

Resumo de IMS

Modelagem de Interações

Introdução

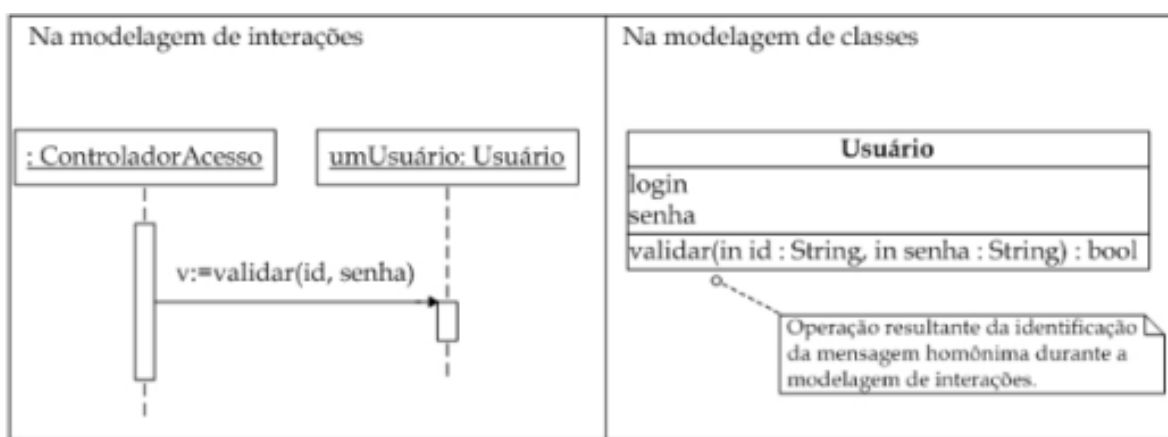
- Modelo de interações preenche as lacunas presentes nos modelos anteriores (casos de uso e de classes).
- O **modelo de interações** representa as mensagens trocadas entre objetos para a execução dos casos de uso e faz parte do modelagem dinâmica do sistema.
- O **diagrama de interações** representa as ações do sistema internamente para que seja realizado um caso de uso.
- O **conjunto de diagrama de interações** do sistema forma seu modelo de interações.
- Principais objetivos:
 - Adquirir informações adicionais e aprimorar outros modelos, especialmente o **modelo de classes**.
 - Fornece uma visão detalhada dos objetos e mensagens na realização de um caso de uso.

Mensagem

- Um sistema orientado a objetos é uma rede de objetos que troca mensagens.
 - Objetos interagem entre si por meio das **mensagens** e realizam as funcionalidades.
 - Um objeto envia uma mensagem para outro

objeto quando o primeiro deseja alguma tarefa do segundo.

- Mensagens de um objeto para outro indicam operações que classes devem ter.
- **Mensagem** é a requisição do objeto remetente a um objeto receptor para que o último realize alguma operação.



- Uma mensagem implica na realização de uma operação no objeto receptor.

Notação das mensagens

Na UML, a mensagem possui a seguinte sintaxe:

`[[expressão-sequência] controle:] [v :=]
nome [(argumentos)]`

O termo **controle** pode ser uma condição ou iteração:

`'*' '[cláusula de iteração]'`
`'['cláusula-condição']'`

Aqui estão alguns exemplos:

Mensagem simples, sem cláusula alguma.

1: adicionarItem(item)

Mensagem com cláusula de condição.

3 [a > b]: trocar(a, b)

Mensagem com cláusula de iteração e com limites indefinidos.

2 *: desenhar()

Mensagem com cláusula de iteração e com limites definidos.

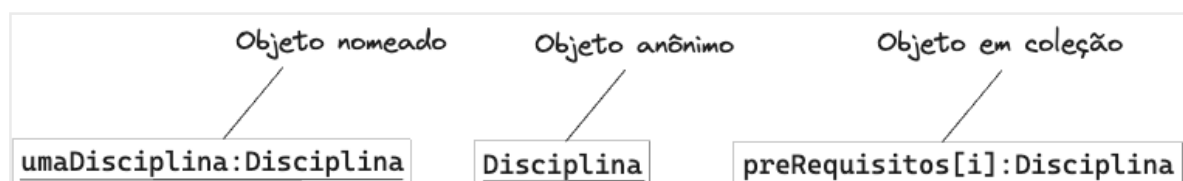
2 *[i := 1..10]: figuras[i].desenhar()

Mensagem aninhada com retorno armazenado na variável x.

1.2.1: x := selecionar(e)

Notação dos objetos

- Podemos representar objetos anônimos ou objetos nomeados.
- Também podemos representar objetos em uma coleção.
- Pode-se representar classes também (para quando mensagens forem enviadas para a classe).
 - Mensagens para uma classe disparam uma operação estática.
 - Representação da classe é a mesma que a do objeto, **porém não é sublinhada**.



Multiobjetos

- **Multiobjeto** na UML designa uma coleção de

- objetos de uma mesma classe.
- É representado por dois retângulos sobrepostos

Tipos de Diagramas de Interação

1. Diagrama de sequência: Foco nas mensagens enviadas no decorrer do tempo.
 2. Diagrama de comunicação: Foco nas mensagens enviadas entre objetos relacionados.
 3. Diagrama de visão geral de interação: Representa uma visão geral de diversas interações entre objetos. Útil para modularizar a construção do diagrama de sequências (ou de comunicação).
- Diagrama de sequência e de comunicação são equivalentes.

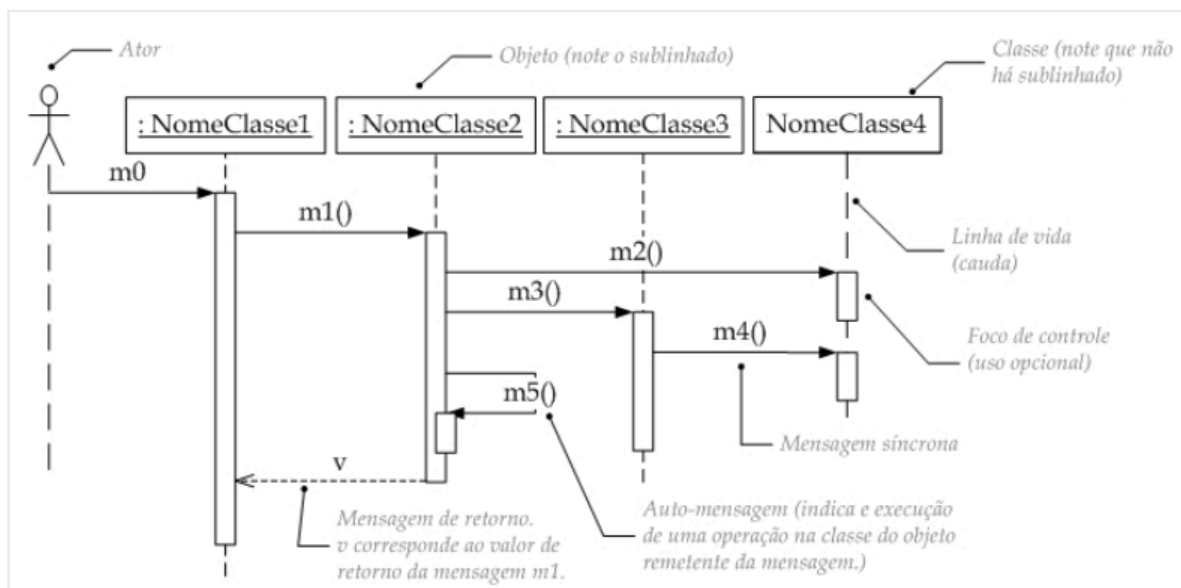
Diagrama de Sequência

- Enfatiza a ordenação temporal das mensagens, sendo baseado em um caso de uso de mesmo nome.
- Objetos são dispostos na horizontal de modo que abaixo de cada objeto existe uma linha de vida.
- Cada linha de vida possui seu foco de controle (quando o objeto está fazendo algo).
- Mensagens são representadas por linhas horizontais rotuladas que parte do remetente para o receptor.
- Representações:
 - Posição vertical das mensagens indica a ordem em que são enviadas

- Deduzimos a ordem de envio a partir das expressões de sequência.
- Podemos indicar criação e destruição de objetos.

Elementos gráficos de um DS

- Atores.
- Objetos, multiobjetos e classes.
- Mensagens.
- Linhas de vida e focos de controle.
- Criação e destruição de objetos.
- Iterações.



Atores

- Representam papéis que interagem com o sistema e seus objetos.
- É um objeto externo ao sistema (está fora do escopo) podendo ser tanto um ser humano quanto um elemento externo.

Objetos

- Elementos que participam das interações no diagrama.
- Representam uma classe ou funcionalidade.
- Interagem por meio das mensagens.
- São representados por um retângulo com seu nome.

Linha de Vida

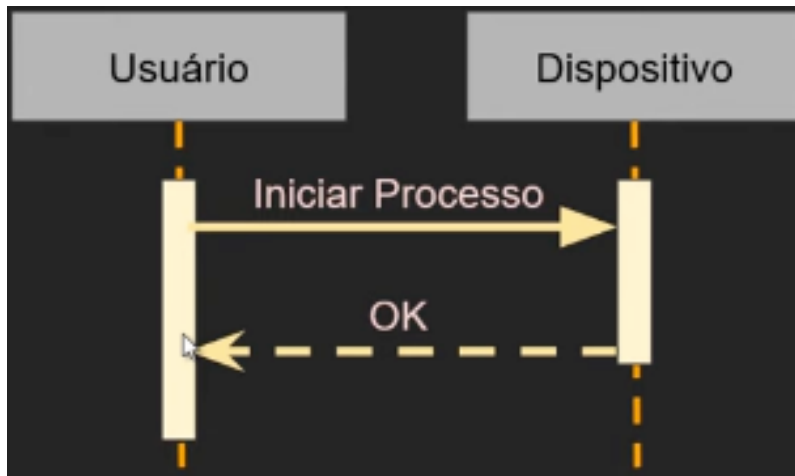
- **Linha de vida** de um objeto representa o período de tempo no qual o objeto existe.
- Quando a linha de vida de um objeto encerra, ele expira e não é mais útil ao sistema.

Foco de Controle

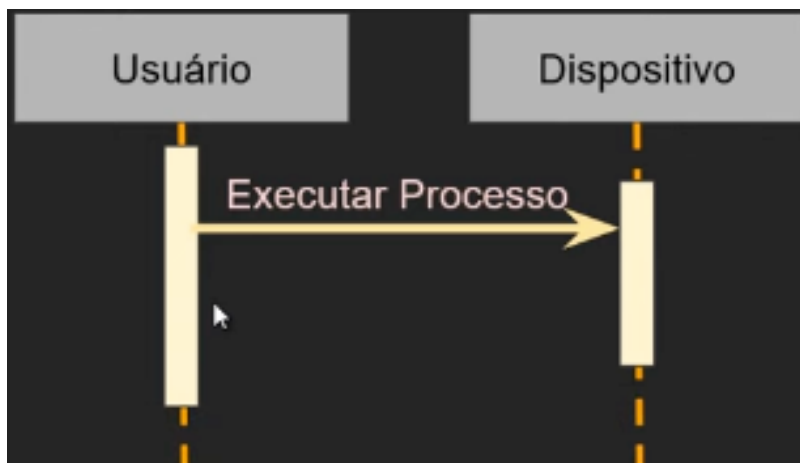
- **Foco de controle** representa o período no qual o objeto está participando ativamente de um processo. É representado na linha de vida por um retângulo vertical.

Tipos de Mensagens

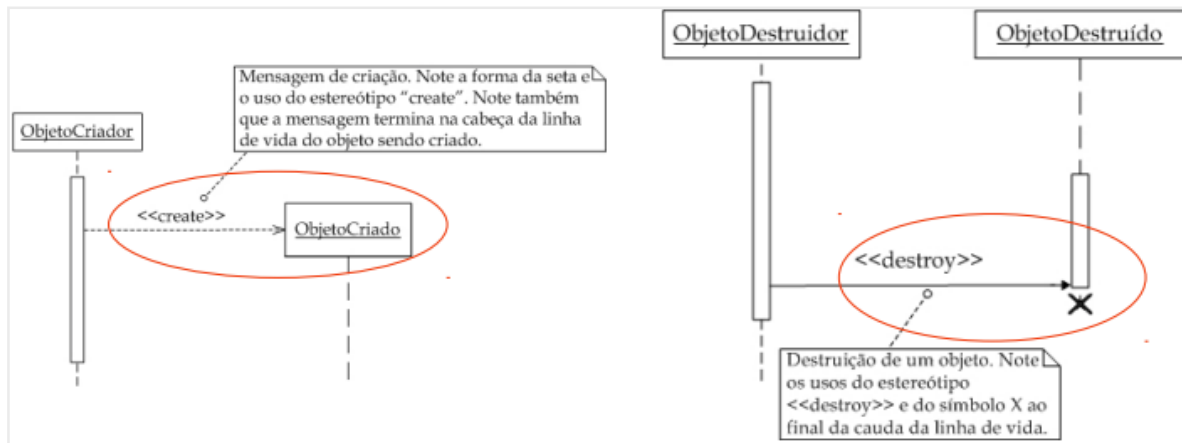
- Mensagem síncrona: Mensagem espera por uma resposta para a interação prosseguir.



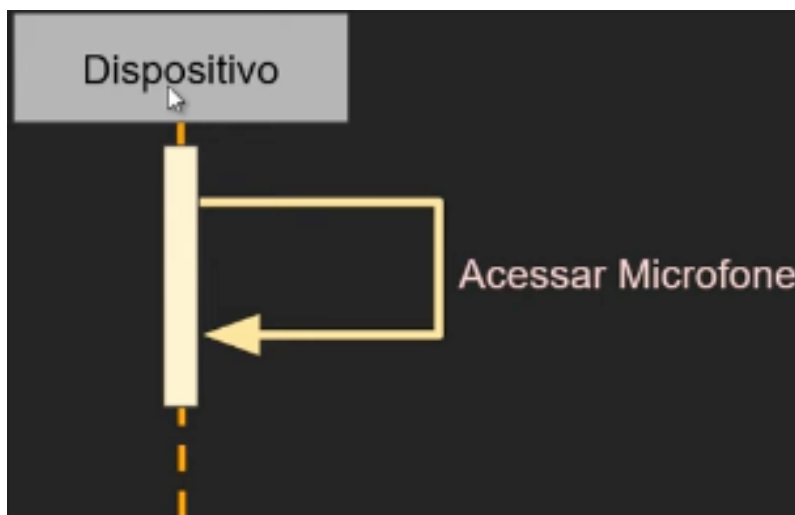
- Mensagem assíncrona: Não espera resposta do destinatário para a interação prosseguir.



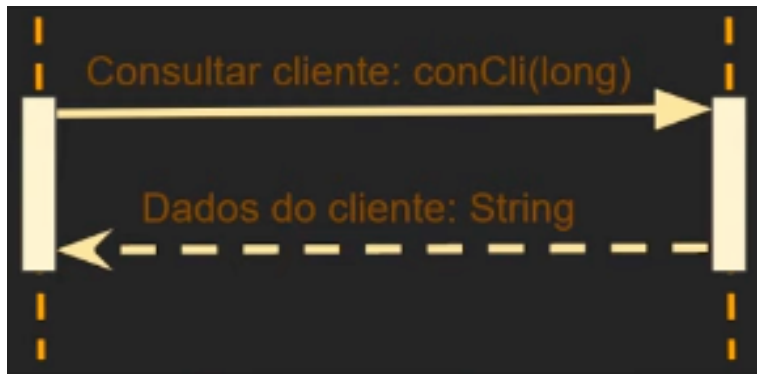
- Mensagem de criação: Permite instanciar um novo objeto no diagrama.
- Mensagem de destruição/exclusão: Permite eliminar um objeto, memória é desalocada.



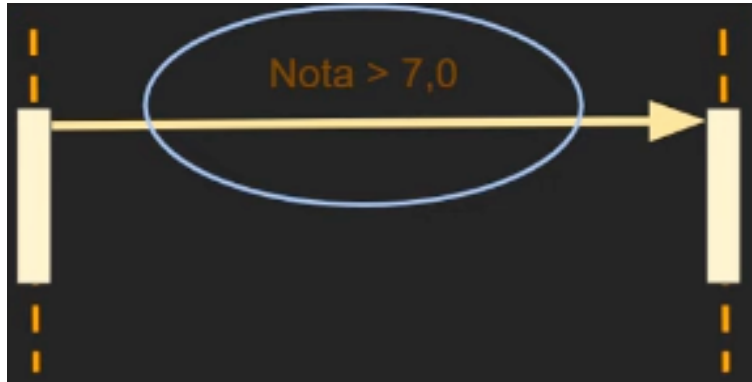
- Mensagem reflexiva (ou auto-mensagem): Aquela em que o remetente é também o receptor. Pode ser síncrona ou assíncrona.



- Mensagem de retorno: Identifica a resposta a uma mensagem enviada para o objeto.



- Nesse caso a mensagem `Consultar cliente` também dispara o método `conCli(long)`.
- Mensagem de guarda: Usadas para modelar condições, úteis quando é necessário restringir o fluxo de uma mensagem de acordo com uma condição.



Gate

- Refere-se ao ponto de conexão entre uma mensagem que esteja dentro de um fragmento de interação com uma que esteja fora.

Quadro de interação

- Também conhecidos como **fragmentos**

combinados, representam diferentes condições de interação entre objetos durante a execução de uma sequência de mensagens.

- Estão atrelados a **operadores de controle estruturado**(ou **operadores de interação**).
 - Os principais operadores são:
 - Execução opcional: **opt**.
 - Execução condicional: **alt**.
 - Execução paralela: **par**.
 - Interação com outro diagrama: **ref**.
 - Quebra de execução: **break**.
-

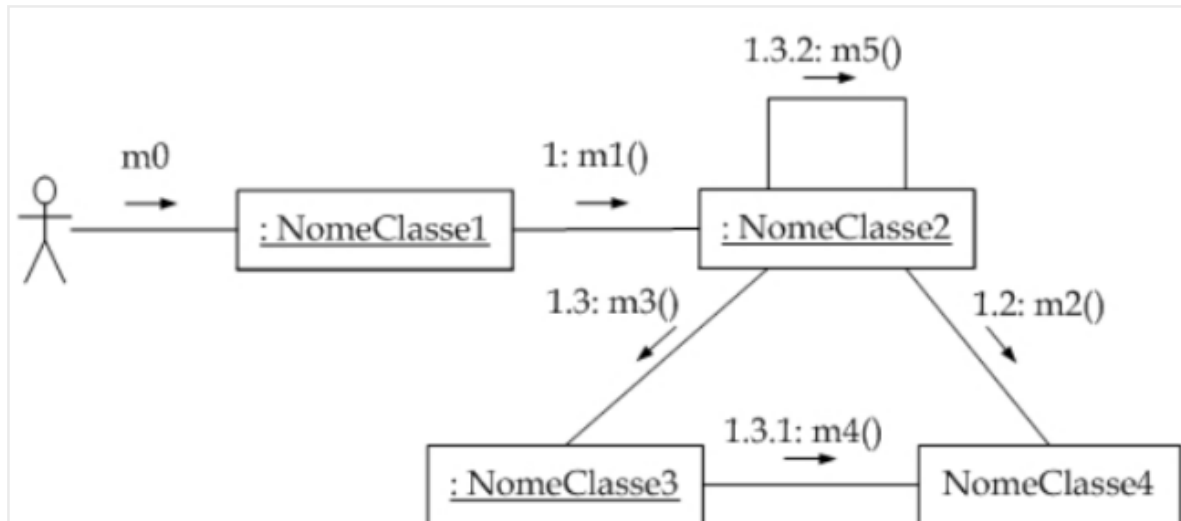
Diagrama de Comunicação

- Semelhante ao diagrama de objetos, no entanto são adicionados setas e rótulos de mensagens nas ligações de objetos.
- Enfatiza a organização estrutural dos objetos que enviam e recebem mensagens.
- Objetos estão dispostos em duas dimensões o que aumenta a legibilidade mas dificulta a identificação da ordem de envio das mensagens.
- A sequência de um diagrama de comunicação não fica tão clara, sendo necessário analisar a numeração dos métodos.

Elementos Gráficos de uma DC

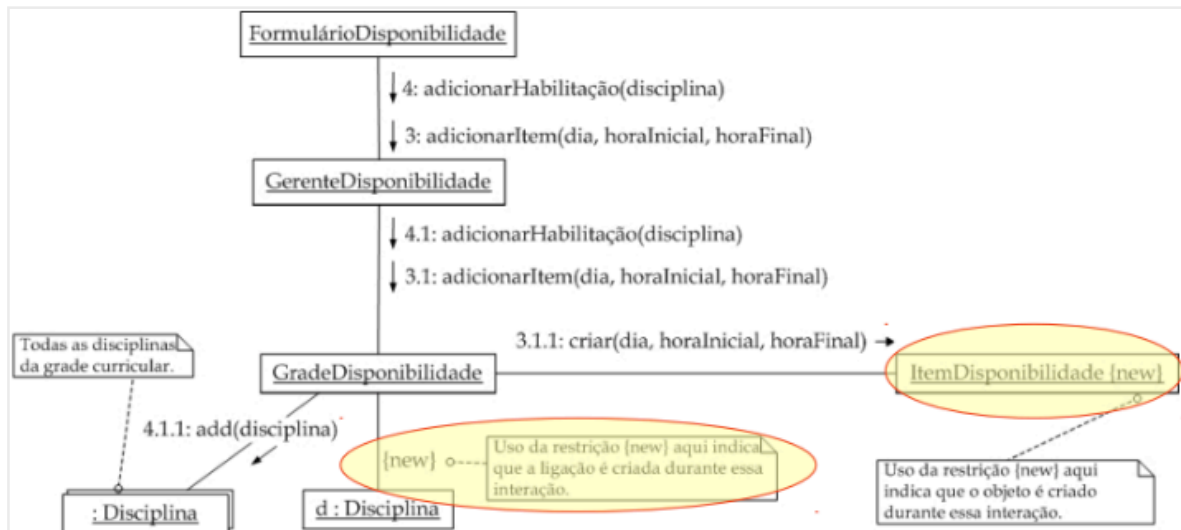
- Atores.
- Objetos, multiobjetos e classes.

- Mensagens.
- Ligações entre objetos.
- Criação e destruição de objetos.
- Iterações.



Criação e Destruição de Objetos em um DC

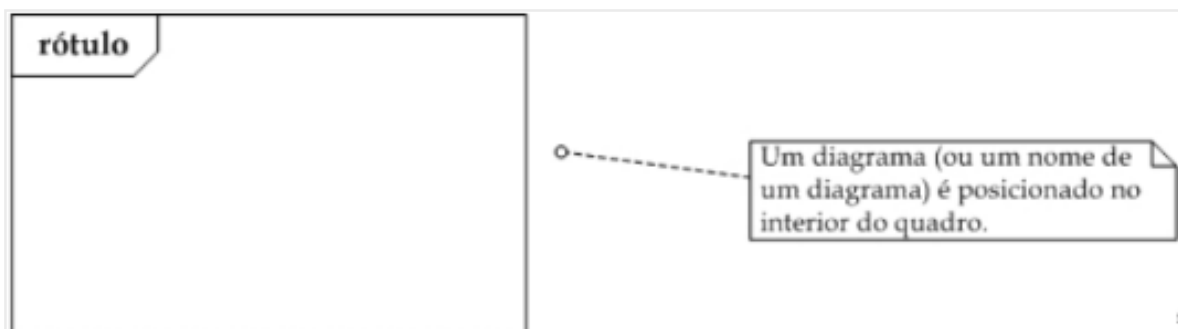
- Durante a execução dos casos de uso, objetos podem ser criados e outros destruídos.
- Foram definidas tags para indicar isso:
 - {new}: Objeto/ligação criada durante a interação.
 - {destroyed}: Objeto/ligação destruída durante a interação.
 - {transient}: Objeto/ligação destruída e criada durante a interação.



Modularização de Interações

Quadros de Interação

- Elemento gráfico utilizado para modularizar a construção de diagramas de sequência/comunicação.
- Objetivos:
 - Dar um **rótulo** ao diagrama no interior do quadro.
 - Fazer referência a esse diagrama separadamente usando o **ref**.
 - Definir um fluxo de interação.



- Fluxos de controle possíveis:

- Alternativa.
- Opções.
- Iterações.

Sequência X Comunicação

- Sequência:
 - Ordem das mensagens.
 - Ligação entre objetos em segundo plano.
- Comunicação:
 - Ligação entre objetos.
 - Ordem em segundo plano.

Construção do Modelo de Interações

- O objetivo da modelagem de interações é identificar **interações** e, mais adiante, **responsabilidades**.

Acoplamento e Coesão

- Dois principais princípios de projetos:
 - **Coesão**: Medida do quão relacionada/focalizada estão as responsabilidades de uma classe.
 - **Acoplamento**: Medida do quão fortemente uma classe está conectada a outras classes.
- **Ideal**: Criação de modelos com **alta coesão** e **baixo acoplamento**.

Dicas para a Construção do MI

- Identifique as classes conceituais que participam de cada caso de uso.
 - Identifique quaisquer classes de software que ajudem a organizar as tarefas a serem executadas.
 - Defina que objetos criam/destróem outros objetos.
 - Verifique a consistência dos diagramas de interação com relação ao MCU e ao modelo de classes.
 - Se certifique de que o objeto de controle realiza apenas a coordenação da realização do caso de uso.
 - Faça o máximo para construir diagramas de interação o mais inteligíveis possível.
 - Utilize o princípio de projeto conhecido como Lei de Demeter.
-

Modelagem de Atividades

Diagrama de Atividade

- Tipo especial de diagrama de estados onde são representados os estados de uma atividade.
- Se trata de um diagrama dinâmico em que é representado o fluxo de controle de uma atividade para outra com uma descrição de ações passo-a-passo.
- São um tipo de fluxograma estendido pois

permitem representar ações concorrentes e sua sincronização.

- Elementos são divididos em 2 grupos: Controle sequencial e controle paralelo.
- Componentes de fluxos sequenciais:
 - Estado de ação.
 - Estado atividade.
 - Estados inicial, final e condição de guarda.
 - Transição de término.
 - Pontos de ramificação.
- Componentes de fluxos paralelos:
 - Barras de sincronização: Barra de bifurcação (**fork**) e barra de junção (**join**).

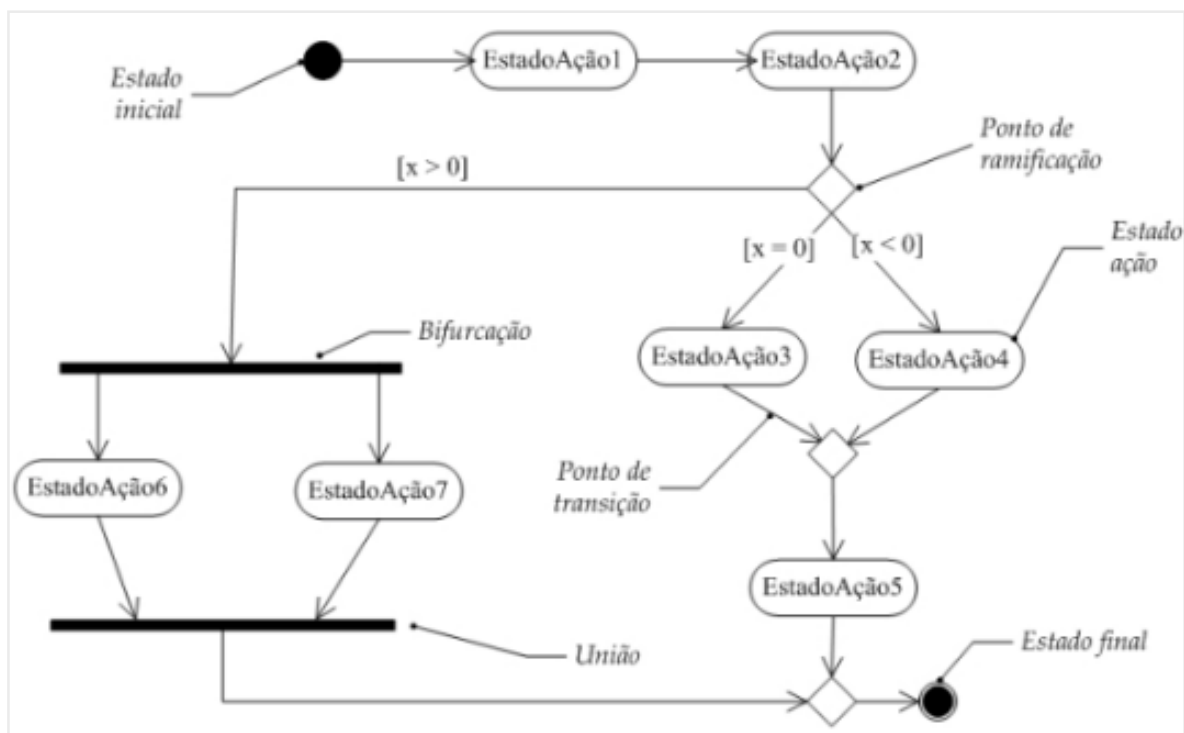
Atividade e Ação

- **Atividade** é um processo de negócio que normalmente descreve a implementação de um caso de uso.
- **Ação** é passo individual (atômico) dentro de uma atividade.

Fluxos de Controle Sequenciais

- Estado podem ser de 2 tipos:
 - Estado atividade: Leva certo tempo para ser finalizado.
 - Estado ação: Realizado instantaneamente.
- Obrigatório ter um **estado inicial** e vários **finais** e **guardas** atreladas a transições. Não ter um estado final pode indicar que o procedimento é **cíclico**.

- **Transição de término** indica o término de um passo e o início de outro.
- **Ponto de ramificação** possui uma única transição de entrada e várias de saída.
 - Para cada saída, há uma transição de guarda associada.
 - Quando o fluxo atinge o ponto de ramificação, somente uma condição deve ser verdadeira.
 - Pode haver transição com **[else]**.
- **Ponto de união** reúne transições que possuem um ponto de ramificação em comum.



Fluxos de Controle Paralelo

- Refere-se a dois ou mais fluxos sendo executados simultaneamente.
- **Barra de bifurcação** recebe uma transição de entrada e cria dois ou mais fluxos de controle paralelos.

- **Barra de junção** recebe duas ou mais transições de entrada e une os fluxos de controle em um único fluxo.
 - Objetivo: Sincronizar fluxos paralelos.
- **Raias de natação** servem para dividir as atividades de um processo entre agentes, subdivide o diagrama em compartimentos.

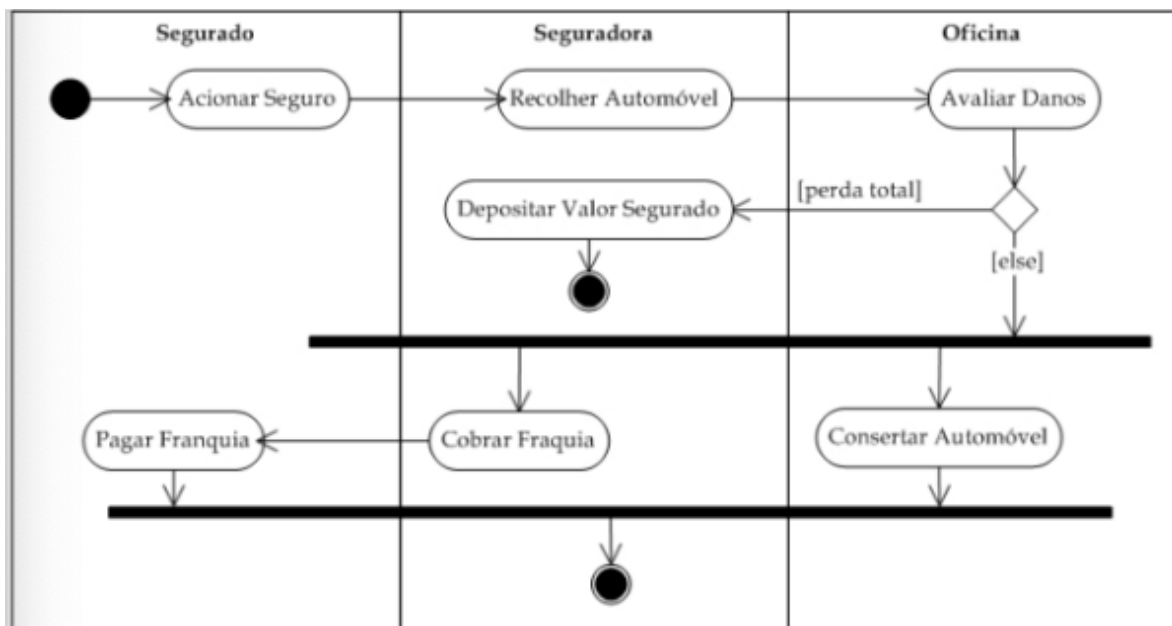


Diagrama de Atividade no Processo de Desenvolvimento Iterativo

Não parece importante..

Modelagem de Estados

- Todo objeto que compõe um sistema de software se encontra em um **estado** particular.
- Um objeto muda de estado quando ocorre algum **evento** interno ou externo ao sistema.

- Na **transição** de um estado para outro, um objeto realiza determinadas ações dentro do sistema.

Diagrama de Transição de Estado

- Diagrama comportamental para descrever como sistema se comporta quando um evento ocorre, considerando os estados, as transições e ações possíveis de um objeto.
- Permite prever todas as possíveis operações realizadas em um sistema, em função de certos eventos, por meio da análise da transição de estados dos objetos.
- Principais elementos de um DTE:
 - Estados
 - Transições
 - Evento
 - Ação
 - Atividade
 - Transições internas
 - Estados aninhados
 - Estados concorrentes

Utilidades

- Ajuda a modelar o comportamento de:
 - Interfaces.
 - Casos de uso.
 - Instâncias de classes
- Porém, normalmente são construídos por classe.

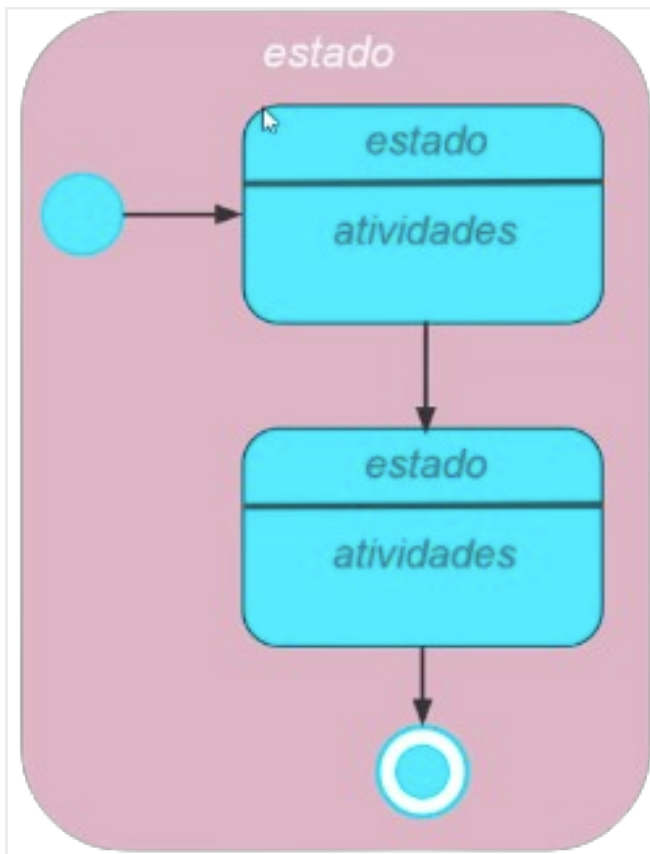
Estado

- Situação na vida de um objeto em que ele satisfaz alguma condição ou realiza alguma atividade. É função dos valores dos atributos e/ou ligações com outros objetos.
- Estados de um sistema: Conjunto de estados de todos os objetos no domínio.
- Estado de objeto no domínio: Conjunto de todos os conceitos onde ele está incluído e todas as ligações dele. Estado é modificado quando objeto entra/sai de um conceito no domínio ou faz/desfaz uma ligação no domínio.
- Estado do modelo de estados: Subconjunto de estados de objetos que são interessantes de destacar.



Estado composto

- Semelhante ao quadro de interação nos diagramas de sequência, refere-se a um estado que possui sub-estados.



Estado Inicial e Estado Final

- **Estado inicial** indica o estado do objeto quando ele é criado, podendo haver apenas um.
- **Estado final** é representado por um círculo eclipsado e sinaliza o fim do ciclo de vida de um objeto.



Transições

- Mudança de um estado para outro estado de um objeto.

- Associa diferentes estados, sendo representado por uma linha entre os estados e uma seta apontando para um deles.
- Uma transição é rotulada da seguinte forma:

evento (lista-parâmetro) [guarda]/ação

- Passos da transição de estado:
 - Elemento se encontra em estado inicial.
 - Evento ocorre.
 - Ação é realizada.
 - Elemento muda de estado.

Eventos

- Uma transição está associada a um evento.
- Evento é algo que acontece em algum momento e que pode modificar o estado de um objeto.
- Pode-se classificar os eventos da seguinte forma:
 - Evento de chamada: Recebimento de uma mensagem/solicitação de outro objeto.
 - Evento de sinal: Recebimento de um sinal de modo que o objeto remetente continua seu processo, diferentemente do evento de chamada.
 - Evento temporal: Passagem de um intervalo de tempo predefinido. É especificado com a cláusula **after** juntamente do intervalo de tempo em parênteses.
 - Evento de mudança: Uma condição se torna verdadeira. É representado por uma expressão de valor lógico (T/F) por meio da cláusula

when.

Eventos resultando em Eventos

- A ocorrência de um evento A pode ocasionar a ocorrência de um evento B relevante para outro objeto.



Condição de Guarda

- Expressão de valor lógico que condiciona o disparo de uma transição.
- A transição correspondente é disparada **se, e somente se**, o evento associado ocorre e a condição de guarda é verdadeira.

Ações

- Ao transitar entre diferentes estados, um objeto pode realizar uma ou mais **ações**.
- Uma ação é uma expressão definida em função dos atributos, operações, associações da classe ou dos parâmetros do evento.
- A ação associada a uma transição é executada **se, e somente se**, a transição for disparada.

Atividades

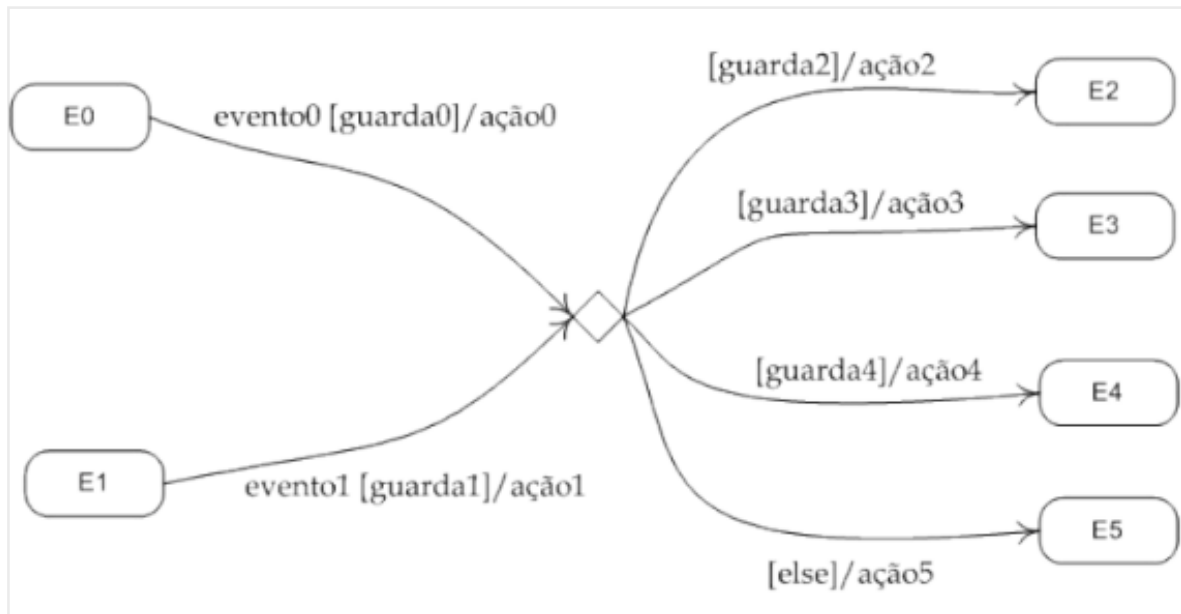
- Semelhante às ações, atividades são algo que deve ser executado.
- Diferenças entre ações e atividades:
 - Atividade pode ser interrompida, uma ação não pode.
 - Atividade está associada a um estado, ação esta associada a uma transição.

Efeitos

- Representa uma ação ou resultado que ocorre durante a transição de um estado para outro.
-

Ponto de Junção

- **Pontos de junção** permitem que duas ou mais transições compartilhem uma "trajetória de transições".
- Dependendo da condição de um objeto, ele pode alternar para diferentes estados por meio das condições de guarda associadas.
- Pode haver transições de saída associadas à cláusula **else**.



Atividades Internas e Cláusulas

- Um objeto pode realizar atividades enquanto está em um estado.
- No compartimento adicional do retângulo de estado, pode-se especificar ações ou atividades a serem executadas.
- Sintaxe geral:

evento/[ação|atividade]

- Há três principais cláusulas: **entry**, **exit**, **do**.
 - **entry**: Especifica uma ação a ser realizada no momento em que o objeto entra no estado. É sempre executada.
 - **exit**: Especifica ações executadas sempre que o objeto sai de um estado. É sempre executada.
 - **do**: Define uma atividade a ser executada quando o objeto passa para um determinado estado.

Paralelismo

- Barras de junção: Ocorre quando duas ou mais transições partem de um mesmo estado.
- Barra de união/junção: Quando duas ou mais transições levam a um mesmo estado.

Identificação de elementos do DTE

- Deve-se identificar os eventos que precedem as transições.
- Outrossim, precisamos verificar se algum fator condiciona o disparo da transição. Se existir, devemos especifica-lo como condição de guarda da transição.
- Eventos externos ao sistema são identificados noas casos de uso.
- De forma geral, toda operação com visibilidade pública de uma classe é um evento em potencial.
- Para identificar os eventos associados às transições podemos analisar as *regras de negócio*.

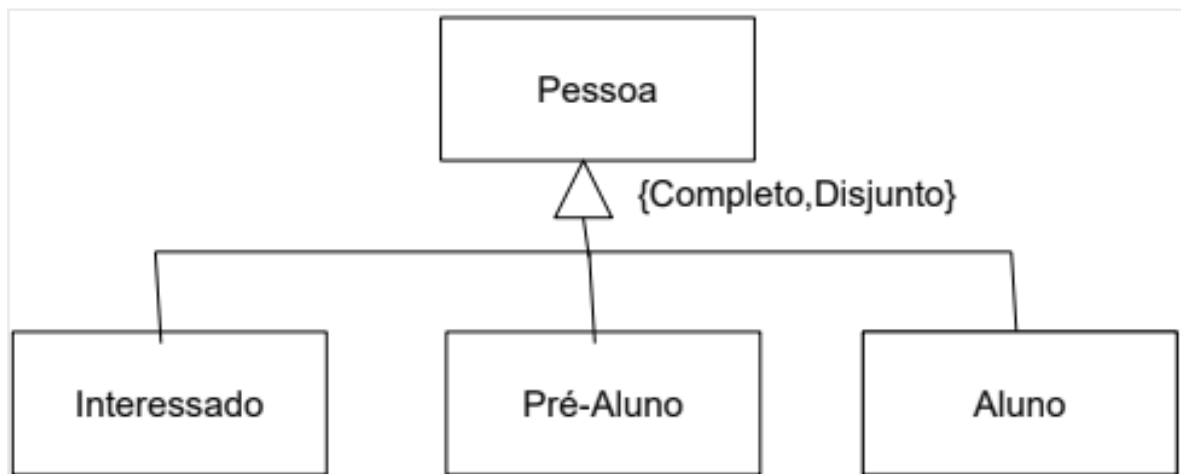
Construção de Diagramas de Transição de Estados

Um DTE para uma classe

- Os diagramas de estado são desenhados **por classe**.
 - Desvantagem: Dificuldade na visualização do

processo como um todo.

- Nem todas as classes precisam de um DTE, apenas objetos rastreáveis e classes com comportamento dinâmico.
- Essas classes tem subclasses que são completas (todas as subclasses foram enumeradas) e disjuntas (herdam uma única subclasses).
- Aqui está um exemplo de um diagrama de classe possível:



- E aqui está o diagrama de estados correspondente:

É considerada pessoa no domínio quando ocorre o evento "Mostrou interesse por frequentar uma turma da escola."

