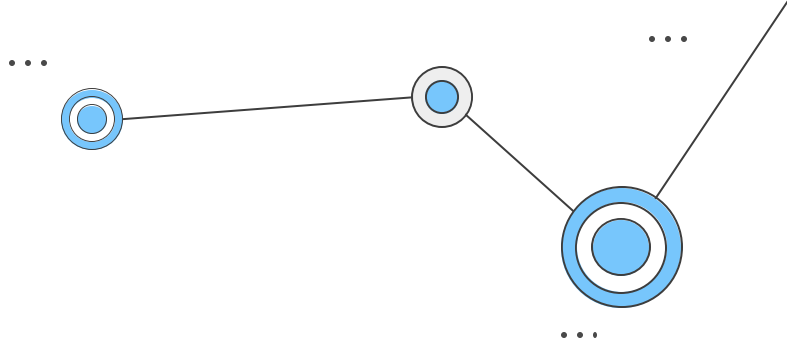




PUC Minas



Projeto de Software

Prof. Dr. João Paulo Aramuni



Unidade 4

Modelagem de Interação

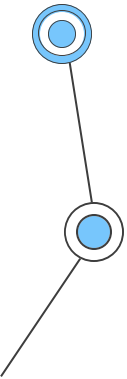
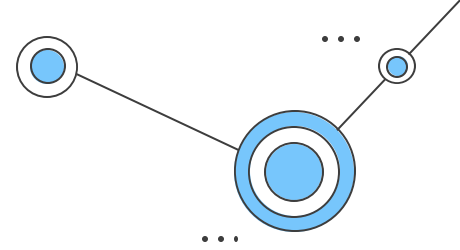
PDS - Manhã /Noite



Modelagem de Interação

Sumário

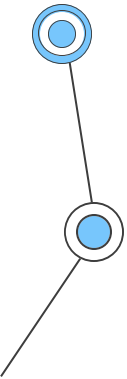
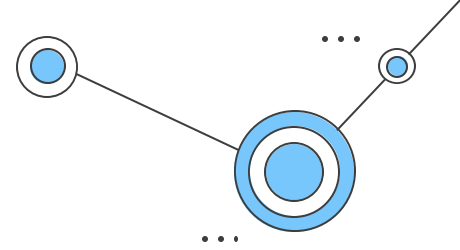
- Conceito modelo de interação
- Diagrama de sequência
- Diagrama de comunicação
- Diagrama estrutura geral
- Considerações



Modelagem de Interação

Realização de caso de uso

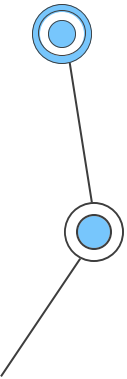
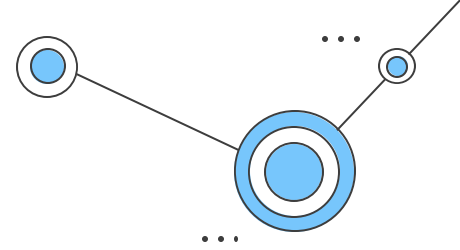
- Realização de um caso de uso mostra a interação entre objetos para dar suporte à funcionalidade de um caso de uso
- A realização de um caso de uso descreve o comportamento do ponto de vista interno ao sistema
- A realização é representada pelos modelos de interação



Modelagem de Interação

Objetivos - Modelagem de Interação

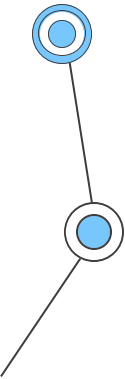
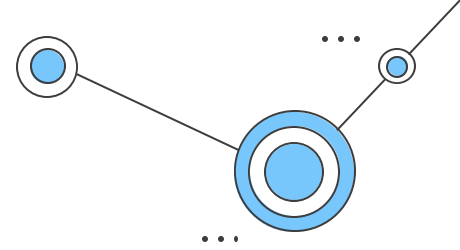
- Descrever o comportamento dinâmico e mostrar como capturá-lo em um modelo.
- Demonstrar como construir, ler e interpretar:
 - Um diagrama de Sequência
 - Um diagrama de Comunicação
 - Um diagrama de Visão Geral de Interação
- Explicar as semelhanças e diferenças entre os diagramas de comunicação e de sequência.



Modelagem de Interação

Motivação - Modelagem de Interação

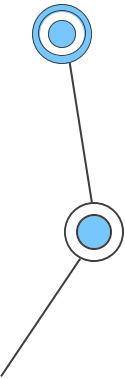
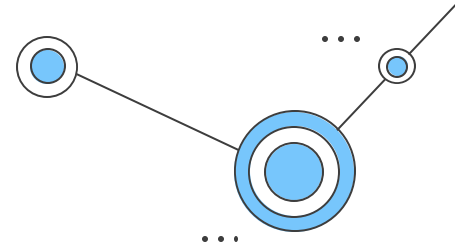
- Objetos são inúteis a menos que possam colaborar para resolver um problema.
 - Cada objeto é responsável pelo seu próprio comportamento e status.
 - Nenhum objeto pode realizar todas as responsabilidades por conta própria.
- Como é que os objetos interagem uns com os outros?
 - Eles interagem através de mensagens



Modelagem de Interação

Motivação - Modelagem de Interação

- O objetivo dos modelos vistos para análise de requisitos (diagrama de classes e diagrama de caso de uso) é fornecer um entendimento do problema correspondente ao sistema a ser desenvolvido.
- Entretanto, esses modelos deixam algumas perguntas sem respostas.
- No modelo de casos de uso:
 - ✓ Quais são as operações que devem ser executadas internamente ao sistema?
 - ✓ A que classes estas operações pertencem?
 - ✓ Quais objetos participam da realização deste caso de uso?

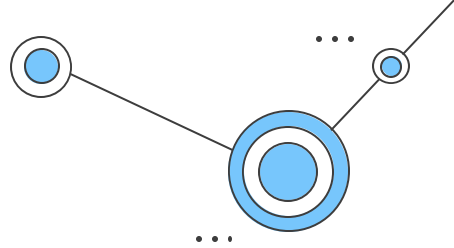


Modelagem de Interação

Motivação - Modelagem de Interação

No modelo de classes de análise:

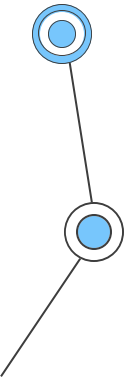
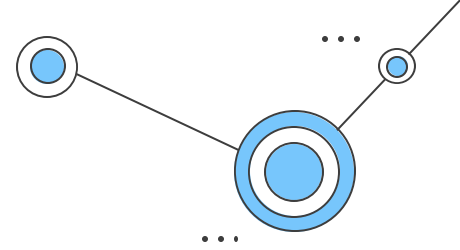
- De que forma os objetos colaboram para que um determinado caso de uso seja realizado?
- Em que ordem as mensagens são enviadas durante esta realização?
- Que informações precisam ser enviadas em uma mensagem de um objeto a outro?
- Será que há responsabilidades ou mesmo classes que ainda não foram identificadas?



Modelagem de Interação

Motivação - Modelagem de Interação

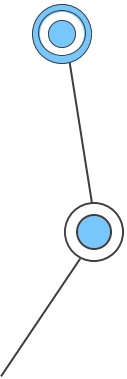
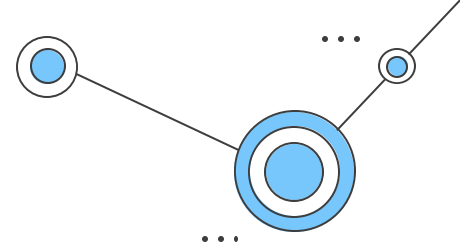
- Para responder às questões anteriores, o modelo de interações deve ser criado.
- Esse modelo representa mensagens trocadas entre objetos para a execução de cenários dos casos de uso do sistema.
- A construção dos diagramas de interação é uma consolidação do entendimento dos aspectos dinâmicos do sistema.
- A modelagem de interações é uma parte da modelagem dinâmica de um SS00.



Modelagem de Interação

Motivação - Modelagem de Interação

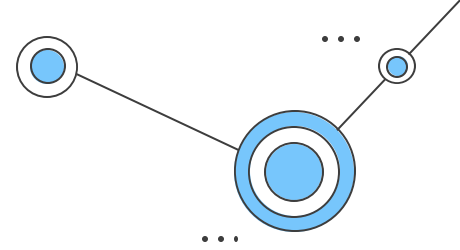
- Diagramas de interação representam como o sistema age internamente para que um ator atinja seu objetivo na realização de um caso de uso.
- A modelagem de um SS00 normalmente contém diversos diagramas de interação.
- O conjunto de todos os diagramas de interação de um sistema constitui o seu modelo de interações.



Modelagem de Interação

Motivação - Modelagem de Interação

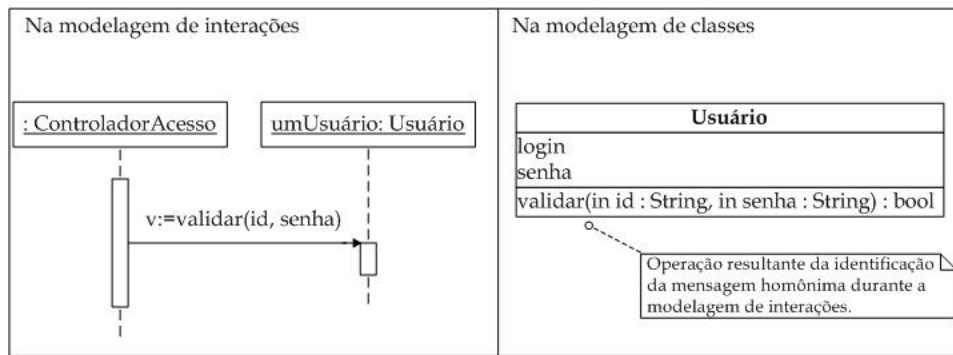
- Os objetivos da construção do modelo de interação são:
 - Obter informações adicionais para completar e aprimorar outros modelos (principalmente o modelo de classes)
 - ✓ Quais as operações de uma classe?
 - ✓ Quais os objetos participantes da realização de um caso de uso (ou cenário deste)?
 - ✓ Para cada operação, qual a sua assinatura?
 - ✓ Uma classe precisa de mais atributos?
- Fornecer aos programadores uma visão detalhada dos objetos e mensagens envolvidos na realização dos casos de uso.



Modelagem de Interação

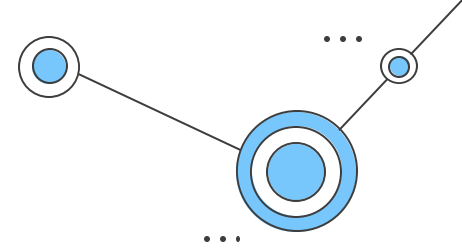
Motivação - Modelagem de Interação

- Qual o objetivo da construção dos diagramas de interação?
 - Identificar **mensagens** e, em última análise, **responsabilidades** (operações e atributos)



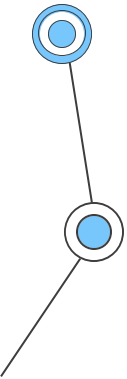
- Uma mensagem implica na existência de uma operação no objeto receptor. A resposta do objeto receptor ao recebimento de uma mensagem é a execução da operação correspondente.

Modelagem de Interação



Tipos de diagrama de interação

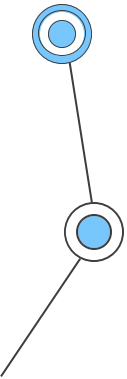
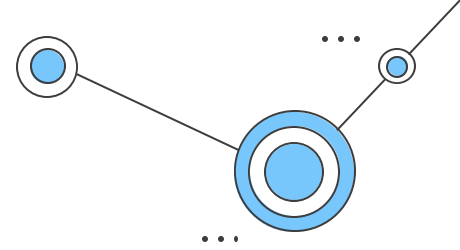
- Há quatro tipos de diagrama de interação na UML 2.0: diagrama de sequência, diagrama de comunicação, diagrama de visão geral da interação e diagrama de temporização. Nessa disciplina serão estudados os três primeiros
 - O diagrama de sequência e o diagrama de comunicação são equivalentes.
- **Diagrama de sequência:** foco nas mensagens enviadas no decorrer do tempo.
- **Diagrama de comunicação:** foco nas mensagens enviadas entre objetos que estão relacionados.
- **Diagrama de visão geral de interação:** Pode ser utilizado para apresentar uma visão geral de diversas interações entre objetos, cada uma delas representada por um diagrama de interação. Diagrama é útil para modularizar a construção do diagramas de sequência (ou de comunicação).



Modelagem de Interação

Diagrama de Sequência

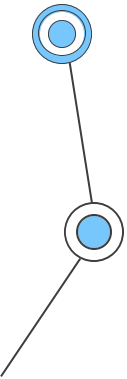
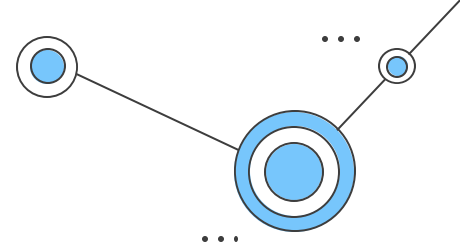
- Diagrama comportamental.
- O objetivo é apresentar as interações entre objetos na ordem temporal em que elas ocorrem.
- Identifica quais mensagens devem ser disparadas entre os objetos e envolvidos e em que ordem.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

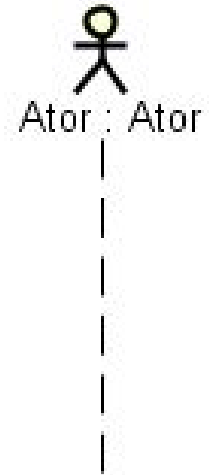
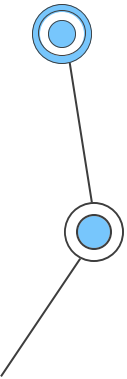
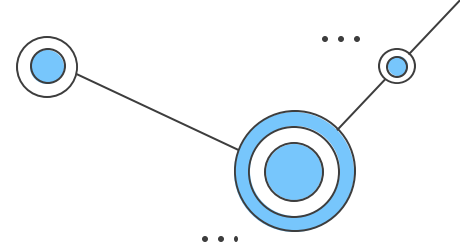
- Elementos gráficos básicos em um diagrama de sequência:
 - Atores
 - Objetos e classes
 - Mensagens
 - Linhas de vida e focos de controle
 - Criação e destruição de objetos
 - Iterações



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

- Ator
 - Entidade externa que
 - ✓ Interage com o sistema;
 - ✓ Solicita serviços.
 - ✓ Tem a mesma representação do diagrama de caso de uso, porém, contendo uma linha de vida (mostrado nos próximos slides)



Modelagem de Interação

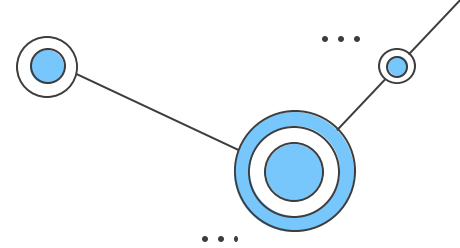
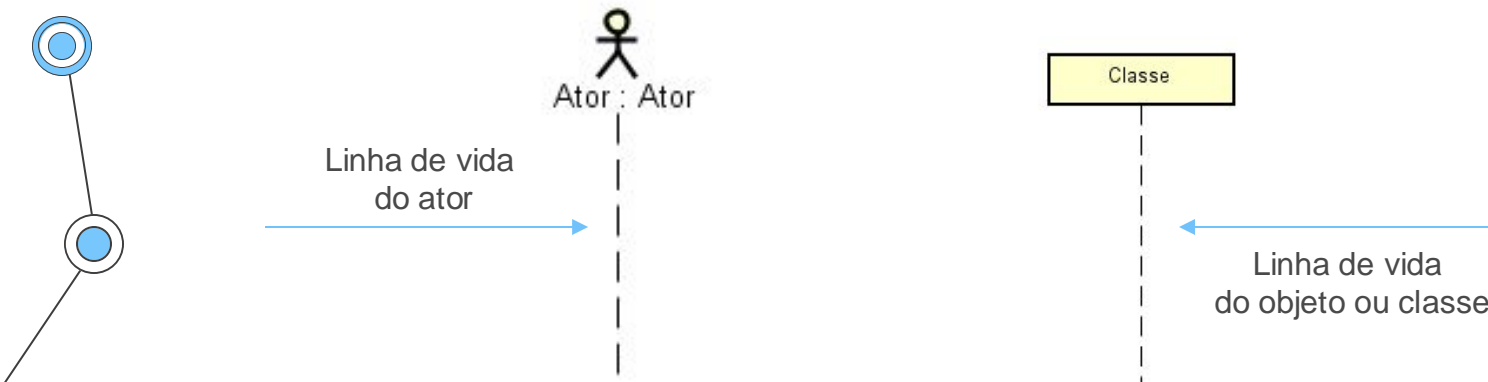


Diagrama de Sequência: Elementos gráficos

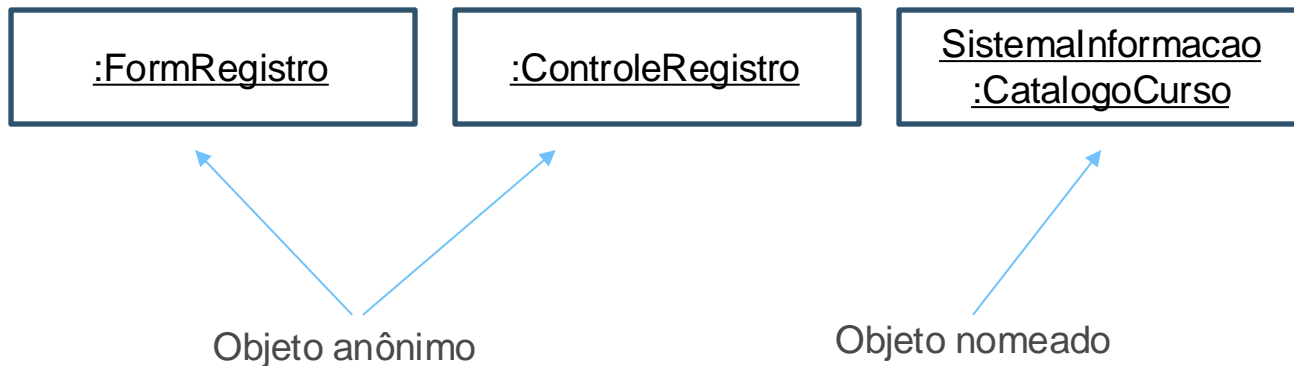
- Linha de Vida (lifelines)
 - Linha vertical tracejada abaixo do objeto;
 - Representa o tempo em que um objeto existe durante o processo;
 - Das linhas de vida partem as mensagens



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

- Objetos
 - são desenhados como retângulos com nomes sublinhados



Modelagem de Interação

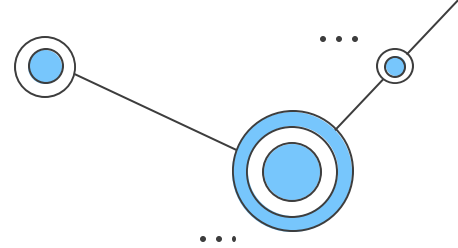
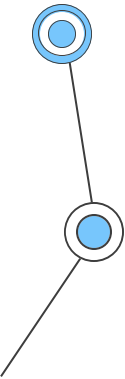


Diagrama de Sequência: Elementos gráficos

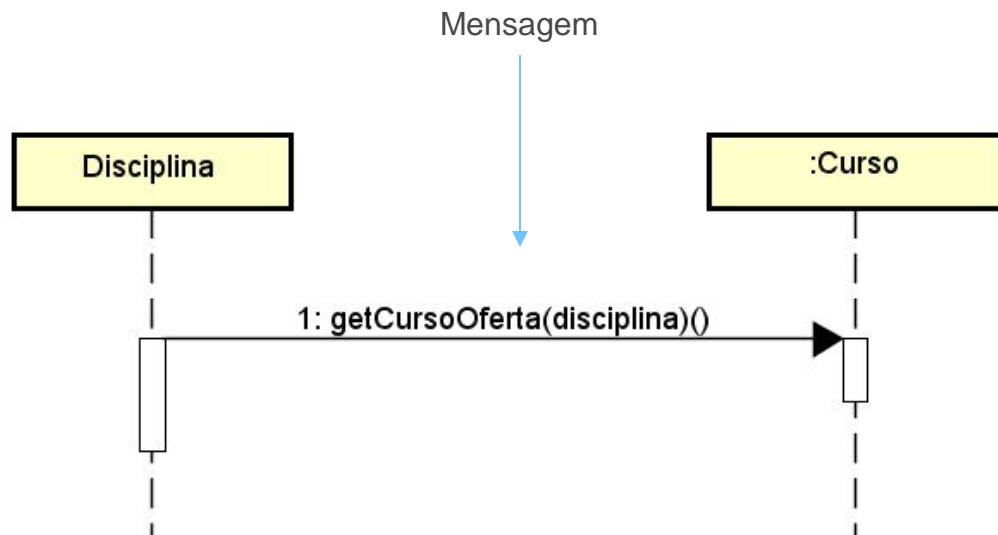
- Mensagem
 - O conceito básico da interação entre objetos é a mensagem.
 - Um sistema OO é uma rede de objetos que trocam mensagens.
 - ✓ Funcionalidades são realizadas pelos objetos, que só podem interagir através de mensagens.
 - ✓ Um objeto envia uma mensagem para outro objeto quando o primeiro deseja que o segundo realize alguma tarefa.
 - O fato de um objeto “precisar de ajuda” indica a necessidade de este enviar mensagens.
 - Na construção de diagramas de interação, mensagens de um objeto a outro implicam em operações que classes devem ter.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

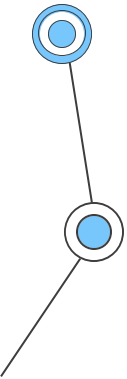
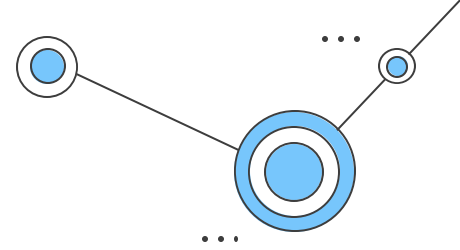
- Mensagens
 - Os objetos interagem com mensagens



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

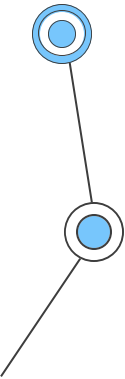
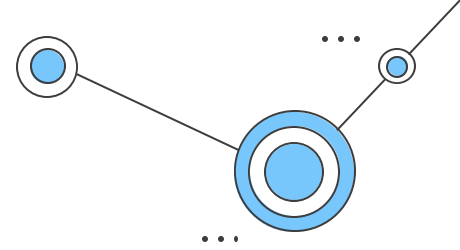
- Tipos de Mensagem:
 - As mensagens podem ser:
 - Simples;
 - Síncrona;
 - Assíncrona;
 - Retorno.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

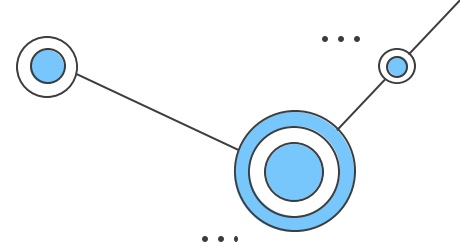
- Mensagem Simples
 - Mostra como o controle é passado de um objeto para outro sem descrever qualquer detalhe sobre a comunicação.
 - ✓ Não se decide se é síncrona, de retorno ou assíncrona.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

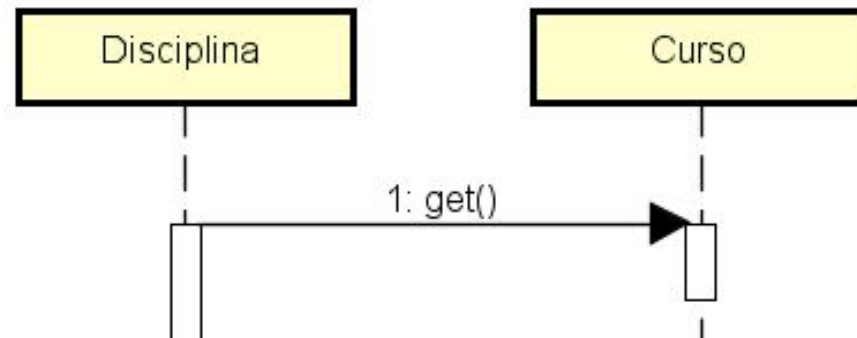
- Mensagem Síncrona
 - São mensagens que implicam um sincronismo rígido entre os estados do objeto que envia a mensagem e os do objeto de destino da mensagem.
 - Ou seja uma mensagem síncrona implica que o objeto que enviou a mensagem aguarde a conclusão do processamento da mensagem (entendida como um sinal de sincronismo) feito pelo objeto destino, para então prosseguir seu fluxo de execução.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

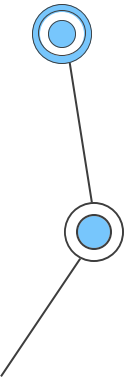
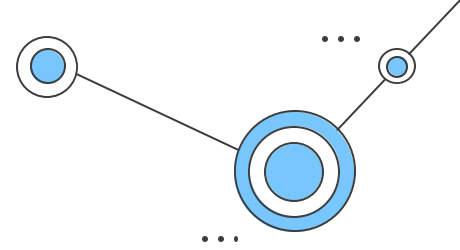
- Exemplo de Mensagem Síncrona:



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

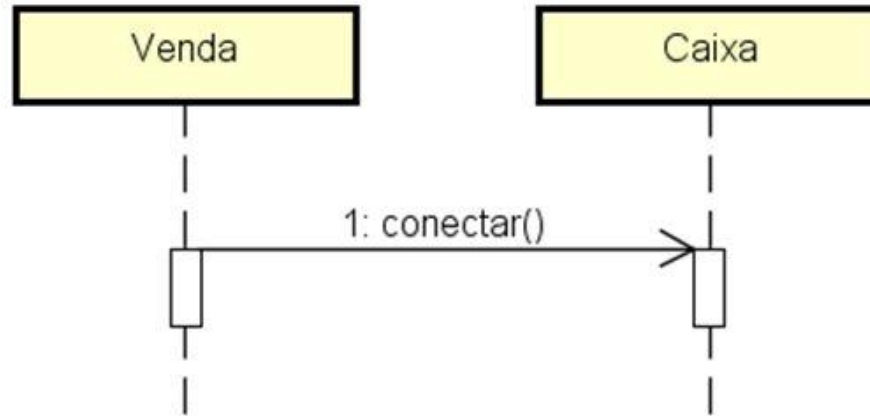
- Mensagem Assíncrona
 - São mensagens enviadas de um objeto a outro sem que haja uma dependência de estado entre os dois objetos.
 - O objeto de origem envia a mensagem e prossegue seu processamento independentemente do tratamento da mensagem feita no objeto destino.
 - A seta é aberta.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

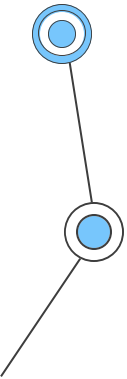
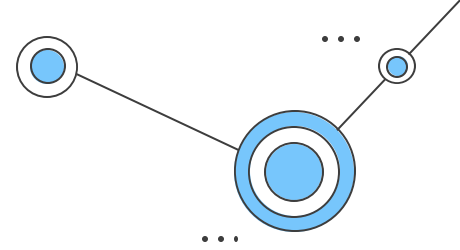
- Exemplo de Mensagem Assíncrona:



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

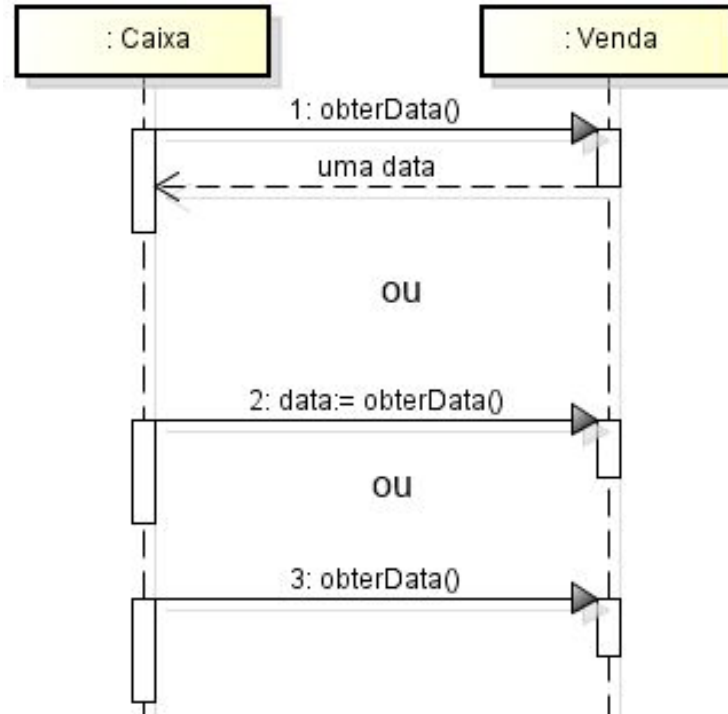
- Mensagens retorno
 - Identifica a resposta a uma mensagem para o objeto que a chamou. Pode retornar informações do método chamado ou apenas um valor indicando se o método foi executado ou não;
 - Além de resposta ao ator, mensagens de retorno podem indicar respostas para objetos;
 - Mensagens de retorno são opcionais em Diagramas de Sequência;
 - Pode retornar informações específicas do método chamado;
 - Usa uma linha tracejada com seta.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

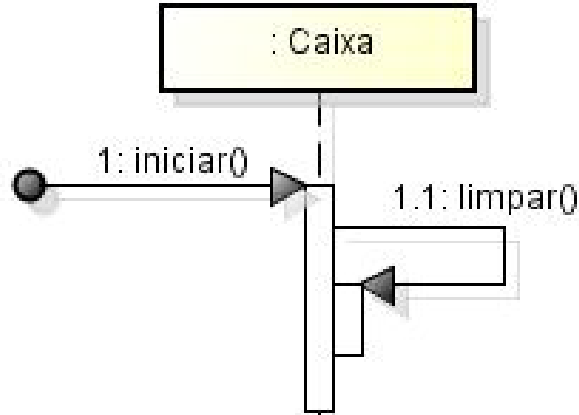
- Exemplo de mensagem de retorno:



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

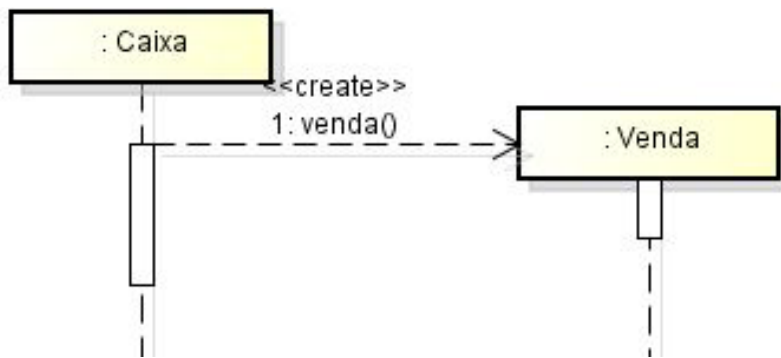
- Mensagens Self ou This
 - Autochamadas são mensagens que um objeto envia para si mesmo



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

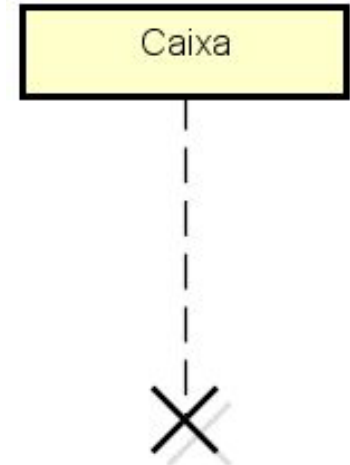
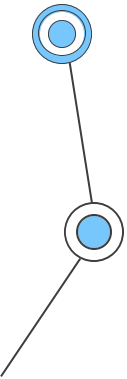
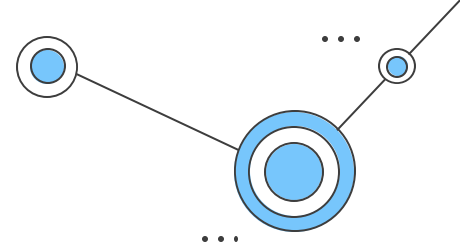
- Mensagens de criação de instância
 - Quando a mensagem cria um novo objeto, a seta atinge o retângulo que representa o objeto, indicando que a mensagem representa um método construtor e que o objeto passa a existir a partir daquele momento



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

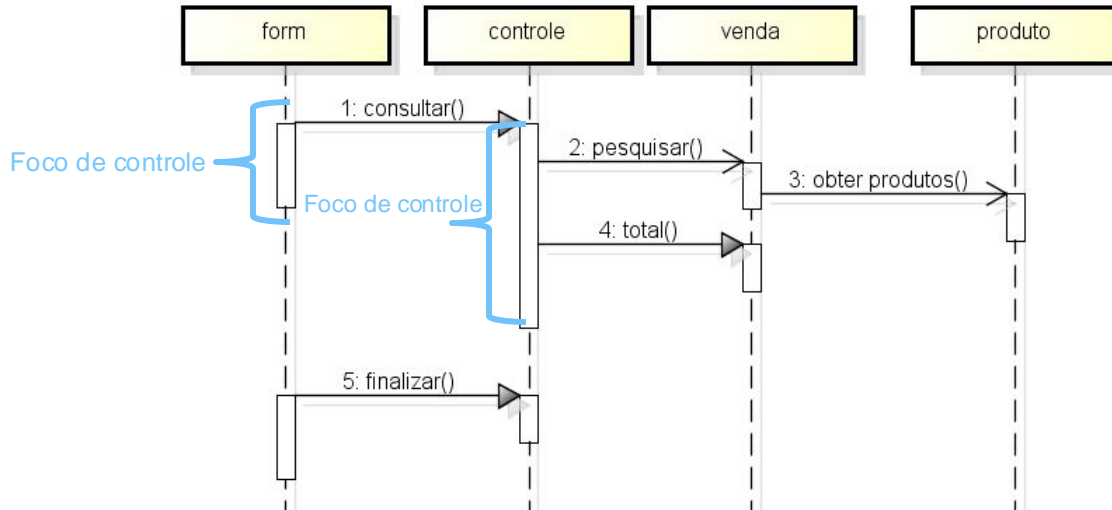
- Destruição de objetos
 - Destruição de objeto é representada por um X no fim da linha de vida do objeto;
 - Mensagem de destruição pode ter estereótipo «destroy»;
 - Pode ocorrer na recepção de mensagem ou no retorno de chamada;
 - Objeto pode auto destruir-se.



Modelagem de Interação

Diagrama de Sequência: Elementos gráficos

- Foco de controle ou ativação
 - Indica os períodos em que um determinado objeto está participando ativamente do processo, ou seja, indica os momentos que um objeto está executando um ou mais métodos utilizados em um processo específico.



Modelagem de Interação

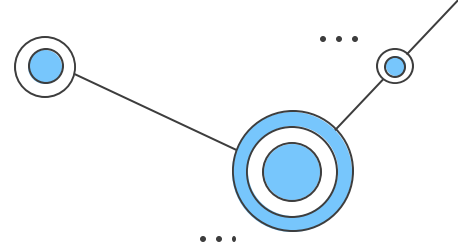
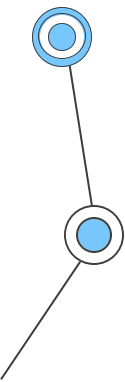


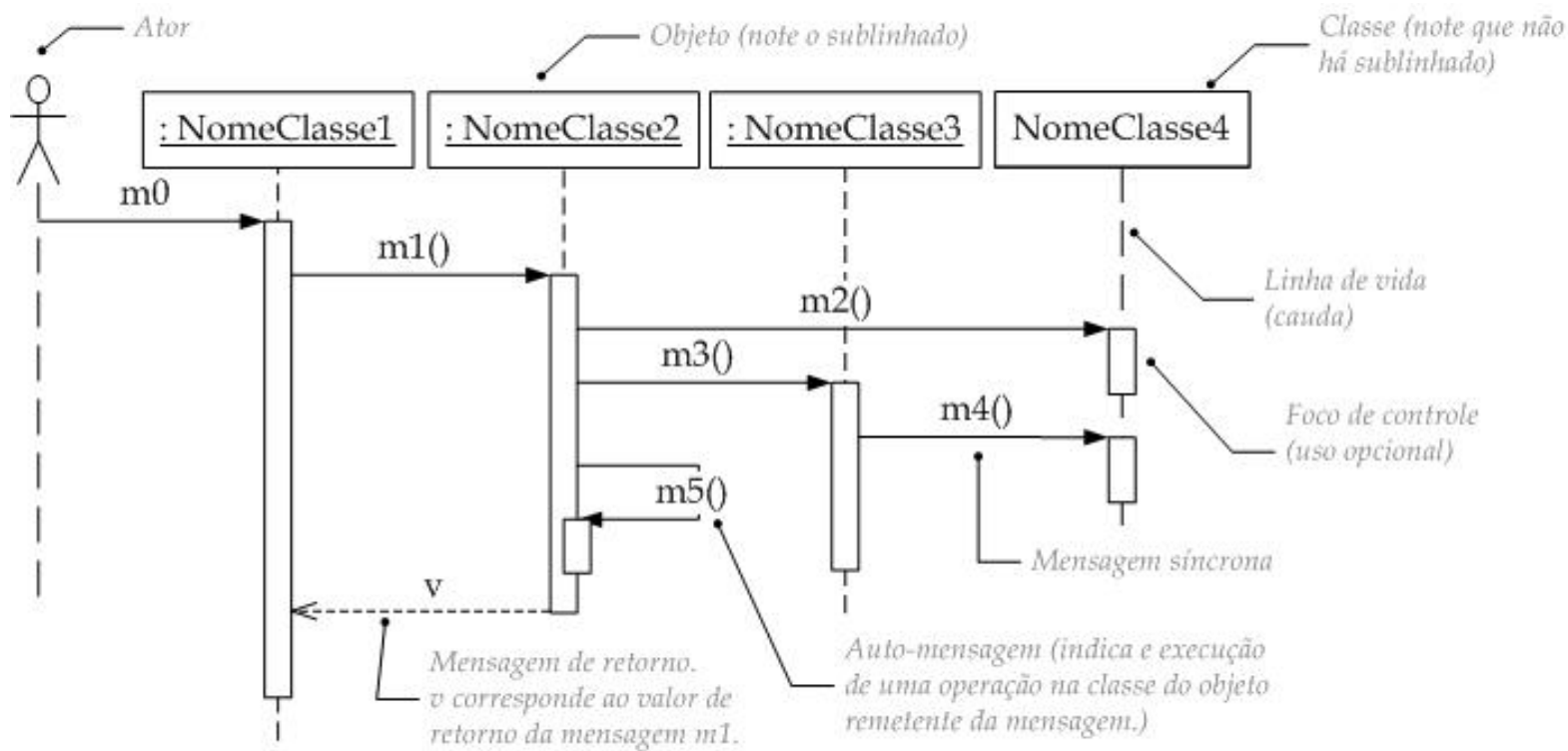
Diagrama de Sequência: Construção

- Os objetos participantes da interação são organizados na horizontal.
- Abaixo de cada objeto existe uma linha (linha de vida)
- Cada linha de vida possui o seu foco de controle.
 - Quando o objeto está fazendo algo.
- As mensagens entre objetos são representadas com linhas horizontais rotuladas partindo da linha de vida do objeto remetente e chegando a linha de vida do objeto receptor.
- A posição vertical das mensagens permite deduzir a ordem na qual elas são enviadas.
- Ordem de envio de mensagens em um diagrama de sequência pode ser deduzida a partir das expressões de sequência.
- Criação e destruição de objetos podem ser representadas.



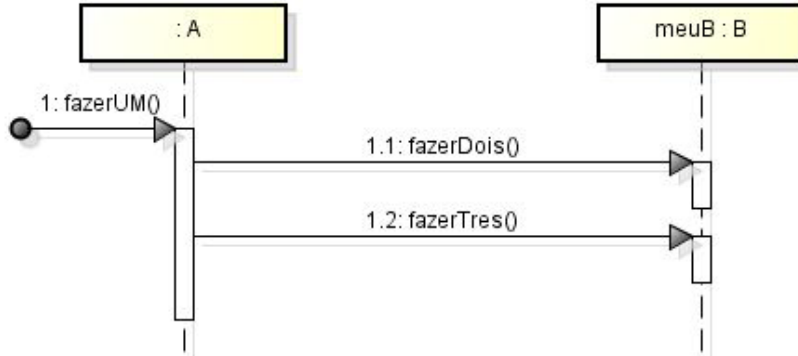
Modelagem de Interação

Diagrama de Sequência: Construção



Modelagem de Interação

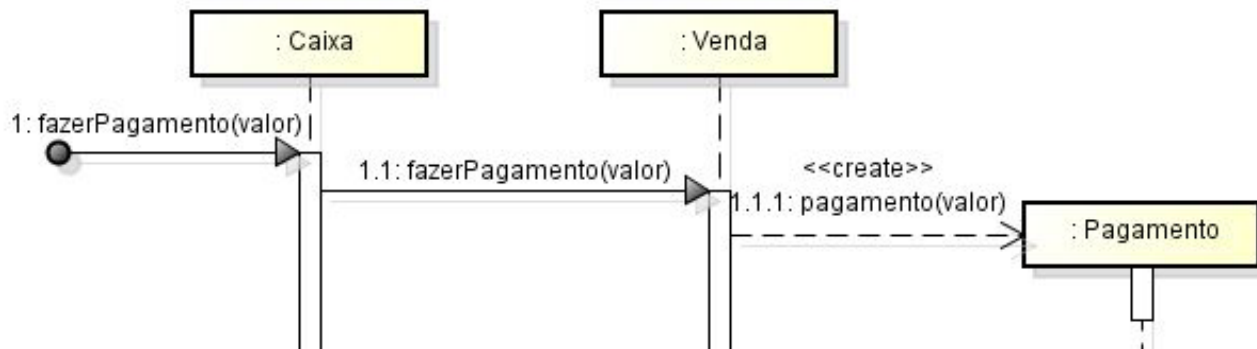
Diagrama de Sequência: Exemplo 1



```
public class A {
    private B meuB = new B();
    public void fazerum() {
        meuB.fazerum();
        meuB.fazerdois();
    }
    ... }
}
```

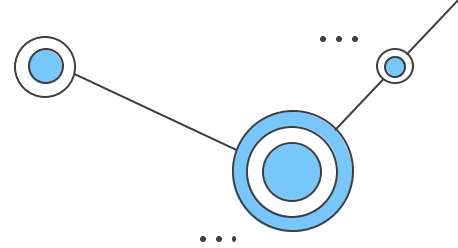
Modelagem de Interação

Diagrama de Sequência: Exemplo 2



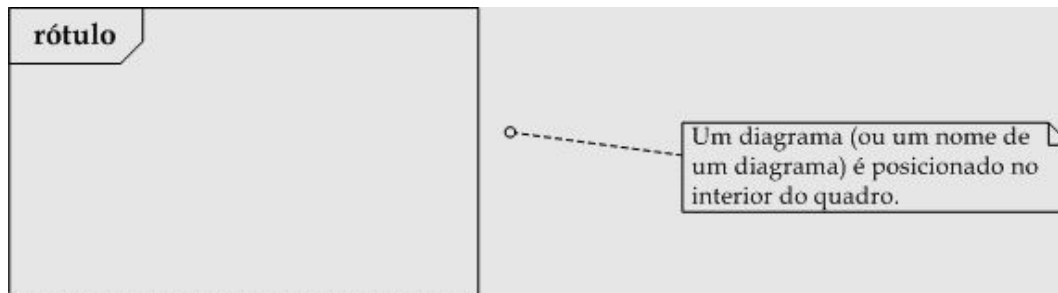
```
public class Venda {
    private Pagamento pagamento;
    public void fazerPagamento(moeda dinheiroentregue) {
        pagamento = new Pagamento(dinheiroentregue)
    }
    ... }
```

Modelagem de Interação



Modularização de Interações: Quadros (molduras) de interação

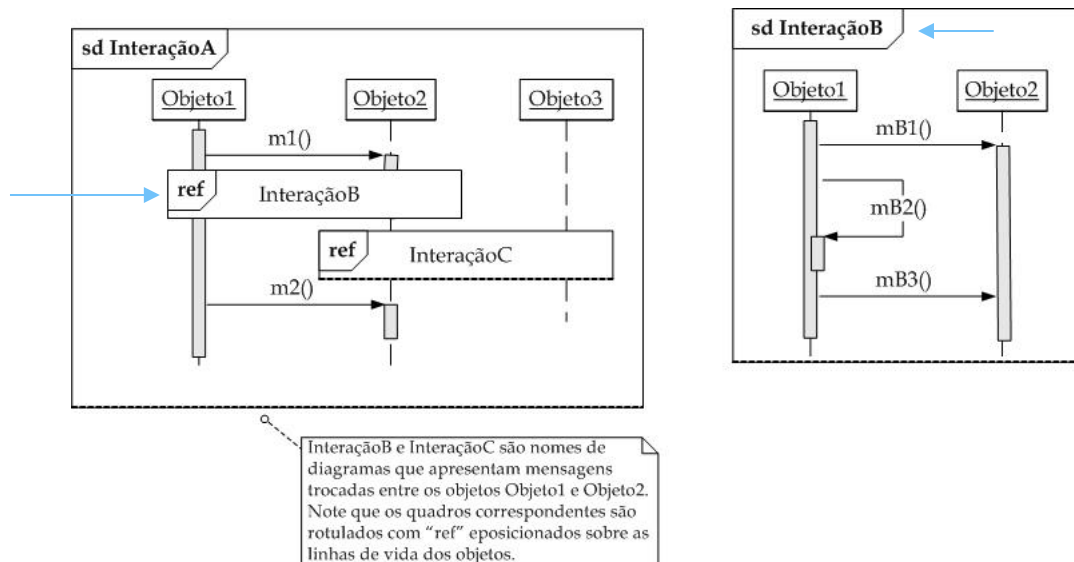
- Elemento gráfico, que serve para modularizar a construção de diagramas de sequência (ou de comunicação).
- Objetivos específicos:
 - Dar um nome ao diagrama que aparece dentro do quadro;
 - Fazer referência a um diagrama definido separadamente;
 - Definir o fluxo de controle da interação.
- Notação:



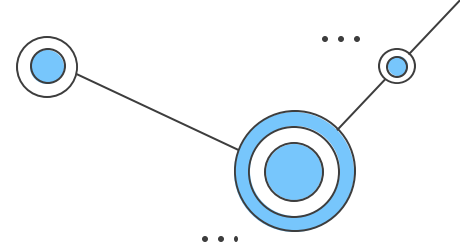
Modelagem de Interação

Modularização de Interações: Ocorrência de interação

- Possibilita que uma interação possa ser referenciada por meio do operador Ref (Referred - referido)
- Significa que se deve procurar por um diagrama cujo nome é o mesmo do nome apresentado após o operador Ref



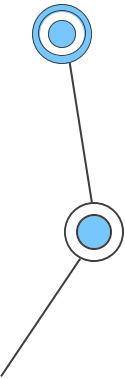
Modelagem de Interação



Modularização de Interações: Operadores

- Existem ainda outros operadores menos utilizados

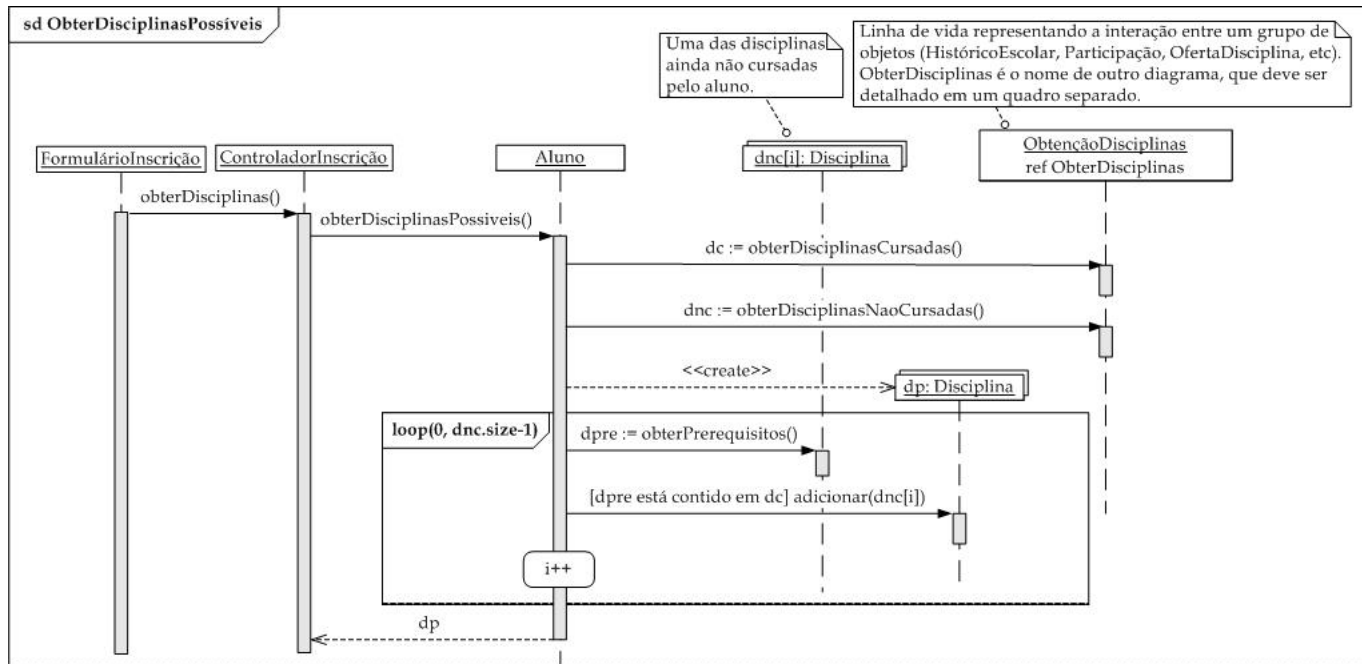
Operador moldura	Significado
alt	Abreviatura Alternatives. Fragmento alternativo que define uma escolha entre dois ou mais comportamentos.
loop	Fragmento de loop enquanto a guarda for verdadeira. Pode-se usar também escrever loop(n) para indicar a iteração de n vezes. Existe discussão de que a especificação será aperfeiçoada para definir um loop FOR , exemplo, loop(1,1,10).
opt	Abreviatura de Option. Fragmento que representa uma escolha de comportamento onde esse comportamento será ou não executado, não havendo uma escolha entre mais de um comportamento possível.
par	Abreviatura de Parallel. Determina que o fragmento combinado representa uma execução paralela de dois ou mais comportamento.
critical	Região crítica. Identifica uma operação atômica que não pode ser interrompida por outro processo até ser totalmente concluída.
Break	Quebra. Indica uma 'quebra' na execução normal do processo. É usado principalmente para modelar o tratamento de exceção.



Modelagem de Interação

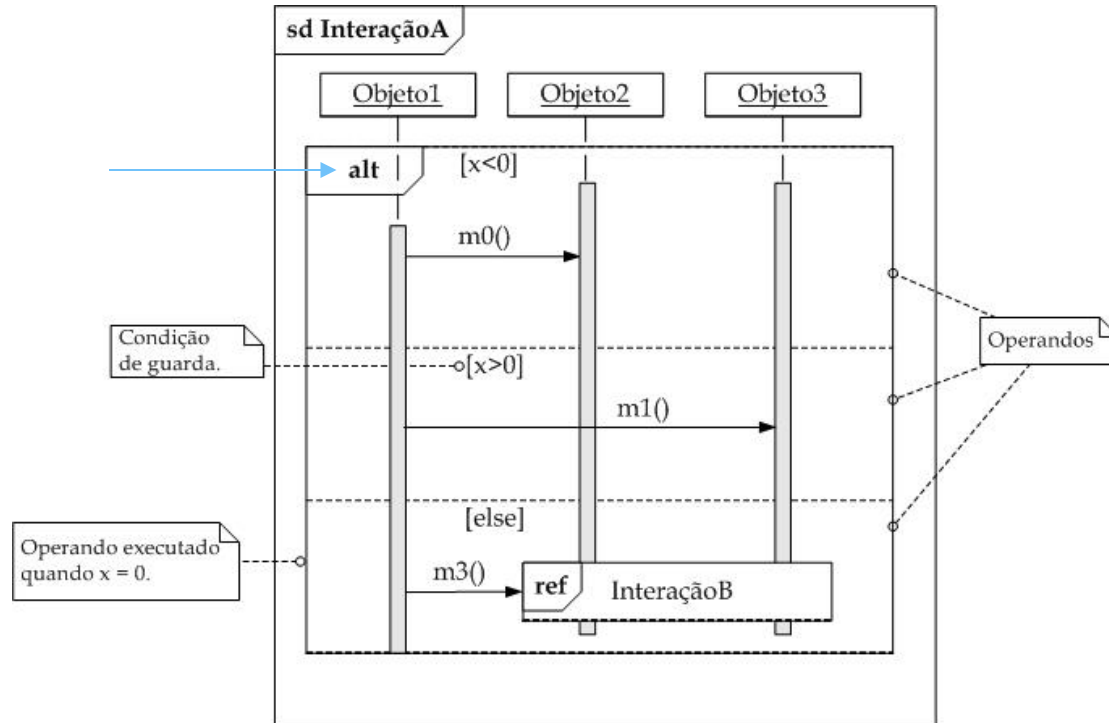
Modularização de Interações: Diagramas nomeados

- Dar um nome ao diagrama que aparece dentro do quadro



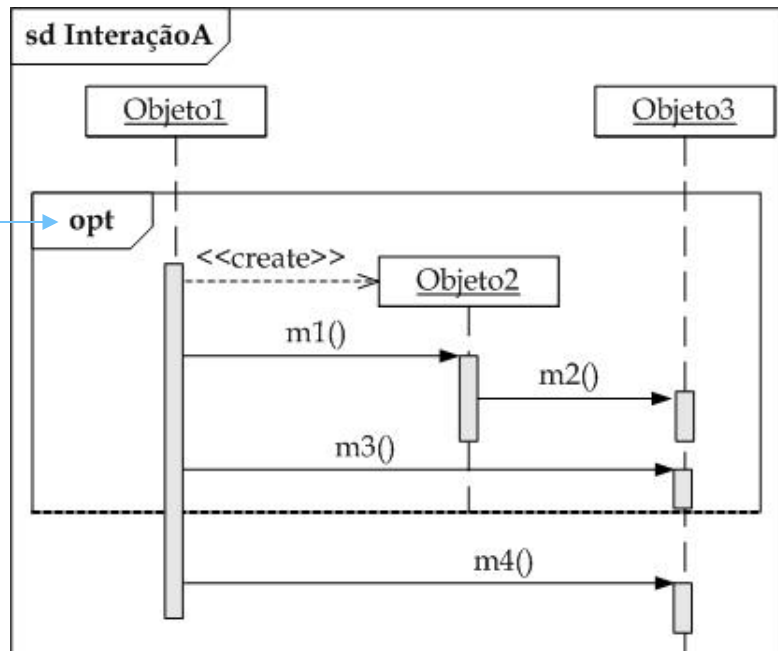
Modelagem de Interação

Modularização de Interações: Fluxo de controle - alternativas



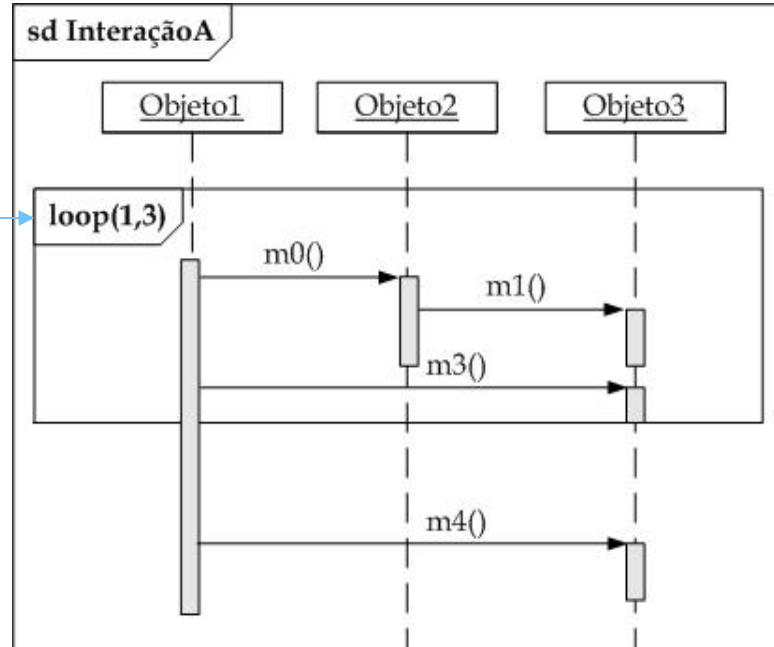
Modelagem de Interação

Modularização de Interações: Fluxo de controle - opções



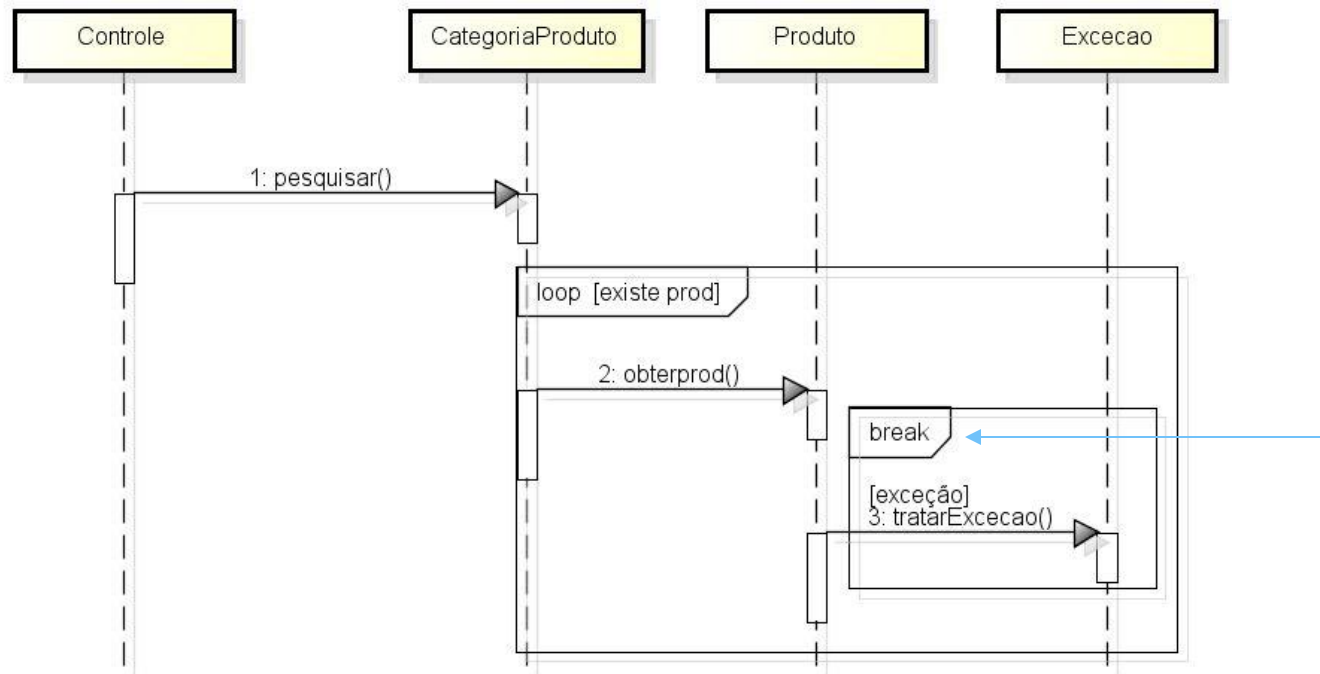
Modelagem de Interação

Modularização de Interações: Fluxo de controle - iterações

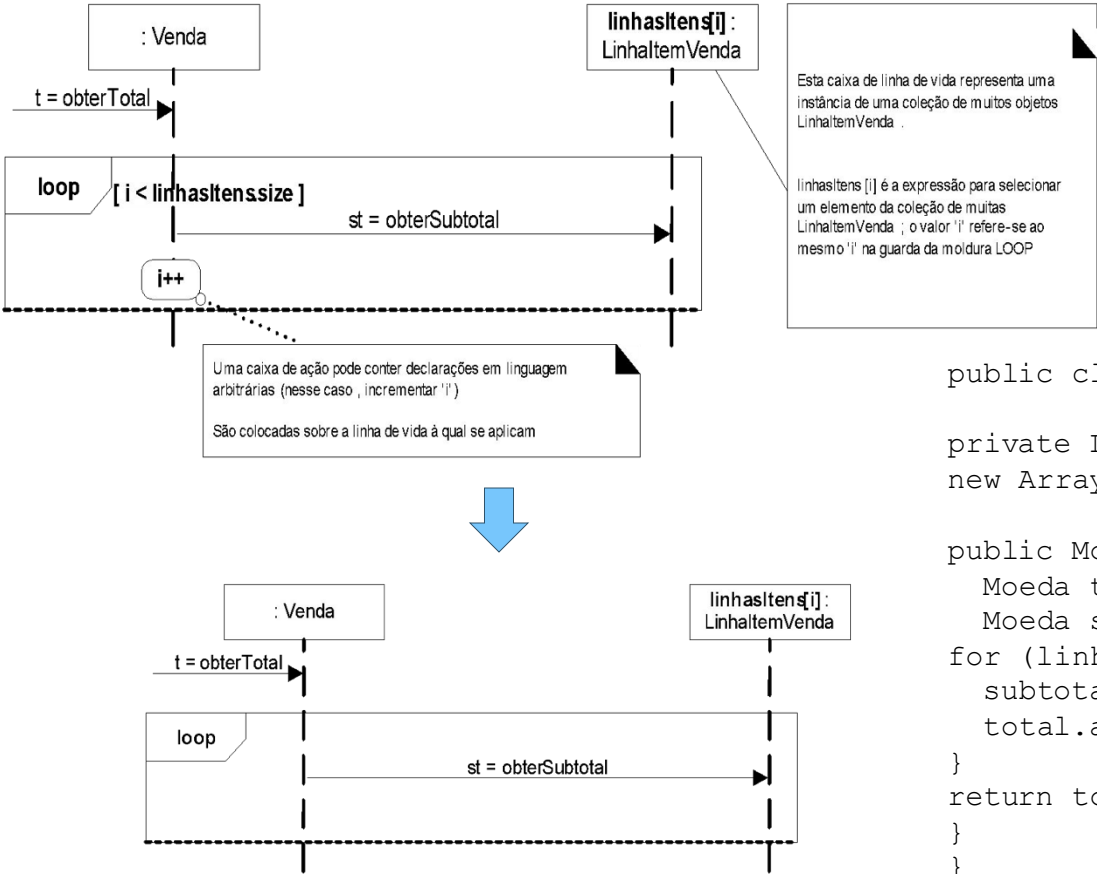


Modelagem de Interação

Modularização de Interações: Fluxo de controle – tratamento de exceção

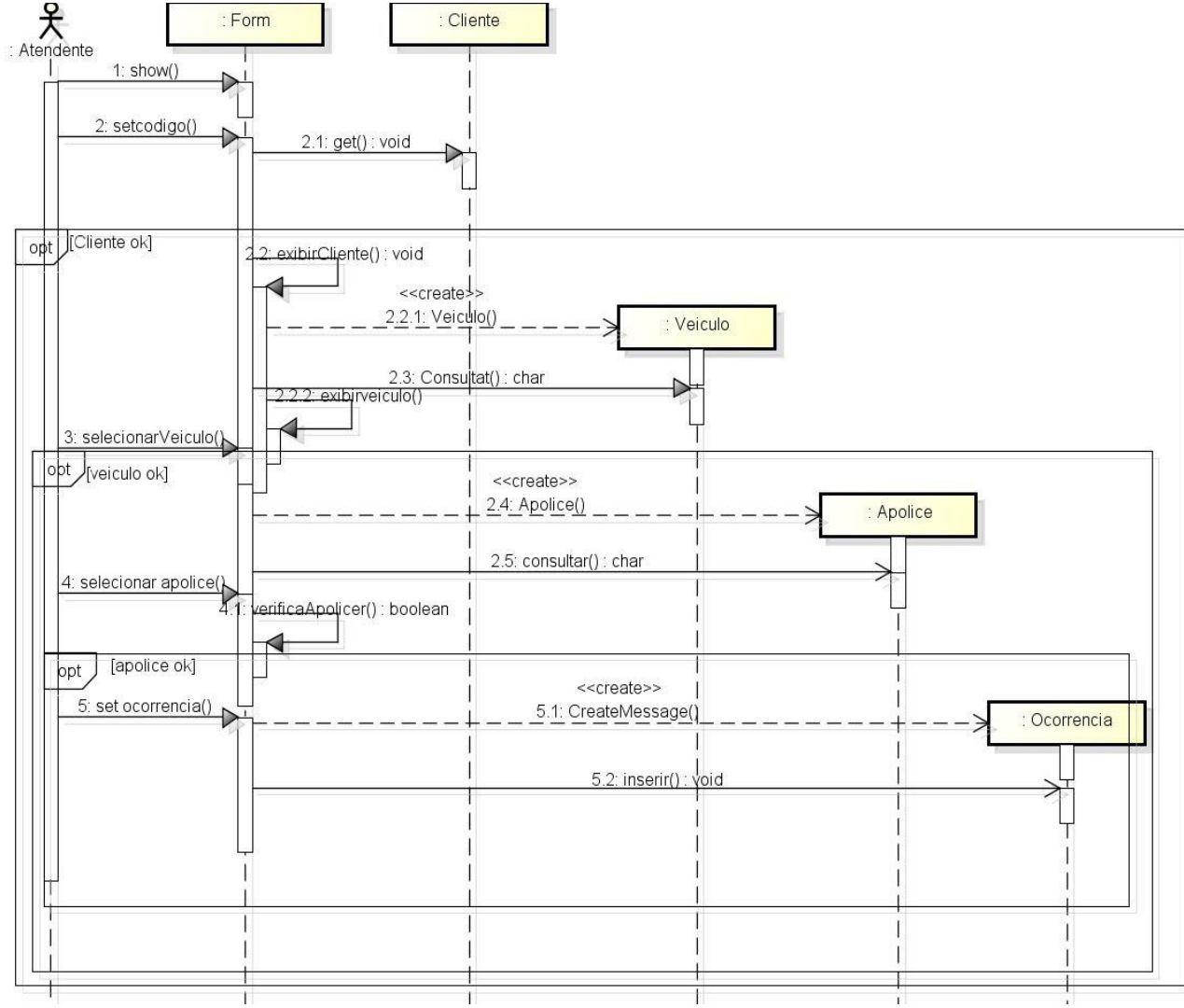


Fluxo de controle – nível de detalhamento iterações



```
public class venda {  
  
    private List<LinhaItemvenda> linhasItem =  
        new ArrayList<LinhaItemvenda>();  
  
    public Moeda obterTotal() {  
        Moeda total = new Moeda();  
        Moeda subtotal = null;  
        for (linhaItemVenda linhaItem : linhasItens){  
            subtotal = linhaItem.obterSubtotal();  
            total.add(subtotal);  
        }  
        return total;  
    }  
}
```

Diagrama de Sequência: Exemplo 3



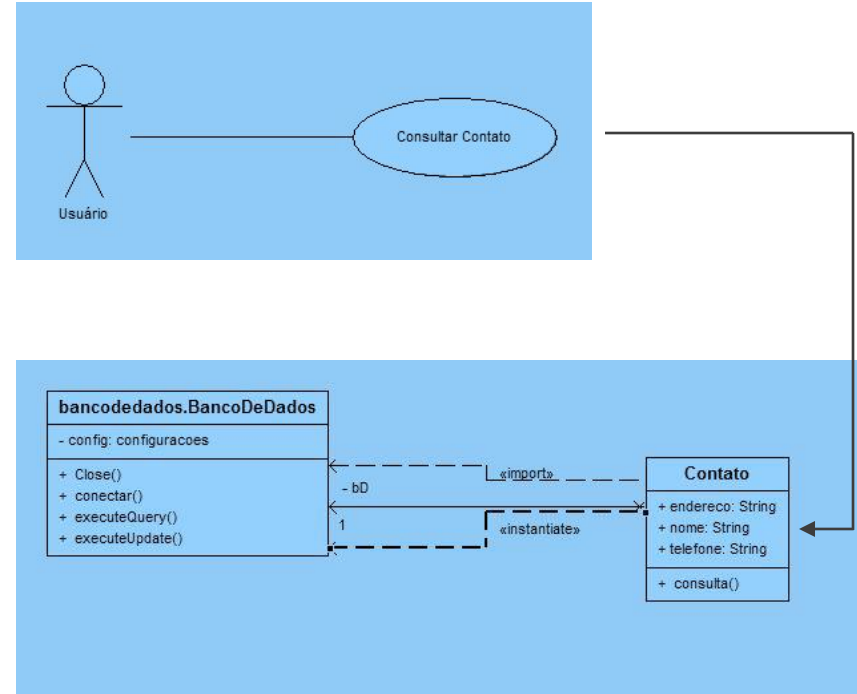
Modelagem de Interação

Diagrama de Sequência: Exemplo 3 – projeto detalhado

- Seja o caso de uso “Usuário Consulta Contato” do Sistema AGENDA

Roteiro

1. Identificamos o caso de uso no modelo da análise (Modelo Lógico – Diagrama de Classes)



Modelagem de Interação

Diagrama de Sequência: Exemplo 3 – projeto detalhado

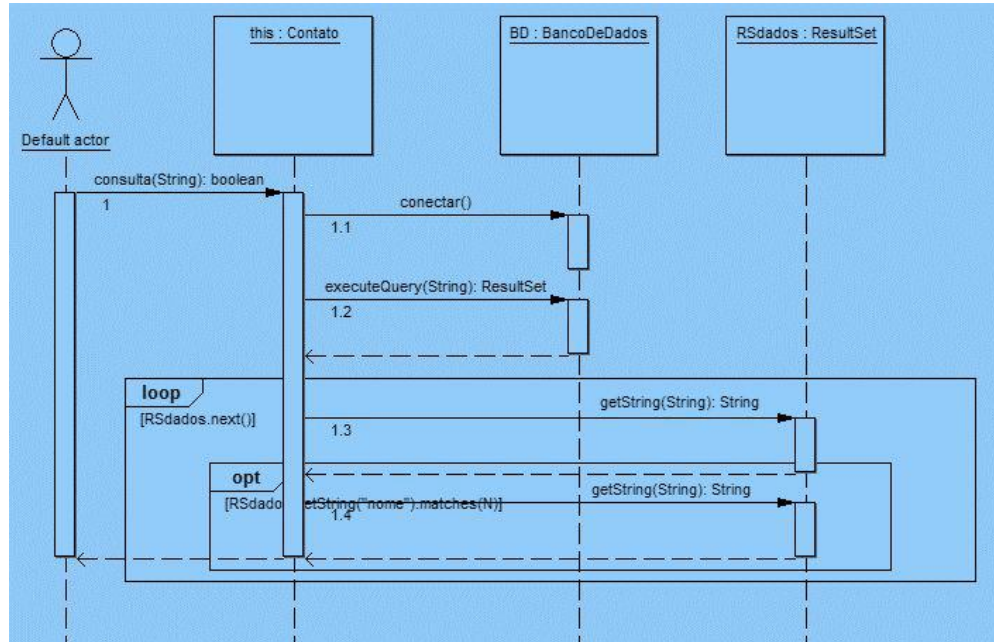
- 2. Criamos o Protótipo de Tela

The image shows a screenshot of a Java Swing application window titled "Sistema Agenda". Inside the window, there is a sub-window titled "Cadastro de Contatos". The sub-window contains three text input fields: "Nome" (with the text "Sandro"), "Telefone", and "Endereço". To the right of the "Nome" field is a button labeled "Consulta". Below the input fields are three buttons: "Cadastrar", "Excluir", and "Cancelar". A blue arrow points from the right side of the image towards the "Consulta" button.

Modelagem de Interação

Diagrama de Sequência: Exemplo 3 – projeto detalhado

- 3. Em seguida, geramos o Diagrama de sequência:



Modelagem de Interação

Diagrama de Sequência: Exemplo 3 – projeto detalhado

- 4. Finalmente, a codificação:

```
package modelo;
import bancodedados.BancoDeDados;
import java.sql.ResultSet;

public class Contato {

    public String nome="";
    public String telefone="";
    public String endereco="";
    private BancoDeDados BD = new BancoDeDados();
    private boolean retorno=false;

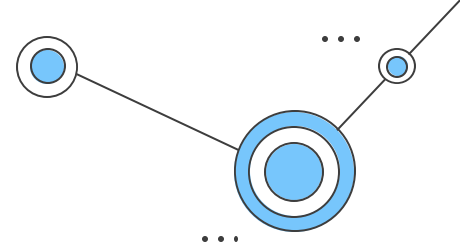
    public boolean consulta(String N) // Método para consulta ao Banco de dados
    {
        try{
            ResultSet RSdados;
            String frase= "select * from contato";
            BD.conectar();
            RSdados = BD.executeQuery(frase);

            while (RSdados.next())
            {
                if (RSdados.getString("nome").matches(N))
                {
                    this.telefone = RSdados.getString("telefone");
                    this.endereco = RSdados.getString("endereco");
                    retorno = true;
                }
            }
        }
        catch(Exception e){ System.out.println("Erro em -> " + e);}
        return retorno;
    }
}
```

Modelagem de Interação

Diagrama de Comunicação

- Um diagrama de comunicação é uma maneira alternativa de representar as mensagens trocadas por um conjunto de objetos.
- Amplamente associado ao diagrama de sequência. As informações mostradas no DC são, com frequência, as mesmas apresentadas no DS, porém o enfoque é na forma como os elementos do diagrama estão vinculados e quais mensagens trocam entre si durante o processo. Não foca na temporalidade do fluxo de mensagem.
- O diagrama de comunicação mostra as interações a partir dos objetos e dos links entre os objetos.



Modelagem de Interação

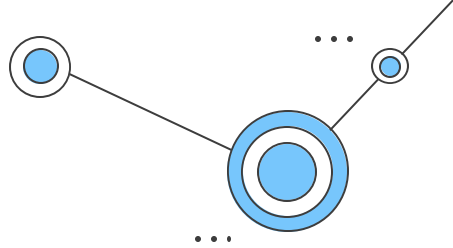


Diagrama de Comunicação

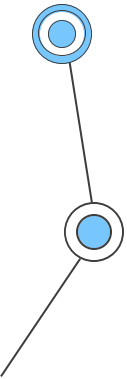
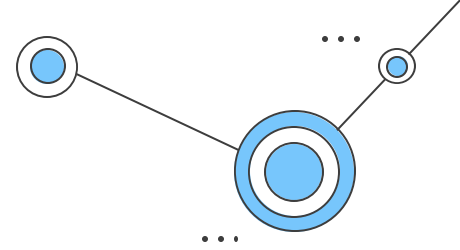
- Estruturalmente, é bastante semelhante a um diagrama de objetos.
 - A diferença é que são adicionados setas e rótulos de mensagens nas ligações entre esses objetos.
- As ligações (linhas) entre objetos correspondem a relacionamentos existentes entre os objetos.
 - Deve haver consistência com o diagrama de classes...
- Os objetos estão distribuídos em duas dimensões
 - Vantagem: normalmente permite construir desenhos mais legíveis comparativamente aos diagramas de sequência.
 - Desvantagem: não há como saber a ordem de envio das mensagens a não ser pelas expressões de sequência.



Modelagem de Interação

Diagrama de Comunicação

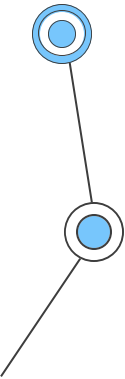
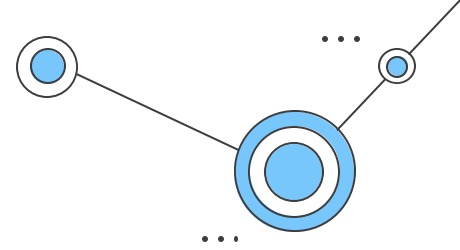
- Direção de envio de mensagem indicada por uma seta próxima ao rótulo da mensagem.
- Os objetos estão distribuídos em duas dimensões
 - normalmente permite construir desenhos mais legíveis comparativamente aos diagramas de sequência.



Modelagem de Interação

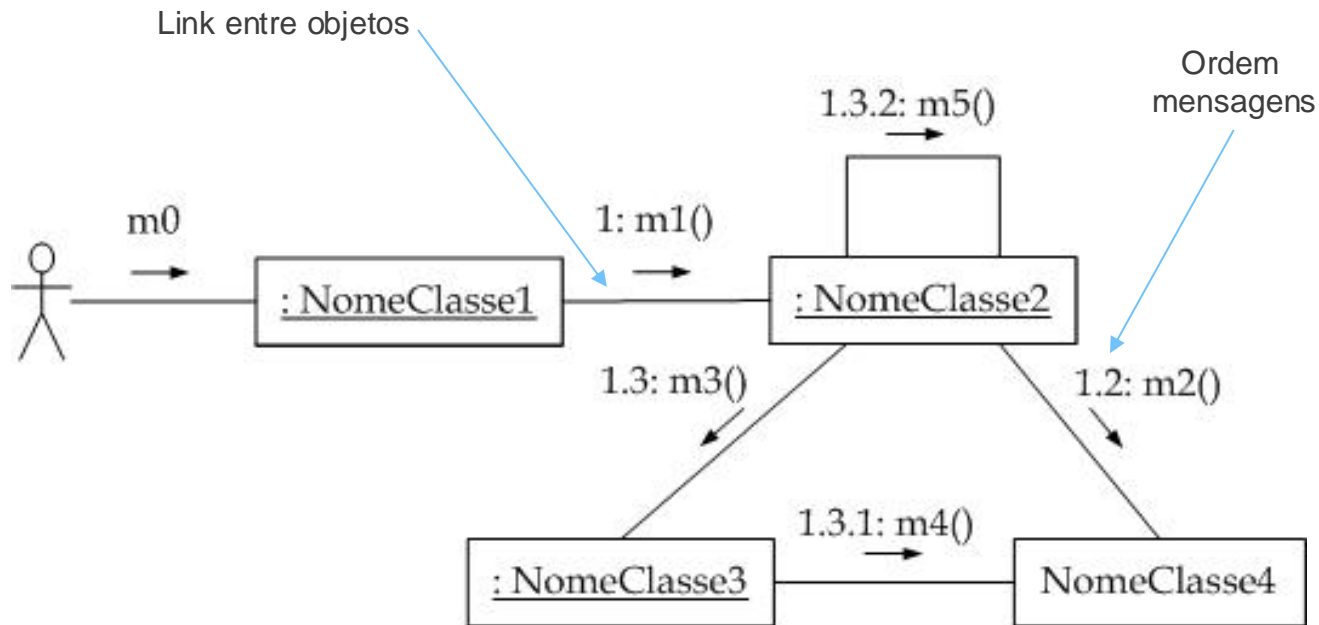
Diagrama de Comunicação: Elementos gráficos

- Elementos básicos em um diagrama de comunicação:
 - Atores
 - Objetos e classes
 - Mensagens
 - Ligações entre objetos
 - Criação e destruição de objetos
 - Iterações



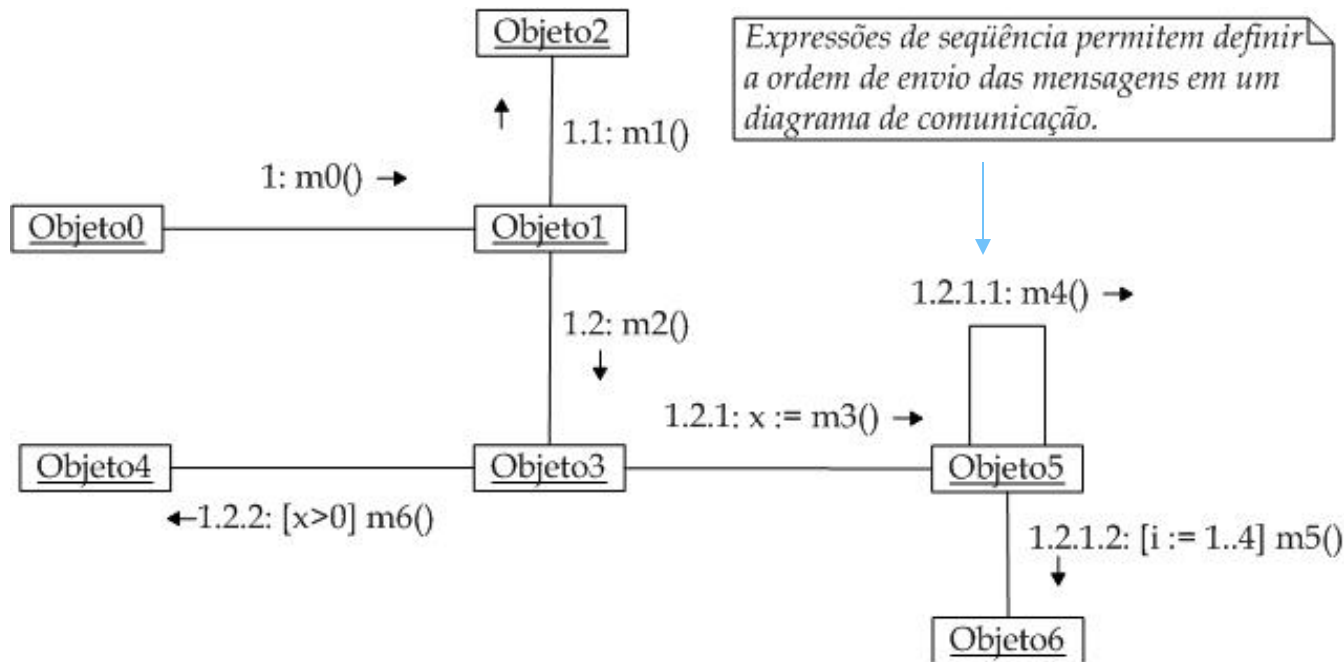
Modelagem de Interação

Diagrama de Comunicação: Elementos gráficos



Modelagem de Interação

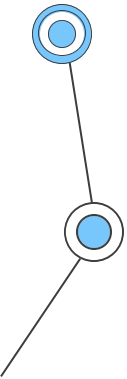
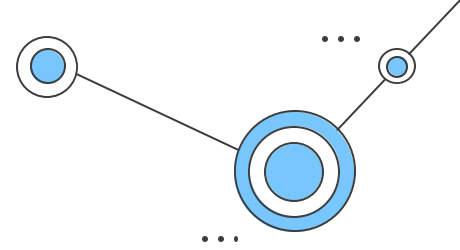
Diagrama de Comunicação: Exemplos (sintaxe UML para mensagens)



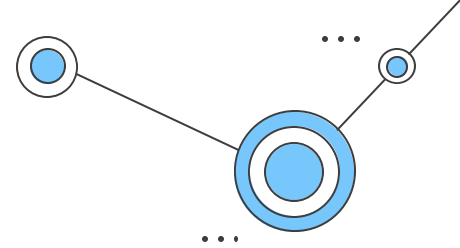
Modelagem de Interação

Escolhendo entre diagramas de sequência e de comunicação

- Diagrama de sequência:
 - Exibe as mensagens ordenadas no tempo.
 - A visualização fica dificultada conforme o número de objetos cresce (disposição em uma dimensão).
- Diagrama de comunicação:
 - Exibe mensagens enfatizando relacionamentos.
 - Melhor utilização do espaço (disposição em duas dimensões).
- Ferramentas CASE: transformação automaticamente.

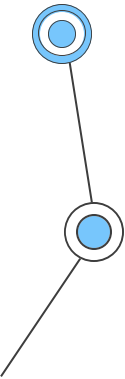


Modelagem de Interação



Diferença entre diagrama de sequência e comunicação

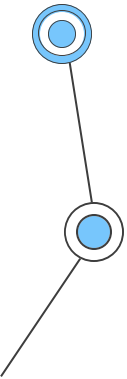
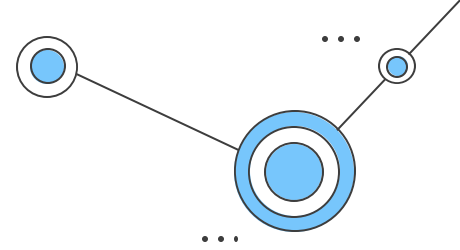
Diagrama de sequência	Diagrama de comunicação
Mostram a sequência explícita de mensagens.	Mostram as relações, além de interações.
Mostram uma ocorrência de execução.	Melhor para a visualização de padrões de comunicação.
Melhor para a visualização do fluxo global.	Melhor para a visualização de todos os efeitos em um determinado objeto.
Melhor para especificações de tempo real e para cenários complexos.	Mais fácil de usar para as sessões de brainstorming.



Modelagem de Interação

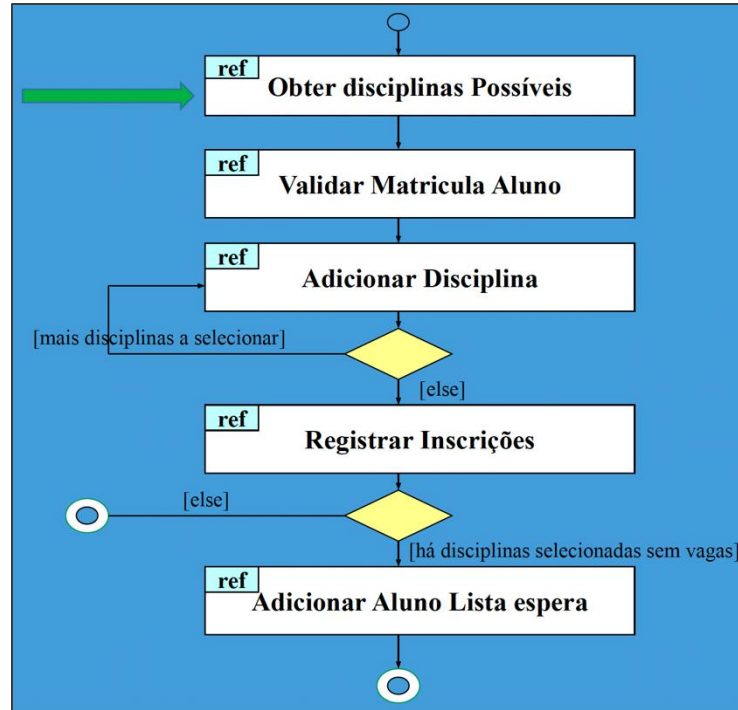
Diagrama de Visão Geral de Interação

- Seu objetivo é fornecer uma visão geral do controle de fluxo.
- É uma variação do diagrama de atividades.
- Utiliza frames no lugar dos nós de atividades.
 - Cada frame é um diagrama de sequencia.



Modelagem de Interação

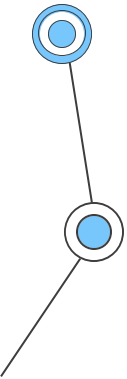
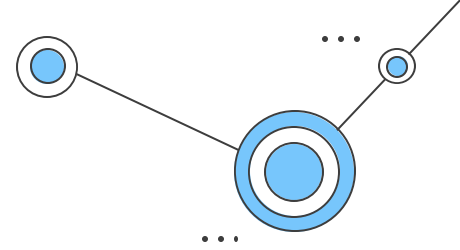
Diagrama de Visão Geral: Caso de uso registrar inscrição



Modelagem de Interação

Procedimento de construção do Modelo de Interação

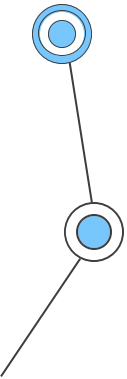
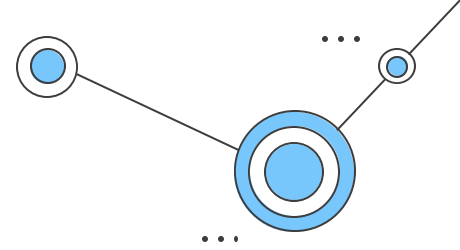
- 1 - Para cada caso de uso, defina um conjunto de cenários relevantes.
- 2 - Para cada cenário definido, faça:
 - Posicione os atores, objeto de fronteira e objeto de controle no diagrama.
 - Para cada passo do cenário selecionado, defina as mensagens a serem enviadas de um objeto a outro.
 - Defina as cláusulas de condição e de iteração, se existirem, para as mensagens.
 - Adicione multiobjetos e objetos de entidade à medida que a sua participação se faz necessário no cenário selecionado.



Modelagem de Interação

Procedimento de construção do Modelo de Interação

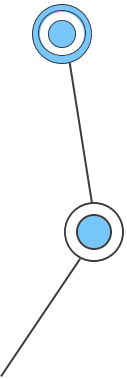
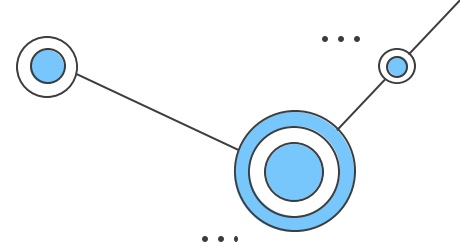
- A definição dos cenários depende da complexidade dos passos do caso de uso.
- Definição de mensagens com base nas responsabilidades de cada objeto envolvido:
 - O nome da mensagem.
 - Os argumentos de cada mensagem, se existirem.
 - O valor de retorno da operação correspondente, se existir.
 - Cláusulas de condição e de repetição, se existirem.
- A maioria dos objetos já devem ter sido identificados durante a construção do modelo de classes de domínio.



Modelagem de Interação

Procedimento de construção do Modelo de Interação

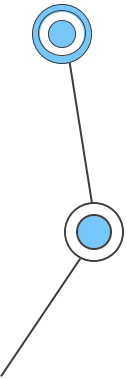
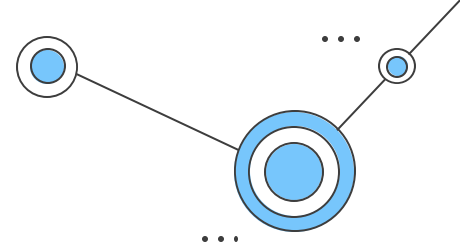
- Consistência em relação ao modelo de casos de uso e ao contexto utilizado do diagrama de classes.
 - Cada cenário relevante para cada caso de uso foi considerado?
 - As mensagens que um objeto recebe estão consistentes com suas responsabilidades?
- Mais de um controlador podem ser criados em um mesmo caso de uso.
- Mensagem de um ator a um objeto de fronteira normalmente é rotulada com a informação fornecida (por exemplo, item de pedido, id e senha, etc).



Modelagem de Interação

Modelo de interações no processo de desenvolvimento

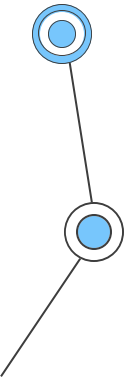
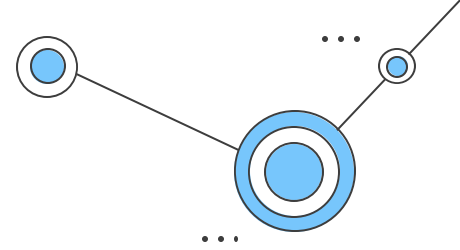
- São utilizados na fase de construção de um ciclo de vida incremental e iterativo.
 - São construídos para os casos de uso alocados para uma iteração desta fase.
- Há controvérsias sobre o momento de início da utilização desse modelo (se na análise ou se no projeto).
- Inicialmente (+análise), pode exibir apenas os objetos participantes e mensagens exibindo somente o nome da operação (ou nome da responsabilidade).
- Posteriormente (+projeto), pode ser refinado.
 - criação e destruição de objetos, tipo e assinatura completa de cada mensagem, etc.



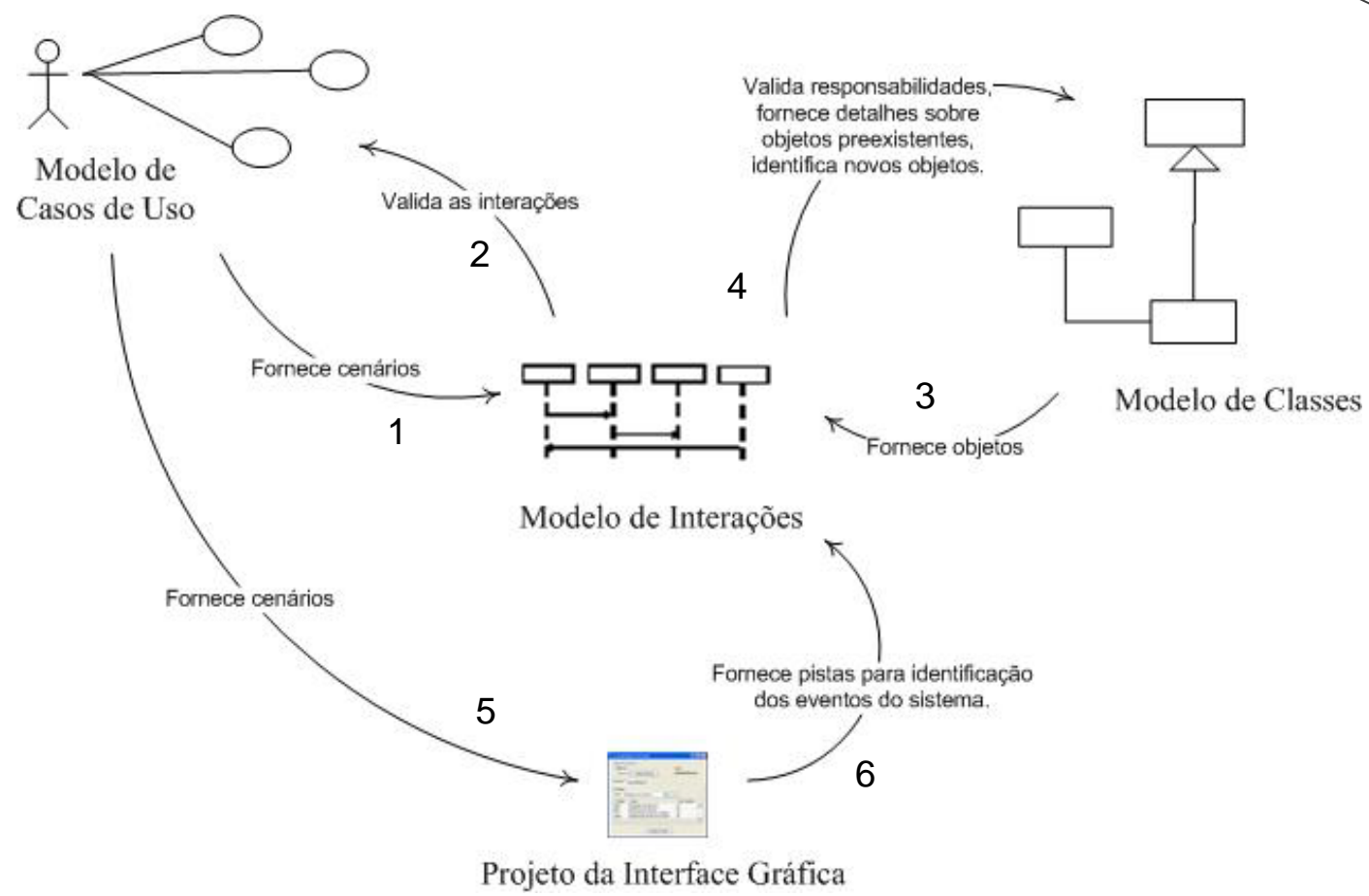
Modelagem de Interação

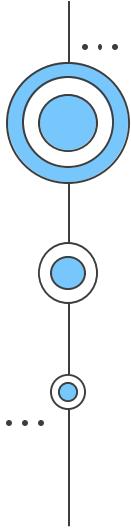
Modelo de interações no processo de desenvolvimento

- Em um processo incremental e iterativo, os modelos evoluem em conjunto.
- Embora estes modelos representem visões distintas do sistema, eles são interdependentes.
- Modelo de classes - modelo de interações.
- Modelo de interações - refinamento do modelo de casos de uso.
- Modelo de interações - operações para o modelo de classes.
- Modelo de interações - novos atributos para o modelo de classes.



Modelo de interações no processo de desenvolvimento

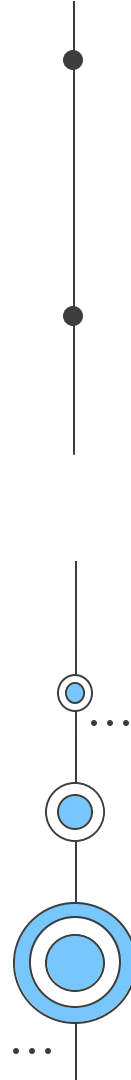


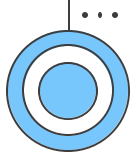


Projeto de Software

Referências básicas:

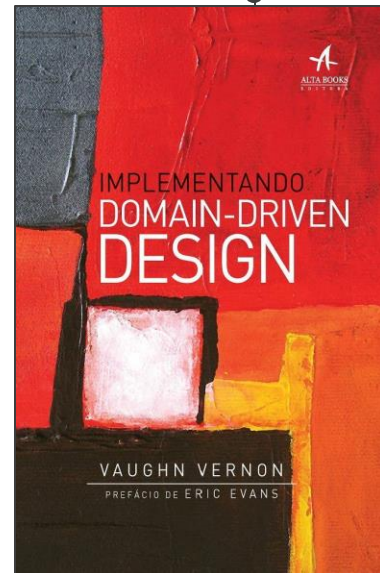
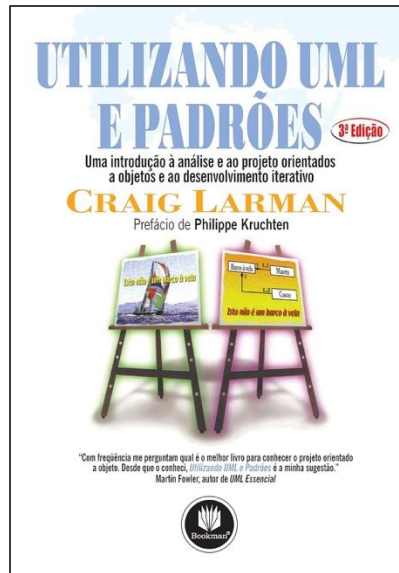
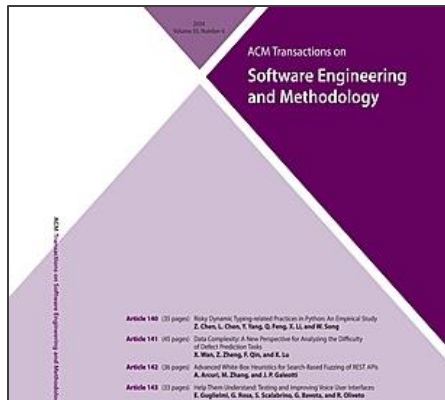
- **ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY**. New York, N.Y., USA: Association for Computing Machinery, 1992-. Trimestral. ISSN 1049-331X. Disponível em: <https://dl.acm.org/toc/tosem/1992/1/2>. Acesso em: 19 jul. 2024. (Periódico On-line).
- LARMAN, Craig. **Utilizando UML e padrões**: uma introdução á análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. E-book. ISBN 9788577800476. (Livro Eletrônico).
- SILVEIRA, Paulo et al. **Introdução à arquitetura e design de software**: uma visão sobre a plataforma Java. Rio de Janeiro, RJ: Elsevier, Campus, 2012. xvi, 257 p. ISBN 9788535250299. (Disponível no Acervo).
- VERNON, Vaughn. **Implementando o Domain-Driven Design**. Rio de Janeiro, RJ: Alta Books, 2016. 628 p. ISBN 9788576089520. (Disponível no Acervo).



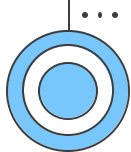


Projeto de Software

Referências básicas:



...

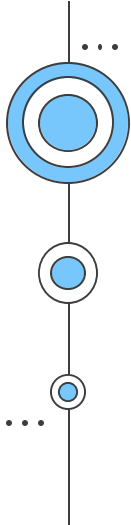


Projeto de Software

Referências complementares:

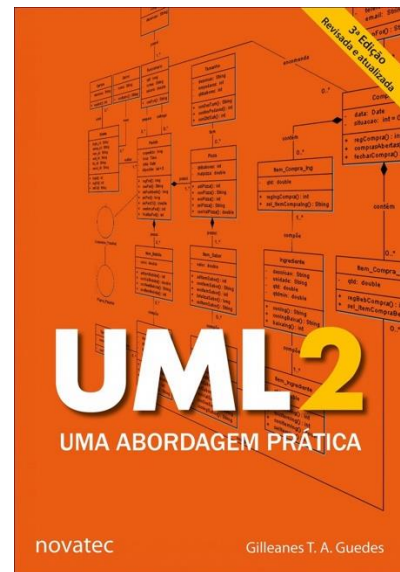
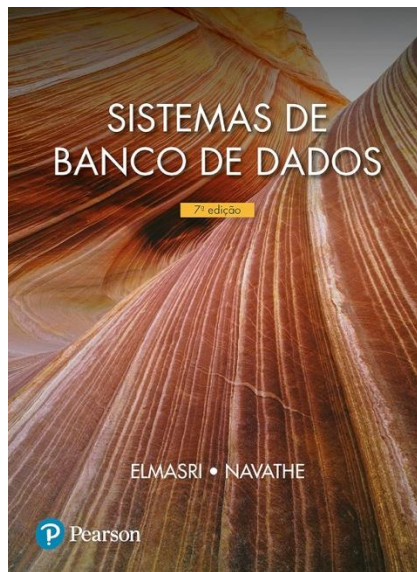
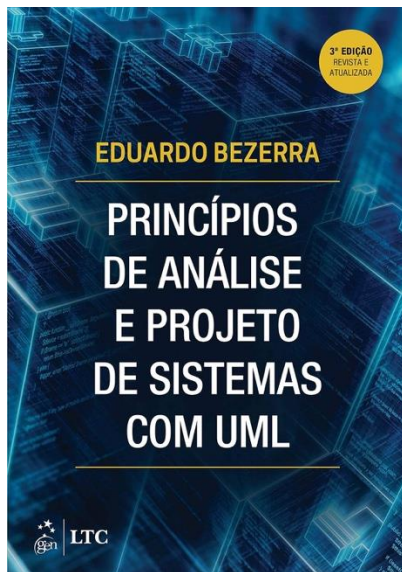
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3. ed. rev. e atual. Rio de Janeiro: Elsevier, 2015. xvii, 398 p. ISBN 9788535226263. (Disponível no Acervo).
- ELMASRI, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**, 7ª ed. Editora Pearson 1152 ISBN 9788543025001. (Livro Eletrônico).
- GUEDES, Gilleanes T. A. **UML 2**: uma abordagem prática. 2. ed. São Paulo: Novatec, c2011. 484 p. ISBN 9788575222812. (Disponível no Acervo).
- **IEEE TRANSACTIONS ON SOFTWARE ENGINEERING**. New York: IEEE Computer Society, 1975-. Mensal,. ISSN 0098-5589. Disponível em: <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32>. Acesso em: 19 jul. 2024. (Periódico On-line).
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, c2019. xii, 756 p. ISBN 9788543024974. (Disponível no Acervo).
- WAZLAWICK, Raul Sidnei. **Análise e design orientados a objetos para sistemas de informação**: modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro, RJ: Elsevier, Campus, c2015. 462 p. ISBN 9788535279849. (Disponível no Acervo).



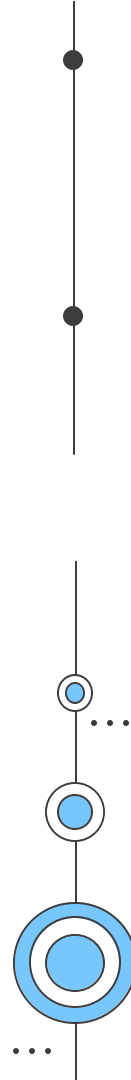


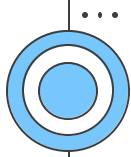
Projeto de Software

Referências complementares:



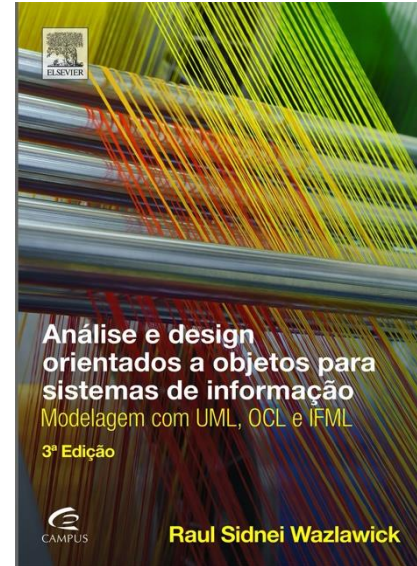
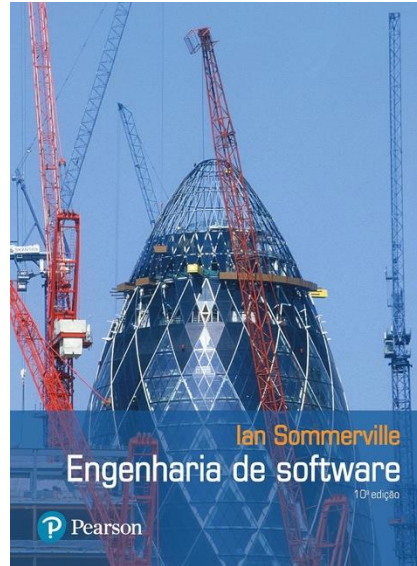
...





Projeto de Software

Referências complementares:



Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...