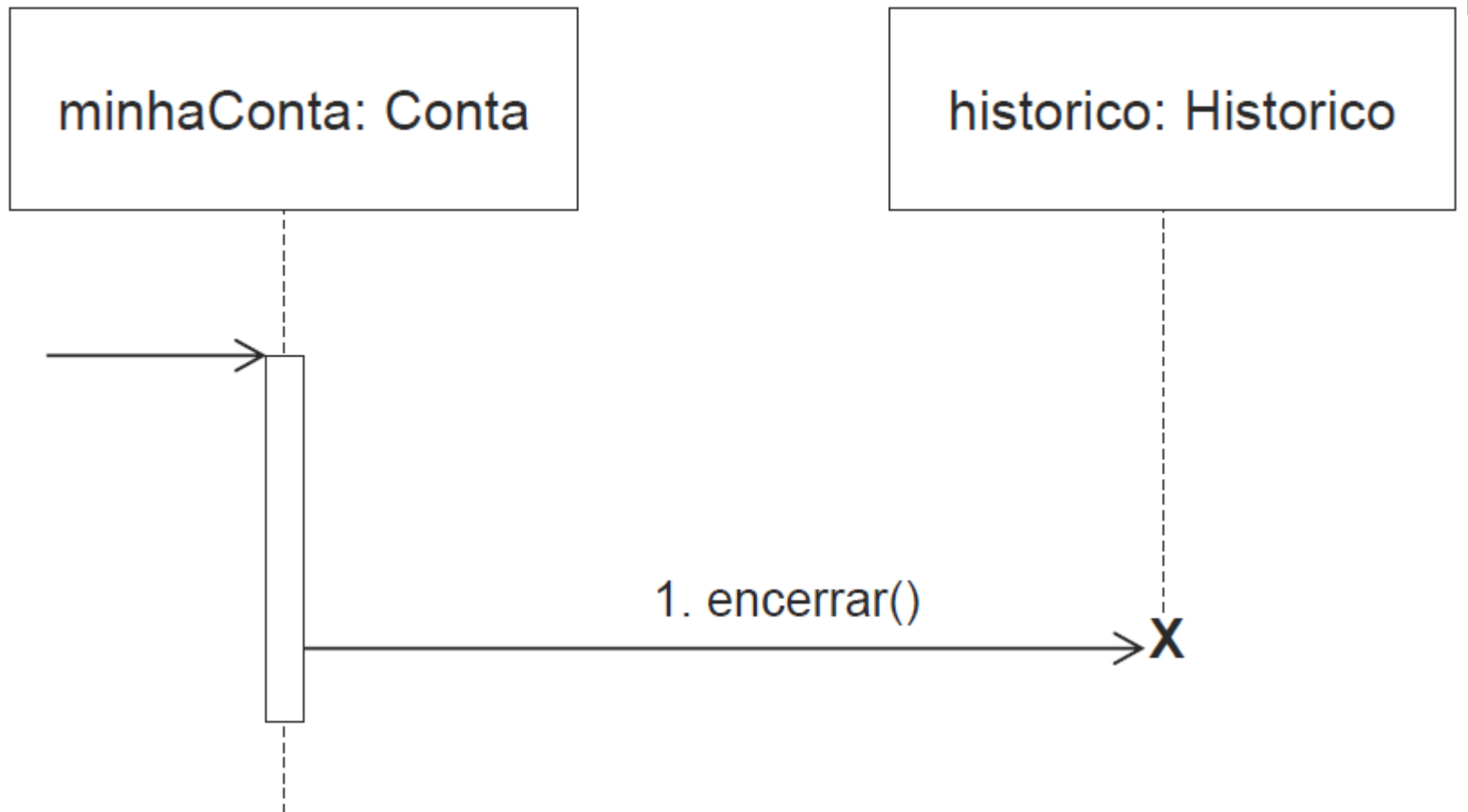
# Diagrama de Sequência “Instanciação (Criação) de um Objeto”



# Diagrama de Sequência “Destruição de um Objeto”

- \* A seta atinge o fim da linha da vida de um objeto
  - \* Um X marca a destruição do objeto
  - \* O objeto deixa de existir a partir daquele momento
- \* A mensagem representa a chamada do método destruidor

# Diagrama de Sequência “Destruição de um Objeto”

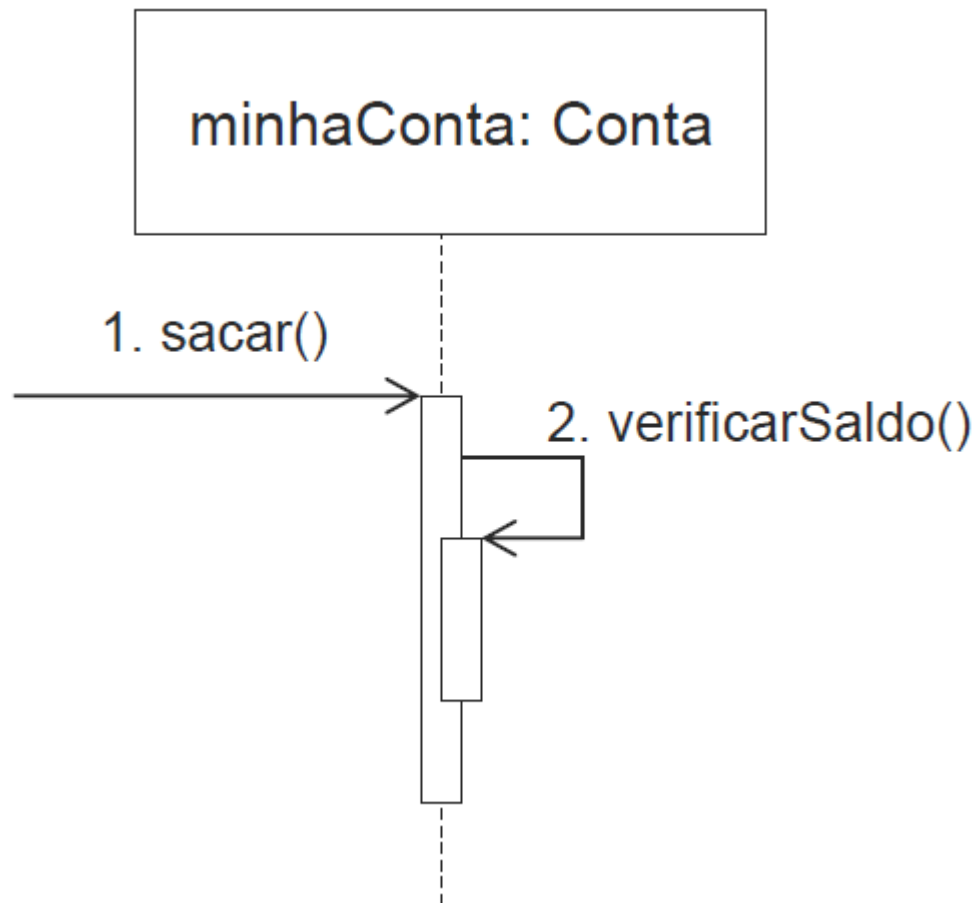


# Diagrama de Sequência

## “Auto Chamadas”

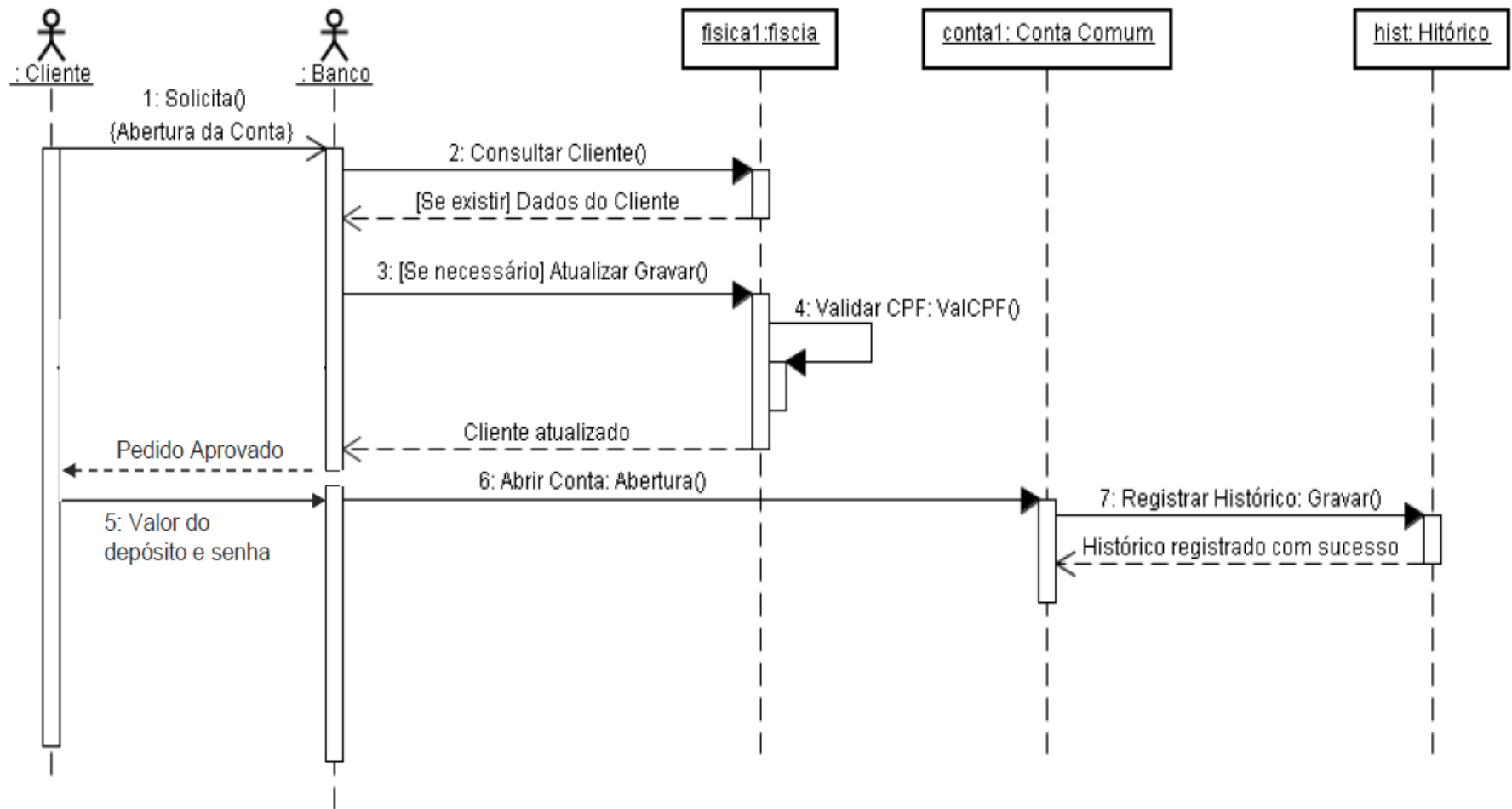
- \* Mensagens que um objeto envia para si mesmo
  - \* A mensagem parte do objeto e atinge o próprio objeto
  - \* Utilizado para indicar que o objeto precisa executar algumas operações relacionadas ao serviço solicitado
  - \* Ou seja, o objeto está chamando os próprios métodos dele

# Diagrama de Sequência “Auto Chamadas”



# Diagrama de Sequência

## “Exemplo – Parte de um Sistema de um Banco”





# Diagrama de Sequência

## “Diagrama de Caso de Uso x Diagrama de Sequência”

- \* Caso de Uso é um processo disparado pelo usuário
- \* O Diagrama de Sequência pode detalhar um caso de uso e mostrar
  - \* A ordem em que os eventos acontecem
  - \* As mensagens que são enviadas
  - \* Os métodos que são chamados
  - \* Como os objetos interagem entre si

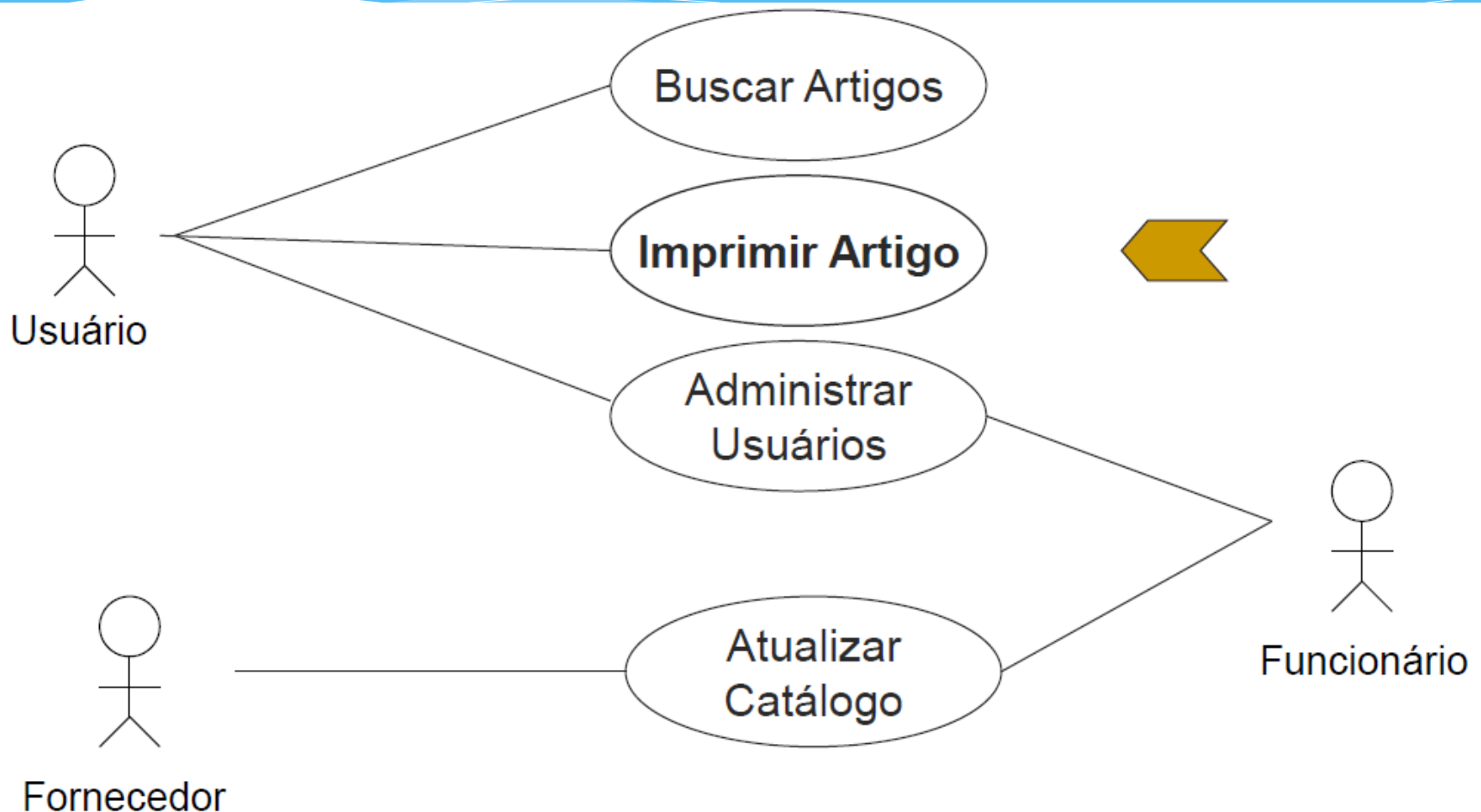
# Diagrama de Sequência

## “Diagrama de Caso de Uso x Diagrama de Sequência”

- \* Um Diagrama de Casos de Uso pode gerar vários Diagramas de Sequência
- \* Nem sempre um Caso de Uso gera um Diagrama de Sequência
- \* Diagramas de Sequência são comuns quando há relacionamentos do tipo <<include>> ou <<extend>>

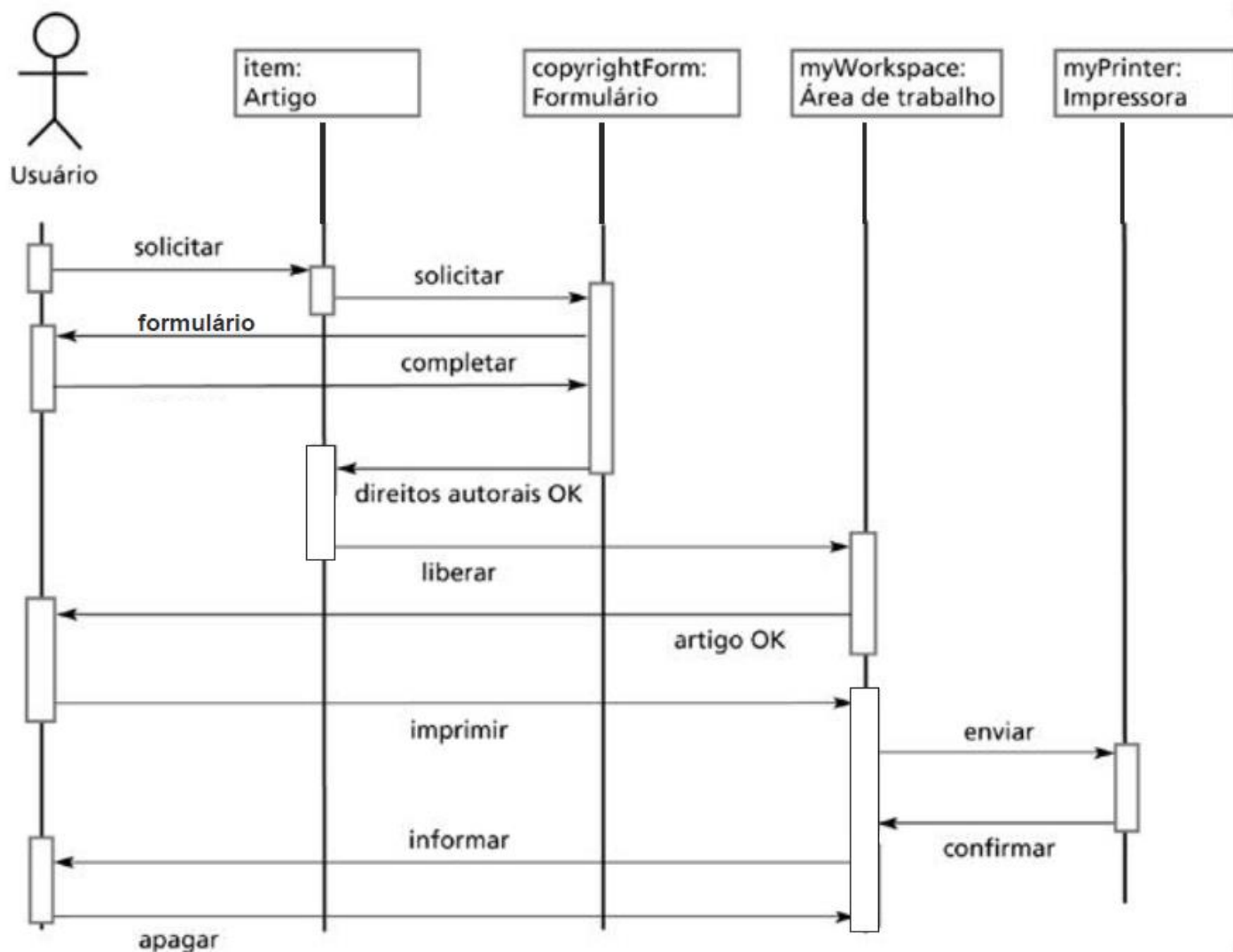
# Diagrama de Sequência

“Exemplo 01 – Diagrama de Caso de Uso de uma Biblioteca”



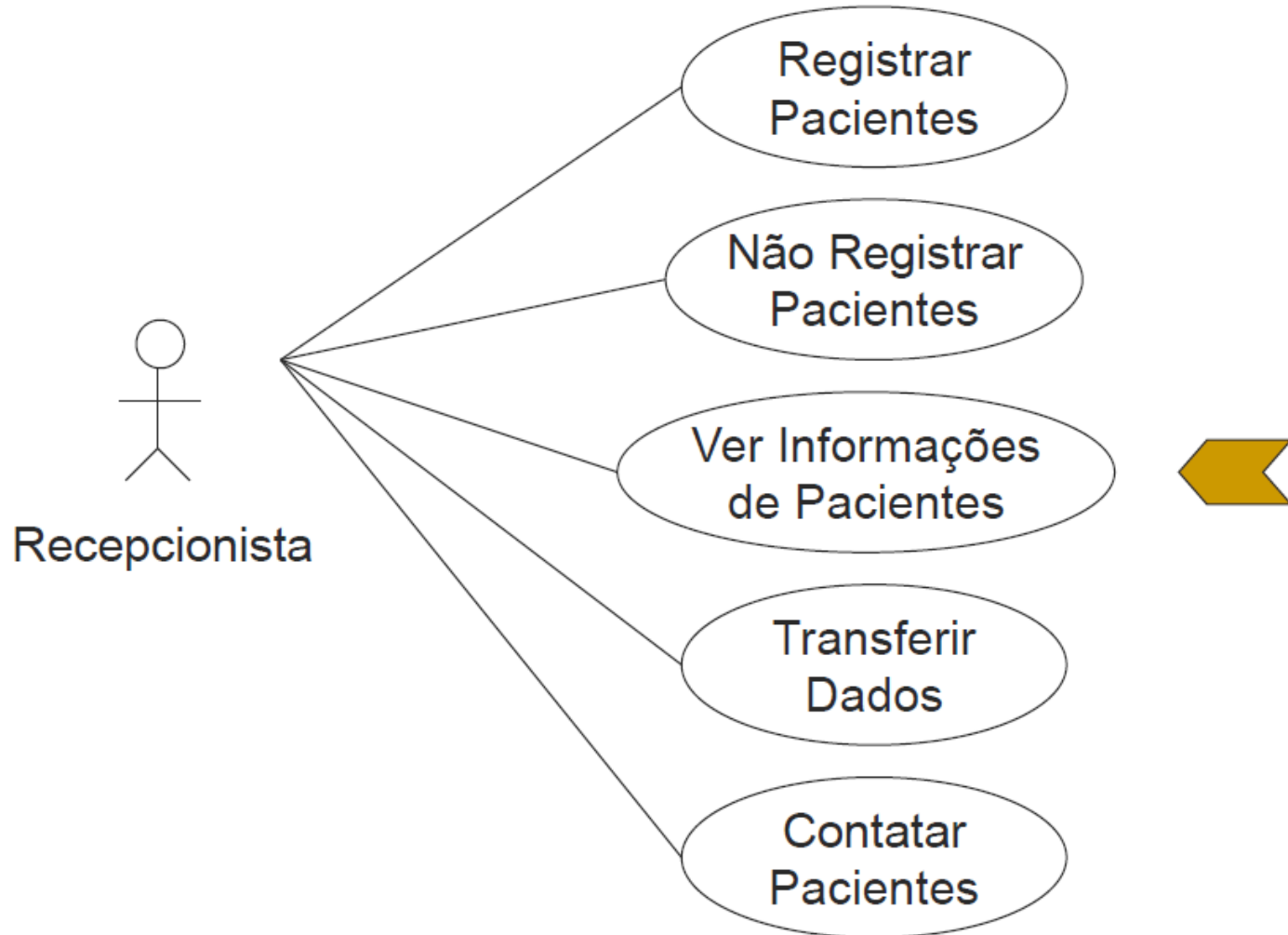
# Diagrama de Sequência

“Exemplo 01 – Diagrama de Sequência sobre o Caso de Uso ‘Imprimir Artigo’ ”



# Diagrama de Sequência

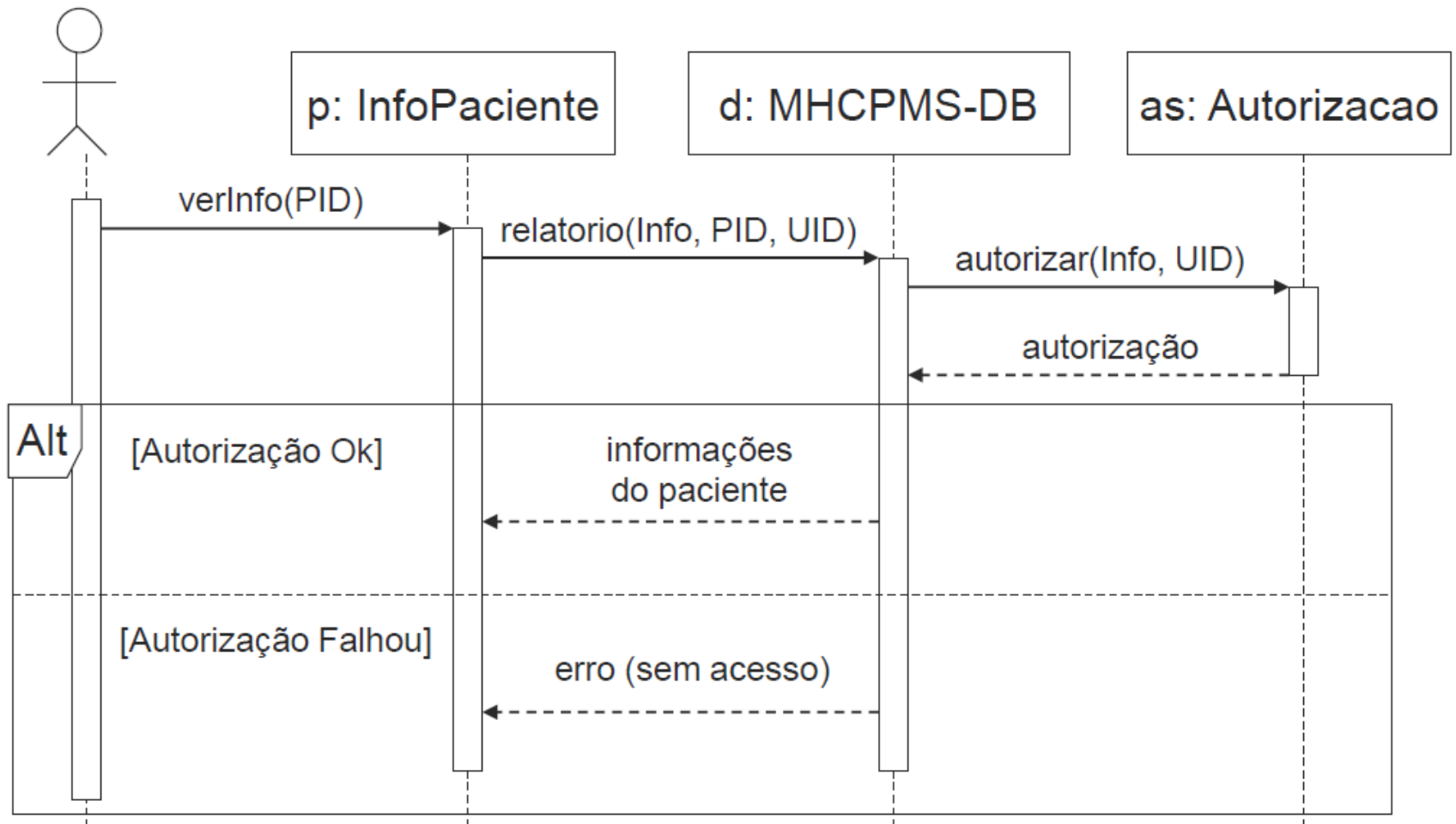
“Exemplo 02 – Diagrama de Caso de Uso de um Consultório”



# Diagrama de Sequência

“Exemplo 02 – Diagrama de Sequência sobre o Caso de Uso ‘Ver Informações do Paciente’ ”

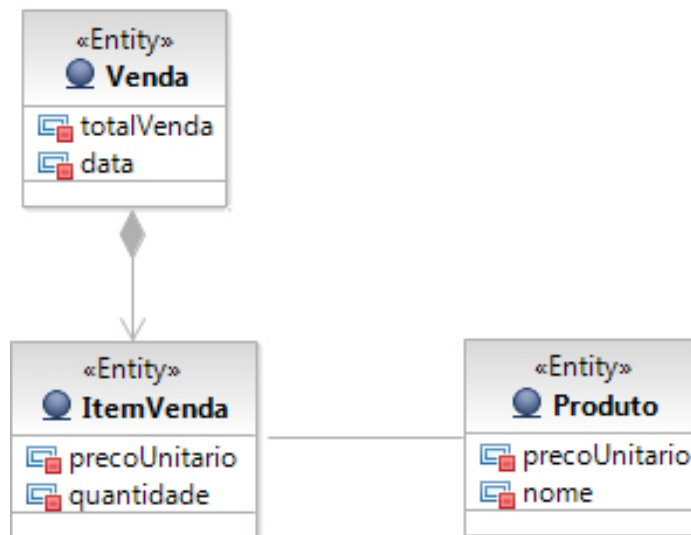
Recepcionista



# Diagrama de Sequência

## Esteriótipos

- \* É um elemento de modelagem que rotula tipos de Classes de Objeto. Uma Classe de Objetos pode ter um ou mais tipos de estereótipos. A notação usada pela UML para Estereótipos, dentro da representação gráfica da Classe de Objeto, é colocá-lo entre << >> na área reservada para o nome da classe e acima deste.



# Diagrama de Sequência

## Esteriótipos

- \* Os principais estereótipos que temos, são:
- \* Boundary – Classes relativas a Tela / Front end;
- \* Control – Classes de Negócio. Controlam objetos entre a Boundary e Entity;
- \* Entity – Classes responsáveis por persistir os dados dos objetos em Base de Dados.

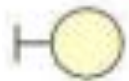


# Diagrama de Sequência

## Esteriótipos - Simbologia

- Fronteira (*boundary*)

- Classes de interface com o mundo externo



(ex: GUI, sistemas externos)

- Controle (*control*)

- Coordenam o comportamento do caso de uso definindo uma interface entre classes fronteira e entidade



- Entidade (*entity*)

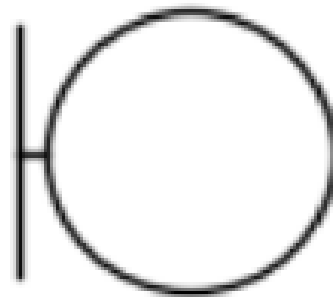
- Classes que armazenam informações manipuladas pelo sistema



# Diagrama de Sequência

## Esteriótipos – Boundary (Classe Fronteira)

É uma classe que modela a comunicação entre a vizinhança do sistema e suas operações internas. Exemplos: Interface tipo Janela, Protocolo de Comunicação, Interface de Impressão, Sensores, etc. No presente estudo de caso, sistema automatizado de Matrícula num Curso, as classes de objeto Formulário em Branco e Sistema de Cobrança são exemplos de estereótipos desta classe.

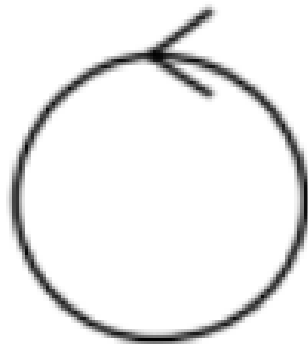


Boundary

# Diagrama de Sequência

## Esteriótipos – Control (Classe Controle)

É uma classe que modela o comportamento de controle específico para uma ou mais “Use Case”. Suas principais características são: Cria, ativa e anula objetos controlados; Controla a operação de objetos controlados; Controla a concorrência de pedidos de objetos controlados; Em muitos casos corresponde a implementação de um objeto intangível.

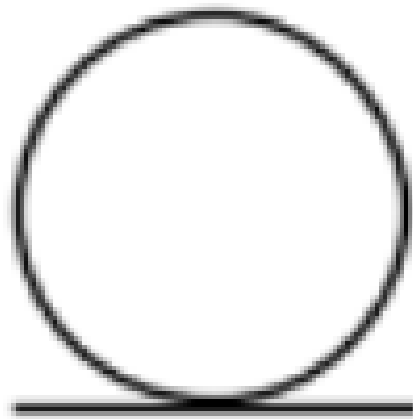


Control

# Diagrama de Sequência

## Esteriótipos – Entity (Classe Entidade)

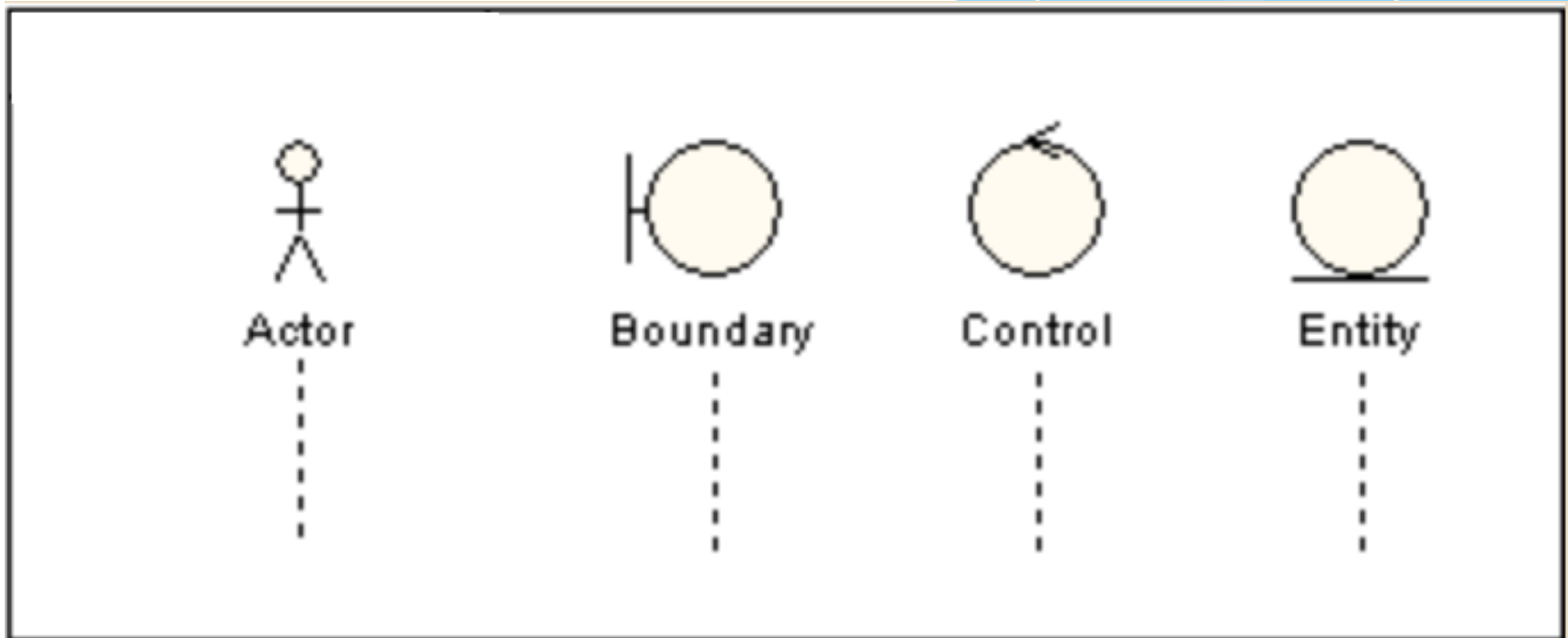
É uma classe que modela objetos cuja informação e o comportamento associados são, de maneira geral, persistentes.



Entity

# Diagrama de Sequência

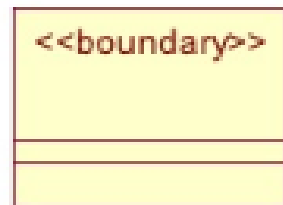
## Esteriótipos – Boundary, Control e Entity



# Diagrama de Sequência

## MVC em UML – Diagrama de Classe e Esteriótipos

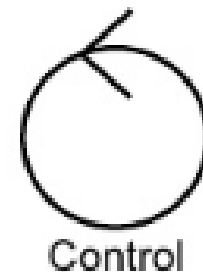
### MVC em UML



=



=

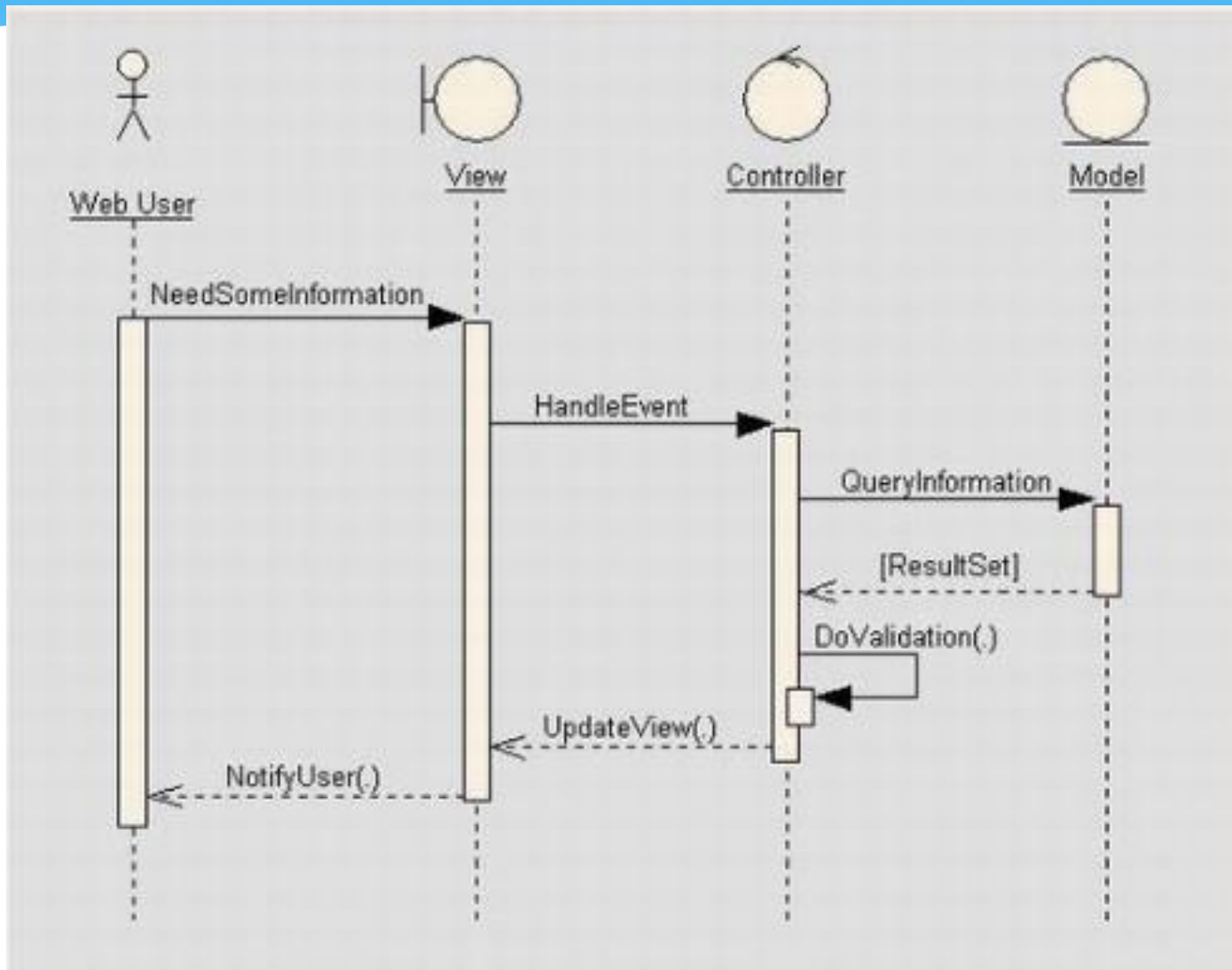


=



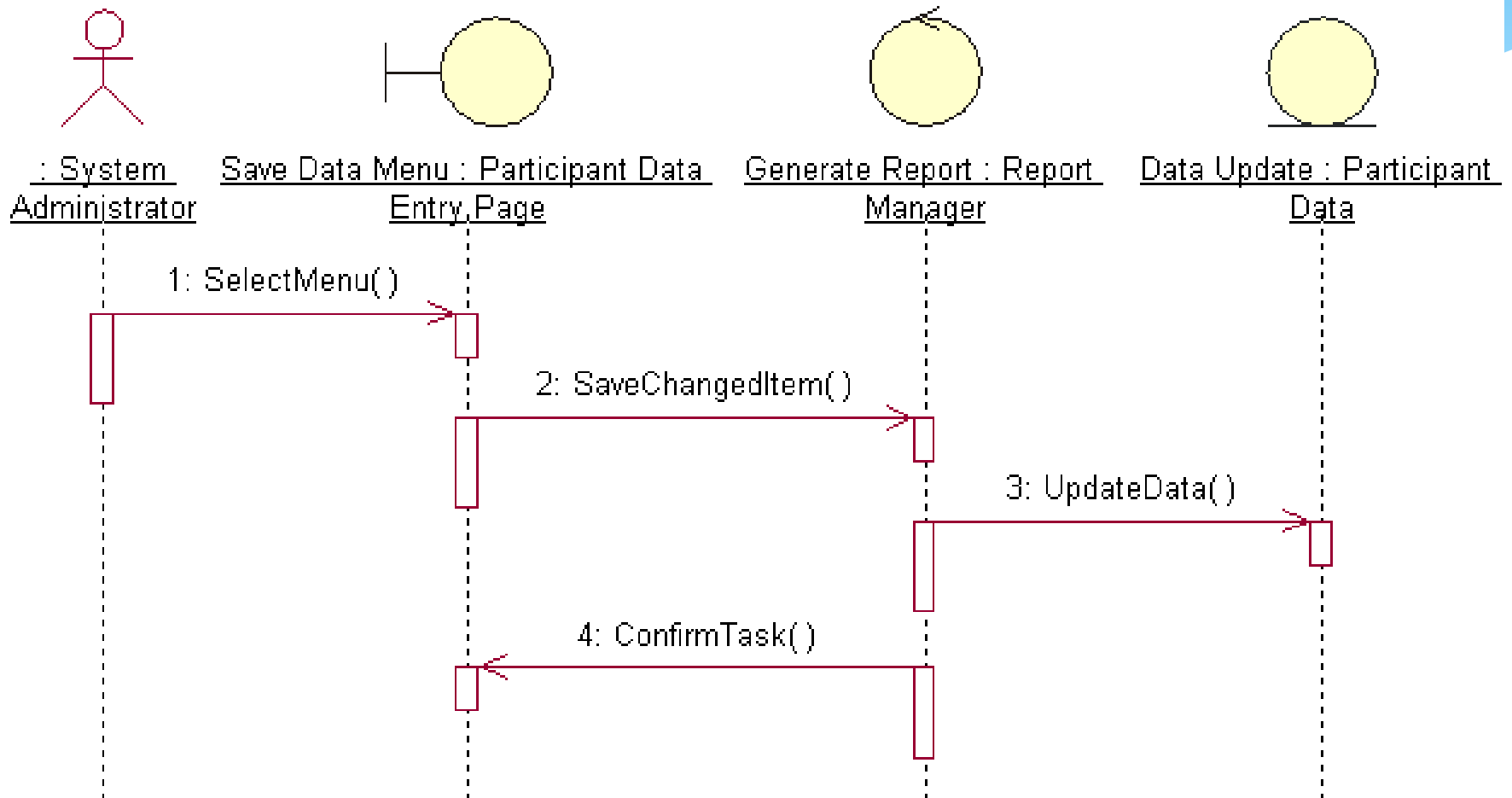
# Diagrama de Sequência

## MVC em UML – Diagrama de Classe e Esteriótipos



# Diagrama de Sequência

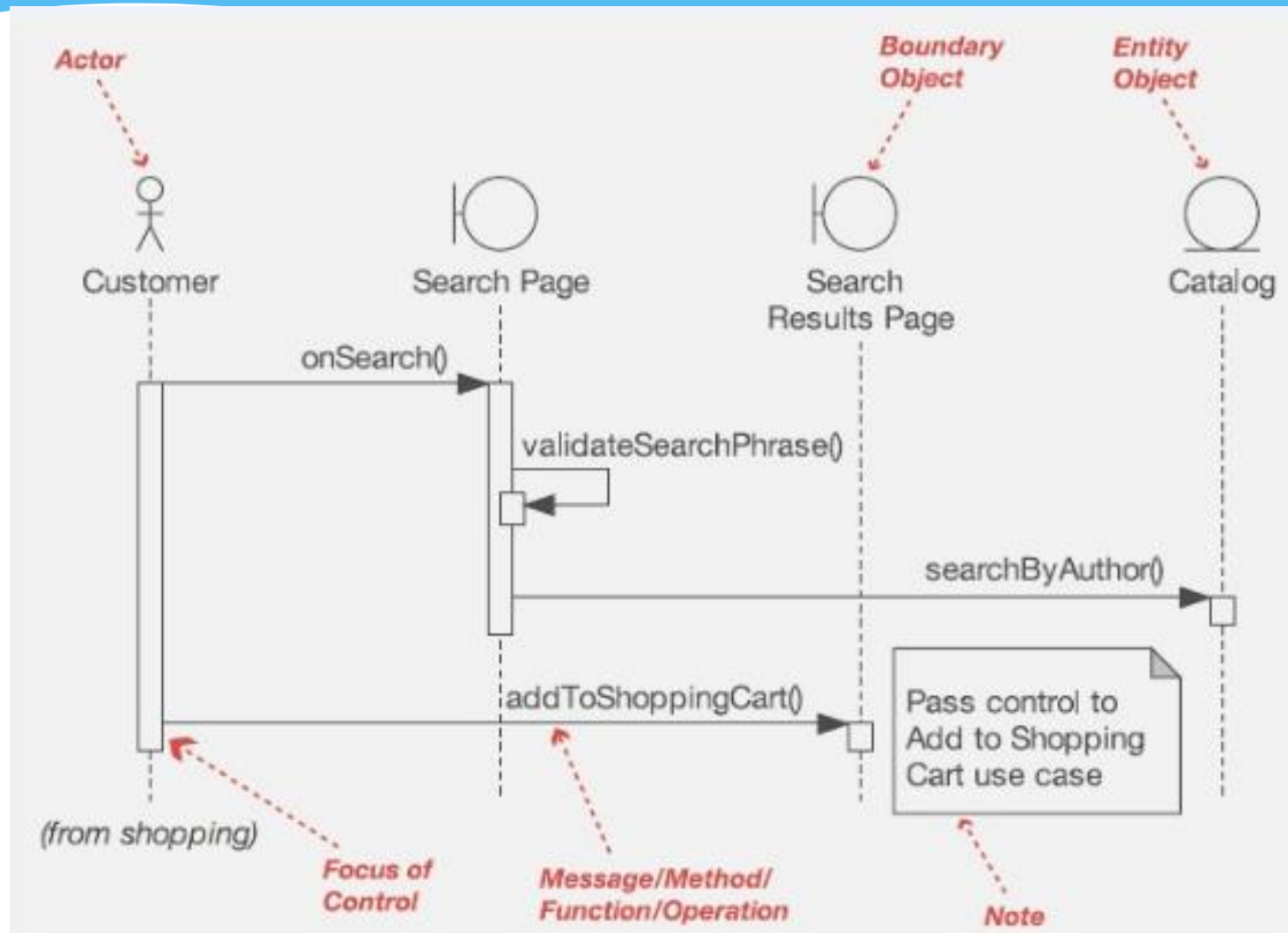
“Exemplo 03 – Diagrama de Sequência utilizando Esteríotipos Boundary, Control e Entity”





# Diagrama de Sequência

“Exemplo 04 – Diagrama de Sequência utilizando Esteríótipos Boundary, Control e Entity ”



# Diagrama de Sequência

## Exercícios

1. Elabore o Diagrama de Sequência para alguns Casos de Uso de um Sistema para Biblioteca Escolar;
2. Elabore o Diagrama de Sequência para alguns Casos de Uso de um Sistema de Consultório Médico;
3. Elabore o Diagrama de Sequência para alguns Casos de Uso de um Sistema de Pizzaria.