



Modelagem de Fluxo de Dados e Comportamental

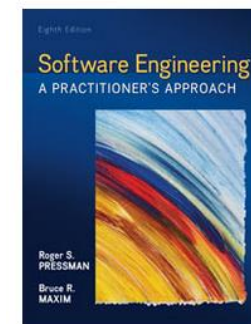
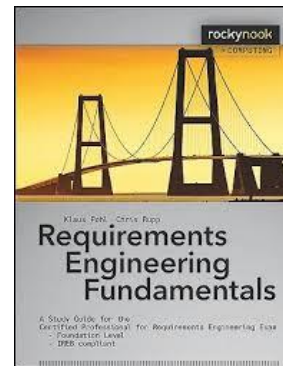
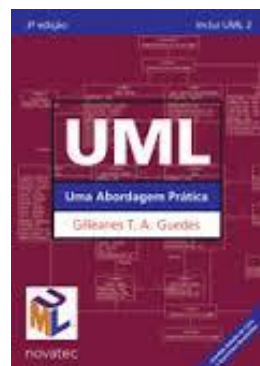
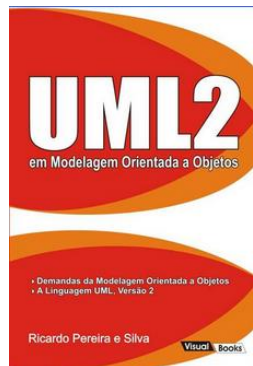


Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP

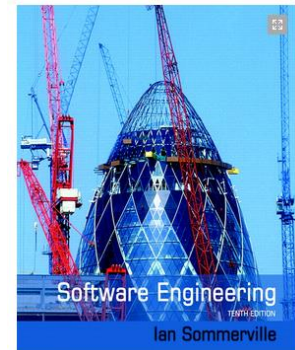


Bibliografia

- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10th edition - 2015**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007
- **UML – Uma abordagem prática – Gilleanes T. A. Guedes – 2004 - Novatec**
- **Fundamentos de Engenharia de Requisitos, Pohl K.,Rupp C. - IREB – T&M , 2012**
- **UML 2 em Modelagem Orientada a Objetos – Prof. Ricardo Pereira e Silva – UFSC, Visual Books, 2007**
- **Como modelar com UML 2 – Prof. Ricardo Pereira e Silva – UFSC, Visual Books, 2009**



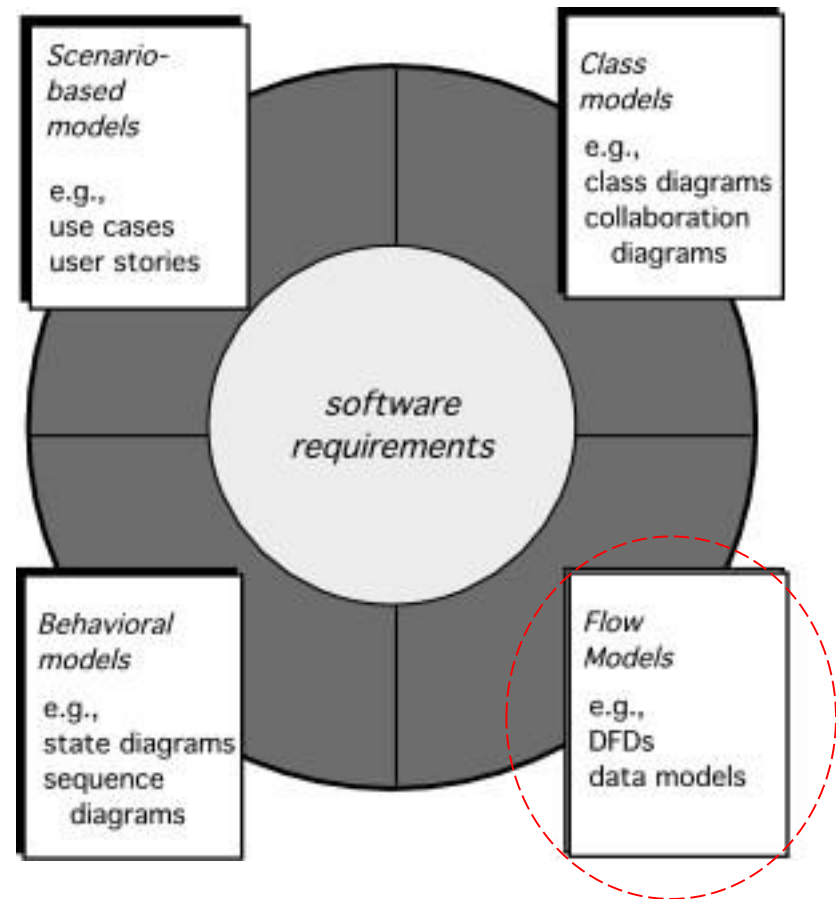
[Software Engineering: A Practitioner's Approach, 8/e](#)





Elementos da Modelagem de Requisitos

- ⊕ Modelagem baseados em Cenários.
- ⊕ Modelagem baseada em Classes.
- ⊕ **Modelagem Orientada a Fluxo.**
- ⊕ Modelagem Comportamental.





Modelagem orientada a Fluxo de Dados

- ✓ Atividade fundamental na Análise Estruturada;
- ✓ Representa como os dados são transformados no software;
- ✓ O Diagrama de Fluxo de Dados (DFD) é a forma diagramática para representar as transformações dos dados;





DFD – Diagrama de Fluxo de Dados

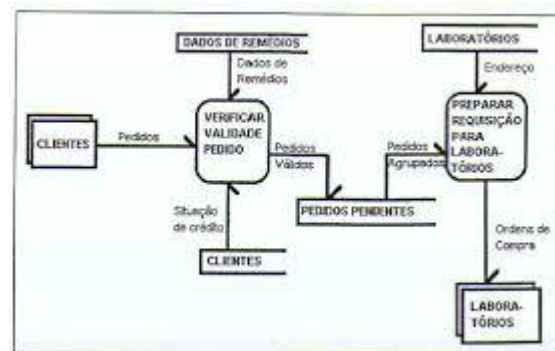
- Modela as funcionalidades do sistema por meio de processos (funções), repositórios de dados, fontes (sources) e destinos (sinks) no ambiente do software, bem como fluxos de dados.
- Concebido pela **Análise Estruturada** (DeMarco, 1978).
- Modelos de fluxos de dados permitem a modelagem do sistema em diferentes níveis de abstração.





Diagrama de Fluxo de Dados

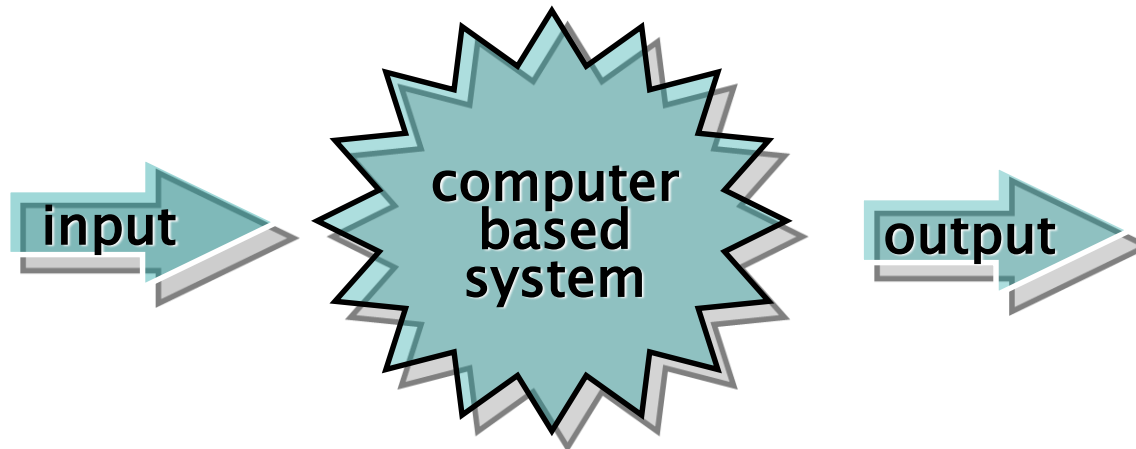
- Embora não façam parte formal da UML, são utilizados para complementar os diagramas UML e darem uma visão adicional sobre o fluxo e requisitos do software;
- O diagrama é representado de forma **hierárquica** (Estruturada);
- O primeiro modelo de fluxo de dados (denominado **DFD nível 0** ou **diagrama de contexto**) representa o sistema como um todo;
- Diagramas de Fluxo de Dados subsequentes **refinam** o diagrama de contexto, fornecendo detalhamento progressivo em cada nível subsequente.





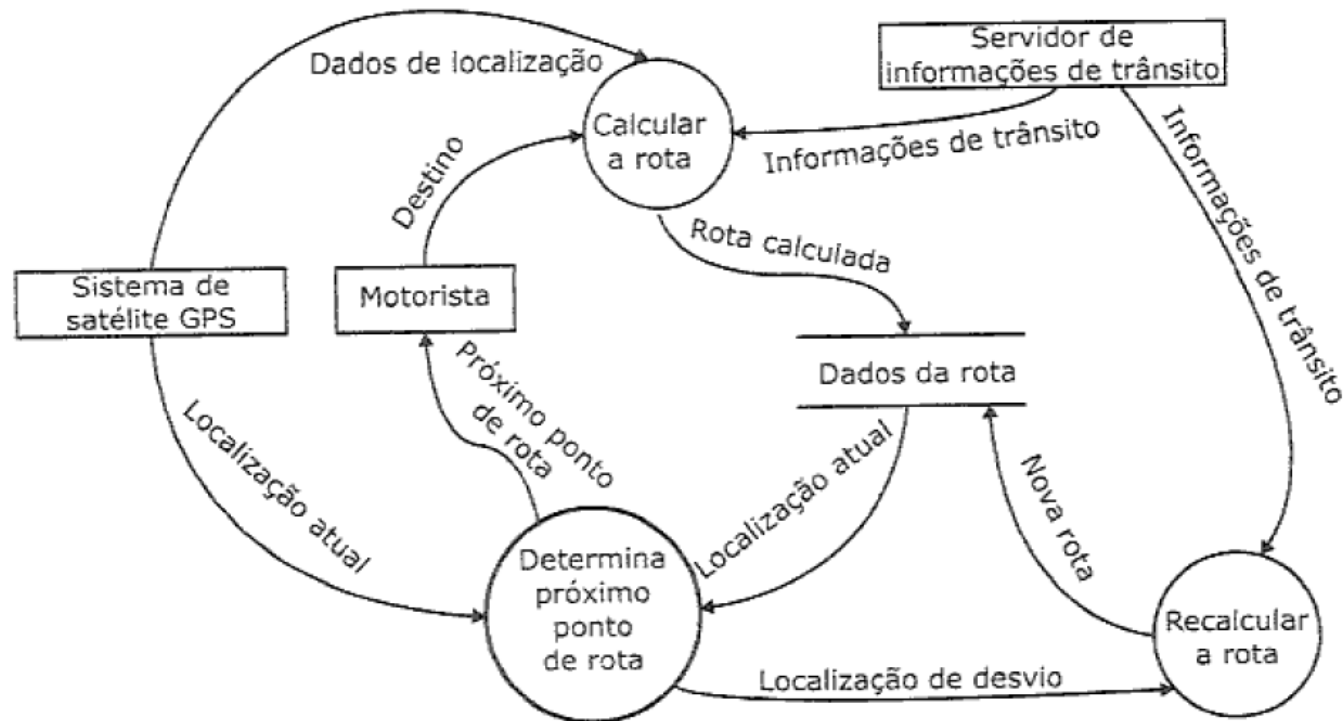
O modelo de Fluxo de Dados

- ✓ Todo sistema baseado em computador é uma transformação de informação.



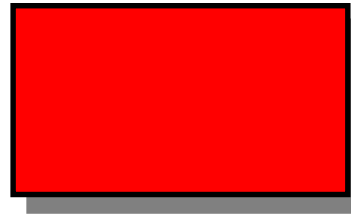


DFD – Exemplo

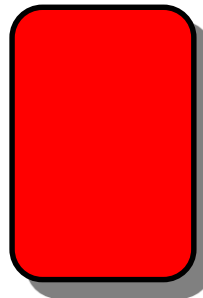




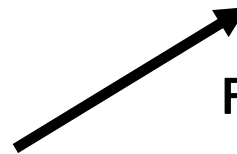
Notação – Modelagem de Fluxo de Dados



Entidade Externa



Processo



Fluxo de Dados



Armazenamento de Dados



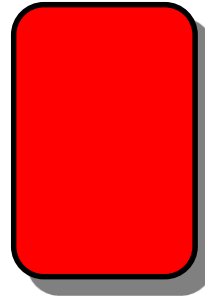
Entidade Externa



- ✓ Um produtor ou consumidor de dados;
- ✓ Exemplos: uma pessoa, um dispositivo, um sistema externo, etc;



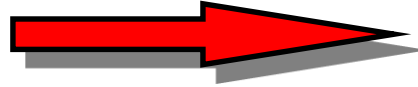
Processo



- ✓ Um transformador de dados (modifica entradas para saídas);
- ✓ Exemplos: Computar taxas, Calcular áreas, Formatar relatório, Exibir gráfico, etc.
- ✓ Dados sempre devem ser processados de alguma forma para que se cumpra funções do software.



Fluxo de Dados



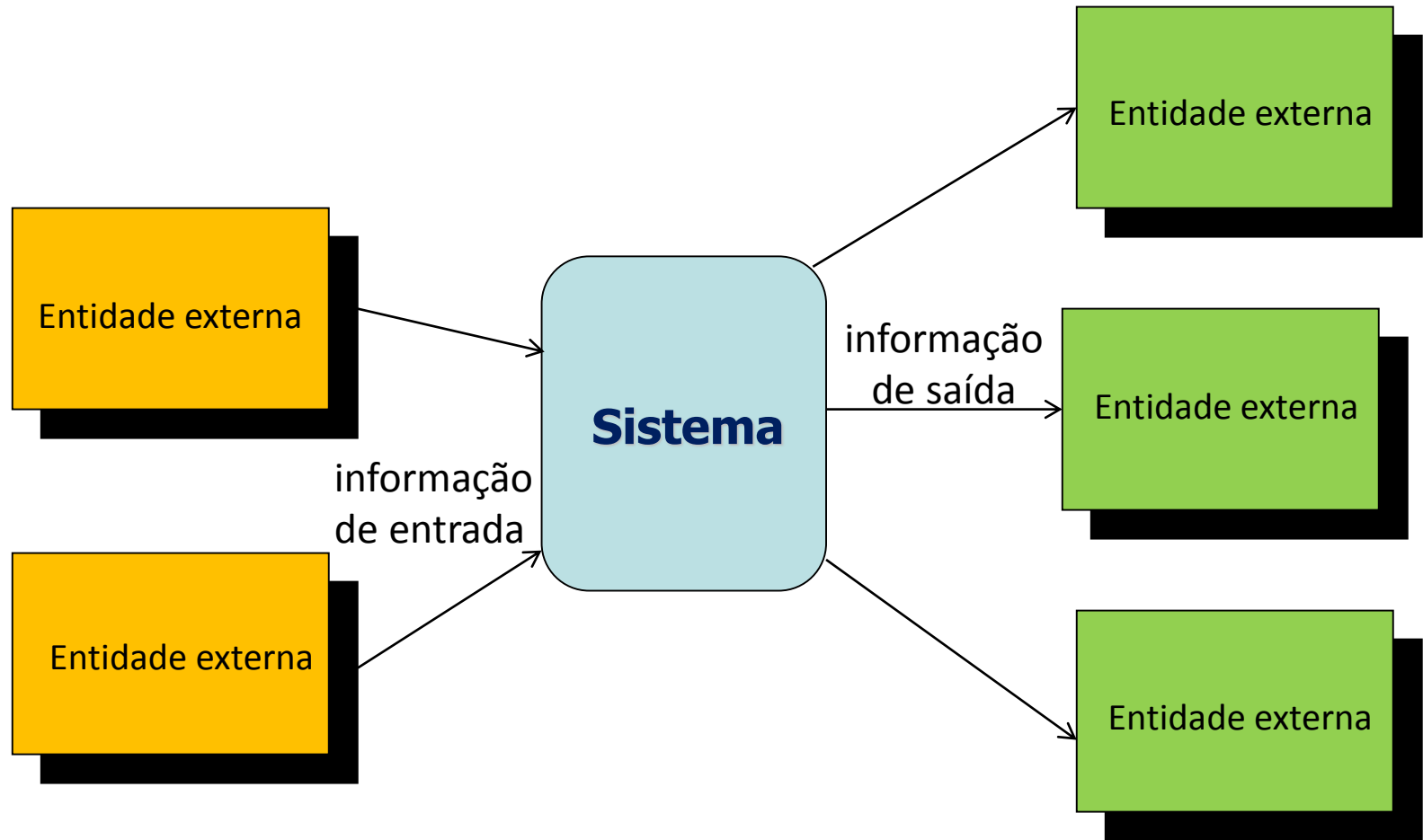
- ✓ Dados fluem através do software, iniciando como input e sendo transformado para um output.





DFD de nível 0

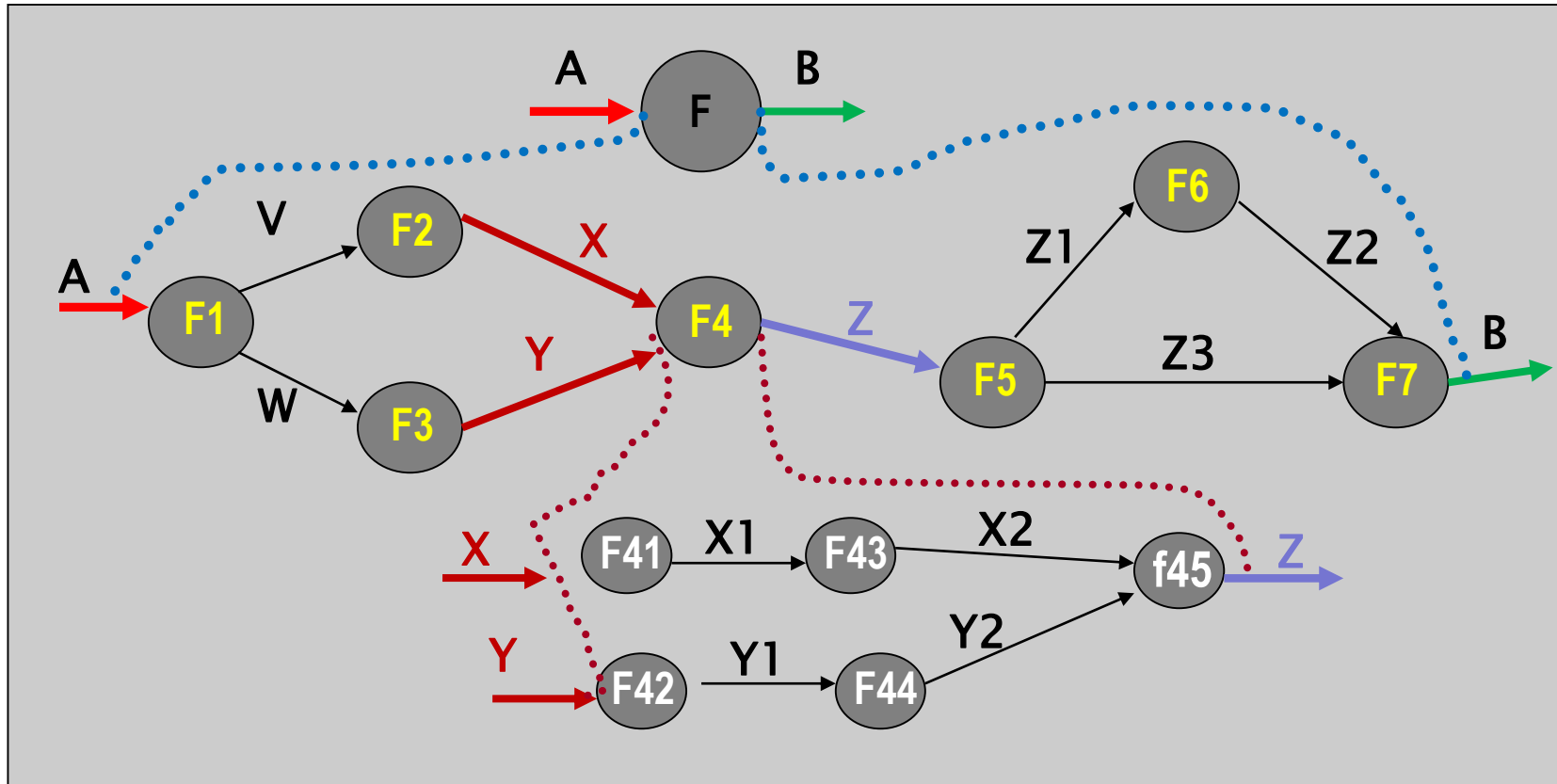
Modelo de contexto





Refinamento do Fluxo de Informação

- ✓ O DFD de nível 0 é dividido em partições para revelar mais detalhes;
- ✓ A continuidade do fluxo de informação deve ser mantida (BALANCEAMENTO).





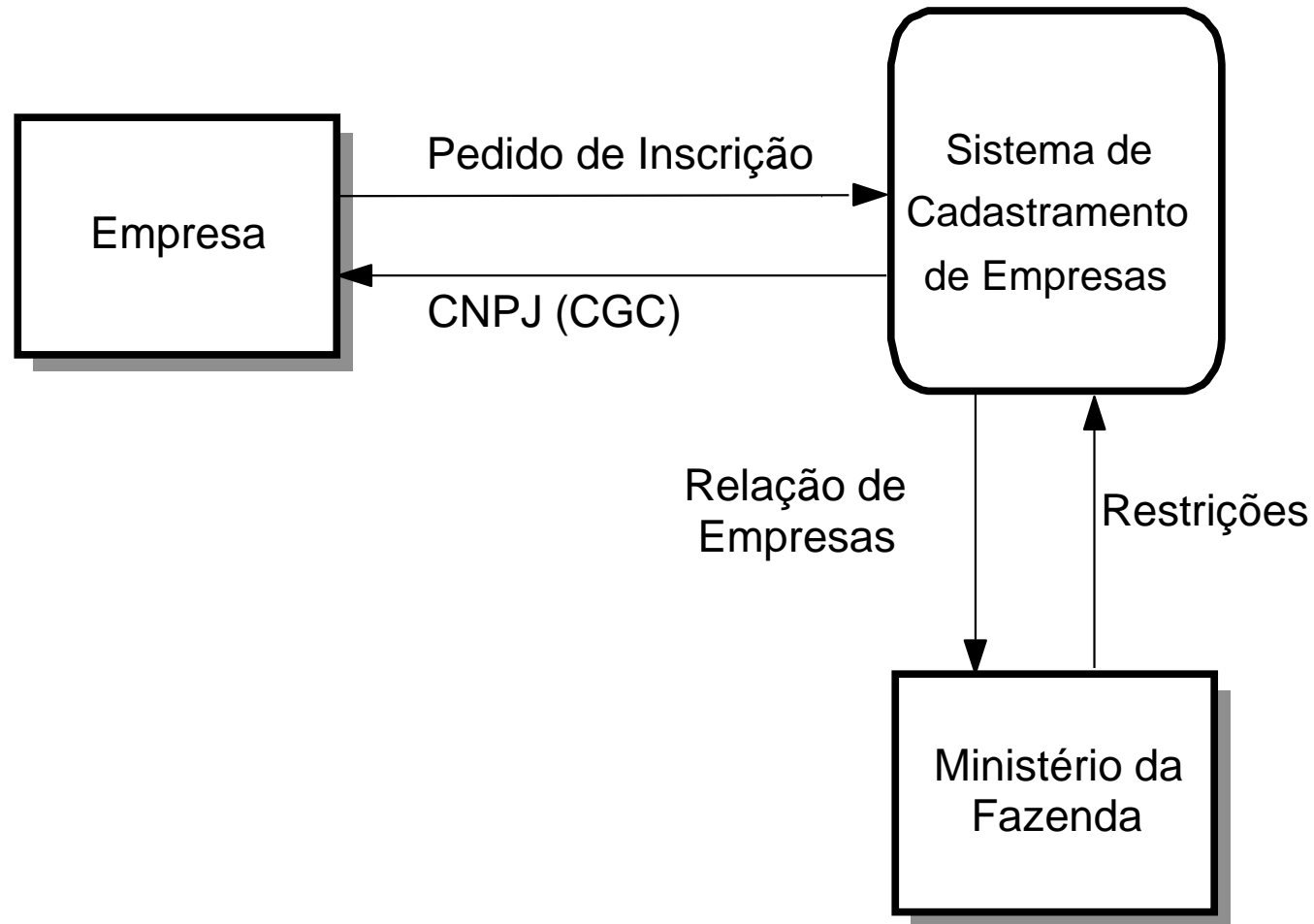
Diretrizes Básicas

- ✓ O DFD de nível 0 deve descrever o software como uma única bolha;
- ✓ O input e o output iniciais devem ser cuidadosamente anotados;
- ✓ O refinamento deve iniciar-se isolando-se possíveis processos, itens de dados e depósitos de dados a serem representados no próximo nível;
- ✓ Todas as setas e bolhas devem ser rotuladas com nomes significativos;
- ✓ A continuidade do fluxo de informação deve ser mantida de nível em nível;
- ✓ Deve ser refinada uma bolha de cada vez.

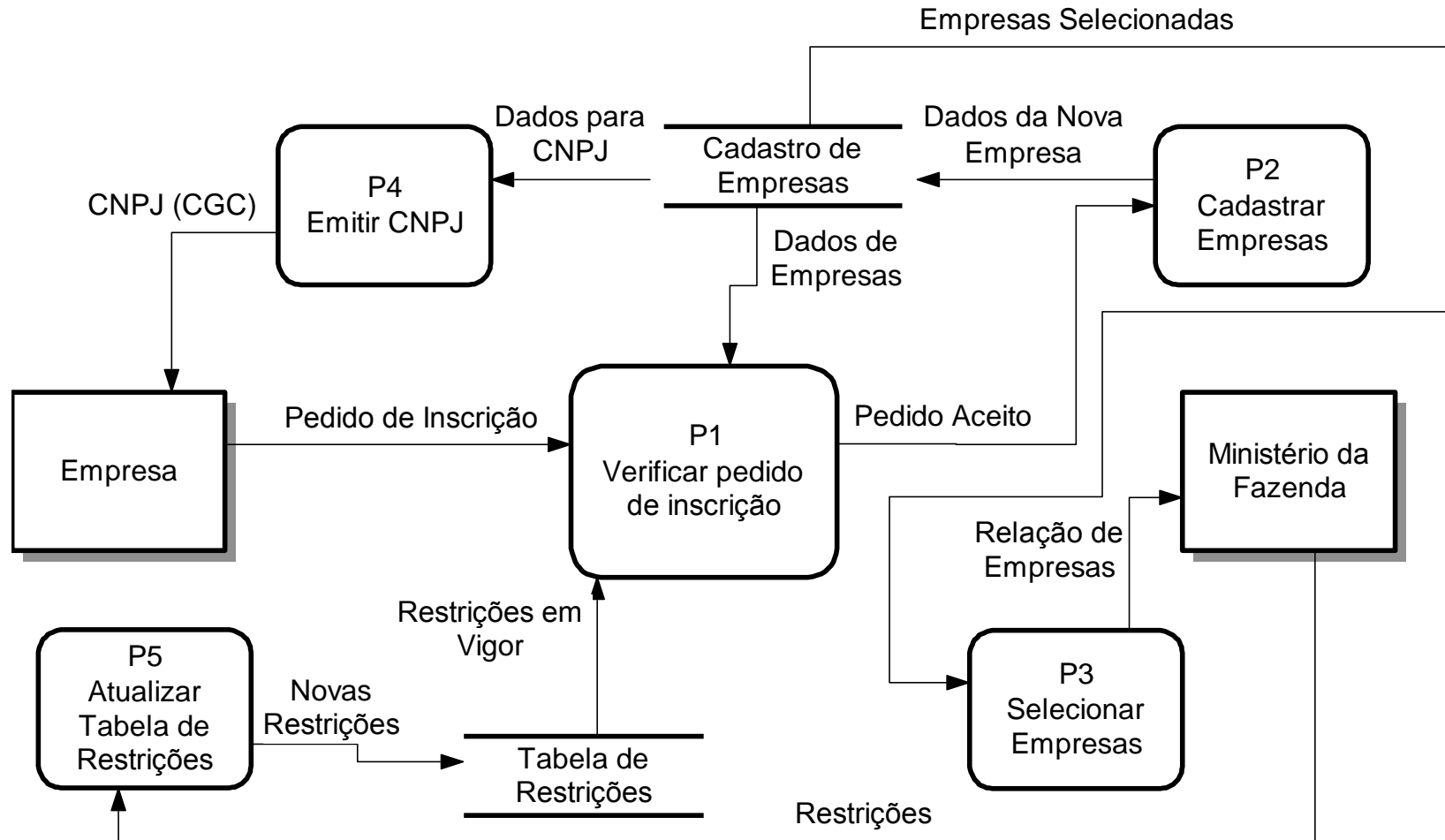




DFD nível O – Diagrama de Contexto

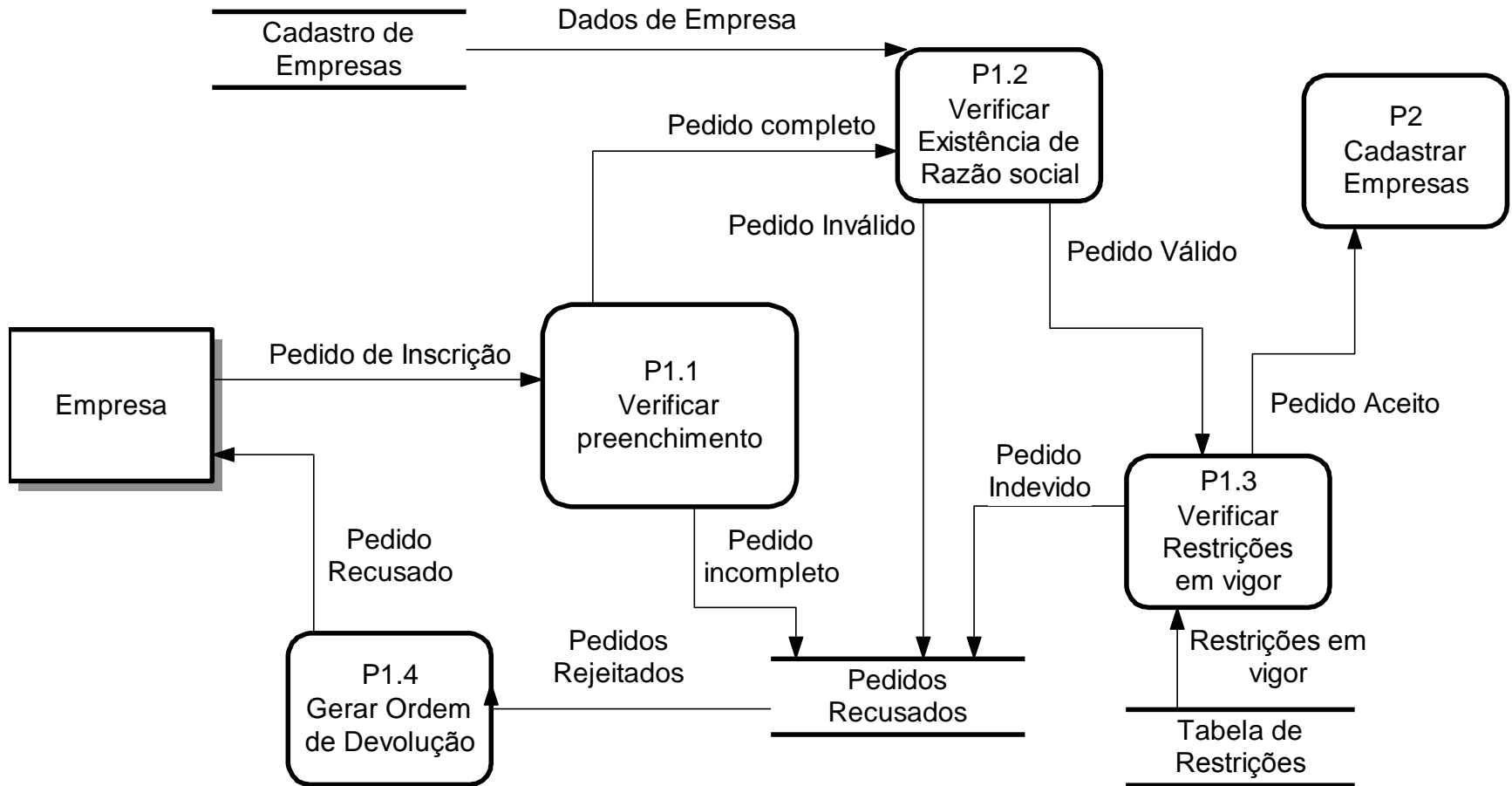


DFD Nível 1





DFD Nível 2 – Explosão do Processo P1



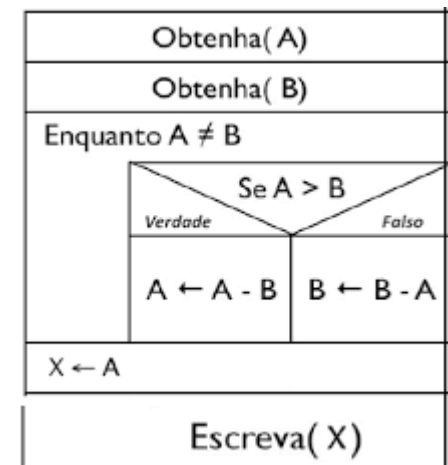


Especificação de Processo

- ✓ A especificação de processo é usada para descrever todos os processos do DFD que aparecem no nível de refinamento final.
- ✓ Pode incluir:
 - *texto narrativo;*
 - *descrição do algoritmo do processo (usando linguagem de projeto de programas - pdl);*
 - *equações matemáticas, tabelas, diagramas ou gráficos.*

```

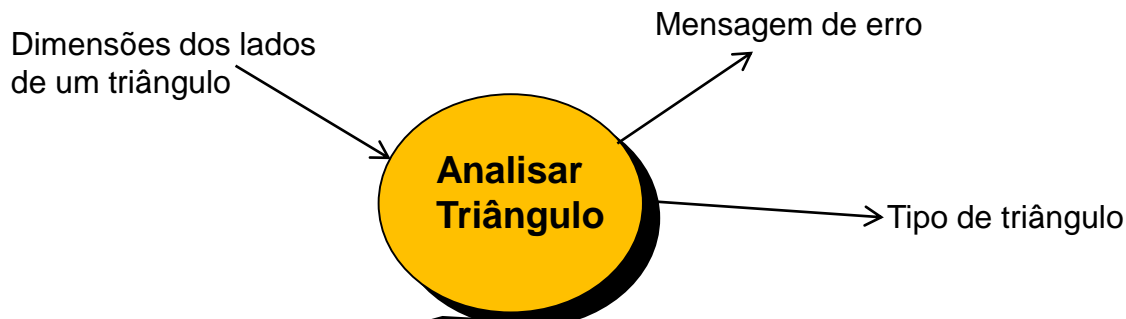
início
  <instruções>
  se <teste> então
    <instruções>
  senão
    <instruções>
  fimse
finalgoritmo
  
```





Exemplo de Especificação de Processo (Narrativa de Processamento)

Diagrama de Fluxo de Dados



Especificação do Processo

O processo analisar triângulo aceita valores A, B e C que representam as dimensões dos lados de um triângulo. O processo testa os valores dimensionais para determinar se todos os valores são positivos. Se um valor negativo for encontrado, uma mensagem de erro é produzida. O processo avalia os dados válidos para determinar se as dimensões definem um triângulo válido e, se assim for, qual tipo de triângulo – equilátero, isósceles ou escaleno. O tipo de triângulo determinado é a saída do processo.



Exemplo de Especificação de Processo (Português Estruturado)

**Analisar
Triângulo**

Especificação do Processo

Procedimento Analisar Triângulo;

Ler dimensões dos lados;

Se qualquer dimensão for negativa

Então Produza mensagem erro;

Se a maior dimensão for menor que a soma das demais

Então

Se três lados forem iguais então o tipo é equilátero;

Se dois lados forem iguais então o tipo é isósceles;

Se nenhum dos lados forem iguais então o tipo é escaleno;

Imprima tipo de triângulo;

Senão imprima tipo=0, indicação de que não existe nenhum triângulo.



O Dicionário de Dados

- ✓ Proposto como gramática quase formal para descrever o conteúdo de objetos definidos durante a análise estruturada.
- ✓ A maioria dos DD contém as seguintes informações:
 - **Nome:** o nome principal do item de dados, do depósito de dados ou de uma entidade externa;
 - **Onde é usado / Como é usado:** listagem dos processos que usam o item de dados e como ele é usado. Ex: entrada ao processo, saída do processo, como um depósito de dados, como uma entidade externa;
 - **Descrição de Conteúdo:** notação para representar o conteúdo;
 - **Informação Complementar:** outras informações sobre tipos de dados, valores previamente estabelecidos (se conhecidos), restrições ou limitações.

Dicionário de Dados									
TABELA 10.1 - Dicionário de Dados									
Descrição dos dados do sistema									
Nome do dado	Origem	Destino	Formato	Unidade	Descrição	Restrições	Observações	Atualizado	Responsável
Nome do cliente	Sistema de vendas	Sistema de vendas	Alfabetico	10	Nome do cliente	Obrigatorio			
Endereço	Sistema de vendas	Sistema de vendas	Alfabetico	30	Endereço do cliente	Obrigatorio			
Cidade	Sistema de vendas	Sistema de vendas	Alfabetico	10	Cidade do cliente	Obrigatorio			
Estado	Sistema de vendas	Sistema de vendas	Alfabetico	2	Estado do cliente	Obrigatorio			
CNPJ	Sistema de vendas	Sistema de vendas	Numerico	14	CNPJ do cliente	Obrigatorio			
CPF	Sistema de vendas	Sistema de vendas	Numerico	11	CPF do cliente	Obrigatorio			
Nome do vendedor	Sistema de vendas	Sistema de vendas	Alfabetico	10	Nome do vendedor	Obrigatorio			
Endereço do vendedor	Sistema de vendas	Sistema de vendas	Alfabetico	30	Endereço do vendedor	Obrigatorio			
Cidade do vendedor	Sistema de vendas	Sistema de vendas	Alfabetico	10	Cidade do vendedor	Obrigatorio			
Estado do vendedor	Sistema de vendas	Sistema de vendas	Alfabetico	2	Estado do vendedor	Obrigatorio			
CNPJ do vendedor	Sistema de vendas	Sistema de vendas	Numerico	14	CNPJ do vendedor	Obrigatorio			
CPF do vendedor	Sistema de vendas	Sistema de vendas	Numerico	11	CPF do vendedor	Obrigatorio			
Nome do produto	Sistema de vendas	Sistema de vendas	Alfabetico	20	Nome do produto	Obrigatorio			
Descrição do produto	Sistema de vendas	Sistema de vendas	Alfabetico	50	Descrição do produto	Obrigatorio			
Valor do produto	Sistema de vendas	Sistema de vendas	Numerico	10	Valor do produto	Obrigatorio			
Quantidade do produto	Sistema de vendas	Sistema de vendas	Numerico	10	Quantidade do produto	Obrigatorio			
Valor total do produto	Sistema de vendas	Sistema de vendas	Numerico	10	Valor total do produto	Obrigatorio			



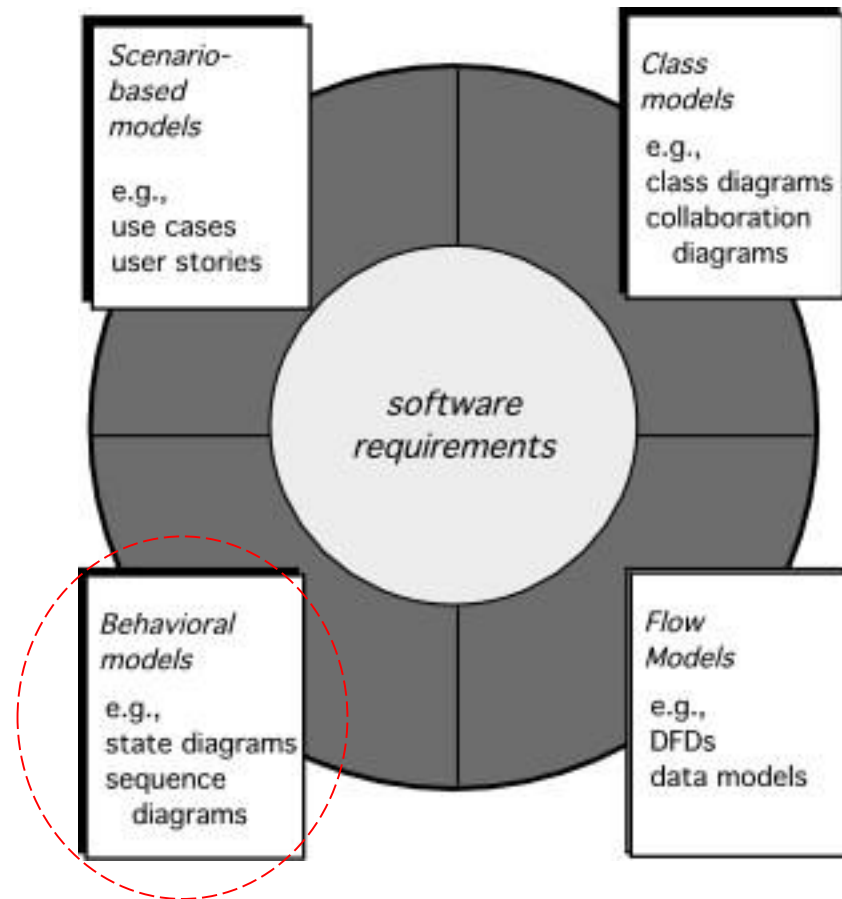
Análise Estruturada – Observações

- ✓ DFD é uma boa ferramenta de comunicação entre usuários e desenvolvedores;
- ✓ Embora seja vista como uma técnica antiga, ainda é muito utilizada em projetos de Engenharia de Software;
- ✓ Tem perdido espaço para as metodologias orientadas a objeto. No entanto, a questão não é qual a melhor, mas sim, qual combinação de representações irá fornecer aos interessados o melhor modelo de requisitos de software e a ligação mais efetiva para o projeto de software [Pressman,2011].



Elementos da Modelagem de Requisitos

- ⊕ Modelagem baseados em Cenários.
- ⊕ Modelagem baseada em Classes.
- ⊕ Modelagem Orientada a Fluxo.
- ⊕ **Modelagem Comportamental.**





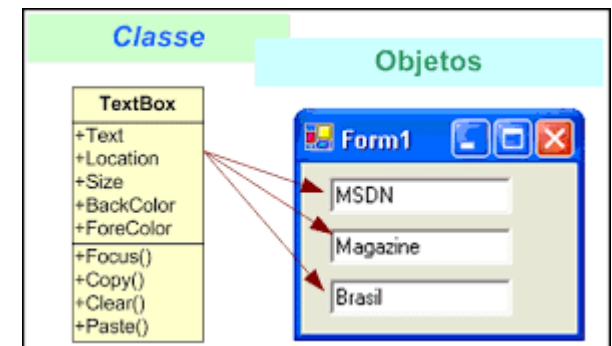
Modelagem Comportamental





Interação de Objetos em UML

- Para se descrever a execução de um programa orientado a objetos, necessita-se modelar o seu comportamento (tempo de execução).
- Objetos podem enviar invocações de métodos a outros objetos .
- Assim, a interação de objetos corresponde a um dos requisitos da modelagem dinâmica de sistemas.





Modelagem de Interação de Objetos



- ⊕ Modelagem de sequências de processamento.
- ⊕ Modelagem de não-determinismos (caminhos alternativos que podem ser seguidos durante o processamento.
- ⊕ Não descreve completamente o processamento que ocorre no programa.
- ⊕ Foca-se apenas nas **comunicações** entre as **instâncias (objetos)** e avaliação de condições entre essas comunicações.
- ⊕ Diagramas de modelagem de interação modelam apenas invocações de métodos, sem menção a outras ações, por exemplo: atribuições. (Não se descreve, portanto, todo o algoritmo de um método...)



Como modelar a interação de objetos num programa orientado a objetos ?





Diagramas de Interação



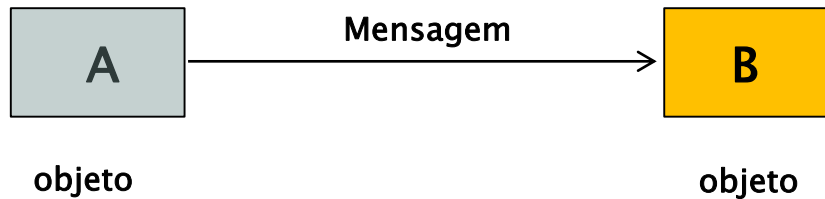
- ⊕ Termo genérico que se aplica a dois tipos de diagramas que enfatizam interações entre os objetos.
- ⊕ Correspondem aos diagramas de **Sequência** e de **Comunicação** (era conhecido por Colaboração até a versão 1.5 da UML)





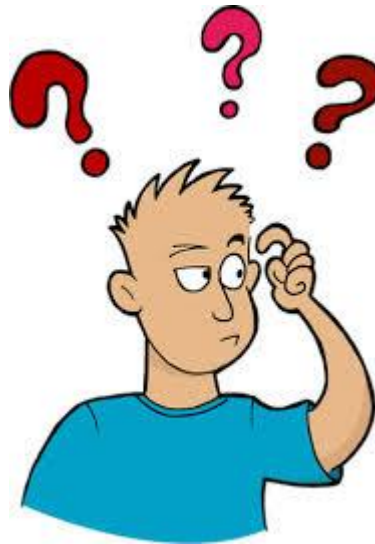
Principais elementos dos Modelagem de Interação

- Objeto
- Mensagem





Qual o uso do Diagrama de Sequência ?





-





Finalidade do Diagramas de Sequência

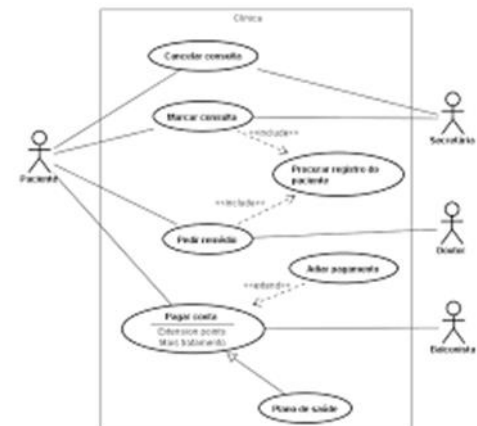
- Modelar objetos interagindo por meio de trocas de mensagens.
- Baseia-se no Diagrama de Casos de Uso.
- Útil para o detalhamento dos Casos de Uso.





Lembrando...

- Casos de Uso descrevem o que o software faz e não como! Portanto, em alto nível de abstração.





- Diagrama de Sequência descreve como o Caso de Uso é executado.
- Descreve o passo-a-passo de como o Caso de Uso é executado.
- Descreve quais objetos interagem para a execução do Caso de Uso.

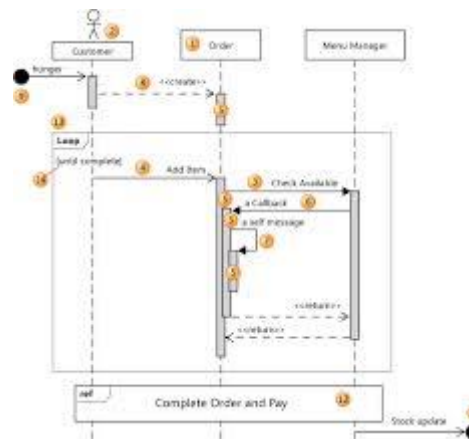




Diagrama de Sequência



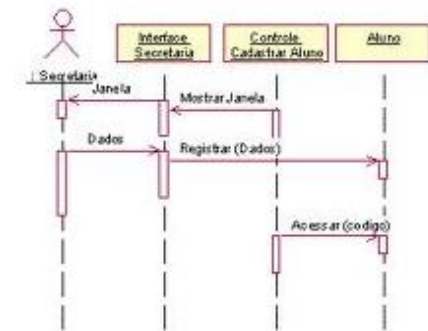
- ⊕ Define a sequência de eventos que ocorrem em um determinado processo.
- ⊕ Ou seja, quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processamento específico.
- ⊕ Assim, os objetivos do Diagrama de Sequência são: determinar a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem entre si dentro de um processo.



Diagrama de Sequência



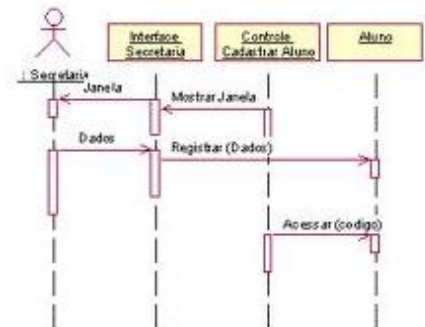
- ⊕ Interação enfatizando o Tempo de Sequência.
- ⊕ Mostra objetos participando em interações de acordo com suas linhas de vida e as mensagens que trocam.
- ⊕ Baseia-se no Diagrama de Caso de Uso. Em geral, cada diagrama de sequência se refere a um determinado Caso de Uso.
- ⊕ Num projeto, pode haver diversos Diagramas de Sequência (em geral, um para cada processo disparado pelo usuário).





Elementos do Diagrama de Sequência

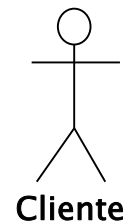
- ✓ Atores
- ✓ Objetos
- ✓ Linhas de Vida
- ✓ Focos de Controle ou Ativação
- ✓ Mensagens ou Estímulos
- ✓ Mensagens de Retorno
- ✓ Condições ou Condições de Guarda.





Atores

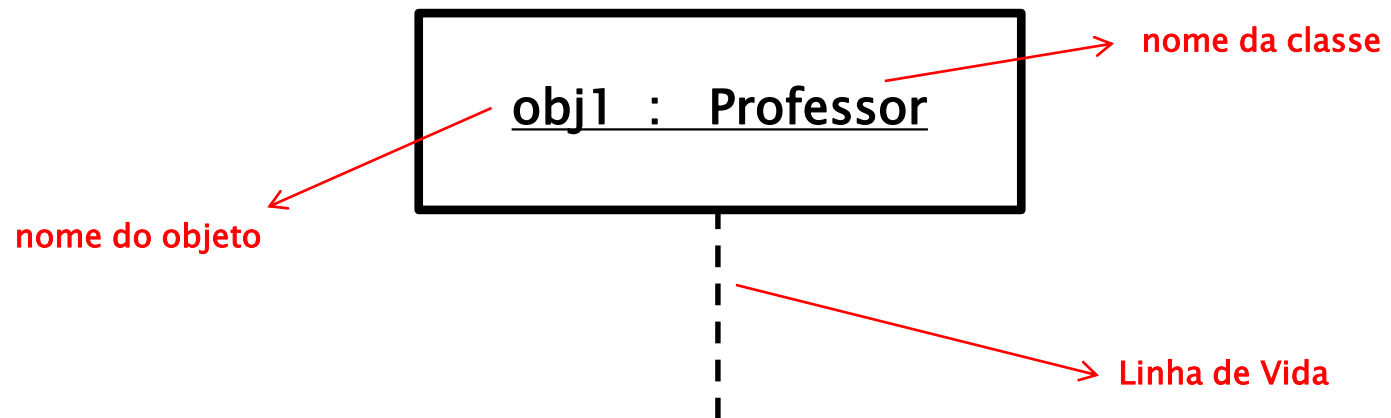
- ⊕ São exatamente os mesmos descritos no Diagrama de Casos de Uso
- ⊕ Representam as entidades externas que interagem com o sistema e que solicitam serviços, gerando eventos que iniciam os processos.
- ⊕ Em geral, os atores são os usuários que utilizam o sistema, mas podem ser representados por outro software ou hardware.
- ⊕ No diagrama de sequência, são representados por um boneco, porém contendo uma linha de vida.
- ⊕ Refletem um aspecto dinâmico do Caso de Uso e facilitam a compreensão de um processo.





Modelagem do Objeto

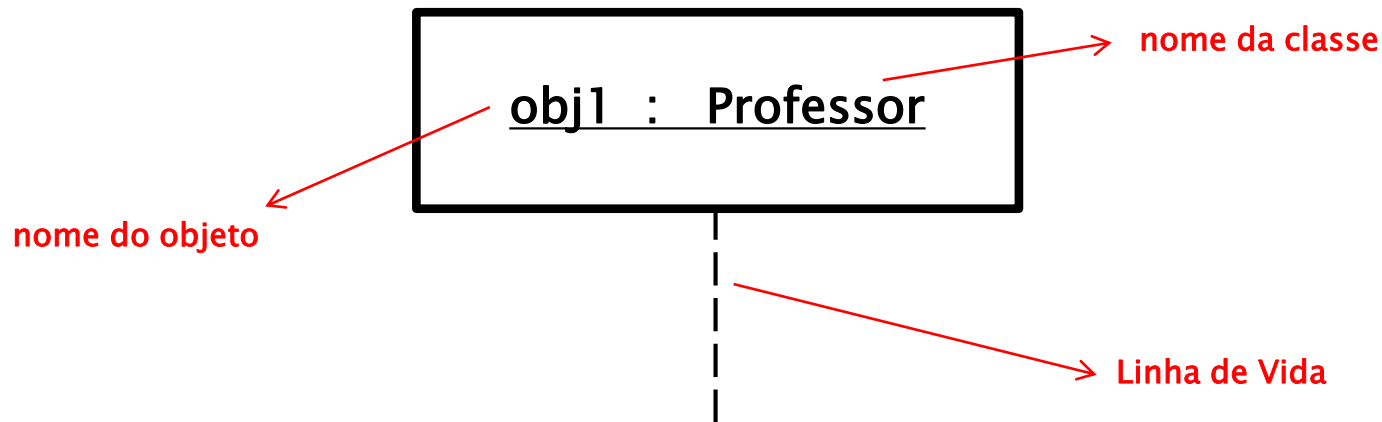
- ✦ Um objeto (instância de uma classe) é representado por um retângulo
- ✦ Internamente ao retângulo, define-se um identificador sublinhado (para denotar uma instância) com uma referência à classe geradora do objeto.
- ✦ Essas duas informações são separadas por : (dois pontos)
- ✦ No diagrama de sequência, os objetos são apresentados na dimensão horizontal do diagrama.
- ✦ No diagrama de sequência, os atributos de um objeto não são representados, mas possuem uma linha vertical tracejada representando a Linha de Vida





Modelagem do Objeto

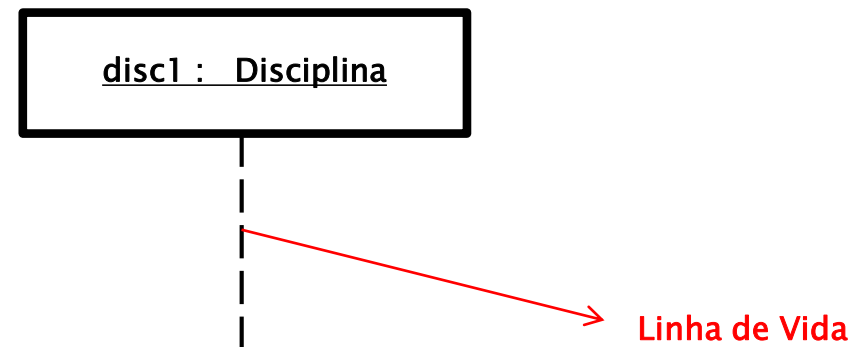
- ✓ Um objeto pode existir desde o início do processo ou ser criado durante o decorrer da execução do mesmo.
- ✓ No primeiro caso, o retângulo que representa o objeto aparecerá na parte superior do diagrama.
- ✓ No segundo caso, o retângulo surgirá na mesma altura em que a mensagem que o criar for chamada.





Linha de Vida

- ✓ Representa o tempo em que um objeto existiu durante um processo.
- ✓ São representadas por linhas finas verticais tracejadas, partindo do retângulo que representa o objeto.
- ✓ A linha de Vida é interrompida com um 'X' quando o objeto for destruído.
- ✓ Um objeto não precisa necessariamente existir quando o processo é iniciado, podendo ser criado durante o processo.

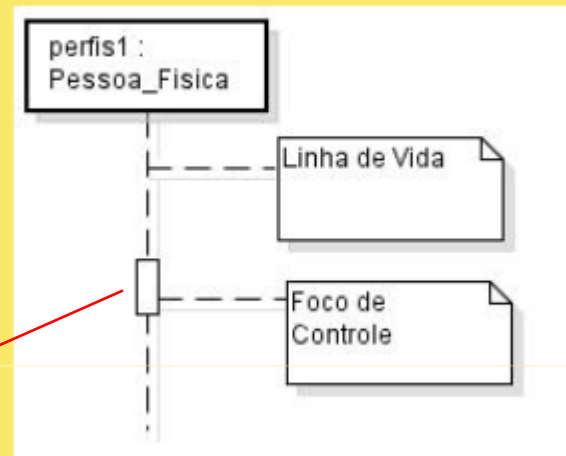




Foco de Controle ou Ativação

- ✓ Indica os períodos em que um determinado objeto está participando ativamente do processo.
- ✓ Ou seja, identifica os momentos em que um objeto está executando um ou mais métodos utilizados em um processo específico.
- ✓ Os Focos de Controle são representados dentro da Linha de Vida de um objeto, porém enquanto as Linhas de Vida são representadas por tracejados finos, o Foco de Controle é representado por extensões mais largas (grossas) da Linha de Vida.

Exemplo:



Foco de Controle

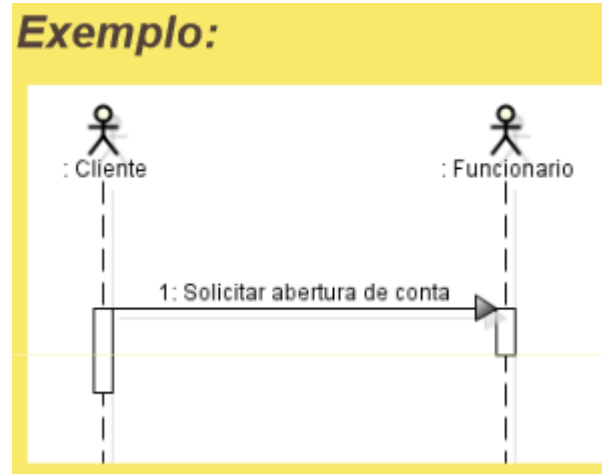


Foco de Controle



Mensagens ou Estímulos

- ✓ Demonstram a ocorrência de eventos que, em geral, forçam a chamada de um método em algum dos objetos envolvidos no processo.
- ✓ No entanto, a mensagem pode ocorrer entre dois atores, o que – neste caso – não dispara nenhum método.





Mensagens ou Estímulos

- ✓ Um diagrama de sequência, em geral, é iniciado por um evento externo, causado por algum ator, o que acarreta o disparo de um método em um dos objetos.
- ✓ A ocorrência deste evento inicia o processo.
- ✓ As mensagens podem ser disparadas entre:
 - ✓ Um ator e outro ator (o que não é muito comum, mas que pode permitir uma compreensão melhor do processo como um todo).
 - ✓ Um ator e um um objeto (um ator produz um evento que força o disparo da execução de um método por um objeto).
 - ✓ Um objeto e um objeto (ocorrência mais comum de mensagens, onde um objeto transmite uma mensagem para outro objeto solicitando a execução de um método).
 - ✓ Um objeto e um ator (normalmente isso ocorre quando um objeto envia uma mensagem de retorno à chamada de um método solicitado, contendo seus resultados).

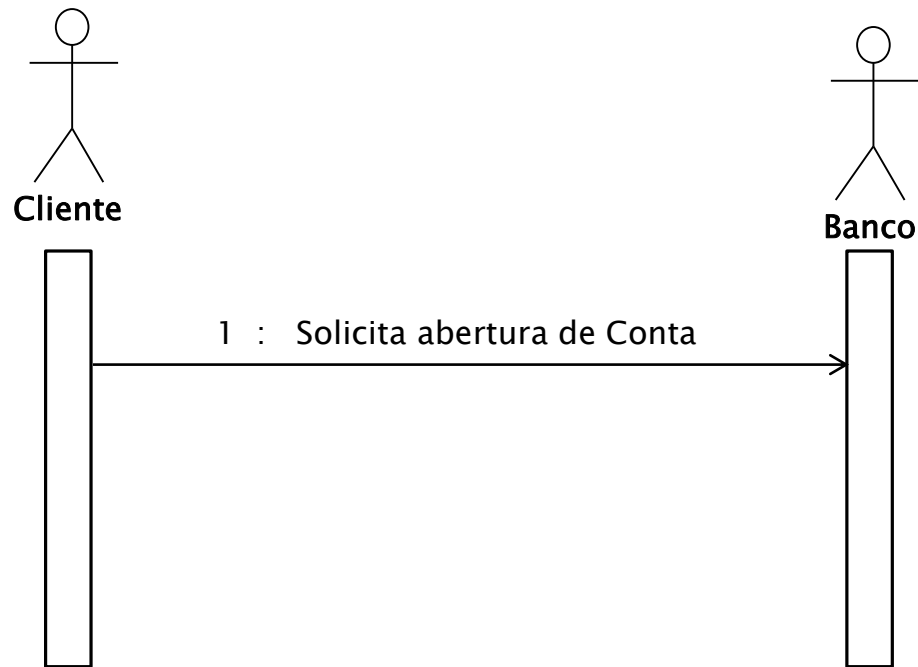


Representação de Mensagens

- ✓ As mensagens são representadas por **retas** entre os dois componentes envolvidos, contendo uma **seta** para indicar qual objeto ou ator disparou a mensagem no outro objeto.
- ✓ O componente posicionado atrás da seta indica qual o gerador da mensagem.
- ✓ O componente para o qual a seta aponta indica o receptor da mensagem.
- ✓ As mensagens são apresentadas na posição horizontal entre as linhas de vida dos componentes (objetos ou atores) e sua ordem sequencial é demonstrada de cima para baixo no diagrama, além de possuírem uma numeração que identifica a temporalidade em que foram sendo disparadas.



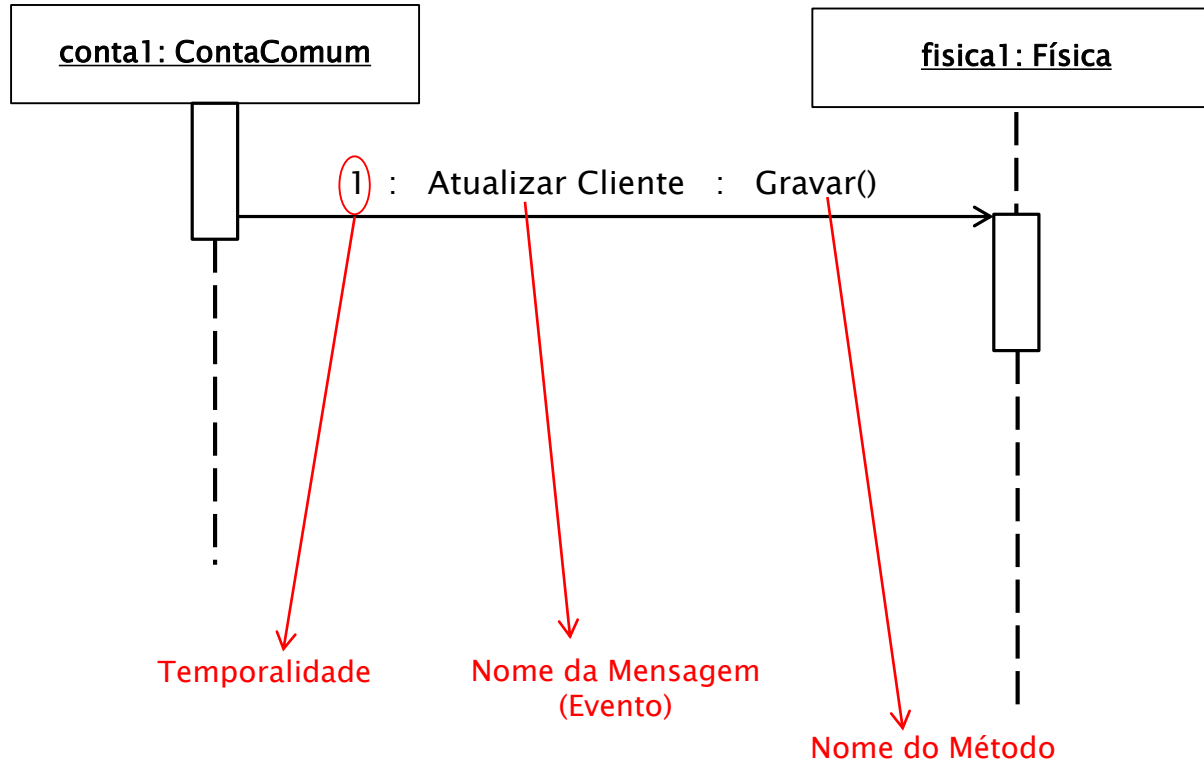
Mensagem sem disparo de Método



Obs. Linha que representa a Mensagem definida com traço fino



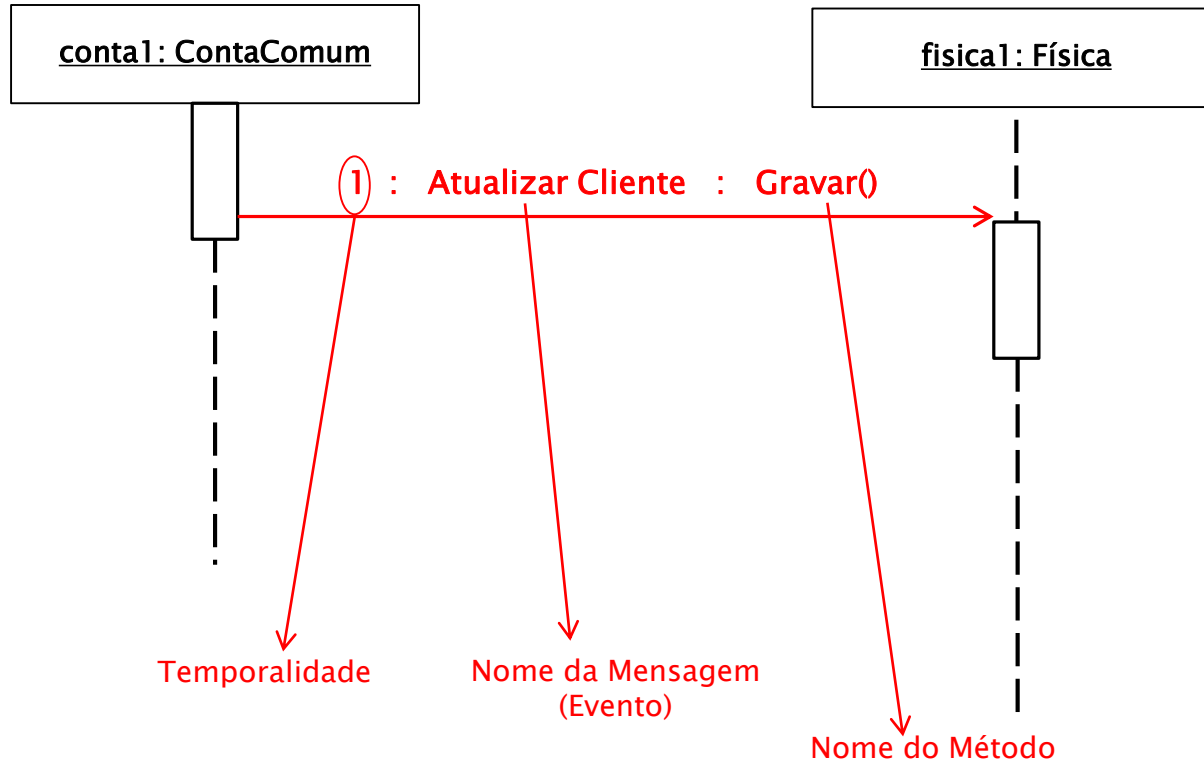
Mensagem com disparo de Método



Obs. Linha que representa a Mensagem definida com traço grosso



Mensagem dirigida a Objetos que já existem



Quando os objetos já existem, a seta da mensagem atinge a Linha de Vida do objeto, identificando que o Foco de Controle está sob o objeto em questão.



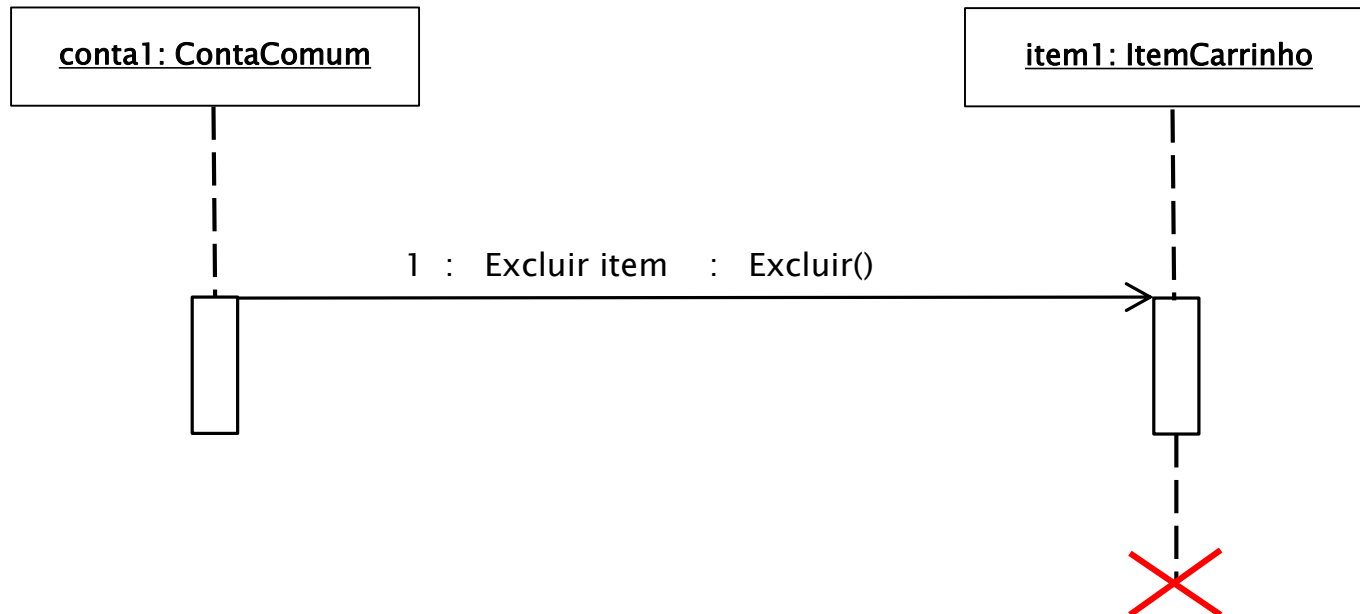
Mensagem que cria novos objetos



Quando a mensagem cria um novo objeto, a **seta da mensagem atinge o retângulo que representa o objeto receptor da mensagem**, indicando que a mensagem representa um método **construtor**. Nesse caso, o objeto passa a existir somente a partir daquele momento.



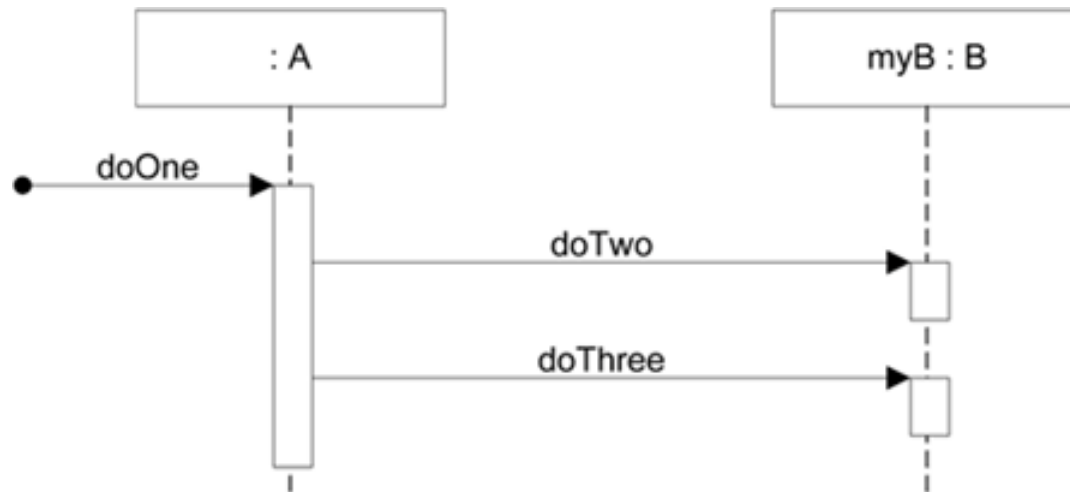
Mensagem destrutora de Objetos



Uma mensagem pode também representar um método destrutor, ou seja, um método que elimina um objeto não mais necessário. Nesse caso, a mensagem atinge a linha de vida de um objeto e a interrompe com um X.



Diagrama x Código



```

public class A {
    private B myB = new B();

    public void doOne() {
        myB.doTwo();
        myB.doThree();
    }
}
  
```

```

class B {
    public void doThree() {
        // ...
    }

    public void doTwo() {
        // ...
    }
}
  
```




Diagrama x Código



```

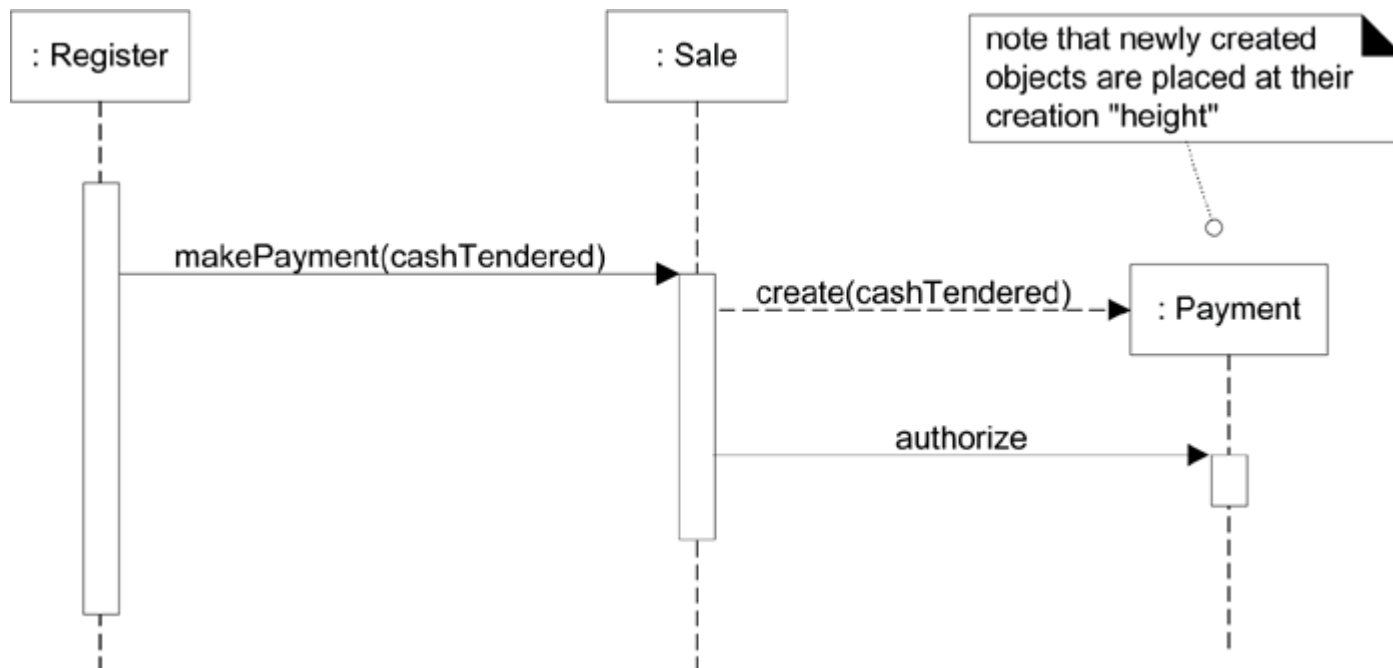
public class Sale {
    private Payment payment;

    public void makePayment(Money cashTendered) {
        payment = new Payment();
        // ...
    }

    // ...
}

```

Diagrama x Código



```

public class Register {
    Sale sale;

    public void makePayment(Money cashTendered) {
        // ...
        sale.makePayment(cashTendered);
        // ...
    }

    // ...
}
  
```

```

class Sale {
    private Payment payment;

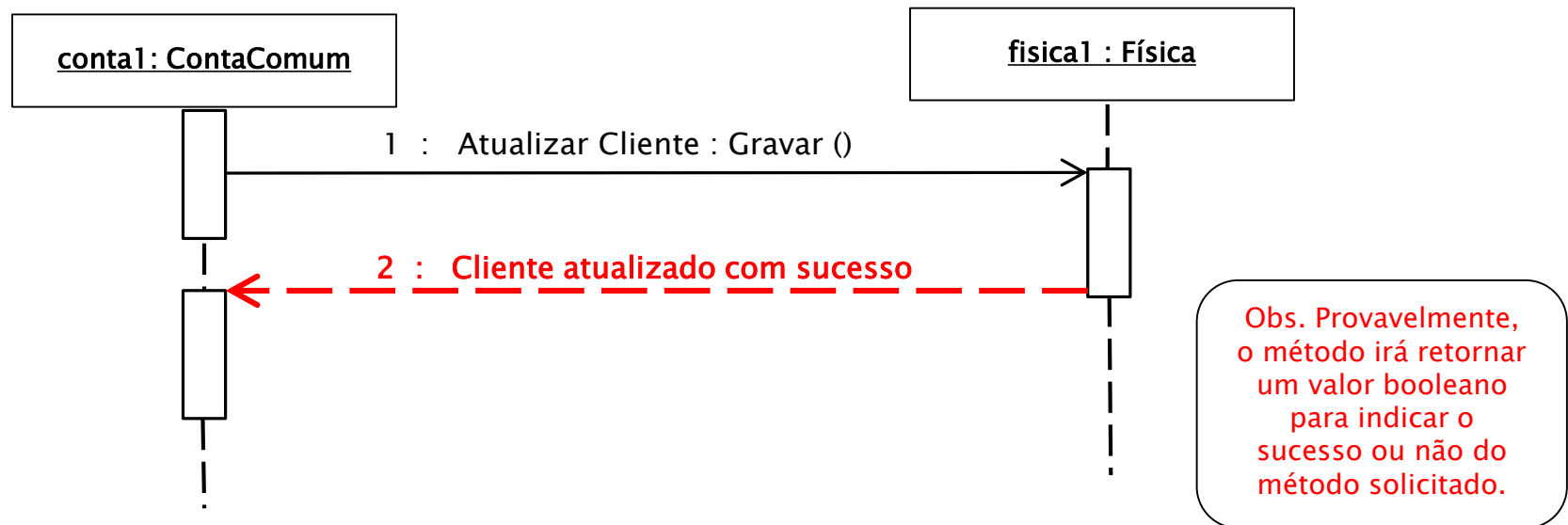
    public void makePayment(Money cashTendered) {
        payment = new Payment();
        payment.authorize();
        // ...
    }
}

class Payment {
    public void authorize() {
        // ...
    }
}
  
```



Mensagens de Retorno

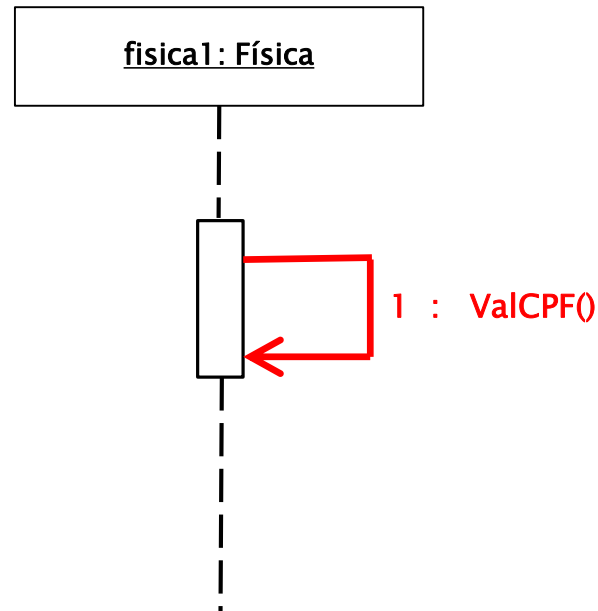
- ✓ Identificam a resposta a uma mensagem para o objeto ou o ator que a chamou.
- ✓ Podem retornar informações específicas do método chamado ou simplesmente um valor indicando se o método foi executado com sucesso ou não.
- ✓ As mensagens de retorno são representadas por uma linha tracejada fina contendo uma seta que aponta para o objeto ou o ator que receberá o resultado do método chamado.





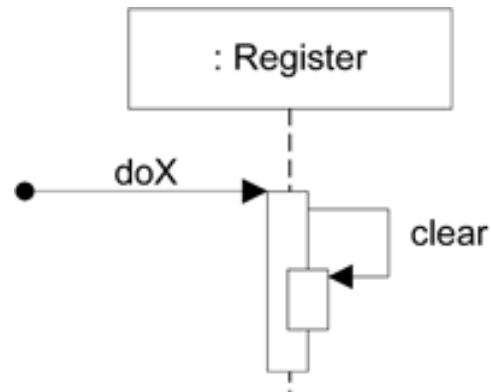
Auto-Chamadas

- ✓ São mensagens que um objeto envia para si mesmo.
- ✓ Nesse caso, chamadas de uma mensagem partem do objeto e atingem o próprio objeto





Chamadas para "this"



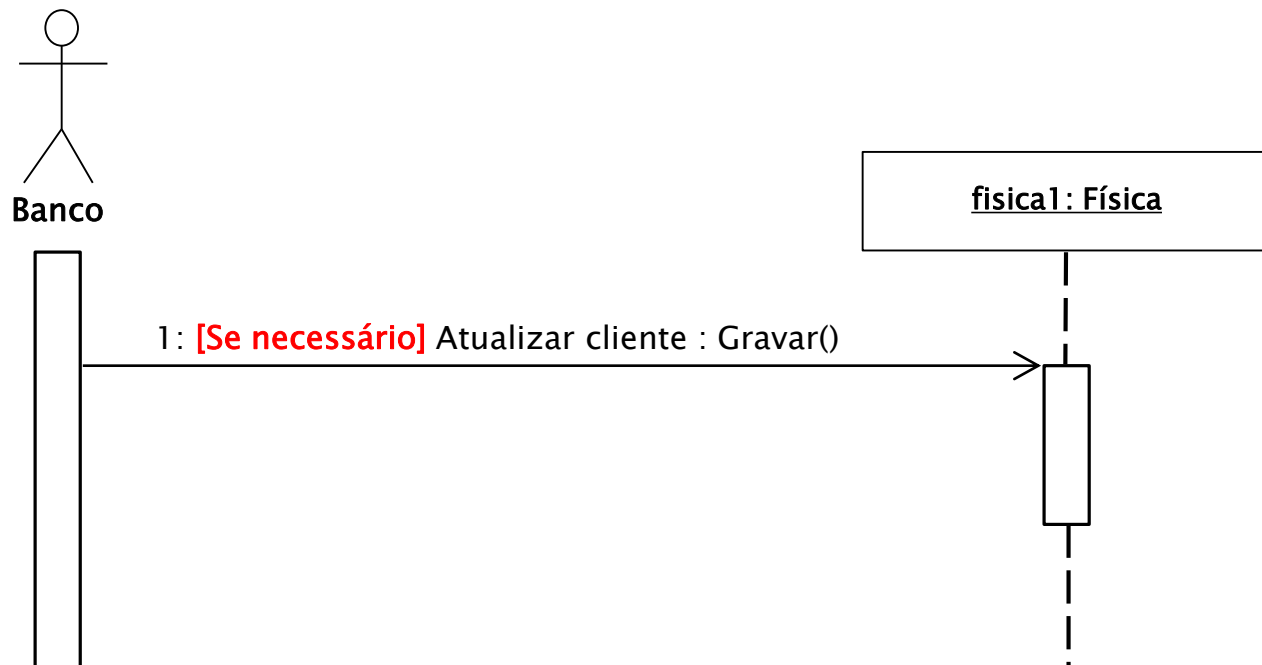
```
public class Register {
    public void doX() {
        // ...
        clear();
        // ...
    }

    public void clear() {
        // ...
    }
}
```



Condições ou Condições de Guarda

- ✓ Indicam que uma mensagem só poderá ser enviada a um objeto se uma determinada condição for verdadeira.
- ✓ As condições são normalmente descritas entre colchetes na mensagem, mas também podem ser representadas por meio de restrições.



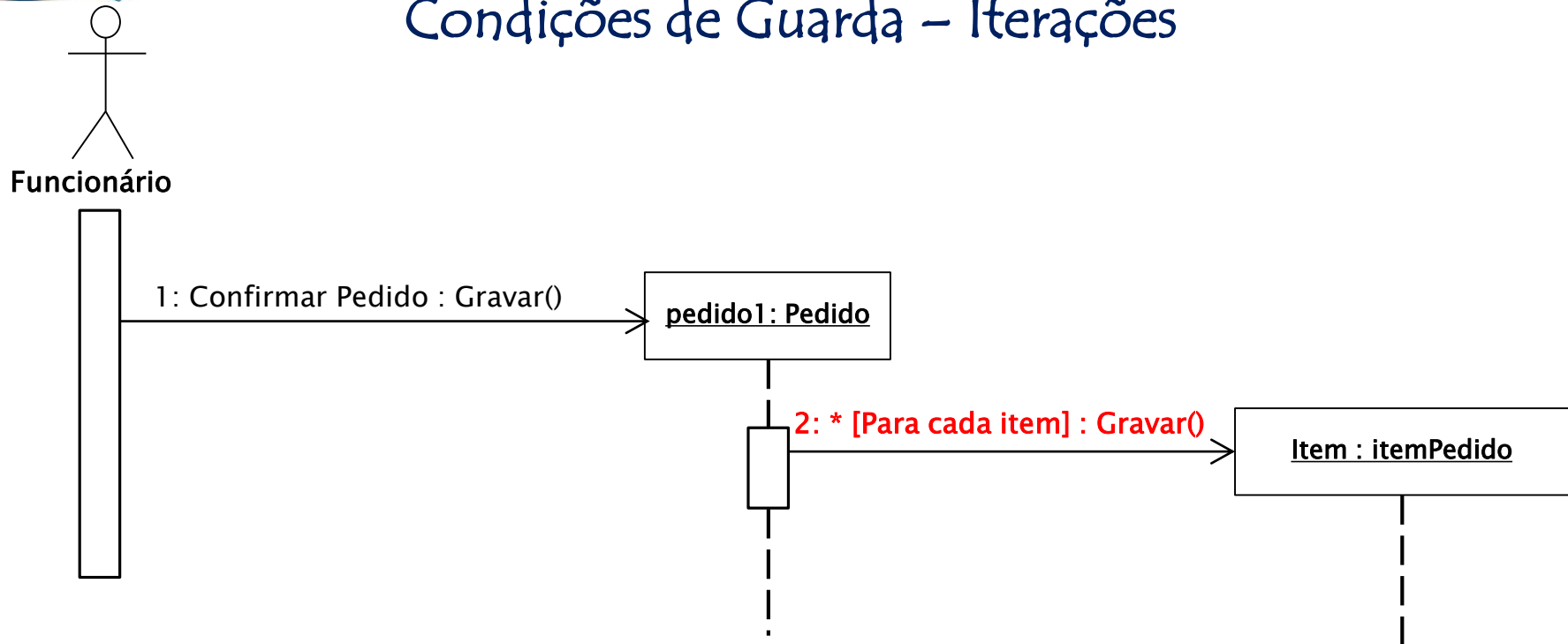
Fonte: UML uma abordagem prática – Gilleanes T. A. Guedes – Novatec 2004



Condições de Guarda – Iterações

- Associada às Condições de Guarda, pode-se representar o disparo de um mensagem a vários objetos, através da utilização do símbolo (*), conhecido por Símbolo de iteração.
- Este símbolo deve estar posicionado antes da Condição de Guarda.

Condições de Guarda – Iterações



Obs. Um pedido pode conter muitos itens de pedido. Assim, caso o pedido venha a ser confirmado, cada um de seus itens precisa ser gerado. Dessa forma, usa-se o Símbolo de Iteração (*) indicando a possibilidade de que um laço venha a ser executado. Insere-se a condição que informa que para cada item do pedido deve ser chamado um método construtor para gerar uma nova instância da classe ItemPedido.

Exemplo

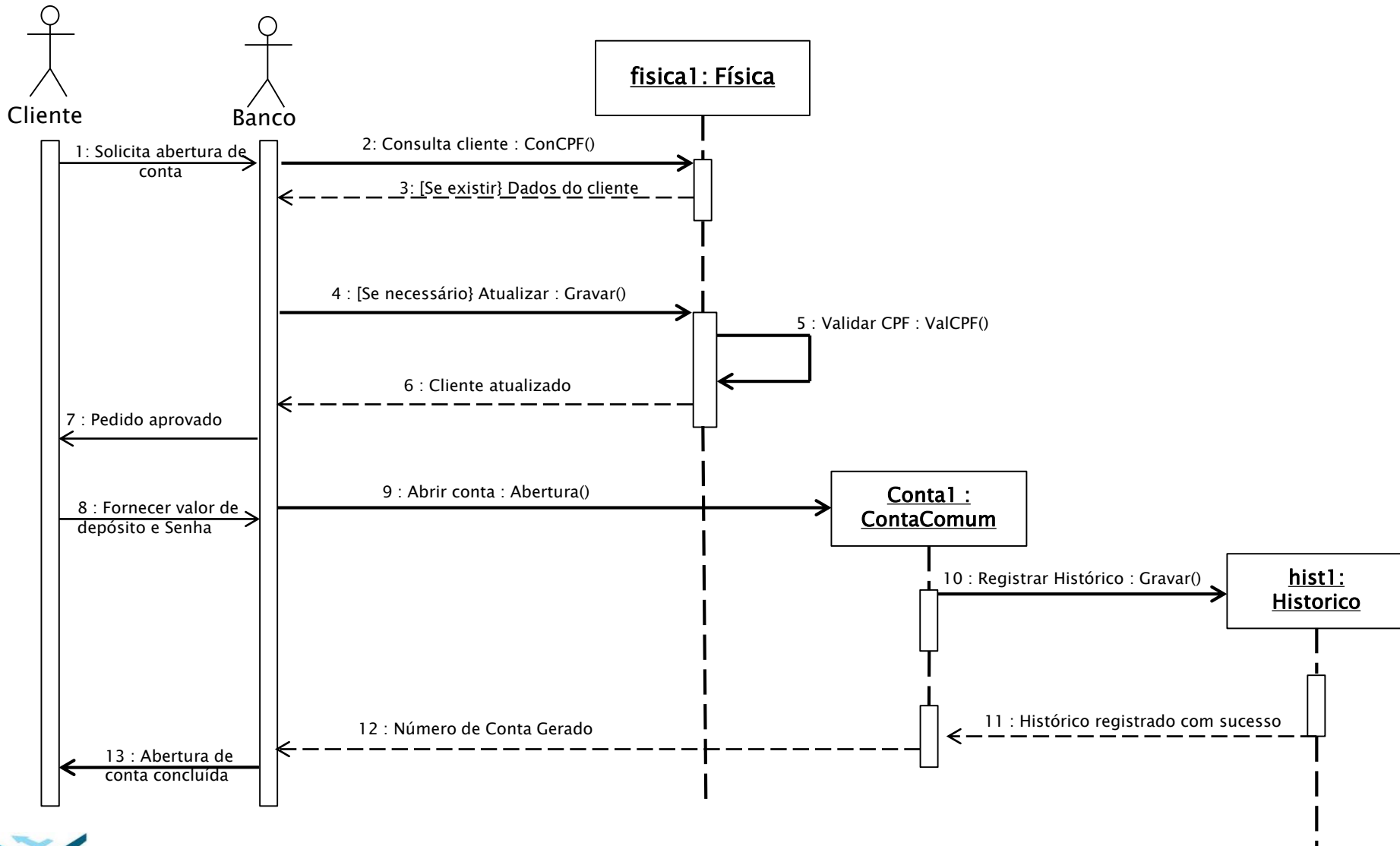




Diagrama de Máquina de Estados

- O comportamento de um objeto em um determinado instante frequentemente depende do **estado** do objeto (valores de seus **atributos** naquele instante);
- Um Diagrama de Estados modela os **estados** de um objeto, as ações executadas e as transições entre os estados do objeto;
- Por meio do Diagrama de Estados, garante-se que todos os **eventos** possíveis para os estados tenham sido levados em conta;
- Especifica a sequência de estados que um objeto pode ter.
- Considera eventos, condições e ações que levam o objeto a alcançar tais estados.

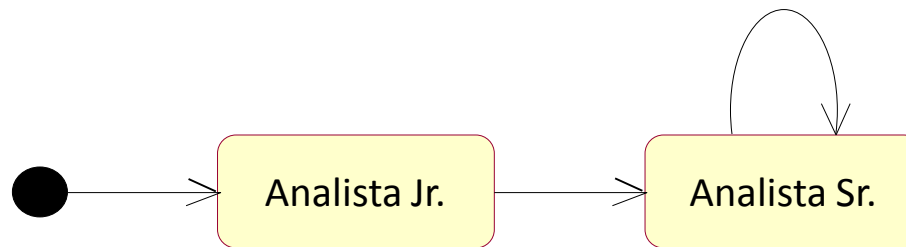




Diagrama de Máquina de Estados e UML

- Incorporado à UML e baseado em [statecharts](#) (David Harel , 1990);
- Pode ser utilizado em qualquer fase do projeto;
- Qualquer elemento de projeto pode ter um diagrama de Estados para melhor compreensão ou exibição de seu comportamento;
- Em geral utiliza-se para se demonstrar os [estados](#) de uma [instância de uma classe](#) ou para representar os estados de um Caso de Uso;
- Recomendado apenas quando existir um certo grau de complexidade referente à transição de estados de objetos envolvidos no processo.



-



Diagrama de Estados – Elemento ESTADO

- Representa a situação em que um objeto se encontra em um determinado instante;
- Determina que ações podem ser feitas sobre o objeto;
- Pode demonstrar:
 - ✓ Espera pela ocorrência de um evento;
 - ✓ Reação a um estímulo;
 - ✓ Execução de alguma atividade;
 - ✓ Satisfação de alguma condição.
- Exemplo: Um ventilador simples pode ter os estados **desligado** e **ligado**. Um ventilador mais elaborado pode ter os estados **desligado**, **ventilando fraco**, **ventilando médio** e **ventilando forte**;
- Representação:





Estados especiais

- O estado inicial é o estado que o objeto se encontra quando ele é criado.
 - ✓ Um estado inicial é obrigatório.
 - ✓ O estado inicial é representado por um círculo sólido.
- Um estado final indica o final do ciclo de vida de um objeto.
 - ✓ Um estado final é indicado por um círculo sólido circunscrito por um outro vazado.
 - ✓ Pode haver mais de um estado final.





Transições

- ✓ Representam um **evento** (significativo para o Sistema) que causa **mudança de estado** no objeto.
- ✓ Representação:

—————→
Evento(lista-parâmetros) [condição] ação

- ✓ Transições **Internas**: não causam mudança de estado no objeto.



Diagrama de Estados

