

# Modelo de Colaboração

Análise de Sistemas e  
Requisitos de Software II

Aula 12

Allan Rodrigo Leite

# Diagrama de Colaboração

- A UML dispõe do diagrama de colaboração para representar a interação dos objetos
  - O diagrama de colaboração também destaca as responsabilidades de cada objeto
- O conceito de responsabilidade parte do princípio que:
  - Cada objeto conhece seu “papel” dentro do sistema
  - Para realizar um evento ou uma consulta no sistema, o objeto atual delega a execução da operação para o objeto que contem a implementação da operação (ou para o que estiver mais próximo)
- É recomendável existir um diagrama de colaboração para cada operação e consulta identificada na análise

# Diagrama de Colaboração

- Diagrama de sequência
  - Ênfase no fluxo da informação
  - Destaca a ordem cronológica das ações
  - Normalmente representado pelas principais classes relacionadas ao caso de uso em questão
- Diagrama de colaboração
  - Ênfase na comunicação entre classes
  - Destaca a responsabilidade e delegação das classes
  - Explora a colaboração entre as classes relacionadas a uma operação ou consulta

# Diagrama de Colaboração

- Características do diagrama de colaboração
  - Extensão do diagrama de objetos
  - Representa os vínculos entre os objetos
  - Não possui representação explícita de tempo
  - Representação mais próxima da implementação

# Diagrama de Colaboração

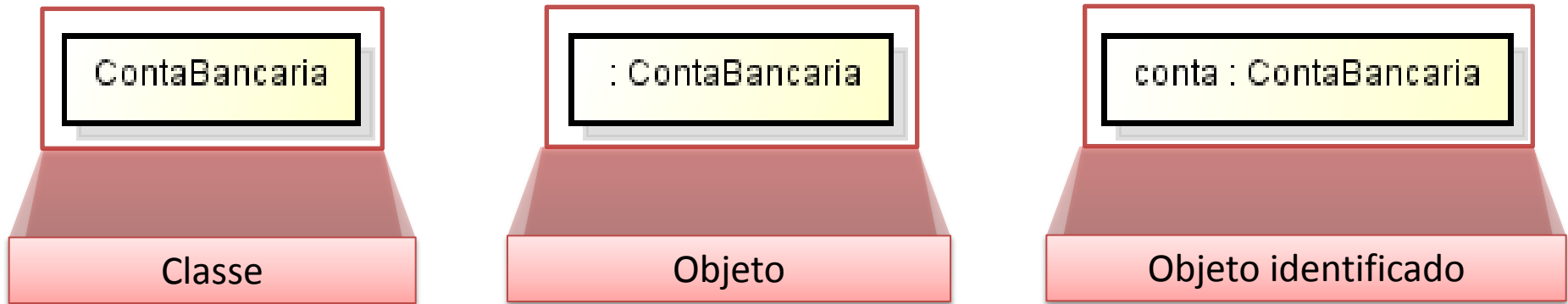
- Um dos principais objetivos do diagrama de colaboração é identificar os vínculos
- Vínculos são ligações existentes entre os objetos envolvidos no processo
  - Envio de mensagens
  - Recebimento de mensagens
  - Ou ambos

# Diagrama de Colaboração

- Elementos do diagrama de colaboração
  - Atores
  - Interface e controlador fachada
  - Objetos e classes
  - Mensagens
  - Ligações entre objetos (colaboração)
  - Criação e destruição de objetos
  - Iterações
  - Condições de guarda

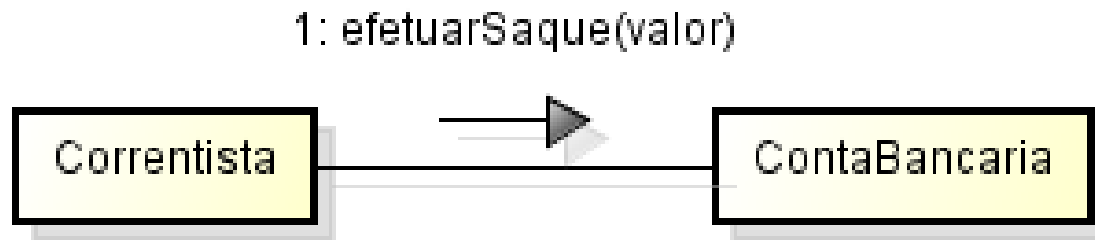
# Diagrama de Colaboração

- Classes e objetos



# Diagrama de Colaboração

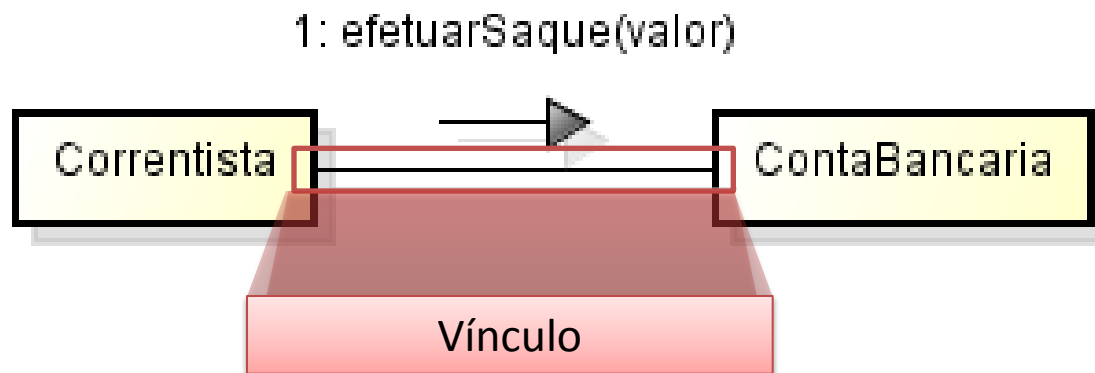
- Colaboração





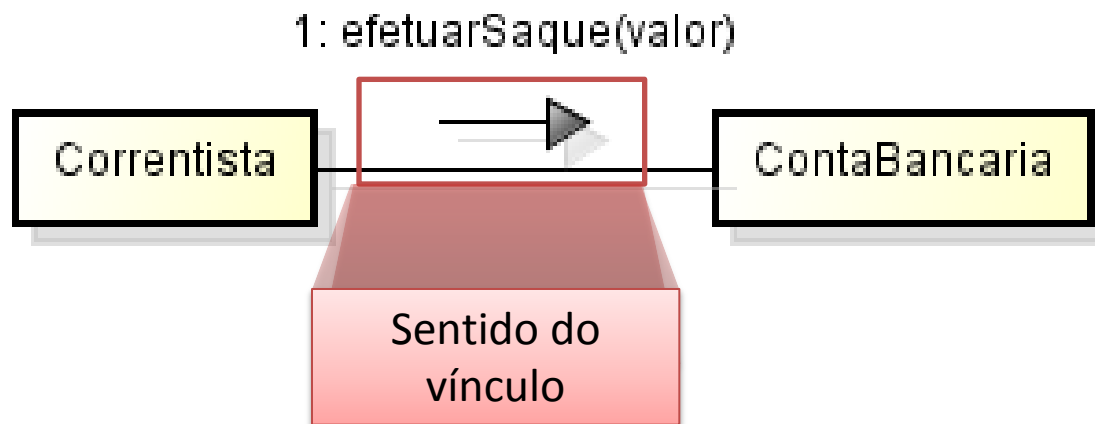
# Diagrama de Colaboração

- Colaboração



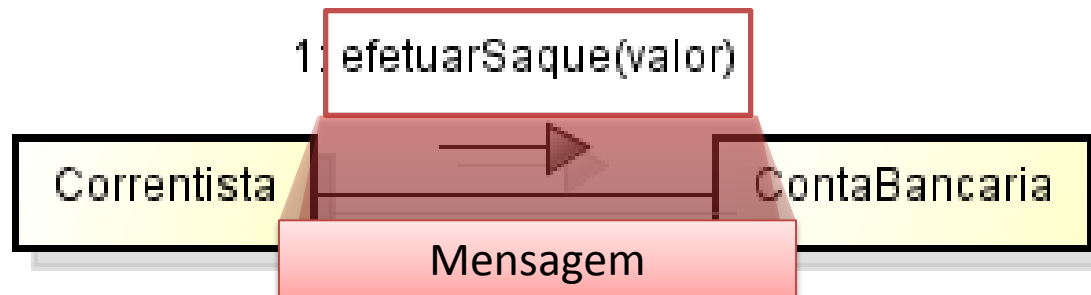
# Diagrama de Colaboração

- Colaboração



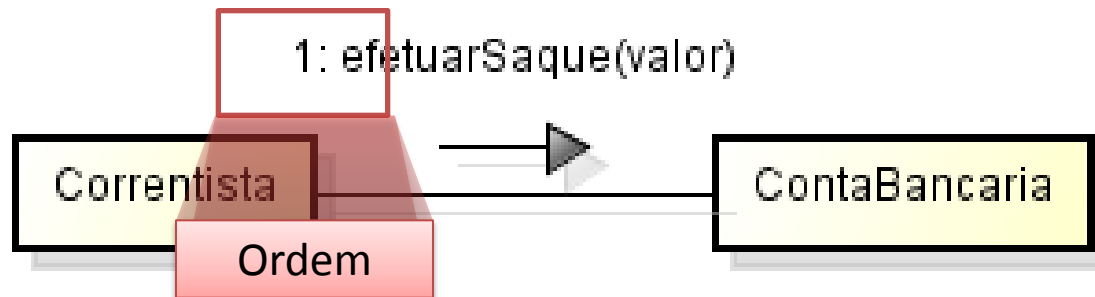
# Diagrama de Colaboração

- Colaboração



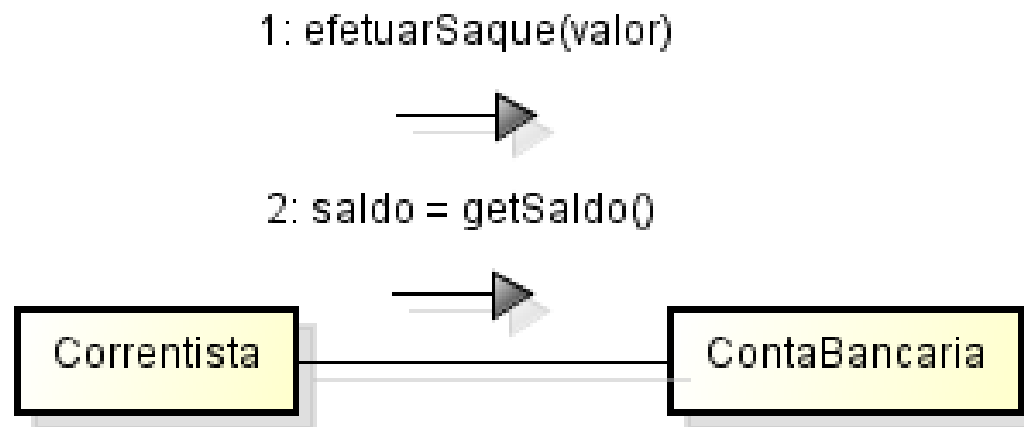
# Diagrama de Colaboração

- Colaboração



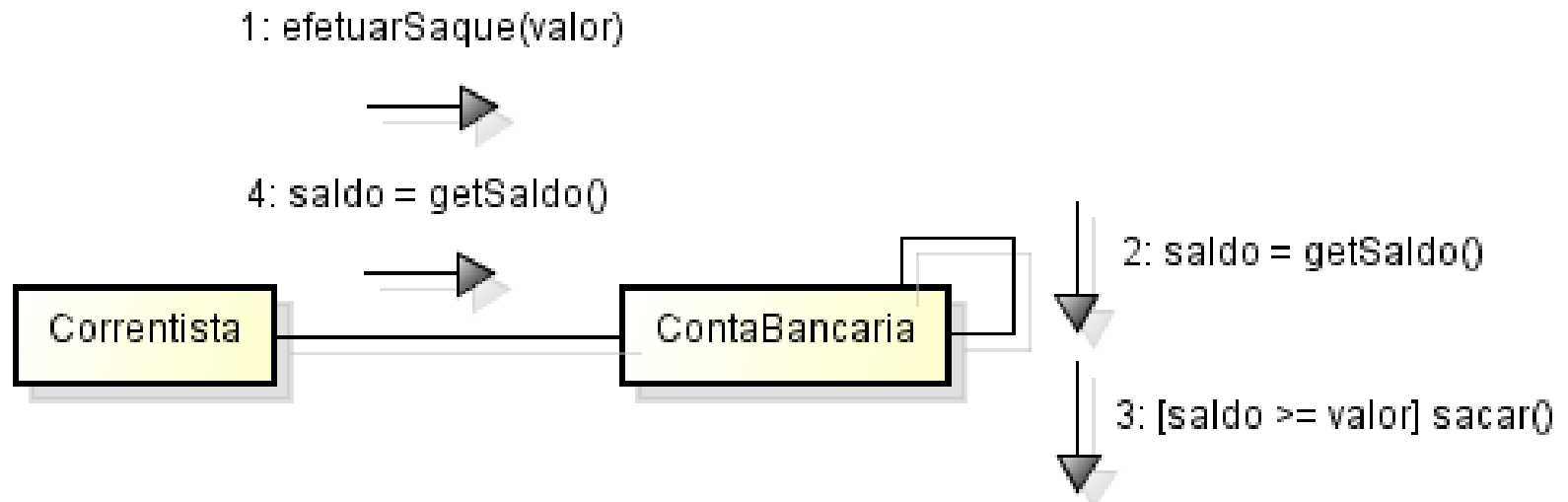
# Diagrama de Colaboração

- Várias mensagens trocadas entre dois objetos



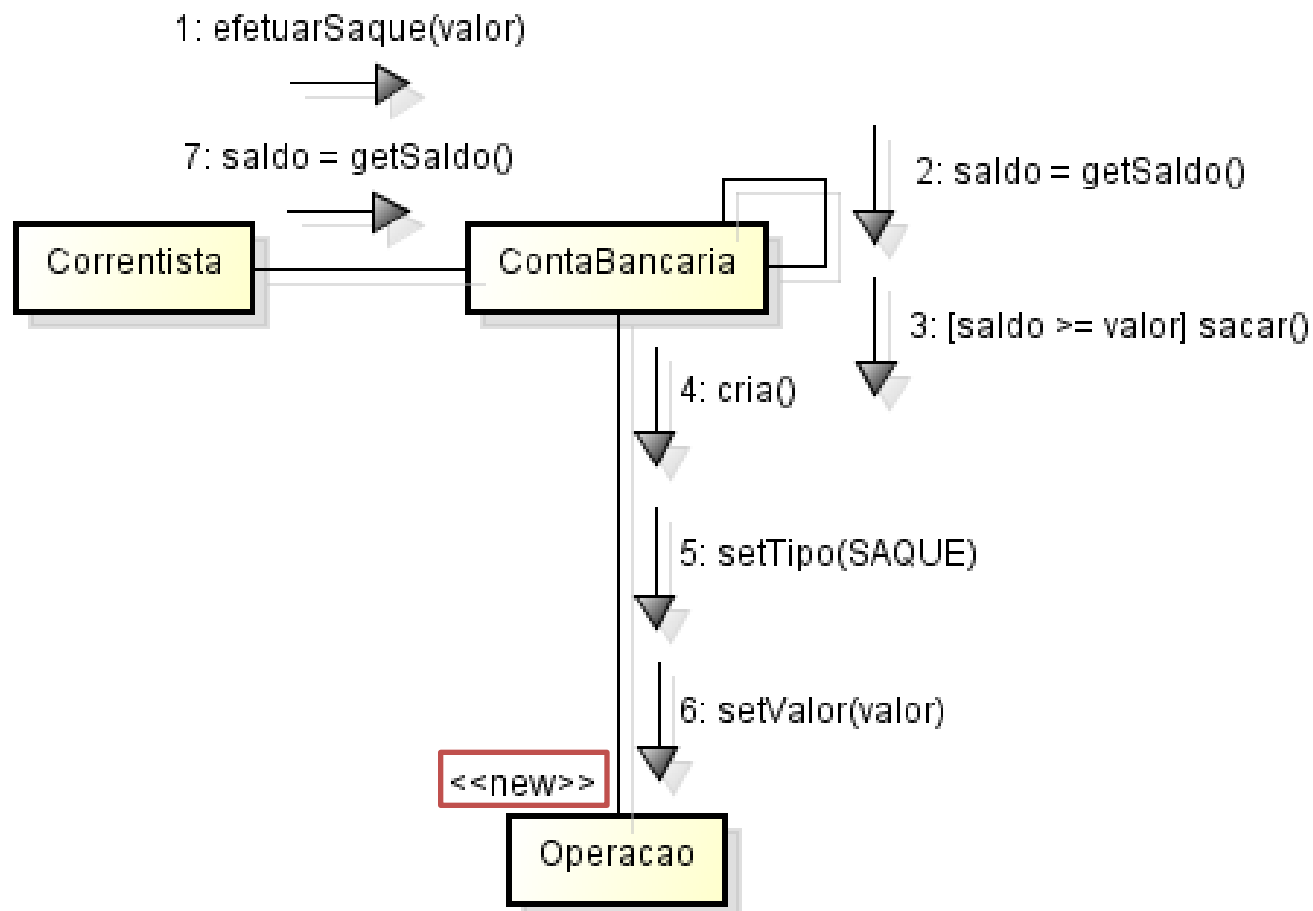
# Diagrama de Colaboração

- Auto-mensagem e condição de guarda



# Diagrama de Colaboração

- Notações ou restrições



# Diagrama de Colaboração

- Notações ou restrições
  - <<new>>
  - <<destroy>>
  - <<transient>>



# Diagrama de Colaboração

- <<create>>
  - Define a criação de uma instância de classe
  - Instância será mantida após a execução da operação
- <<destroy>>
  - Define a destruição de uma instância de classe
- <<transient>>
  - Define a criação e destruição de uma instância de classe
  - Instância existe apenas durante a execução da operação

# Diagrama de Colaboração

- Tipos de mensagens

- Síncrona



- Assíncrona



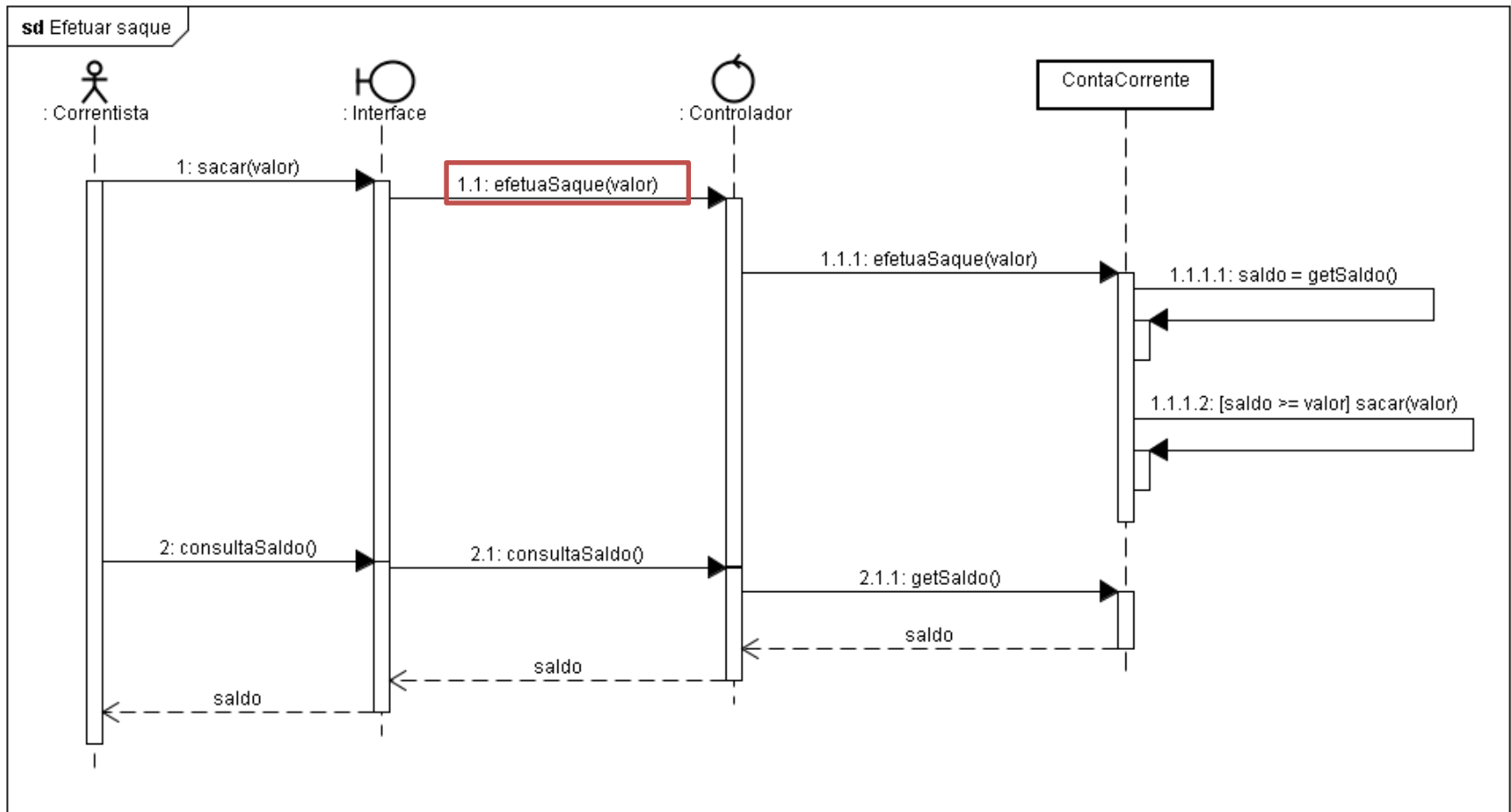
- Para mensagens assíncronas, quando há necessidade de sincronismo deve ser utilizado o caractere “/” antes da operação

# Como Desenvolver os Diagramas de Colaboração

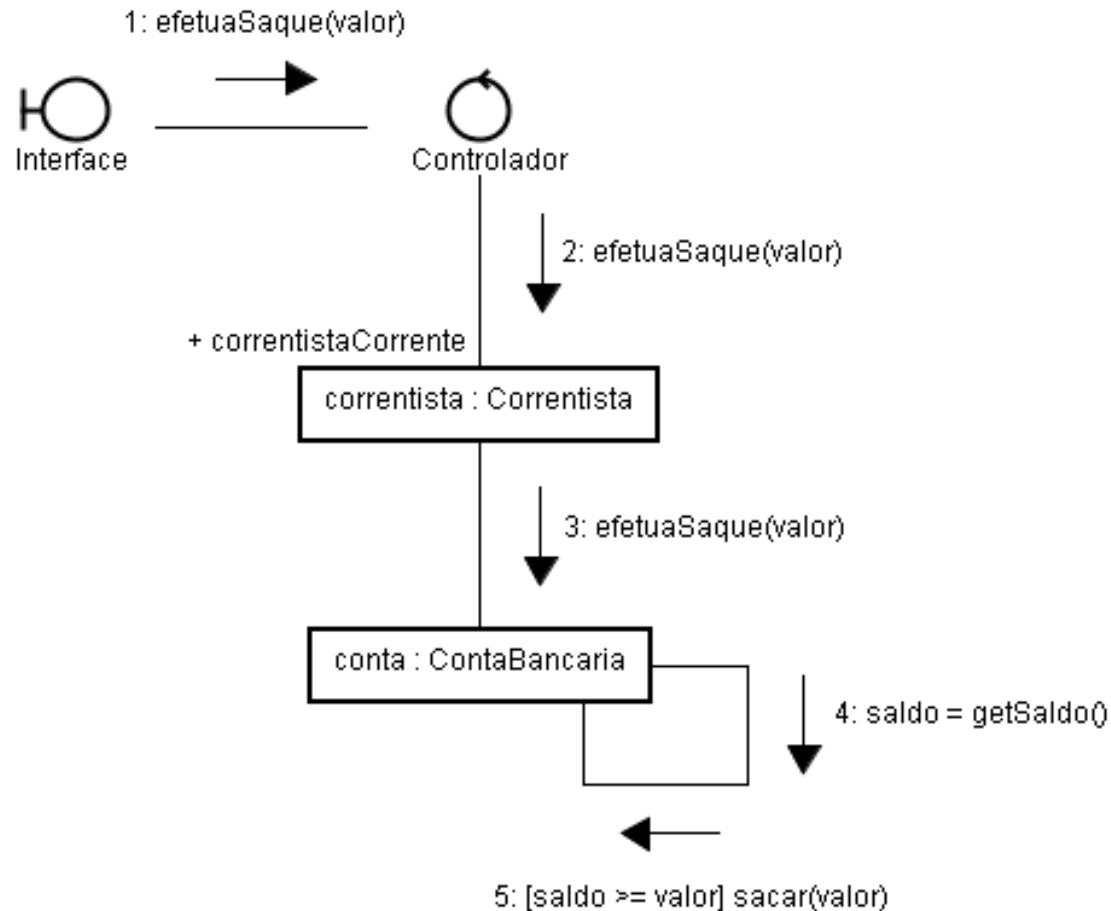
- Analisar os diagramas de seqüência, identificando as possíveis operações ou consultas complexas
- Para cada operação ou consulta mapeada, deve ser descrita todas as ações relacionadas
- Identificar as classes responsáveis por cada ação
  - Analisar questões como:
    - Coerência
    - Baixo acoplamento

# Diagrama de Colaboração

- Diagrama de colaboração para a operação efetuarSaque



# Diagrama de Colaboração



# Modelo de Colaboração

Análise de Sistemas e  
Requisitos de Software II

Aula 12

Allan Rodrigo Leite