



# Engenharia de Software I

**Prof<sup>a</sup>. Me. Cynara Leão Garcia**

**[cynara.garcia@unicesumar.edu.br](mailto:cynara.garcia@unicesumar.edu.br)**



# Diagrama de Classes

## Exercício

# Resolução Exercício Teórico

Uma estação ferroviária é composta por 1 ou mais linhas ferroviárias. Em uma linha ferroviária podem estar estacionados diversos recursos ferroviários. Recursos ferroviários são vagões, locomotivas ou trens. Um trem é formado por vagões e locomotivas. Podem existir vagões e locomotivas estacionadas em uma linha sem estarem na formação de um trem. Uma estação ferroviária tem uma sigla e uma descrição (que não precisa obrigatoriamente possuir um valor). Uma linha ferroviária tem um número (que a diferencia de outra linha dentro da mesma estação), uma extensão em metros e uma descrição (que não precisa obrigatoriamente ter um valor). Um vagão é descrito por um número de série, tipo, capacidade de carga (valor default igual a 3000 ton), comprimento entre esteiras e comprimento dos engates (um único valor correspondendo aos dois lados). Uma locomotiva é descrita por um número de série, capacidade de tração e comprimento. Um trem é descrito por um prefixo (ex: NAG1010) e data/hora de formação. Um trem é formado em uma estação ferroviária de origem e tem como destino, uma outra estação ferroviária, ou seja, a estação de origem não pode ser igual à estação de destino. Todo trem é formado por pelo menos uma locomotiva e um vagão. Um trem não pode ter mais do que 150 recursos (vagões e locomotivas).

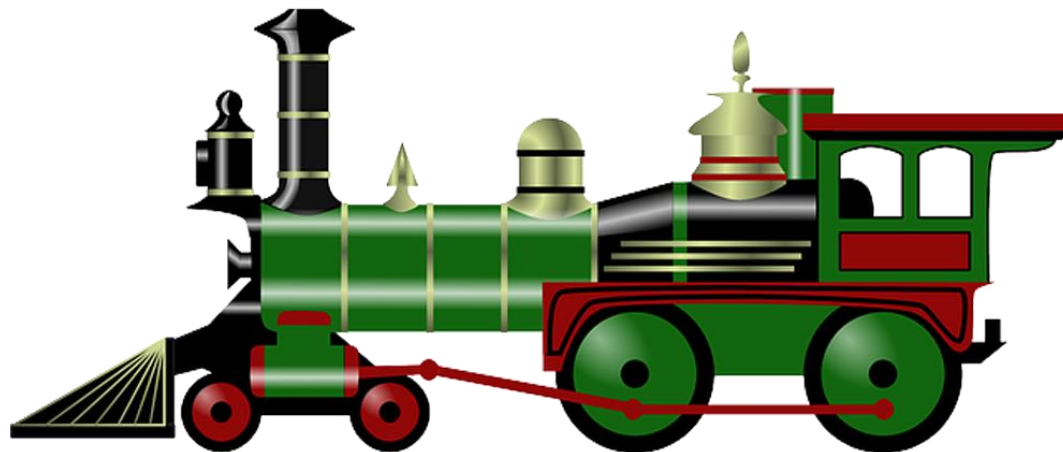
Elabore um diagrama de classes (UML) ilustrando os conceitos (classes, atributos e relacionamentos) e restrições mencionados no texto. Não esqueça de colocar os tipos, valor default e multiplicidade dos atributos.

# Resolução Exercício Teórico

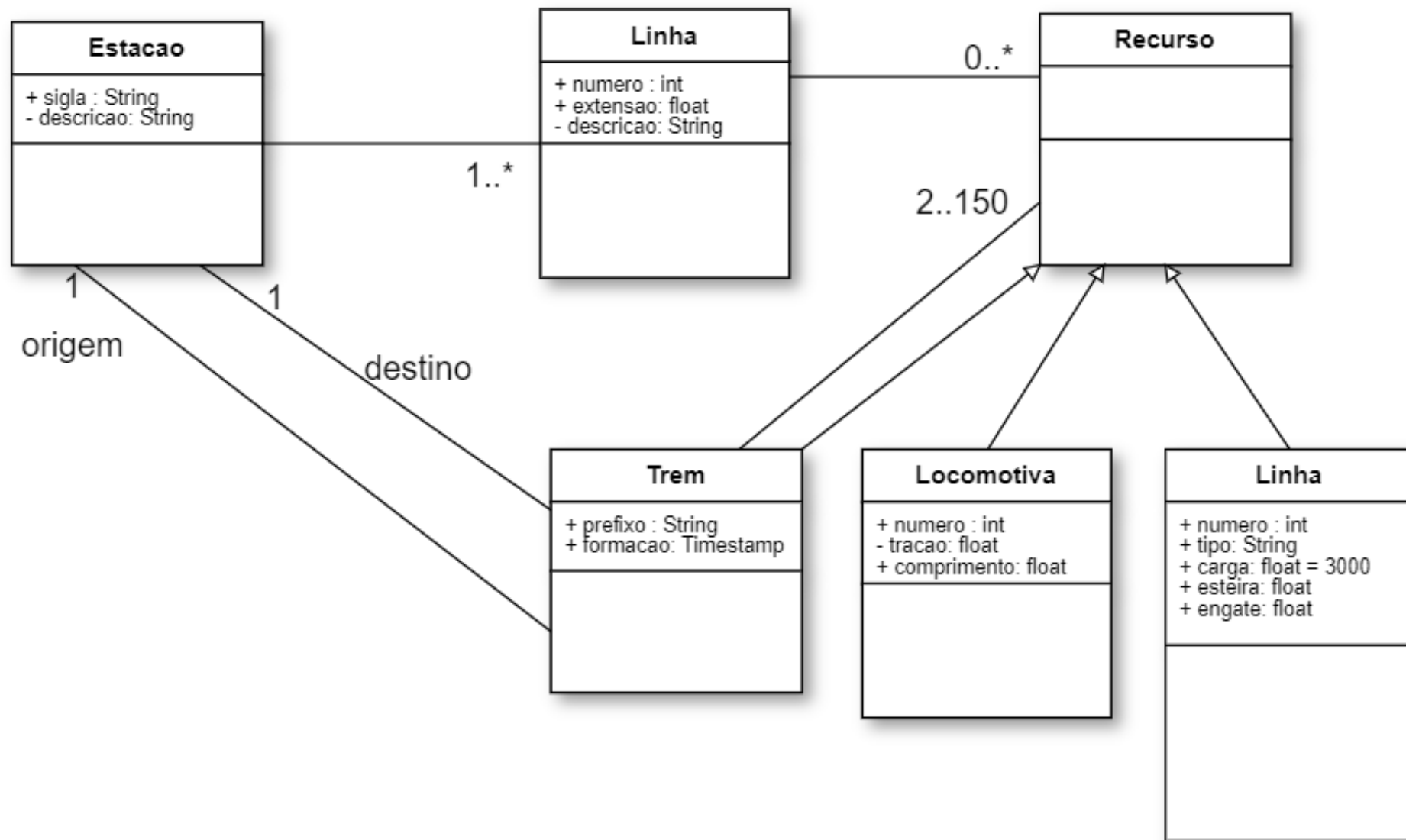
Um trem é formado por vagões e locomotivas estacionados na mesma linha

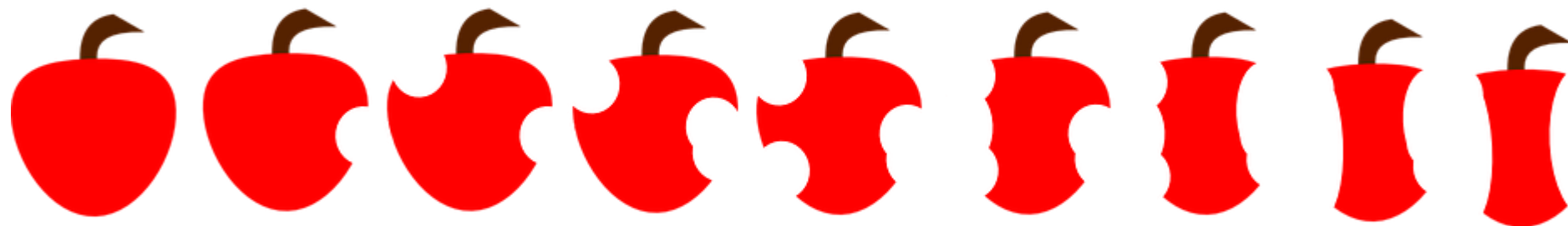
Podem existir vagões e locomotivas estacionadas em uma linha sem estarem na formação de um trem

A estação de origem não pode ser igual à estação de destino



# Resolução Exercício Teórico





## Diagrama de Sequência

# Diagramas Principais

- Cinco diagramas UML representam a essência da maioria dos sistemas
  - Diagrama de Casos de Uso
  - Diagrama de Classes
  - Diagrama de Sequência
  - Diagrama de Atividades
  - Diagrama de Estados

# Diagramas de Interação

- Representam como objetos interagem para executar um serviço
- Captura o comportamento de uma funcionalidade
  - Exemplo, de um caso de uso
- Uma interação é representada através da troca de mensagens



# Tipos de Diagramas de Interação

- **Diagrama de Sequência**
  - Enfatiza a ordem temporal das ações
- **Diagrama de Comunicação**
  - Enfatiza o relacionamento estrutural entre os objetos participantes da ação

# Diagrama de Sequência

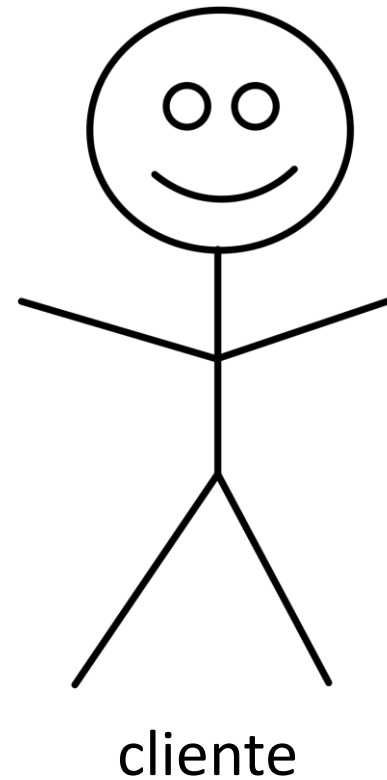
- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode ser usado para detalhar um Caso de Uso
- Identifica
  - O evento gerador da funcionalidade modelada (ator responsável pelo evento)
  - Os objetos envolvidos na ação

# 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

# Atores

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



# Objetos

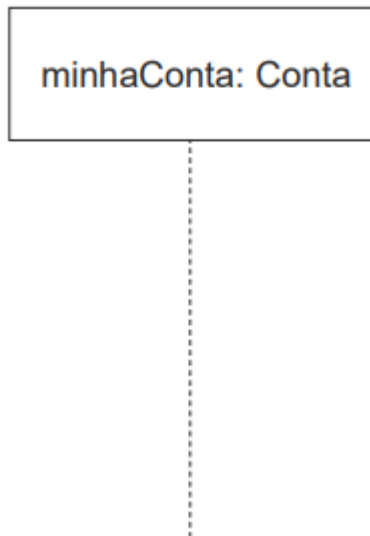
- Indicam instâncias de uma classe envolvidas no processo
  - As classes são mostradas em Diagramas de Classes
- Representados por retângulos
  - Nome do objeto (inicial minúsculo)
  - Nome da classe (inicial maiúscula)
  - Separação por dois pontos (:)



minhaConta: Conta

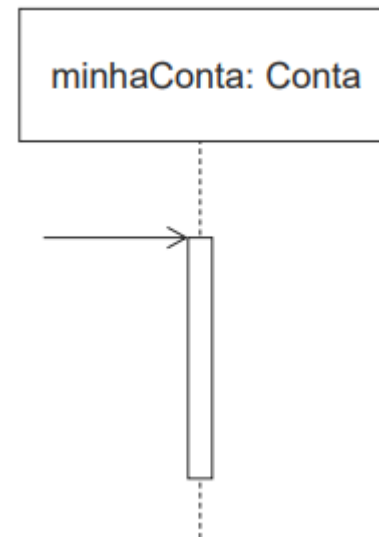
# Linha de Vida

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



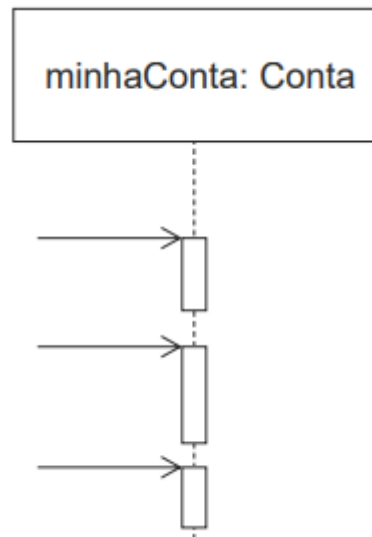
# Ativação do Objeto

- 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



# Ativação do Objeto

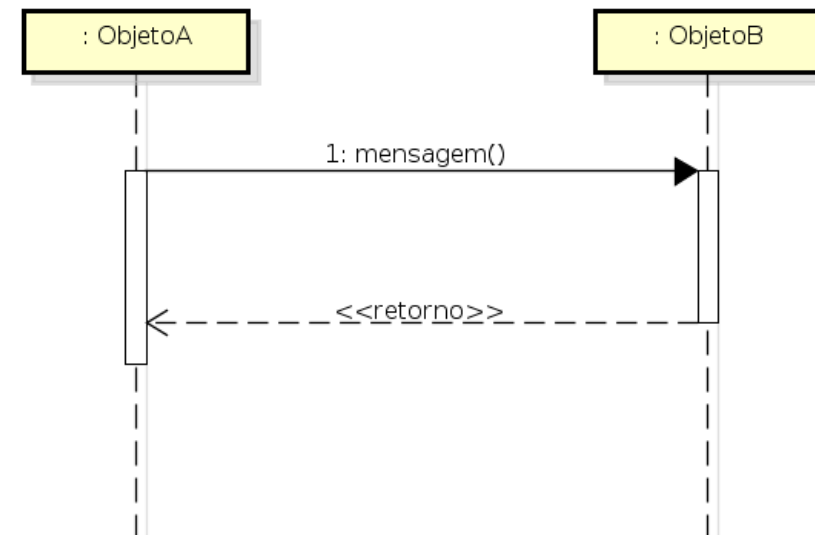
- Podem haver vários períodos em que o objeto se encontra ativo.





# Mensagens do Diagrama

- 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

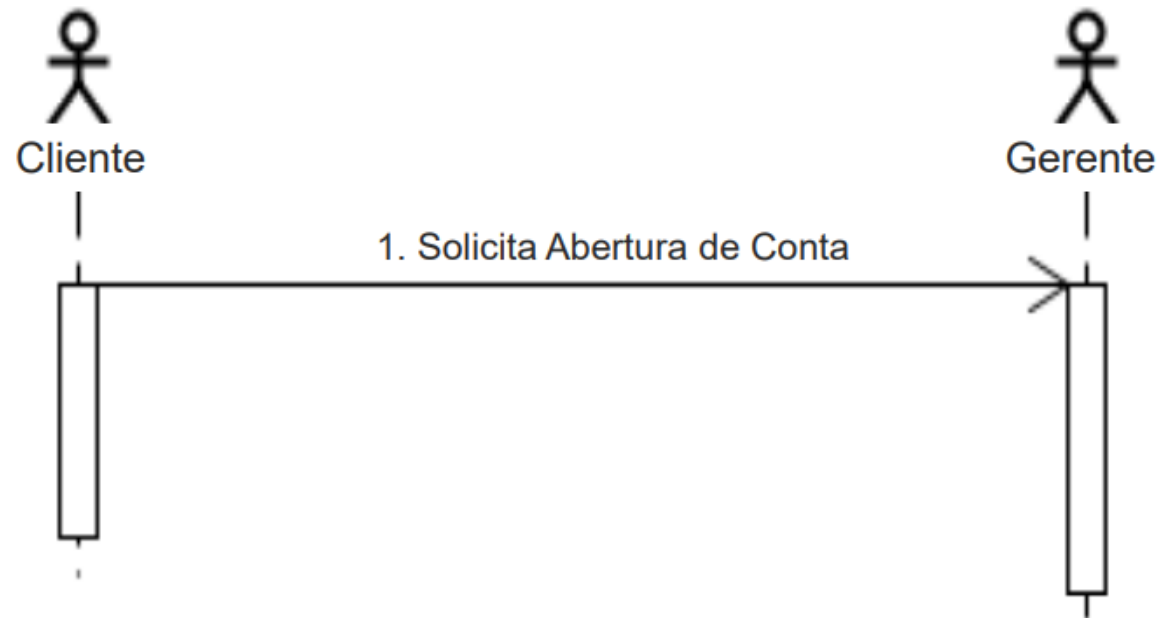


# Tipos de Mensagens

- Ator para Ator
- Ator para Objeto
- Objeto para Objeto
- Objeto para Ator

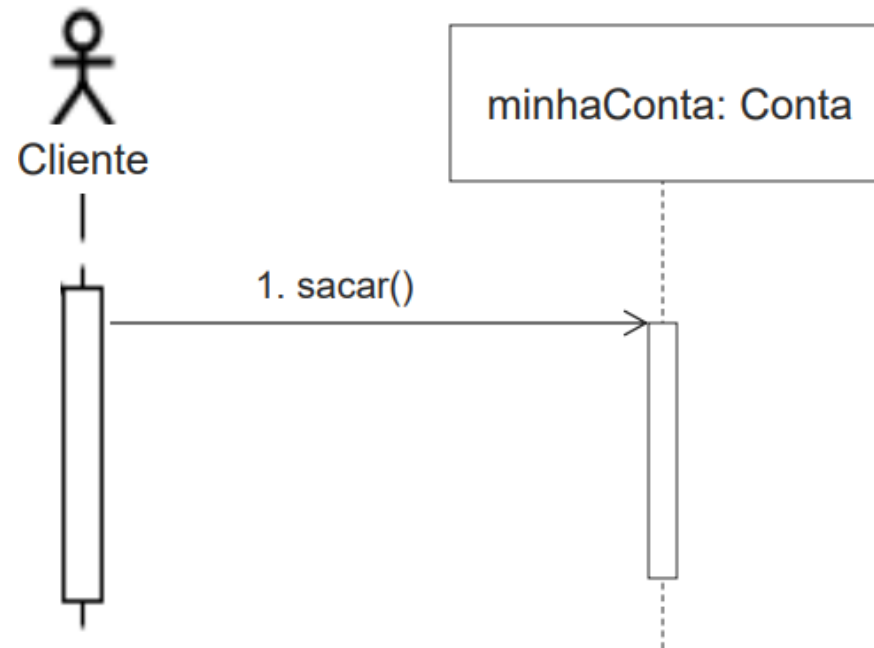
# Mensagem entre Atores

- Indica a conversa entre atores
- Os atores podem não fazer parte do sistema
- Mas, facilita a compreensão do processo
- Não é muito comum de se modelar



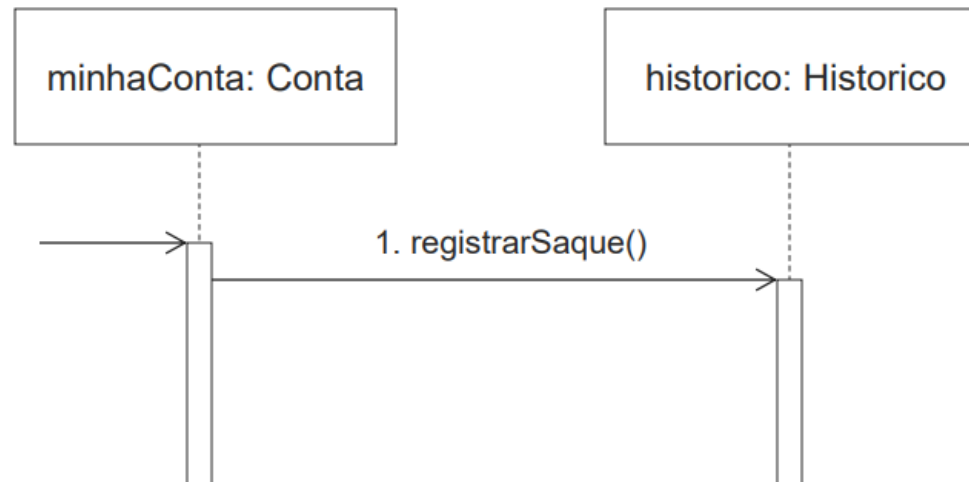
# Mensagem Ator-Objeto

- Indica uma solicitação de serviço feita pelo ator ao sistema
- O ator produz um evento que força o disparo de um método
- Tipo comum quando se modela casos de uso



# Mensagem entre Objetos

- Indica que um objeto transmite uma mensagem para outro objeto
  - Exemplo, solicitando a execução de um método
- Tipo mais comum de troca de mensagens

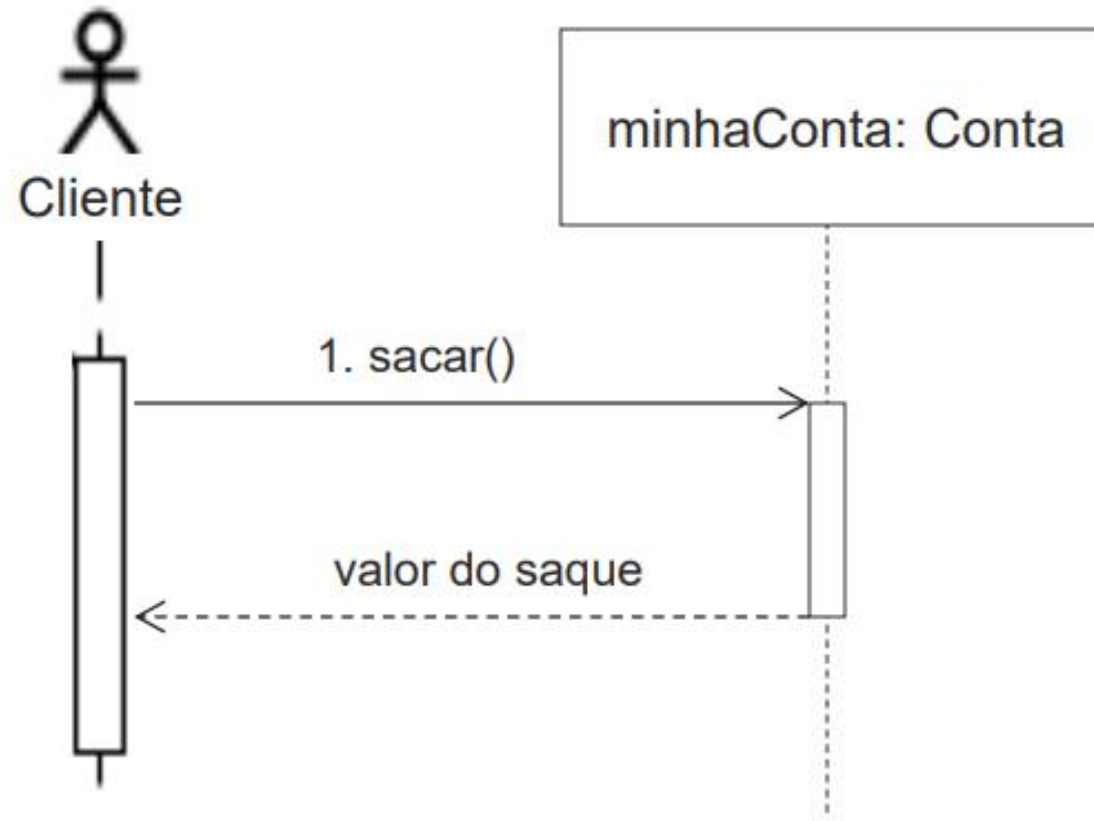


# Mensagem Objeto-Ator

- Indica a resposta de uma solicitação de serviço feita pelo ator
  - Ocorre normalmente quando o objeto envia uma mensagem de retorno
- Mensagens de retorno são representadas por linhas tracejadas
  - Pode conter legenda indicando o retorno

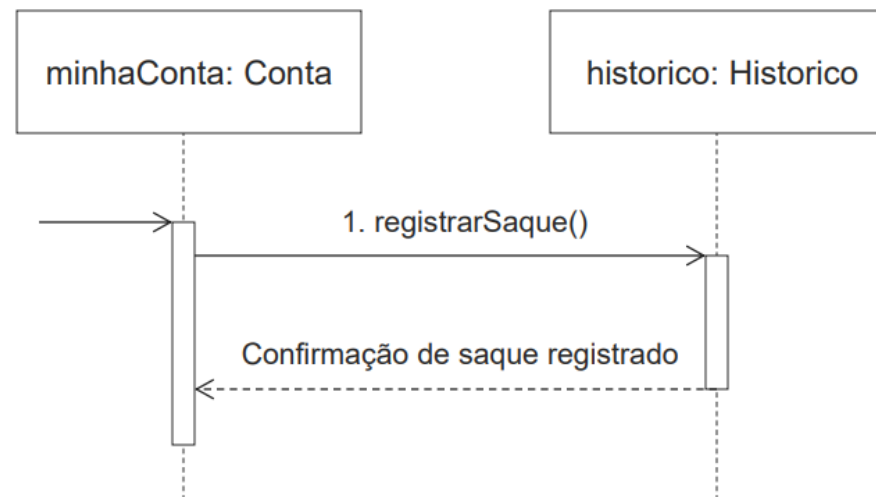


# Comunicação entre Objeto e Ator



# Mensagem de Retorno

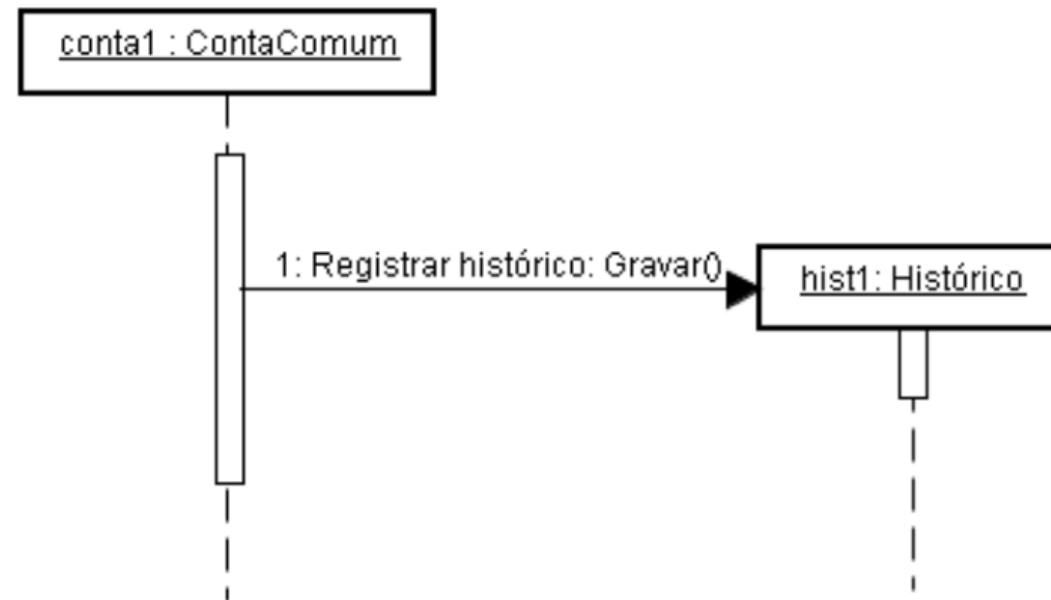
- Além de resposta ao ator, mensagens de retorno podem indicar respostas para objetos
  - Pode retornar informações específicas do método chamado
- Mensagens de retorno são opcionais em Diagramas de Sequência





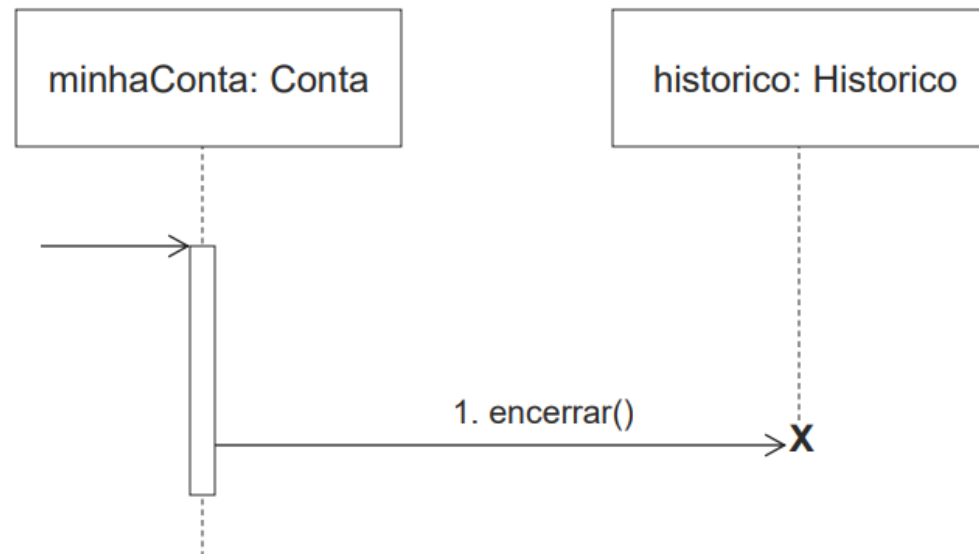
# Instanciação de um Objeto

- 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



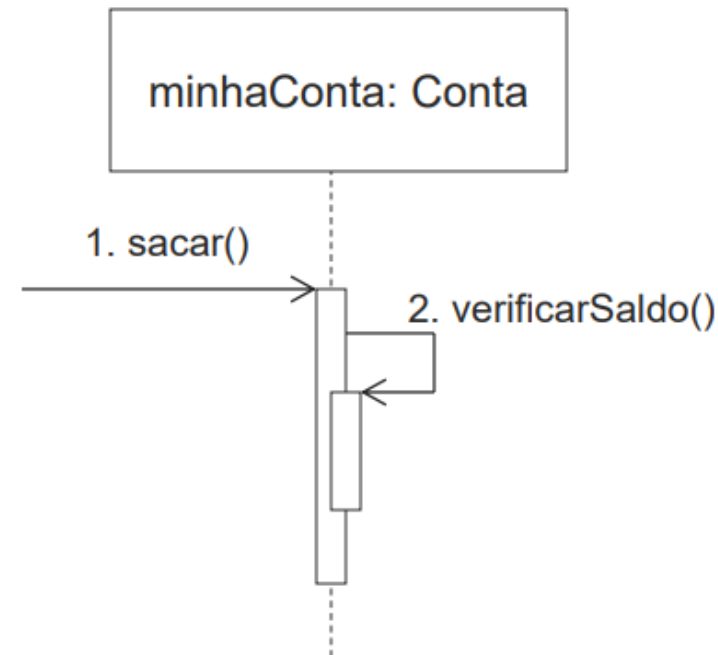
# Destruição de um Objeto

- 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

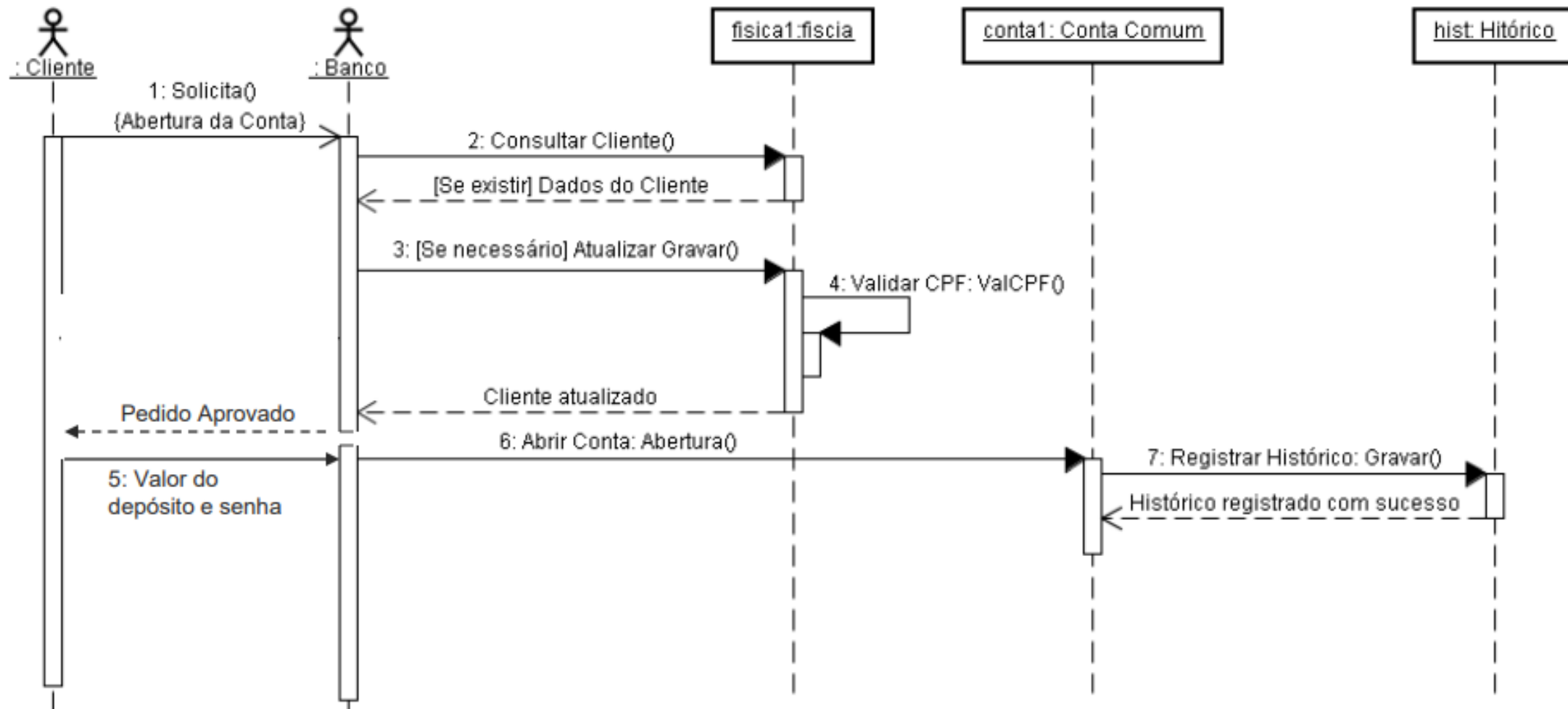


# Auto-Chamadas

- 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



# Diagrama de Sequência



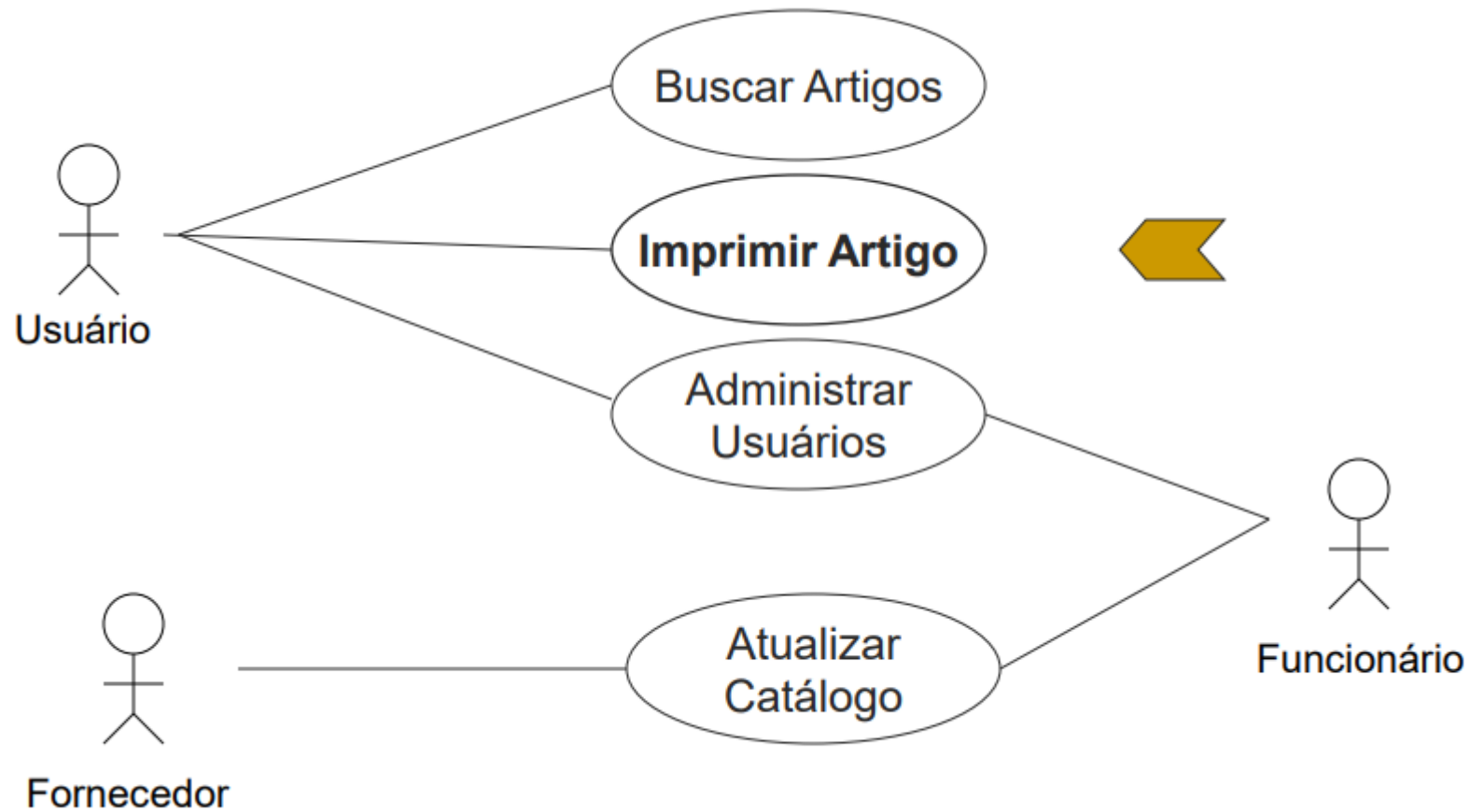
# Detalhe de um Caso de Uso

- 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

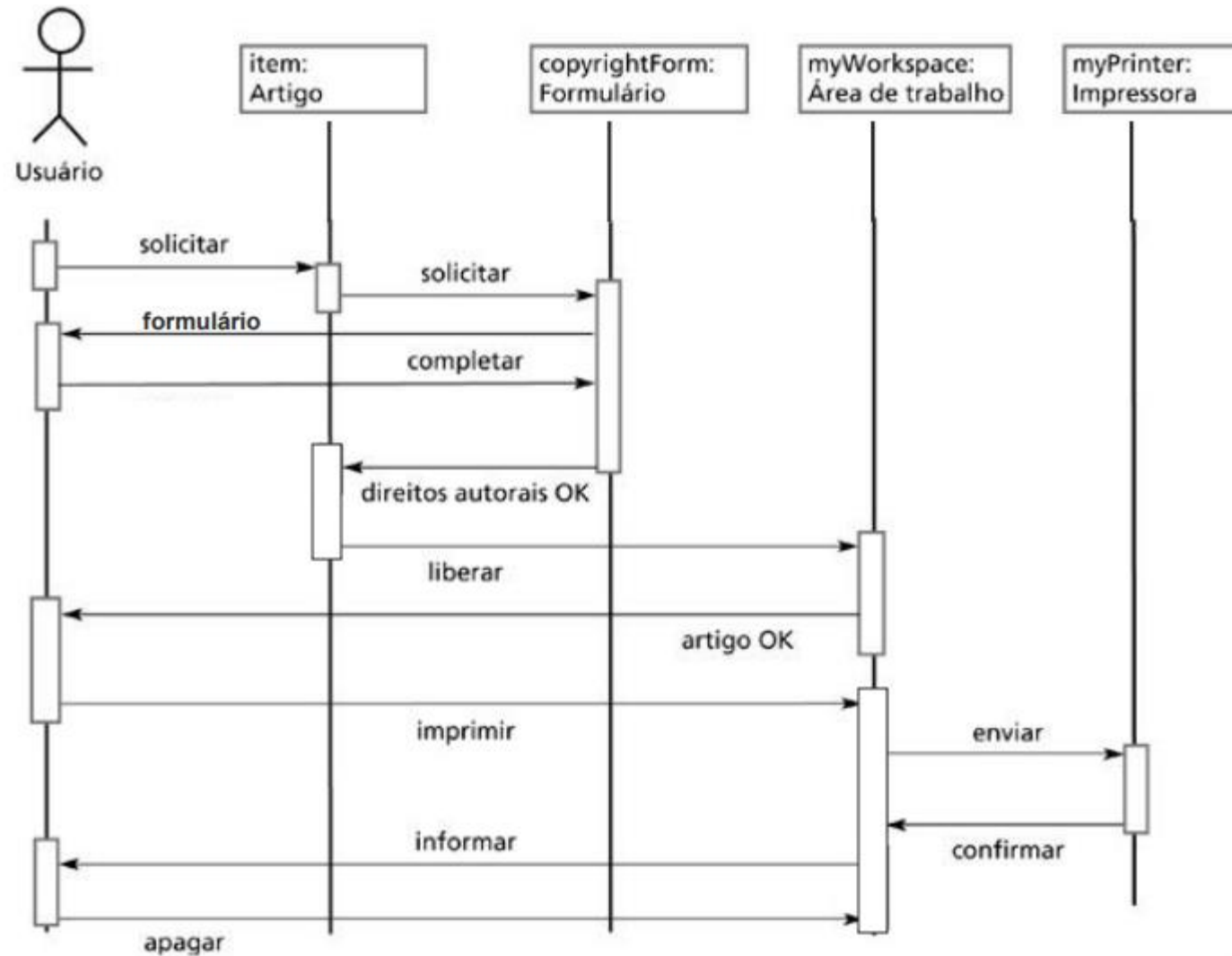
# Caso de Uso x Sequência

- 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>>*

# Exemplo: Casos de Uso



# Diagrama de Sequência: Imprimir Artigo







# Engenharia de Software I