

# Especialização em Desenvolvimento Java

## **UML e Padrões de Projetos**

Prof. Vinícius de Paula

<https://github.com/viniciusdepaula/aulas-uml-padroes>

# Diagrama de Sequência

- Diagrama comportamental que procura determinar a **sequência de eventos** que ocorrem em um **determinado processo**.
  - Determina a ordem em que os eventos ocorrem;
  - Identifica as mensagens que são enviadas e os métodos que são chamados;
  - Esclarece como os objetos interagem dentro de um determinado processo.
- Mostra os eventos que **partem do ator** e **chegam ao sistema**.
  - Para cada evento recebido o sistema irá executar uma operação em resposta.

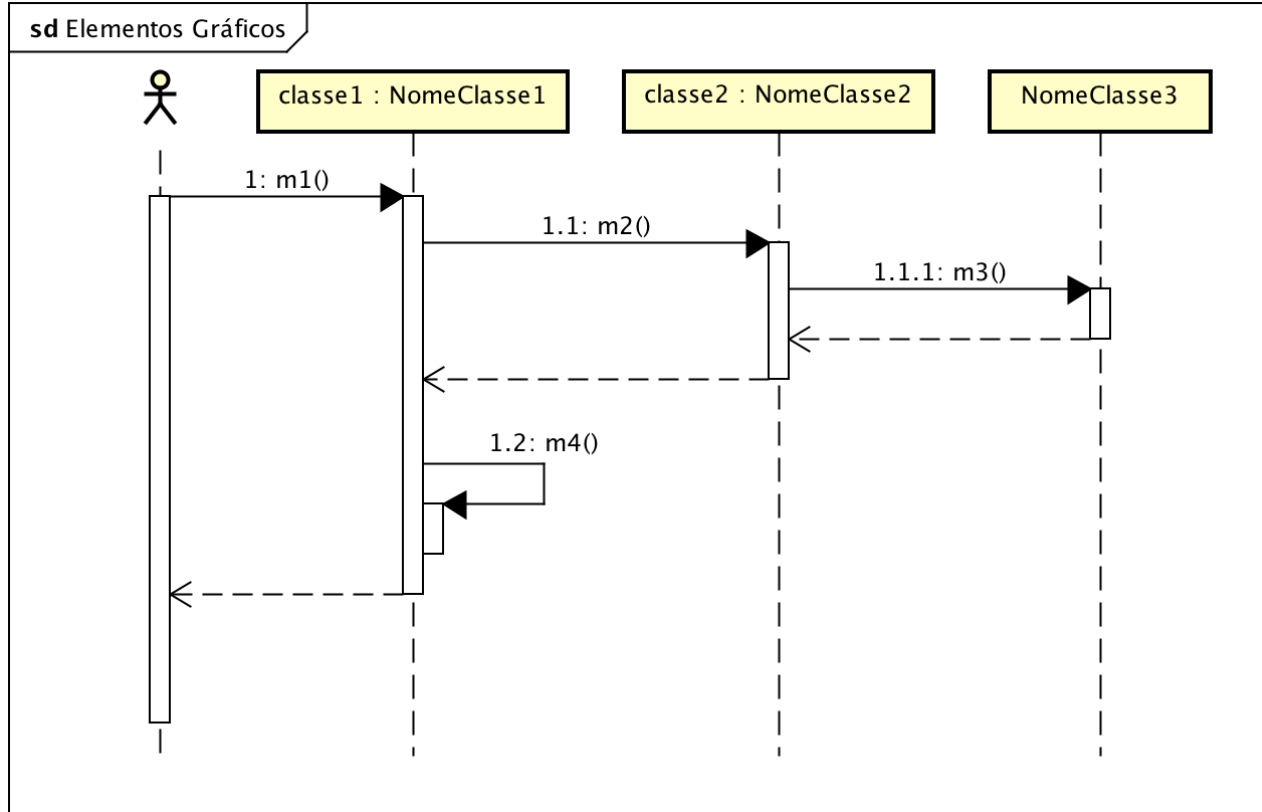
# Diagrama de Sequência

- Baseia-se no diagrama de casos de uso.
  - Normalmente existe um diagrama de sequência para cada caso de uso declarado.
- Obviamente, existe uma dependência do diagrama de classes, onde as classes dos objetos utilizados no diagrama de sequência estão descritas nele.
  - Pode ser utilizado para complementar o diagrama de classes, pois é ao modelar o diagrama de sequência que se percebe quais métodos são necessários declarar em que classes.

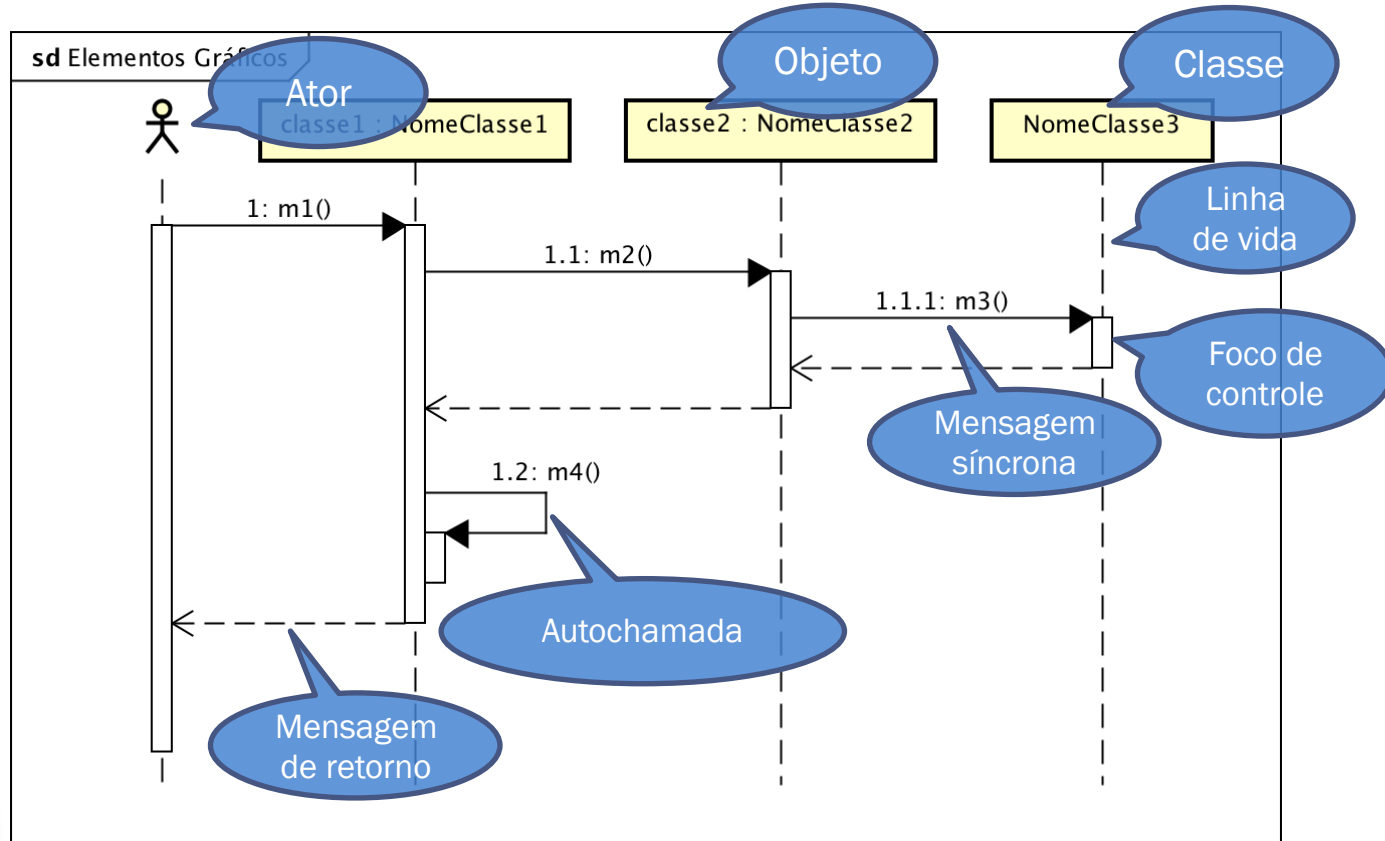
# Elementos Básicos

- Elementos básicos em um diagrama de sequência:
  - Atores
  - Objetos
  - Linhas de vida e focos de controle
  - Mensagens
  - Iterações

# Elementos Gráficos

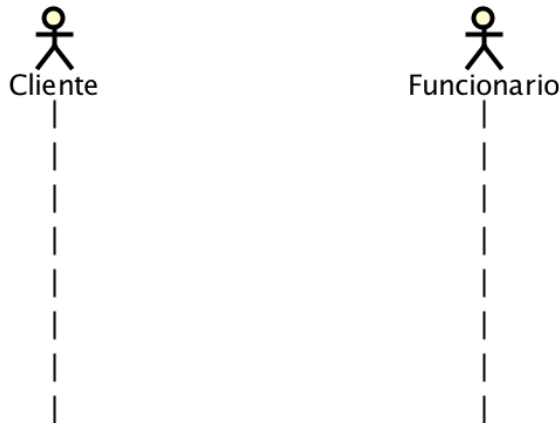


# Elementos Gráficos



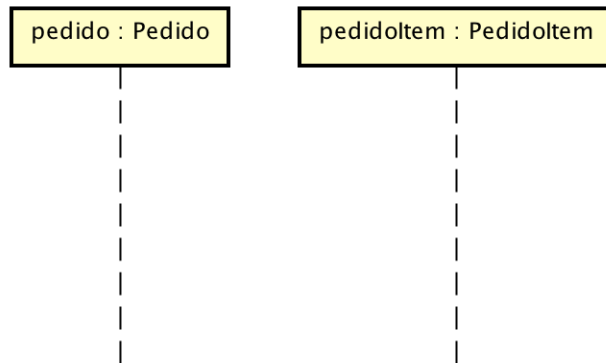
# Atores

- São os mesmos do diagrama de casos de uso e possuem a mesma representação, mas diferenciam-se por apresentar uma “linha de vida”.
- Na maioria das vezes, um diagrama de sequência é iniciado por um evento externo gerado por um ator.



# Objetos

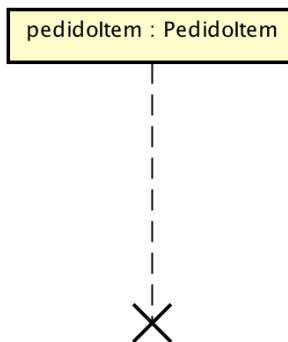
- Representam as instâncias das classes envolvidas no processo ilustrado pelo diagrama de sequência.
- Assim como os atores, os objetos possuem uma “linha de vida” vertical tracejada.
- Um objeto pode existir desde o início do processo ou ser criado durante sua execução.





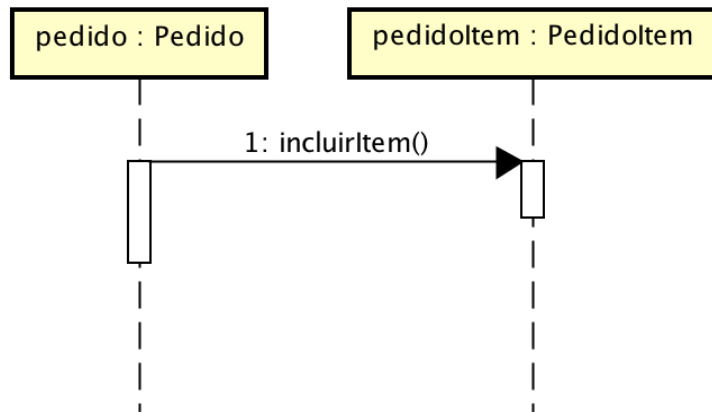
# Linha de Vida

- Representa o tempo em que um objeto existe durante um processo.
  - Representada por uma linha fina tracejada partindo do objeto.
  - Quando o objeto é destruído, a linha de vida é interrompida com um "X".
  - Um objeto não precisa necessariamente existir quando um processo é iniciado.



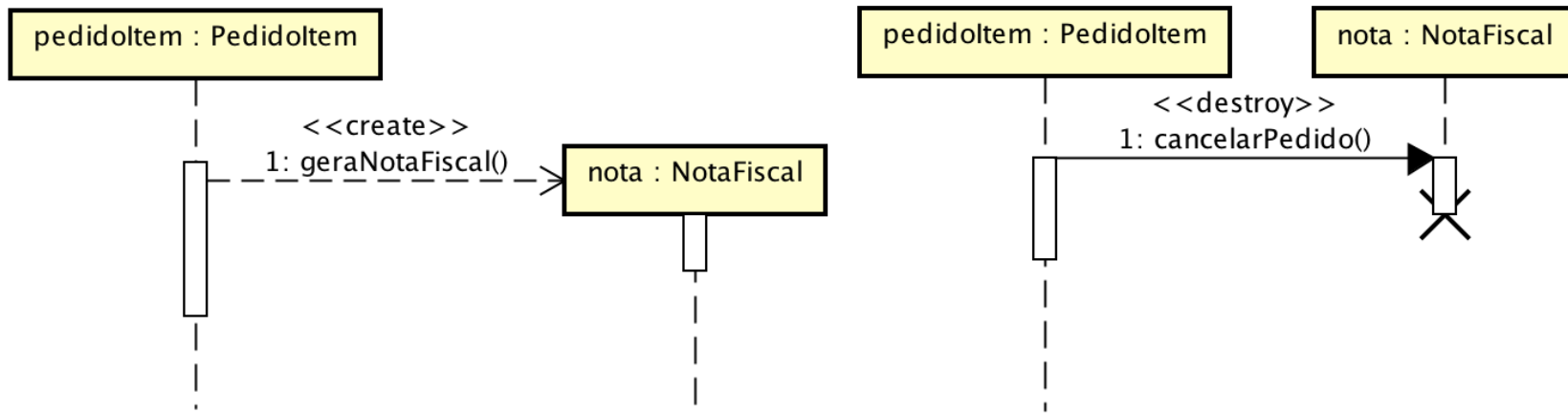
# Foco de Controle

- 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 grossa.



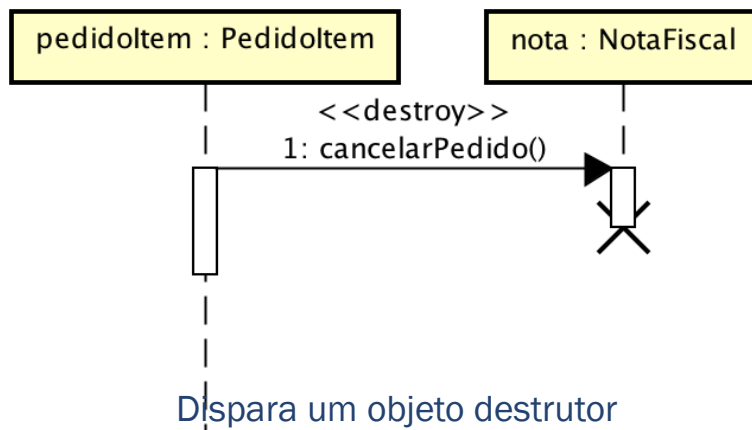
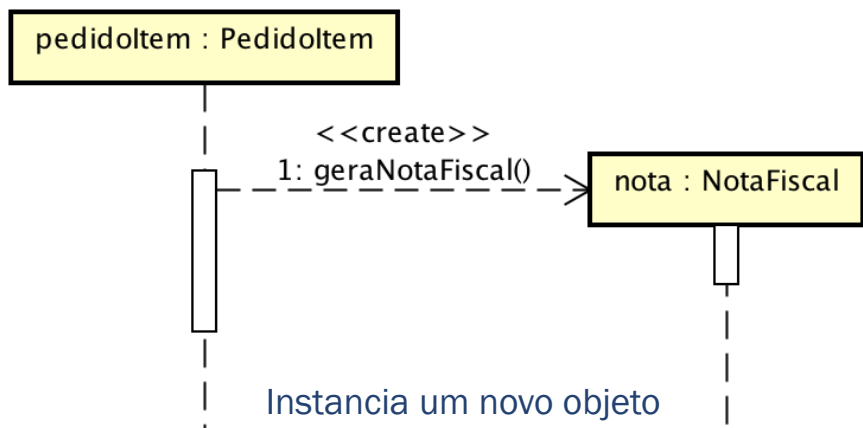
# Mensagens

- Demonstram a ocorrência de eventos que, comumente, fazem chamadas a um método de algum objeto envolvido no processo.
  - São representadas por uma seta e indica qual componente enviou a mensagem e qual a recebeu.



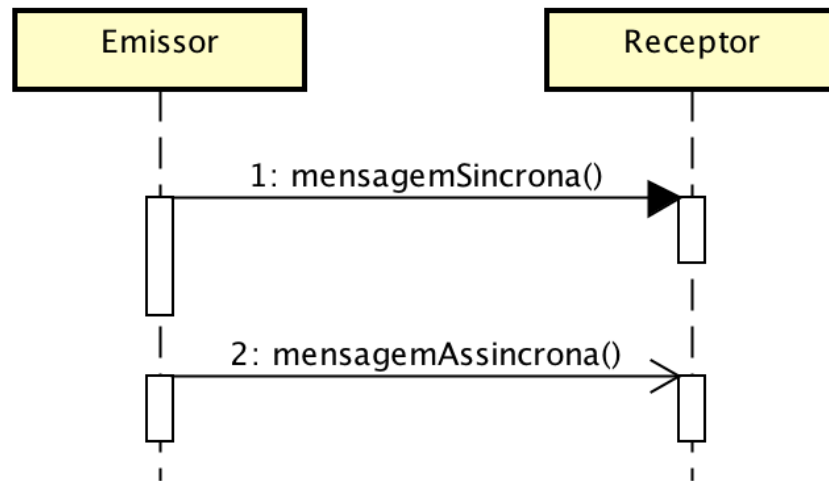
# Mensagens

- Demonstram a ocorrência de eventos que, comumente, fazem chamadas a um método de algum objeto envolvido no processo.
  - São representadas por uma seta e indica qual componente enviou a mensagem e qual a recebeu.



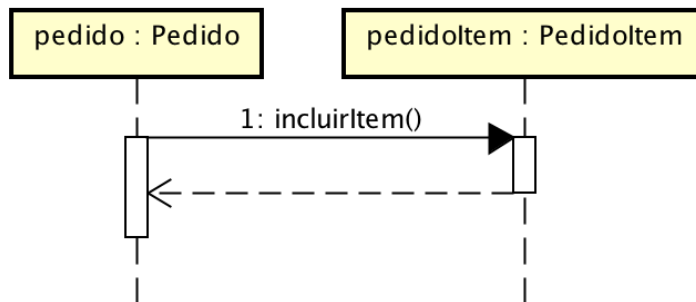
# Tipos de Mensagens

- **Síncrona:** o emissor da mensagem fica bloqueada até o receptor receber e tratar a mensagem.
- **Assíncrona:** o emissor continua a emitir mensagens, não há dependências.



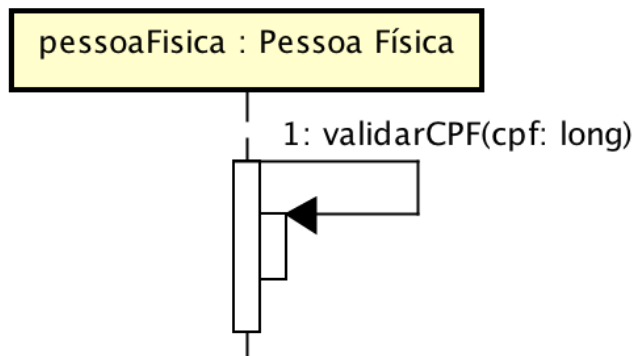
# Mensagens de Retorno

- **Identifica a resposta a uma mensagem para o objeto ou ator que a chamou.**
  - Pode retornar informações específicas do método chamado ou apenas um valor indicando que o método foi executado com sucesso ou não.
  - São representadas por uma linha tracejada contendo uma seta fina que aponta para o objeto ou ator que recebe o resultado do método chamado.



# Autochamada

- São mensagens que um objeto envia para si mesmo.
- Uma mensagem parte da linha de vida do objeto e atinge a linha de vida do próprio objeto.



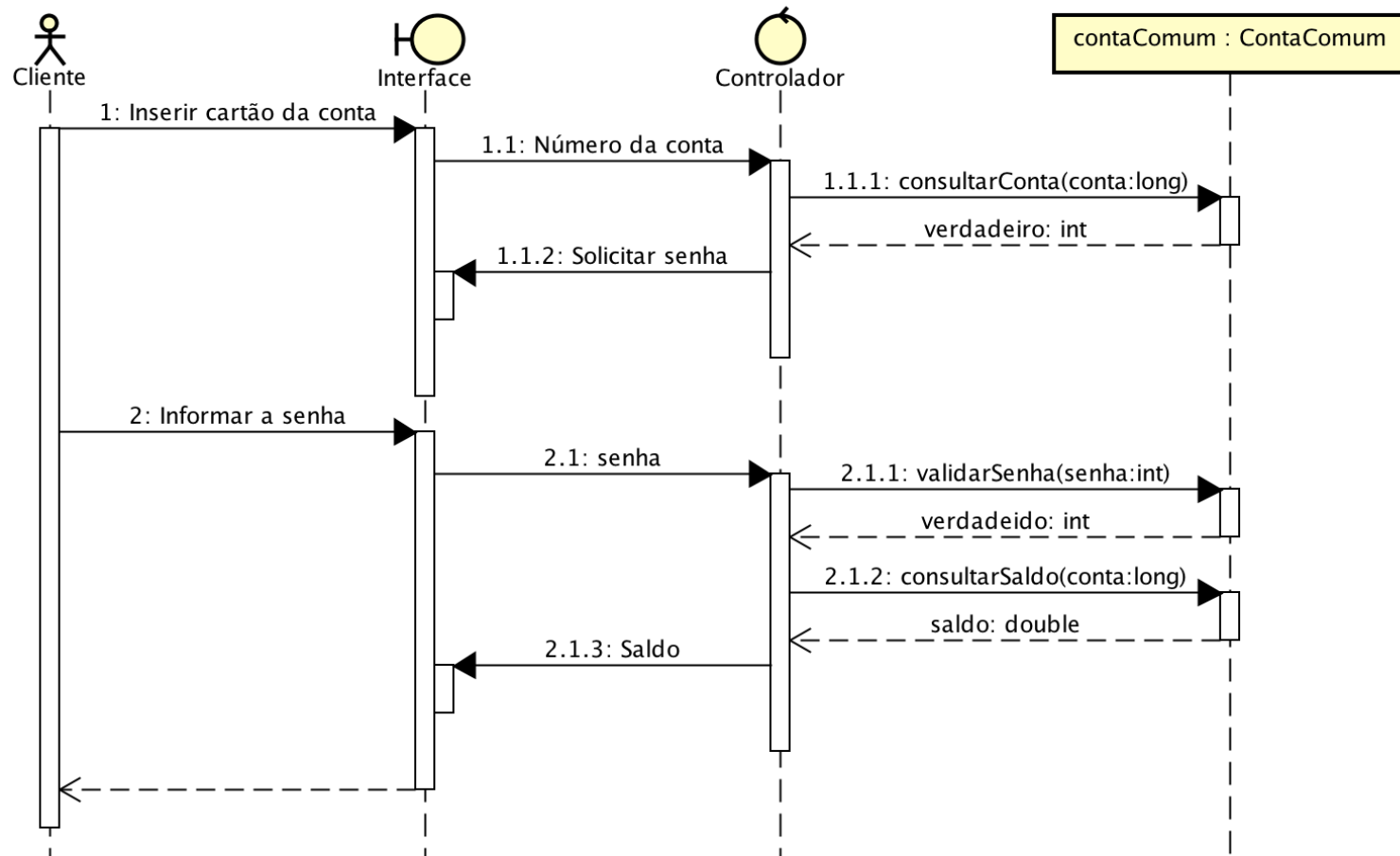
# Estereótipos

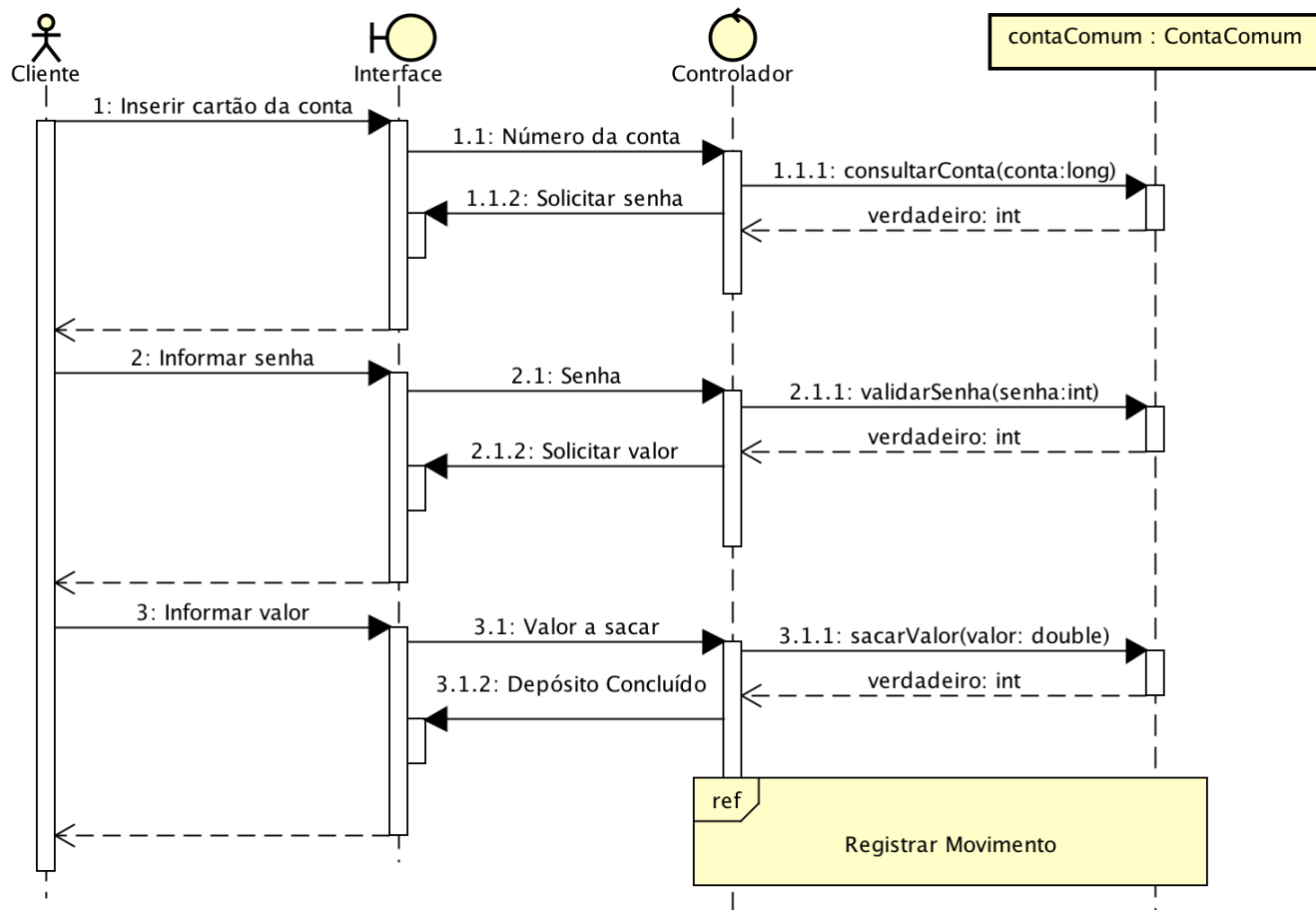
- **<<boundary>>** identifica uma classe que serve de comunicação entre os Atores externos e o sistema.
  - Comumente é associada à própria interface e interage com classes do tipo **<<control>>**
- **<<control>>** intermediária entre as classes **<<boundary>>** e outras do sistema.
  - Responsável por interpretar eventos ocorridos sobre os objetos **<<boundary>>** e retransmiti-los para as outras classes do sistema.



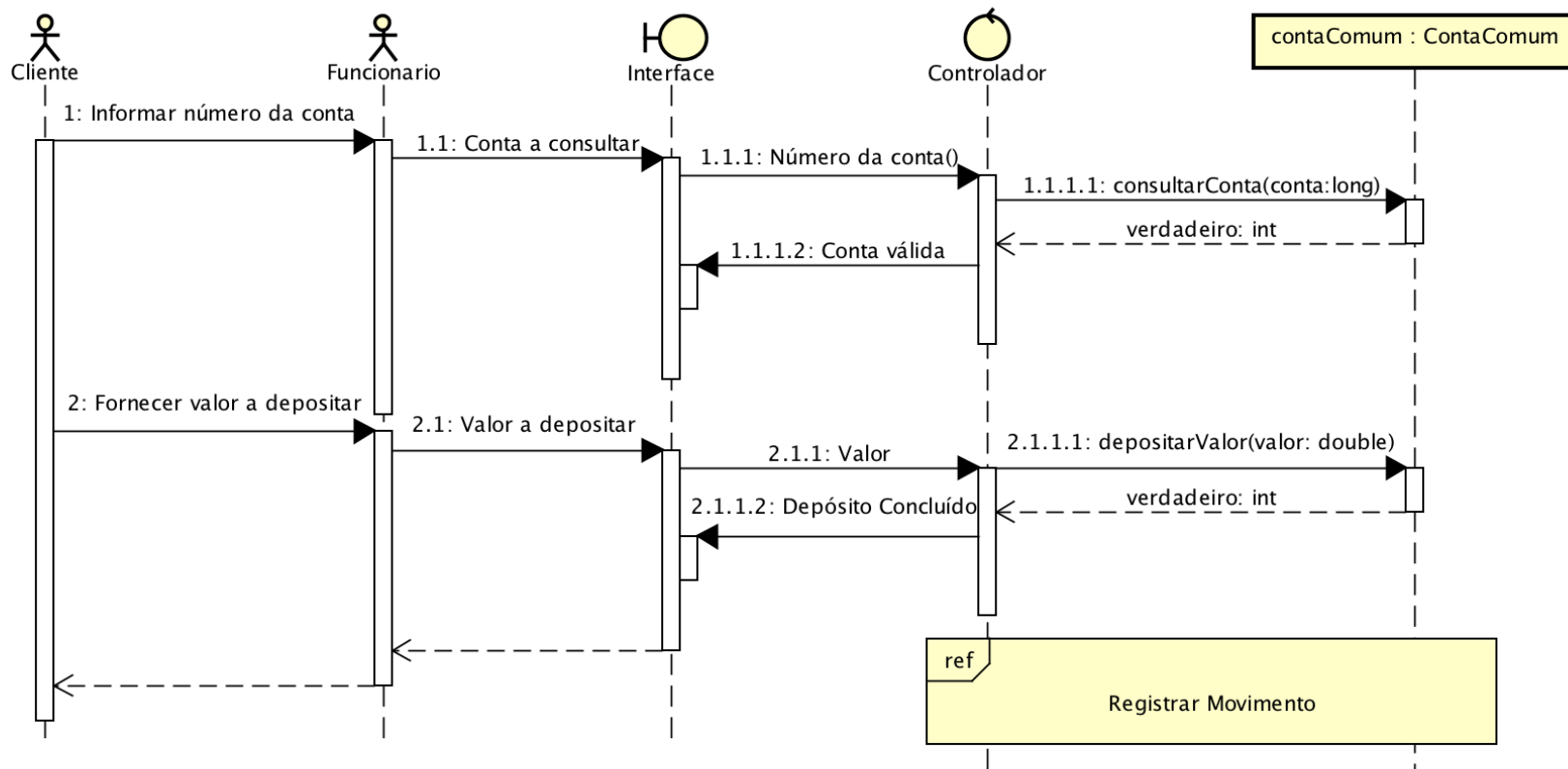
# Exemplos de Diagramas de Sequências

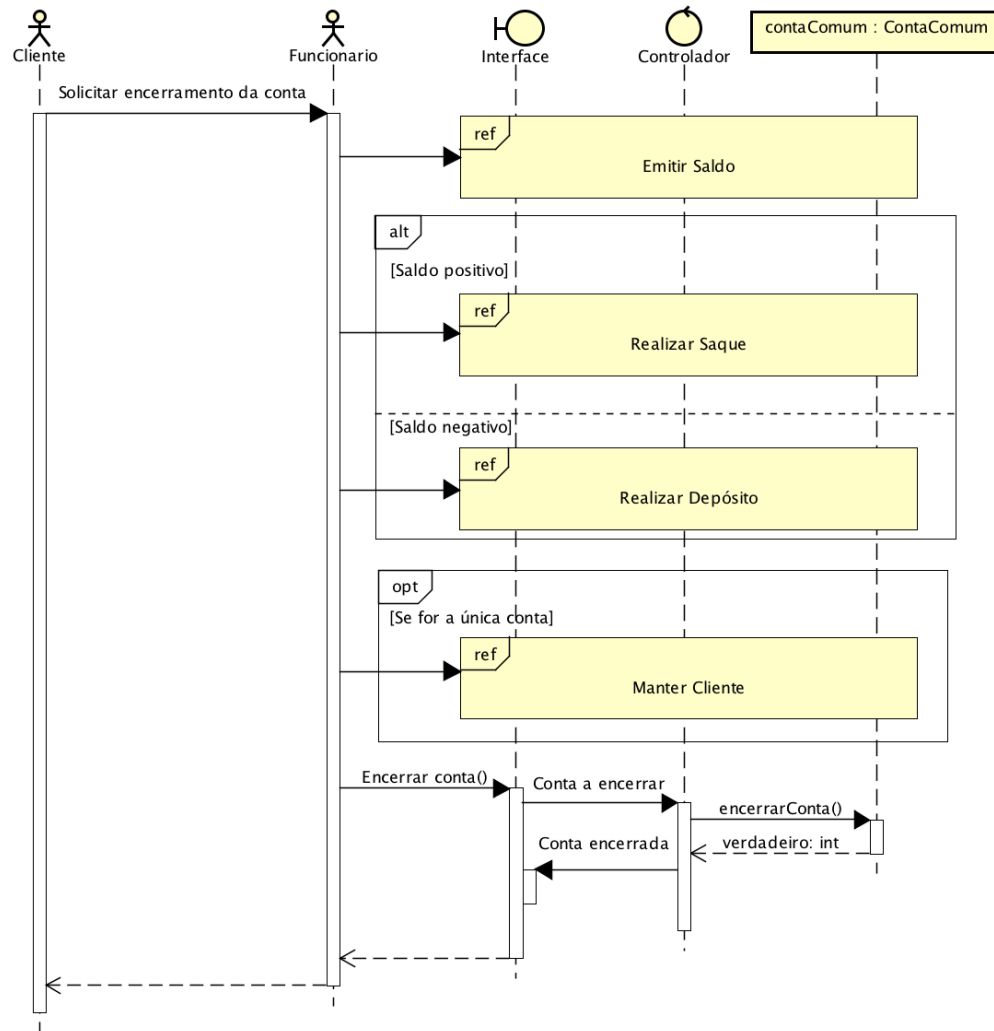
- Nos slides a seguir serão exemplificados alguns dos diagramas de sequência referentes ao sistema de controle bancário.





# sd Realizar Depósito





# Lista de Exercícios III



UML e Padrões de Projeto - Lista de Exercícios III.pdf



45min

# Bibliografia

- GUEDES, Gilleanes. UML Uma Abordagem Prática. Editora Novatec. São Paulo, 2014.
- FURLAN, José. Modelagem de Objetos através da UML. Editora Makron Books.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML Guia do Usuário. Editora Campus.