

## Diagramas de Interação: Sequência e Colaboração

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>  
[dcc603@gmail.com](mailto:dcc603@gmail.com)

09 Abril 2012

## Tópicos da Aula

- Diagrama de Sequência
- Diagrama de Colaboração

## Alguns Diagramas UML

- Diagramas Estruturais (Estáticos)
  - Diagrama de Casos de Uso
  - Diagrama de Classes
  - Diagramas de Objetos
  - Diagrama de Componentes, etc.
- Diagramas Dinâmicos
  - Diagrama de Sequência
  - Diagrama de Colaboração
  - Diagrama de Estados
  - Diagrama de Atividades, etc.

Aulas anteriores

Aula de Hoje:  
Diagramas de Interação

## Diagramas Principais

- A maioria dos usuários UML acreditam que cinco diagramas representam a essência de qualquer sistema
  - Diagrama de Casos de Uso
  - Diagrama de Classes
  - Diagrama de Sequência
  - Diagrama de Atividades
  - Diagrama de Estados

## Diagramas de Interação

- Representa 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

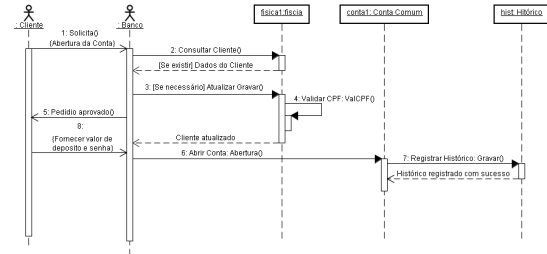
## Tipos de Diagramas de Interação

- Diagrama de Sequência
  - Enfatiza a ordem temporal das ações
- Diagrama de Colaboração
  - 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

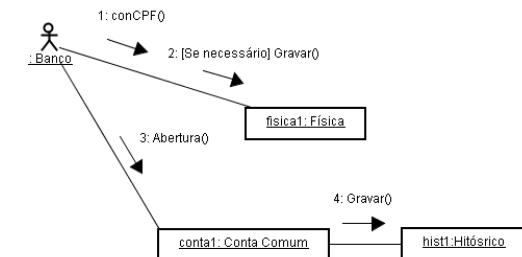
## Diagrama de Sequência



## Diagrama de Colaboração

- Não se preocupa com a temporalidade
- Define a estrutura de como os objetos estão vinculados
  - Como o Diagrama de Classes
- Indica quais mensagens são trocadas entre objetos
  - Como o Diagrama de Sequência

## Diagrama de Colaboração



## 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

## [ Atores ]

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



## [ Objetos ]

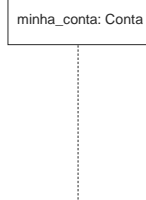
- Indicam instâncias de uma classe envolvidas no processo
  - As classes são mostradas em Diagramas de Classes
- Representados por retângulos
  - Nome do objeto (em minúsculo)
  - Nome da classe (inicial maiúscula)
  - Separação por dois pontos (:)

minha\_conta: Conta

Um retângulo contendo o texto "minha\_conta: Conta", onde "minha\_conta" é o nome do objeto em minúsculo e "Conta" é o nome da classe em maiúscula, separados por dois pontos.

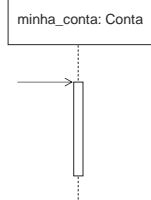
## [ Linha de Vida ]

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



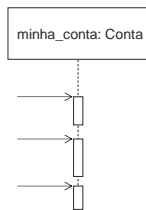
## [ Ativação do Objeto ]

- 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.



## [ Troca de Mensagens ]

- Representam a comunicação entre objetos e/ou atores
- 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

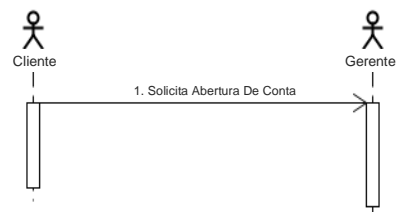
## [ Tipos de Mensagens ]

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

## [ Mensagens 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

## [ Comunicação entre atores ]



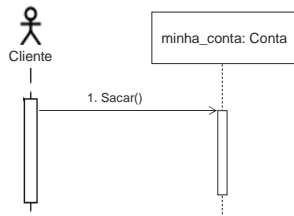
## [ Tipos de Mensagens ]

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

## [ 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

## Comunicação entre ator e objeto



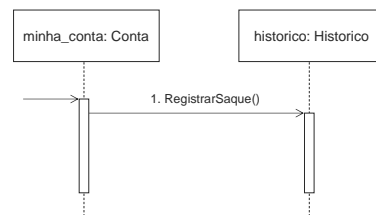
## Tipos de Mensagens

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

## 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

## Comunicação entre objetos



## Tipos de Mensagens

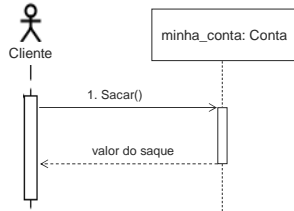
- Ator para Ator
- Ator para Objeto
- Objeto para Objeto
- **Objeto para Ator**

## Mensagem Objeto-Ator

- Indica a resposta de uma solicitação de serviço feita pelo ator
  - O objeto envia uma **mensagem de retorno**
- Representadas por linhas tracejadas
  - Pode conter legenda indicando o retorno

←-----

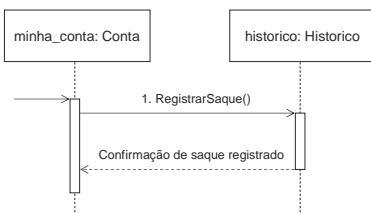
## Comunicação entre objeto e ator



## 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

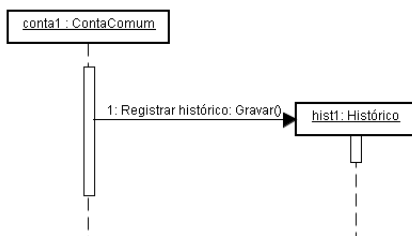
## Mensagem de Retorno



## Instanciando 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

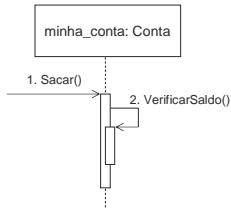
## Exemplo de instanciação



## Auto-chamadas de um objeto

- 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

## Exemplo de auto-chamadas



## Detalhando um Caso de Uso

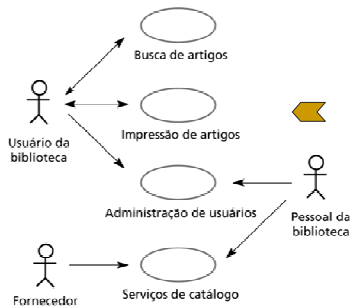
## Detalhe de um Caso de Uso

- 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 é um processo disparado pelo o usuário

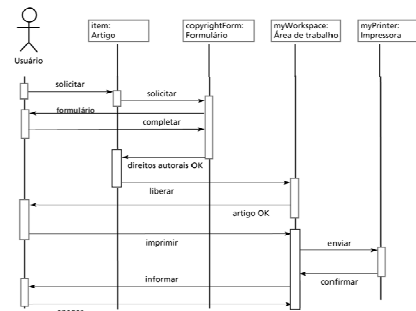
## Caso de Uso x Sequência

- 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>>`

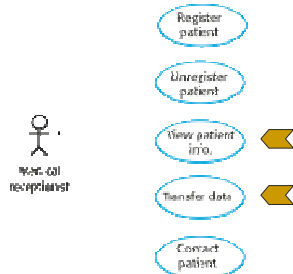
## Exemplo: Casos de Uso



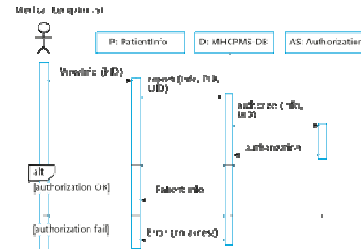
## Sequência: Impressão de Artigo



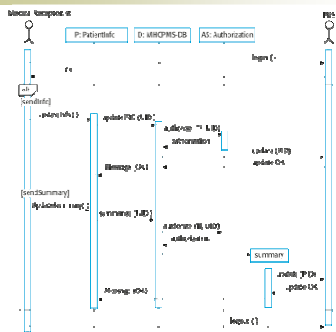
## Exemplo 2: Casos de Uso



## Sequência: *View Patient Info*



## Sequência: *Data Transfer*



## Diagrama de Colaboração

## Sequência x Colaboração

- Semelhante ao Diagrama de Sequência
  - Diagrama de Sequência se concentra na ordem temporal em que as mensagens são trocadas
- **Diagrama de Colaboração** se preocupa com a organização estrutural dos objetos

## Objetos

- Semelhante a representação do Diagrama de Sequência
  - Não há linha da vida nem ativação

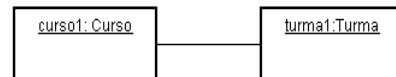
minha\_conta: Conta

## Vínculos

- 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

## Representação de Vínculo

- Um vínculo é representado por uma linha unindo dois objetos
  - Deve existir equivalente no Diagrama de Classes



## Mensagens

- As mesmas definidas no Diagrama de Sequência
  - Geralmente representam chamadas de métodos
- No Diagrama de Colaboração não existe a preocupação com a ordem
  - O importante é quem dispara a mensagens

## Mensagens

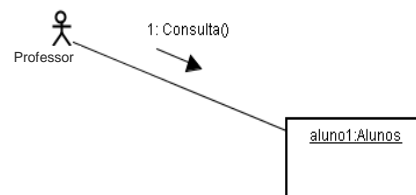
- Não existe mensagens de retorno



## Atores

- Mesmos do Diagrama de Sequência
  - Consequentemente, os mesmos do Diagrama de Casos de Uso
- Um ator possui vínculos com outros objetos ou outros atores
- Um ator envia e recebe mensagens através de vínculos
  - Assim como objetos

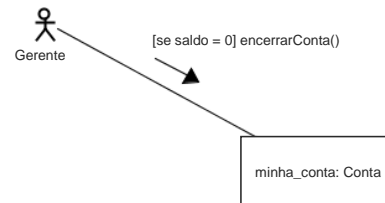
## Exemplo de Vínculo de Ator



## Condições

- Indica que uma mensagem só será enviada quando uma determinada condição for satisfeita
- A condição vem entre colchetes antes da mensagem

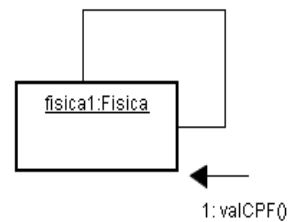
## Exemplo de Condição



## Auto-chamadas

- Um objeto pode disparar uma mensagem para si mesmo
  - Como no Diagrama de Sequência
- A mensagem parte do objeto para si próprio
- Indica que o objeto tem que fazer aquela tarefa para completar o serviço solicitado

## Exemplo de auto-chamada



## Referências

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
  - Seções 5.1 a 5.4 (Modelagem de sistemas)
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.
  - Capítulos 15 e 18
- M. Fowler. **UML Essencial**, 3a Edição. Bookmann, 2004.
  - Capítulo 4

## Próxima Aula

- Laboratórios 2011 / 2012
  - Se o CRC concluir a instalação do ArgoUML