

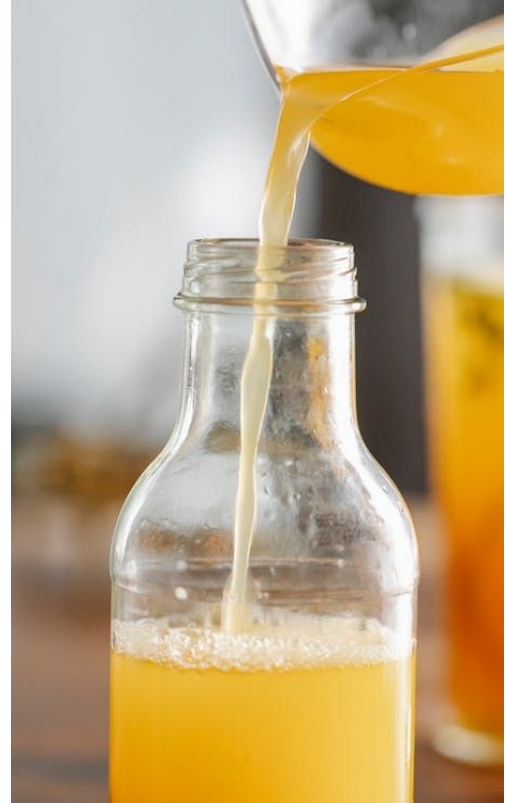
Diagrama de Estados

Projeto de Software

Profª Aline Brito

Estados

Uma jarra está cheia de suco



<https://www.pexels.com/pt-br/foto/anonimo-apetitoso-tentador-borrao-5946785/>

Estados

Um cachorro está cansado



<https://www.pexels.com/pt-br/foto/fotografia-de-close-up-de-um-cao-adulto-de-pelo-curto-bronzeado-e-branco-dormindo-em-tecido-cinza-durante-o-dia-731022/>

Estados

Os objetos de um sistema de software também possuem estados

Estados

Quando um objeto muda de estado dizemos que ele realizou uma transição de estados

Estados

Os estados e as transições definem o ciclo de vida do objeto

Estados

Para mudar de estado um objeto realiza ações

Modelagem de Estados

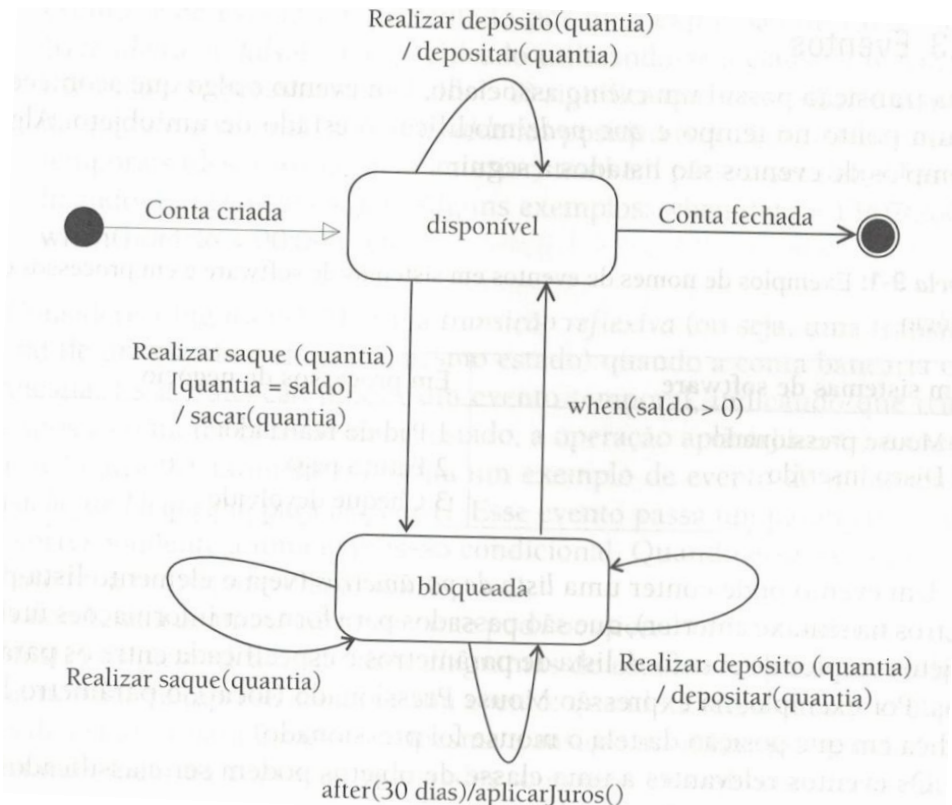
Através da análise das transições entre estados dos objetos podemos prever as possíveis operações realizadas, em função dos eventos

Diagrama de Transição de Estados

Modelagem de Estados

Podemos utilizar o diagrama de transição de estado para realizar a análise das transições

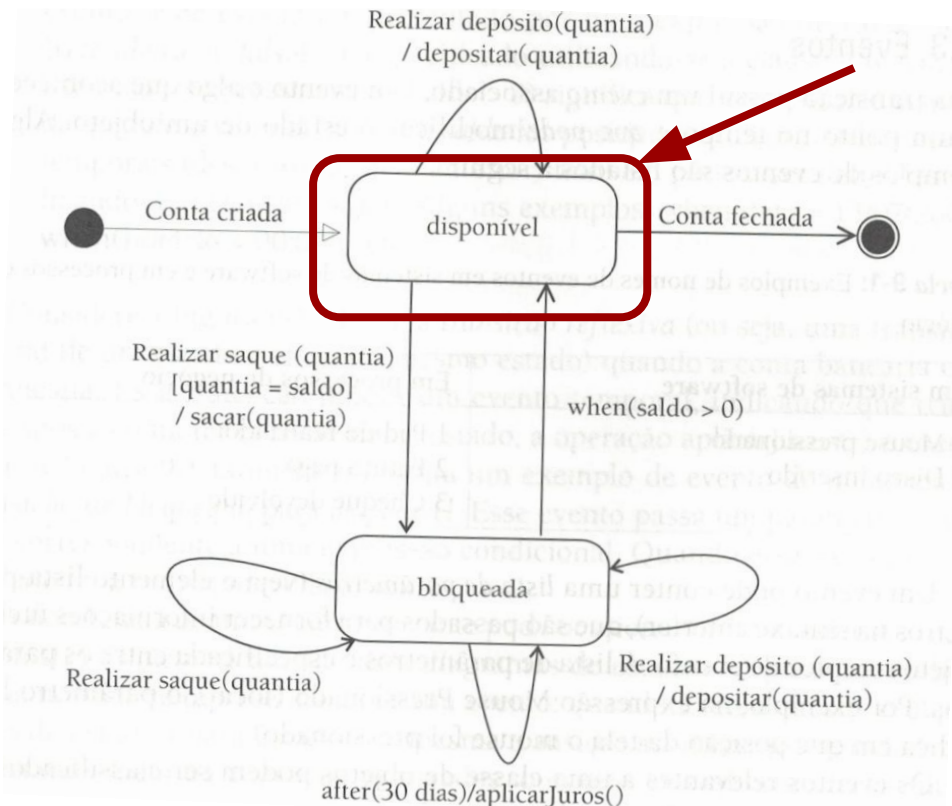
Diagrama de Transição de Estados



Exemplo para a classe
ContaBancaria

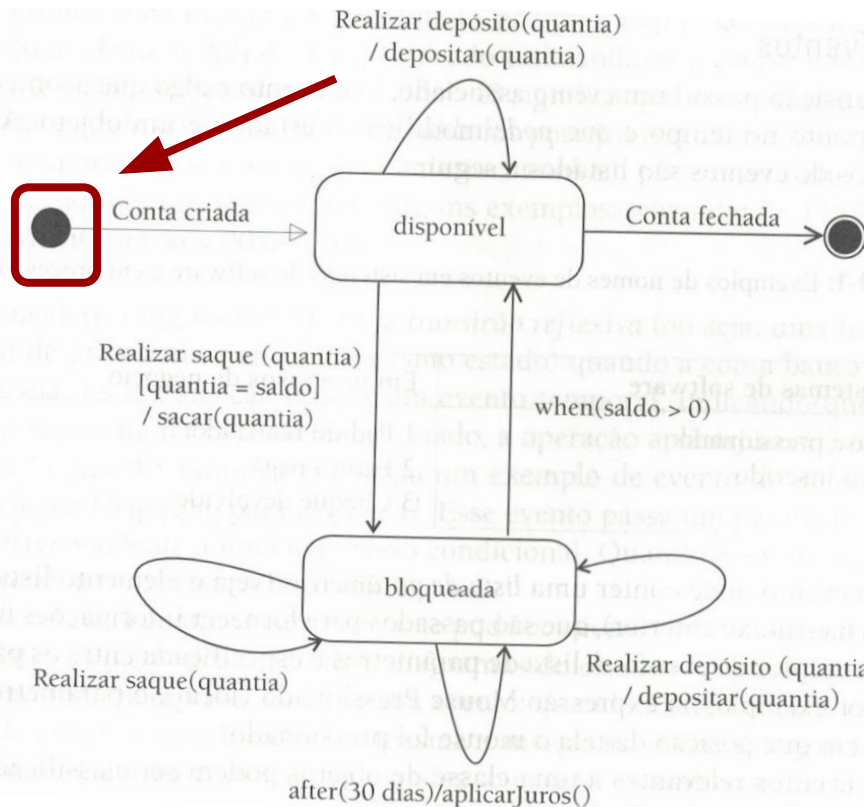


Diagrama de Transição de Estados



Representamos um estado com um retângulo com bordas arredondadas

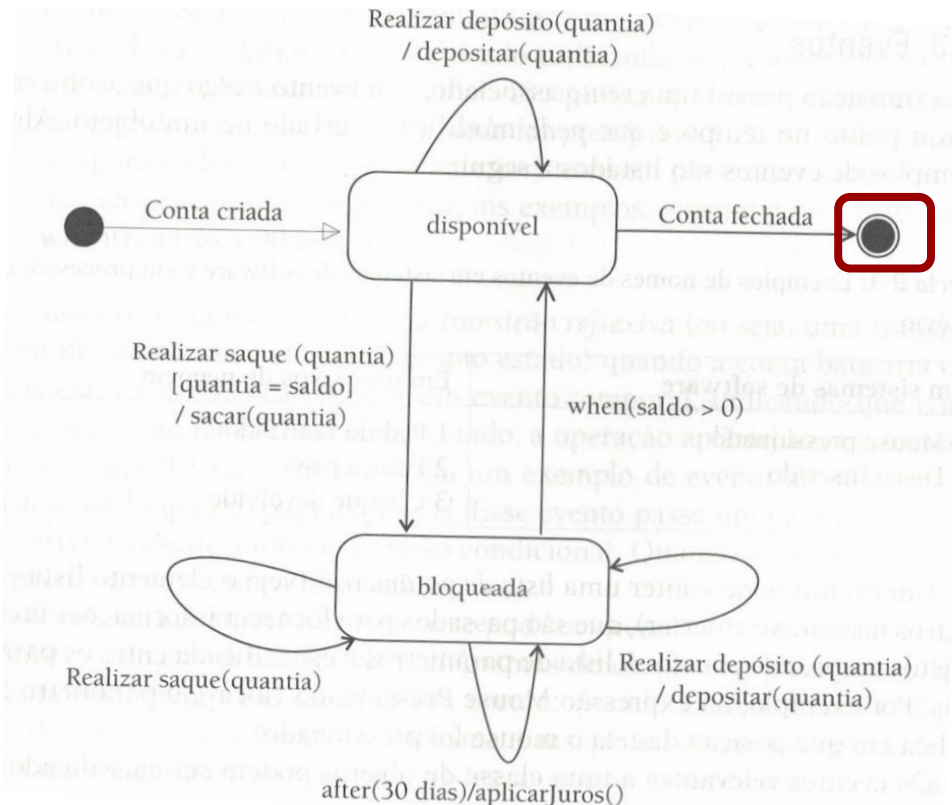
Diagrama de Transição de Estados



O estado inicial é apresentado com um círculo preenchido

Indica estado do objeto quando ele é criado

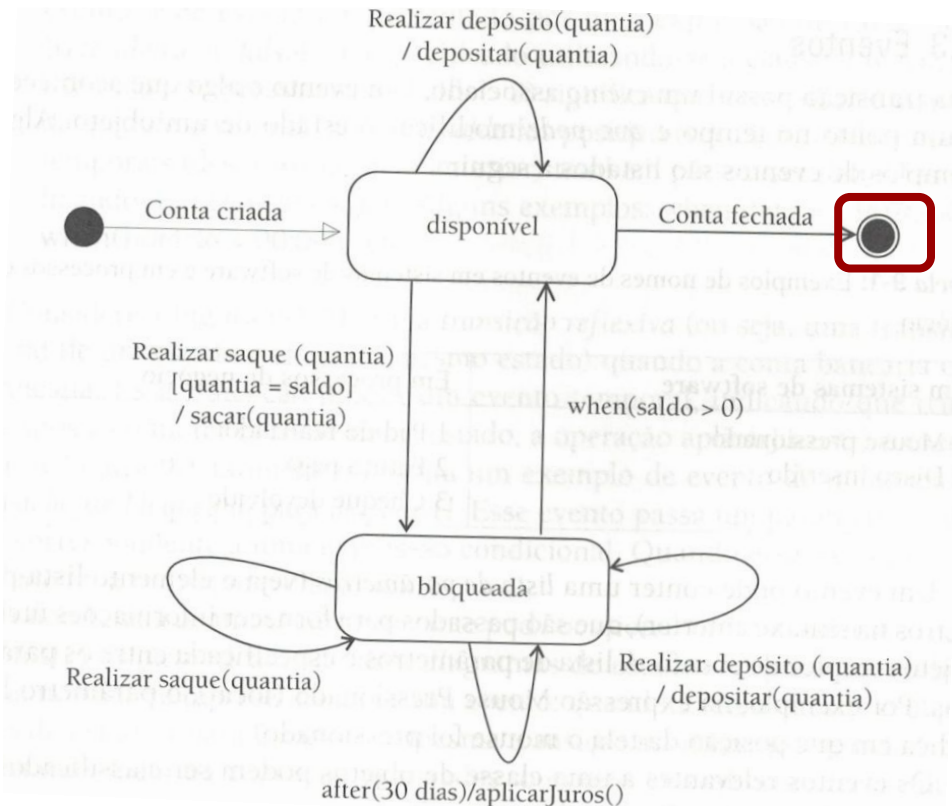
Diagrama de Transição de Estados



O estado final é representado por um círculo com borda

Indica o fim do ciclo de vida do objeto

Diagrama de Transição de Estados

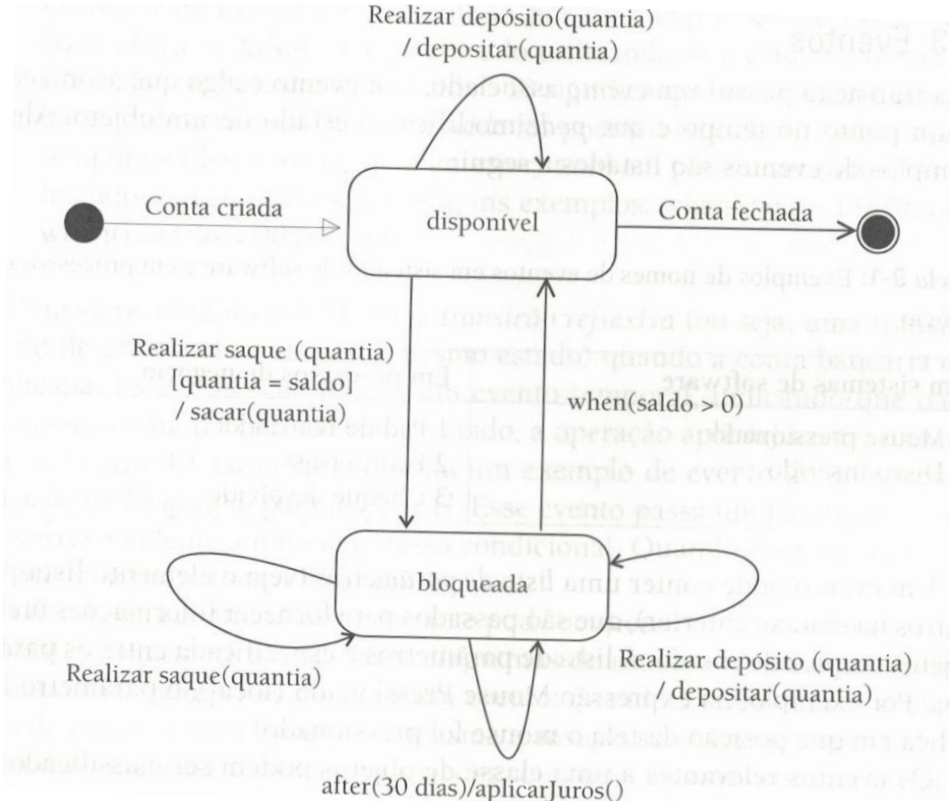


É um estado opcional

Podemos ter mais de um estado final no diagrama

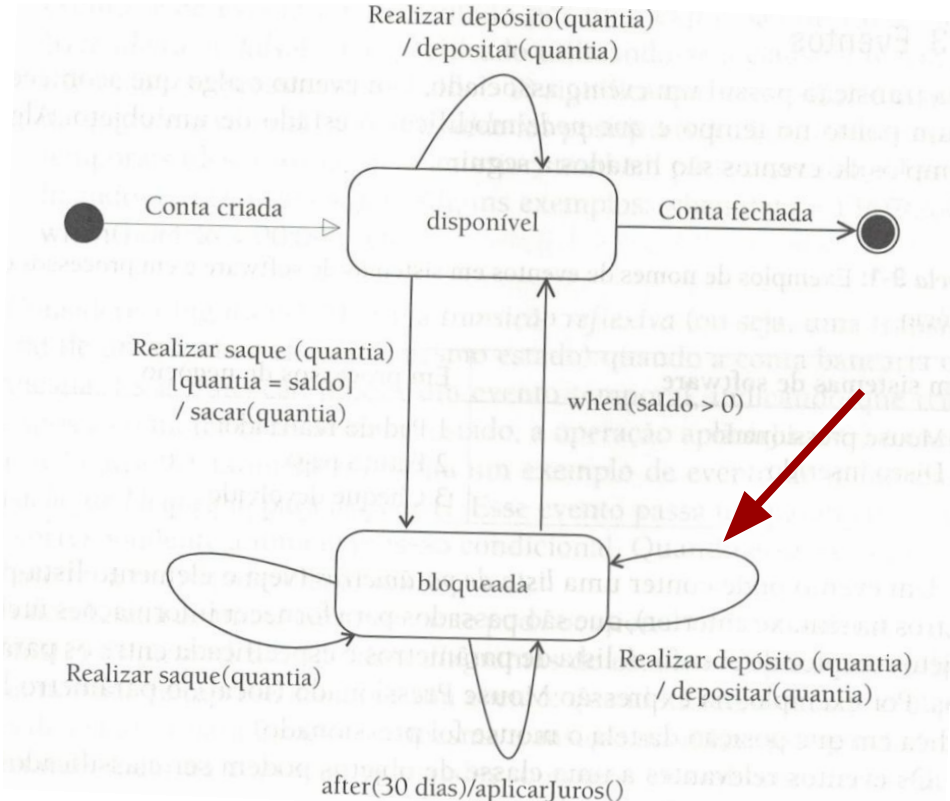
Transições

Transições



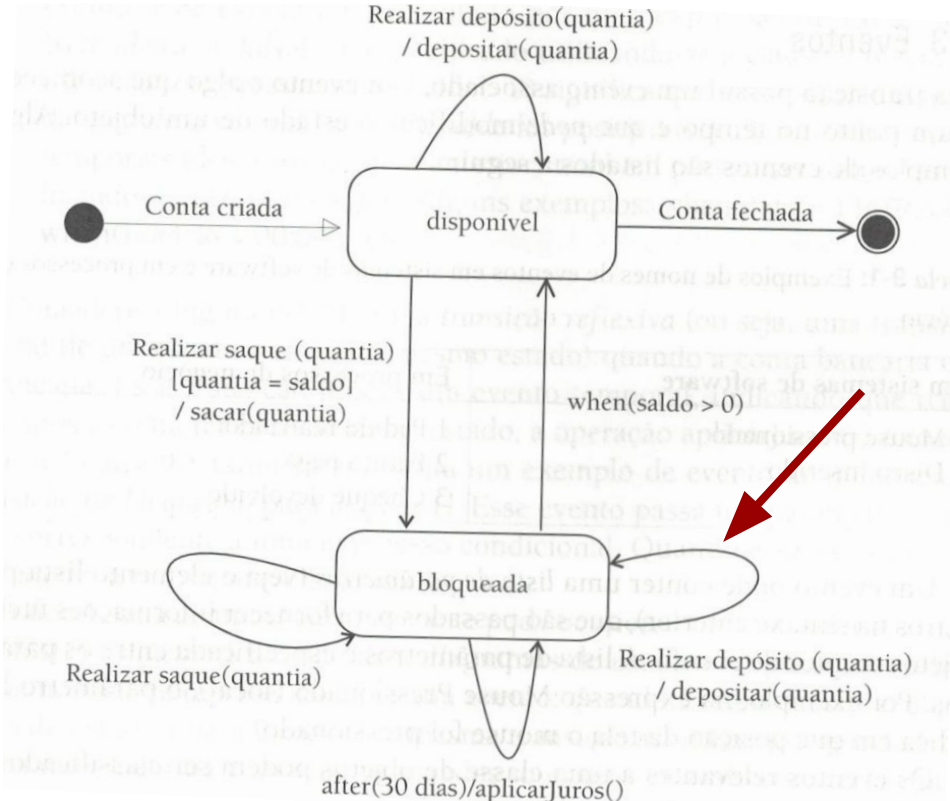
Os estados são associados por transições

Transições



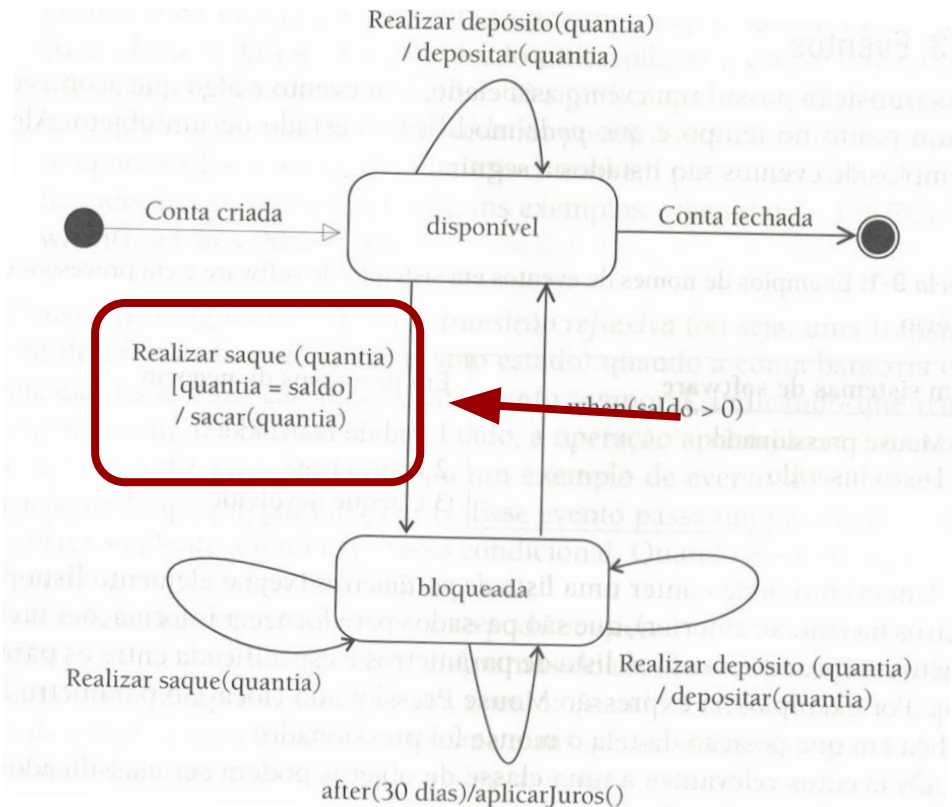
A transição é representada por uma seta apontando para um estado

Transições



Observe que o estado subsequente pode ser igual ao anterior

Transições



Podemos rotular uma transição

Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação

Exemplo: Realizar saque (quantia) [quantia = saldo] / sacar(quantia)

Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação



Algo que acontece, podendo modificar o estado do objeto. Exemplo:
Realizar saque

Rótulo das Transições

evento **(lista-parâmetros)** **[guarda]** / **ação**



Fornecem informações úteis para o objeto receptor do evento

Exemplo: Mouse pressionado (local). O parâmetro local indica em qual posição na tela o mouse foi pressionado

Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação



A condição de guarda é uma expressão lógica, podendo ser definida utilizando a lista de parâmetros ou atributos da classe. Podemos testar também o valor de um estado. Exemplo: `quantia <= saldo`

Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação



A transição ocorre se o evento é disparado e a condição de guarda é verdadeira.

Rótulo das Transições

Realizar saque (quantia) [quantia = saldo] / sacar(quantia)



- quantia é um parâmetro recebido
- saldo é um atributo da classe
- o evento Realizar saque ocorre se a condição de guarda for verdadeira

Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação



Um objeto pode realizar ações ao transitar entre os estados. Exemplo:
sacar(quantia)

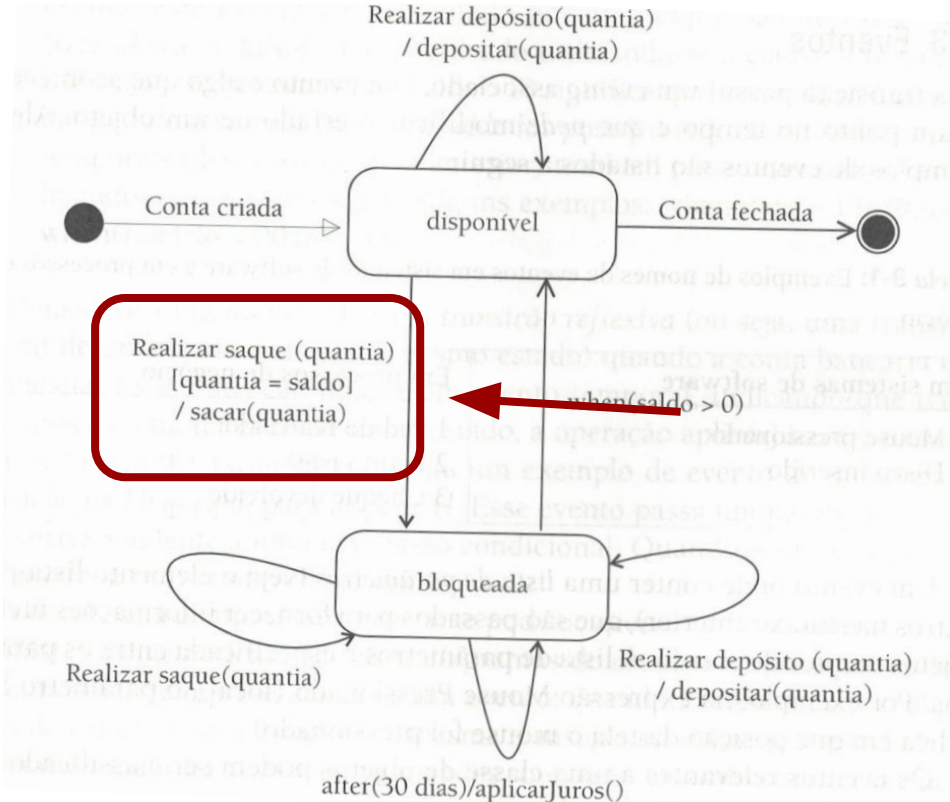
Rótulo das Transições

evento (lista-parâmetros) [guarda] / ação



A ação é precedida por uma barra

Transições



Realizar saque (quantia)
[quantia = saldo] / sacar(quantia)

Rótulo das Transições

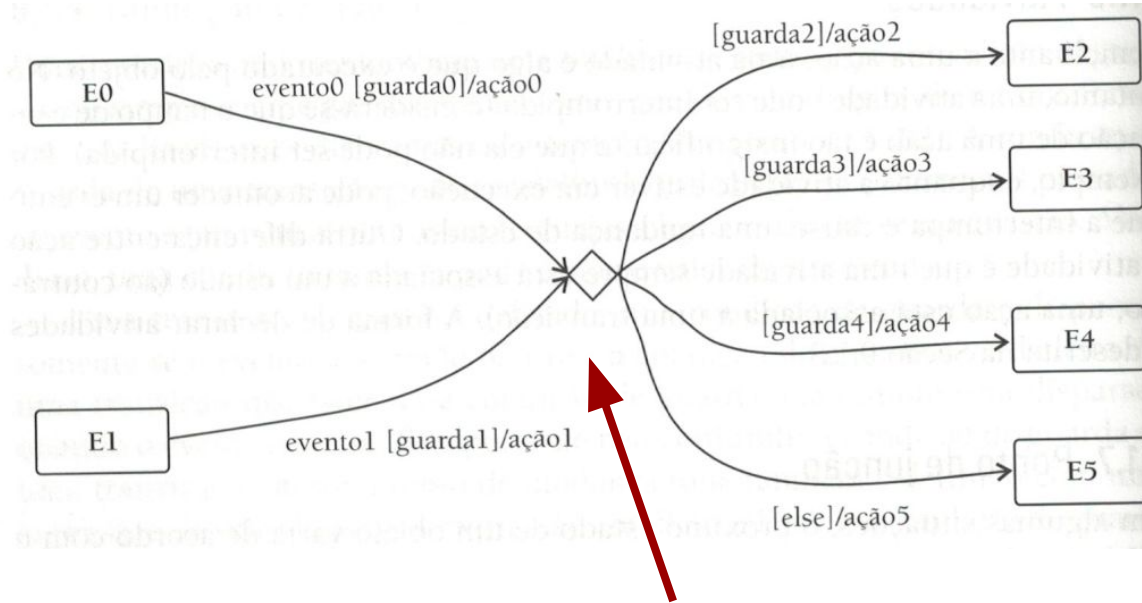
Realizar saque (quantia) [quantia = saldo] / sacar(quantia)



- quantia é um parâmetro recebido
- saldo é um atributo da classe
- o evento Realizar saque ocorre se a condição de guarda for verdadeira

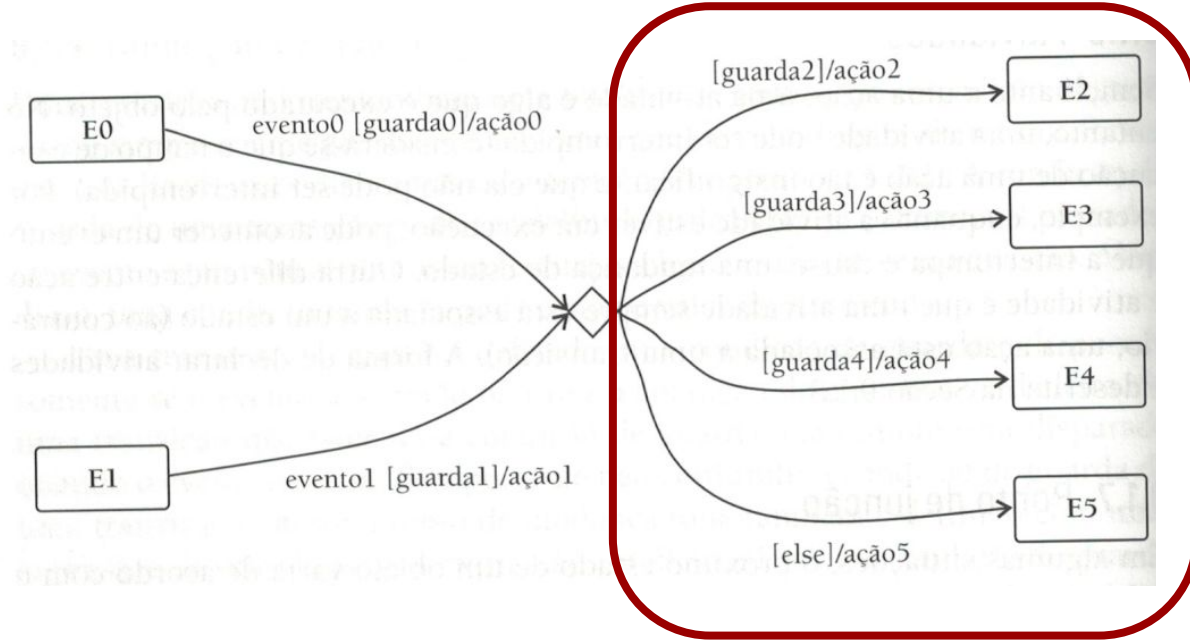
Ponto de Junção

Ponto de Junção

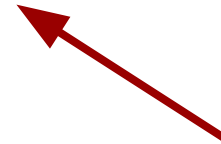


Podemos utilizar ponto de junção quando um objeto varia de acordo com o valor da condição de guarda

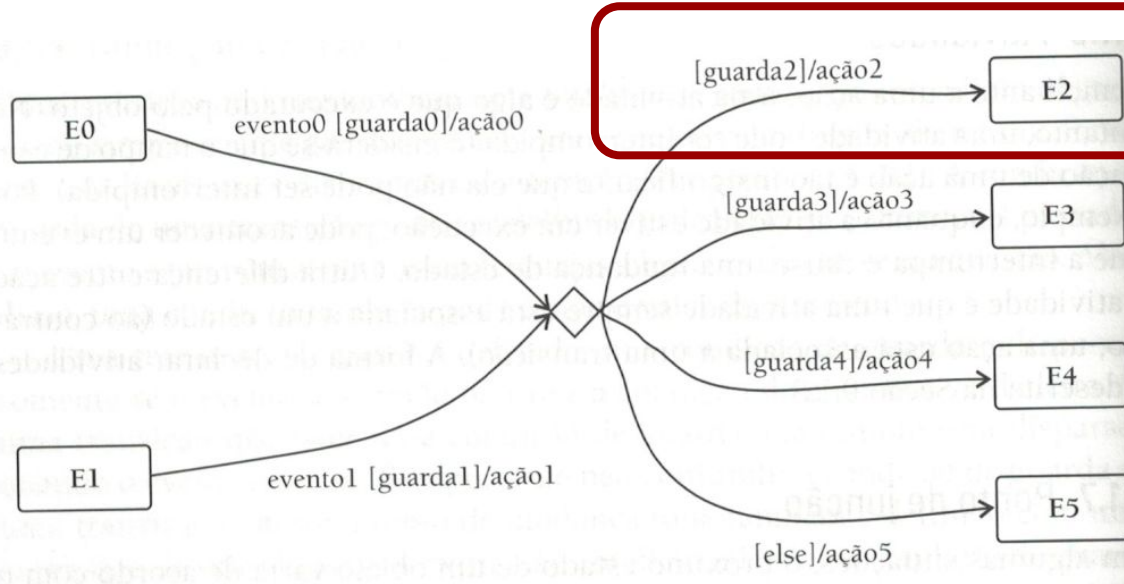
Ponto de Junção



Quatro transições associadas a uma condição de guarda

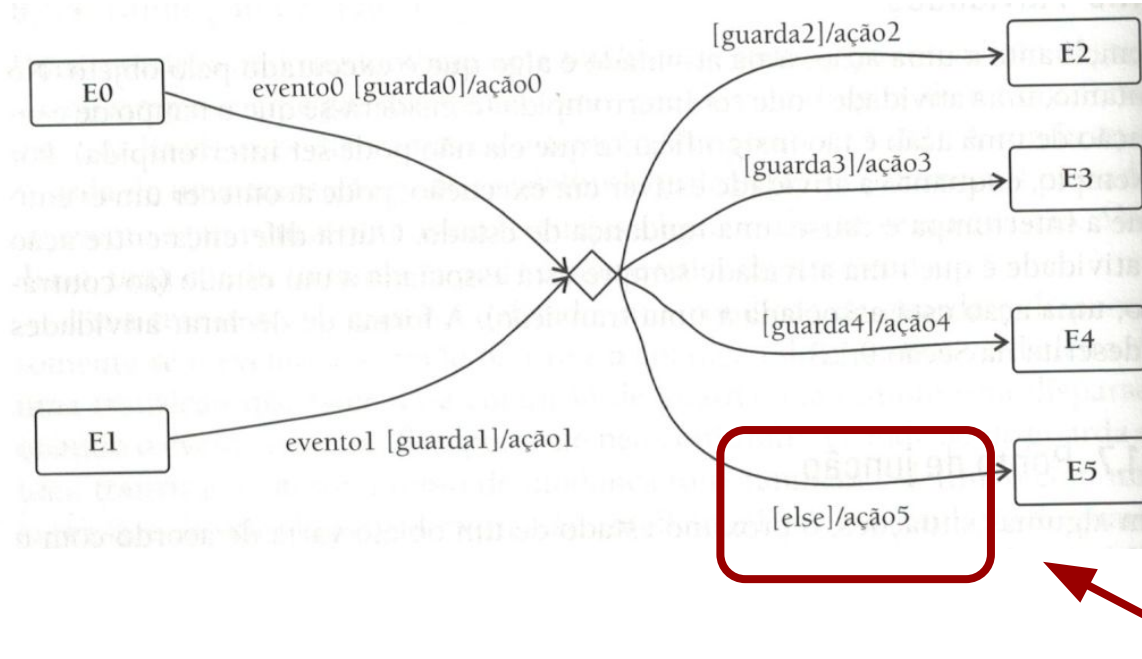


Ponto de Junção



Quando uma condição é verdadeira, as demais são falsas

Ponto de Junção

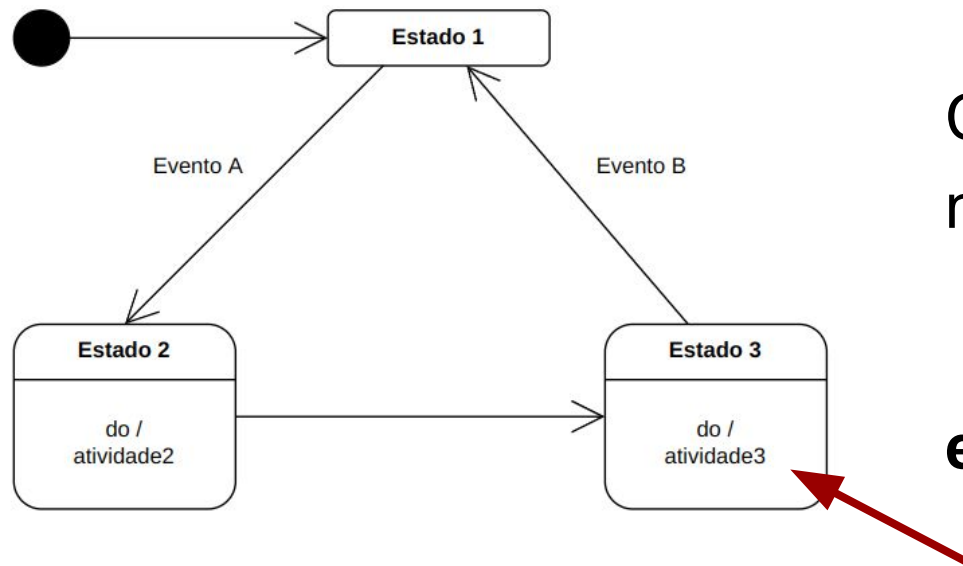


Transição rotulada com “else”

Disparada se todas as outras condições de guarda são falsas

Cláusulas entry, exit e do

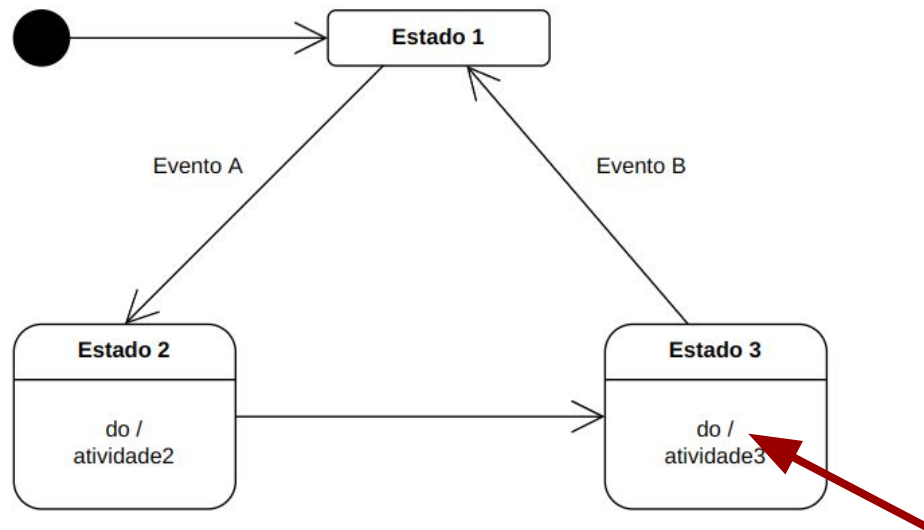
Cláusulas entry, exit e do



Cláusulas especificadas
no interior do retângulo

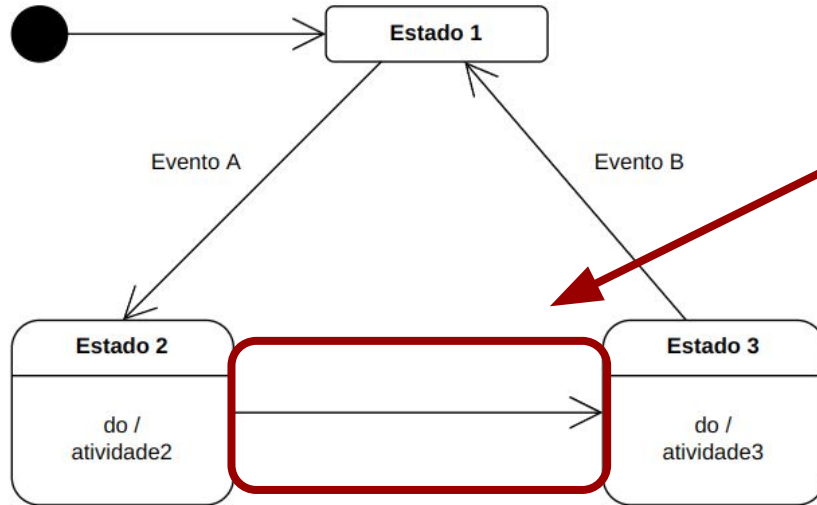
evento/ [ação | atividade]

Cláusulas “do”



Cláusula “do” define uma atividade a ser executada quando o objeto passa pelo estado

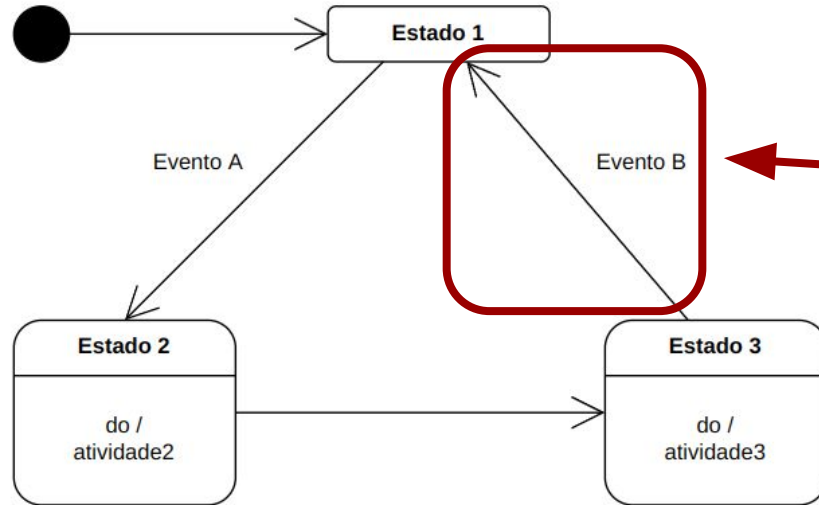
Cláusulas “do”



Sem evento

Assim que a “*atividade2*” for finalizada, a transição ocorre

Cláusulas “do”



Se evento B ocorrer, a transição do Estado 3 para Estado 1 pode ocorrer antes do término da atividade

Cláusula “entry”

Pode ser usada para especificar uma ação a ser realizada no momento em que o objeto entra em um estado

A ação desta cláusula é sempre executada, independentemente do estado do qual o objeto veio (é como se a ação especificada estivesse associada a todas as transições de entrada no estado)

Cláusula “exit”

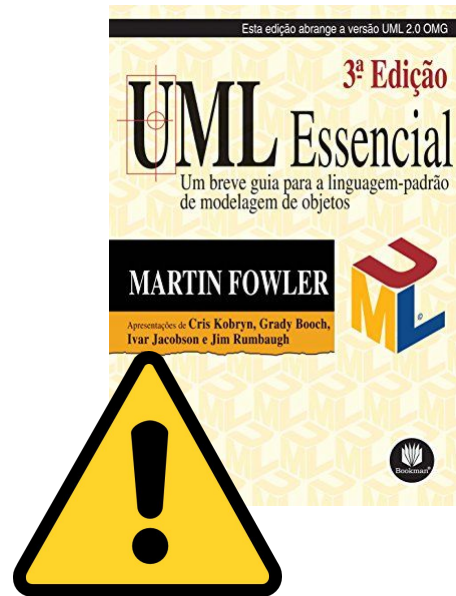
Serve para declarar ações que são executadas sempre que o objeto sai de um estado

Da mesma forma que a cláusula entry, a ação da cláusula exit é sempre executada, independentemente do estado para o qual o objeto vai (é como se a ação especificada estivesse associada a todas as transições de saída do estado).

Ações x Atividades

“A UML 1 usava o termo ação para atividades normais e usava atividades somente para a modalidade realizar-atividades (*do-activity*)

A UML 1 separava as ações de vida curta das atividades de vida longa. A UML 2 chama as duas de atividades e usa o termo realizar-atividade para as atividades de vida longa”



Ações x Atividades

Ao transitar de um estado para outro, um objeto pode realizar uma ou mais ações

Uma ação é uma expressão que pode ser definida em termos dos atributos, das operações ou das associações da classe. Os parâmetros do evento também podem ser utilizados. Uma ação pode também corresponder à execução de uma operação

A ação é representada na linha da transição e deve ser precedida por uma barra inclinada para a direita (símbolo “/”)

Ações x Atividades

Semelhante a uma ação, uma atividade é algo que é executado pelo objeto

No entanto, uma atividade pode ser interrompida (considera-se que o tempo de execução de uma ação é tão insignificante que esta não pode ser interrompida).

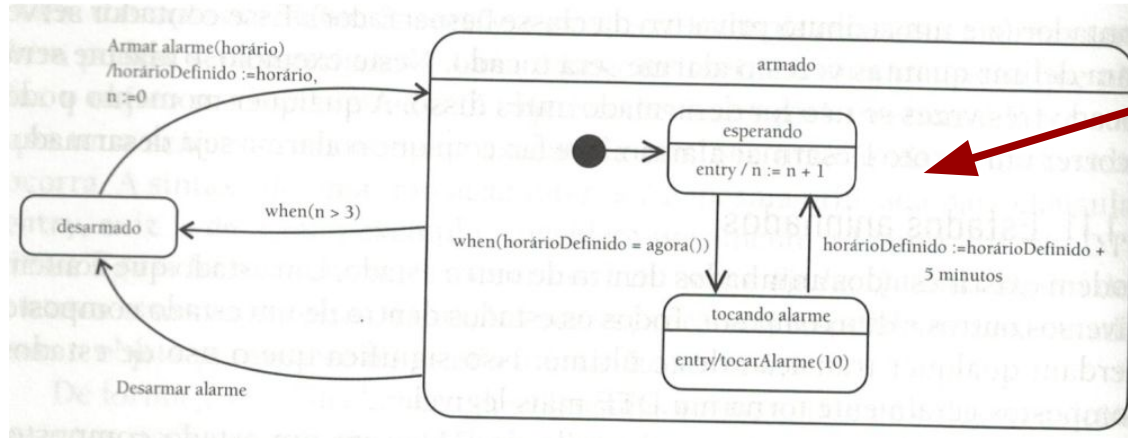
Ações x Atividades

Por exemplo, enquanto a atividade estiver em execução, pode acontecer um evento que a interrompa e cause uma mudança de estado

Uma outra diferença entre ação e atividade é que uma atividade sempre está associada a um estado (ao contrário, uma ação está associada a uma transição)

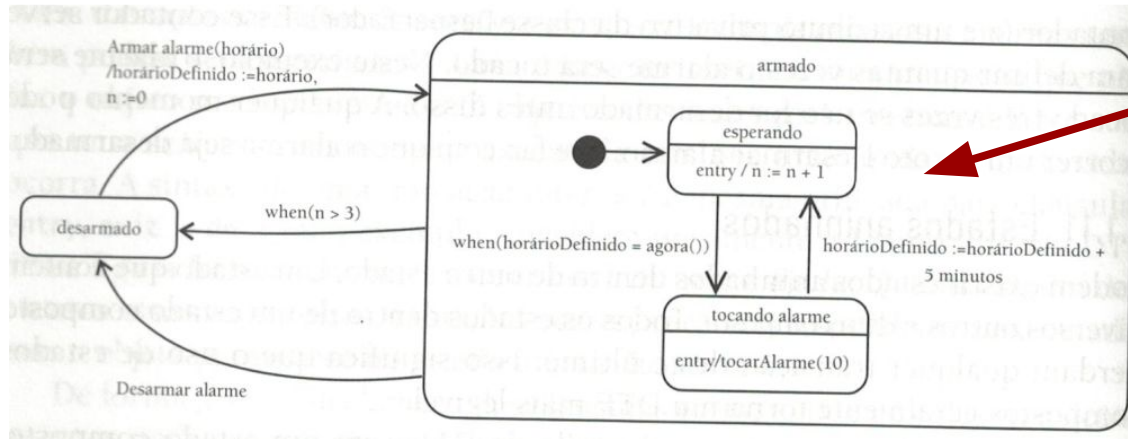
Estados Concorrentes e Aninhados

Estados Aninhados



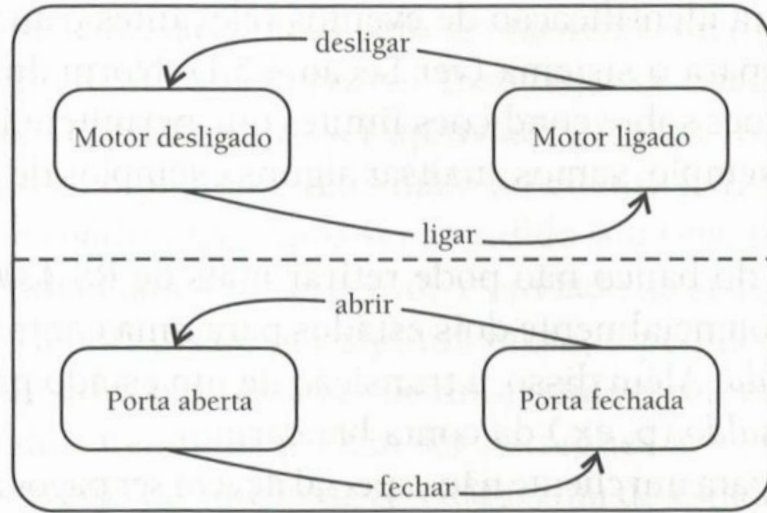
Podem existir estados aninhados dentro de outro estado

Estados Aninhados



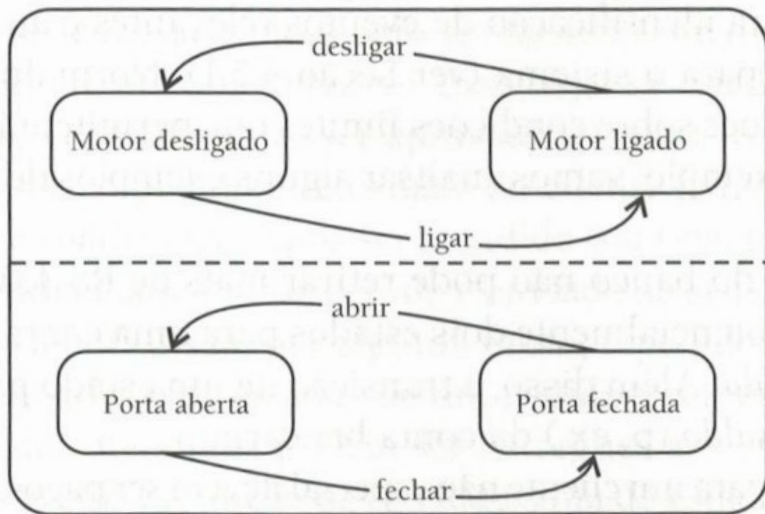
O estado possui dois estados aninhados: "esperando e tocando alarme"

Estados Concorrentes



O objeto pode se encontrar em dois ou mais estados independentes

Estados Concorrentes



Exemplo: estado composto para a classe Refrigerador

O objeto pode estar com a porta aberta ou fechada

Independente disso, o objeto pode estar com o **motor ligado ou desligado**

Diagrama de Transição de Estados

Os diagramas de transição de estados não são definidos para todas as classes do sistema, mas apenas para aquelas que possuem um número finito de estados conhecidos e quando o comportamento das classes de objetos é afetado e modificado por diferentes estados



Exercício

Exercício

Construir um diagrama de transição de estados para a classe Turma de um Sistema de Controle Acadêmico

Turma
<ul style="list-style-type: none">-ano : Integer-semestre : Integer-qtdAlunos : Integer-capacidadeMáxima : Integer-status
<ul style="list-style-type: none">+cancelar()+fechar()+abrir()-lotar()

Classe Turma

Quais são os estados para a classe?

Classe Turma

Quais são os estados para a classe?

- Aberta
- Lotada
- Fechada
- Cancelada

Classe Turma

Quais são os eventos relevantes
para a classe?

Classe Turma

Quais são os eventos relevantes para a classe?

- Inscrição
- Abertura, cancelamento, e fechamento da turma

Classe Turma

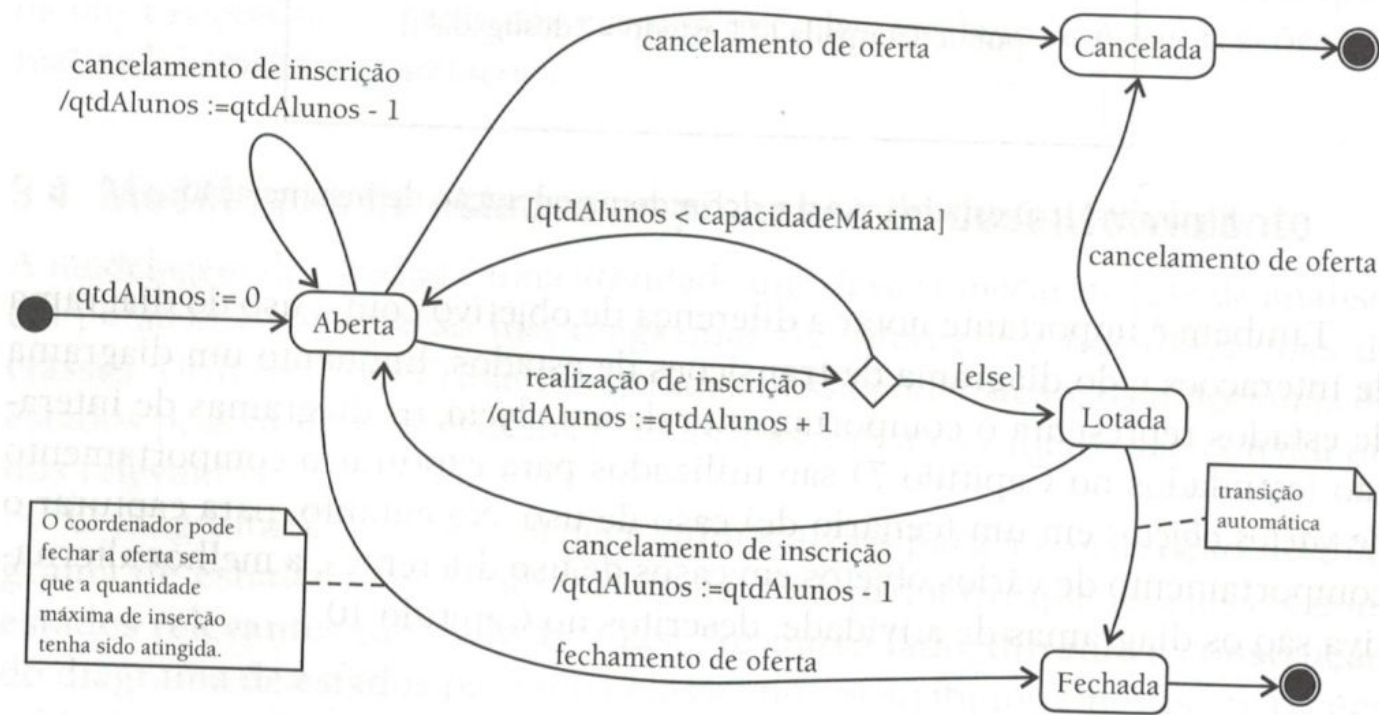
Precisamos monitorar algum atributo da turma?

Classe Turma

Precisamos monitorar algum atributo da turma?

Sim, a quantidade de alunos inscritos

Exercício: Gabarito



Referências

Princípios de Análise e Projeto de Sistemas com UML. Eduardo Bezerra. 3ª Edição. Cap 9 - Modelagem de Estados

UML Essencial. Martin Fowler. 2ª Edição. Cap 8 - Diagramas de Estados

UML Essencial. Martin Fowler. 3ª Edição:

- Cap 10 - Diagramas de Máquina de Estados.
- Apêndice: Modificações nas versões da UML

Diagrama de Estados

Projeto de Software

Profª Aline Brito