

# Diagrama de Transição de Estados

João Pedro Oliveira Batisteli

# Estado de Objetos

- A jarra está *cheia*
- O livro está *reservado*
- A conta está *encerrada*
- A pessoa está *cansada*
- A prova está *corrigida*
- O curso está *concluído*
- O aluno está *aprovado*

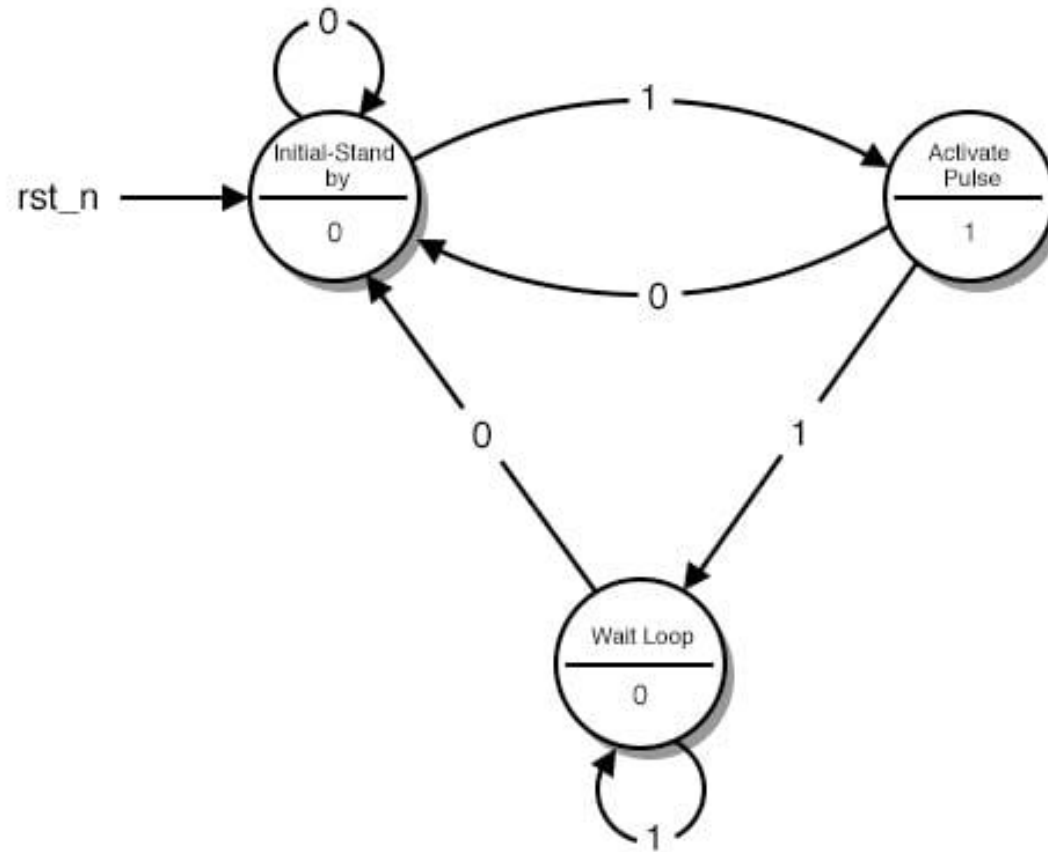
# Transição de Estados

- Um objeto muda de estado quando acontece algum **evento** interno ou externo ao sistema.
- Uma mudança de um estado para outro é chamada de **transição entre estados**
  - Quando um **objeto transita de um estado para outro**, o **sistema no qual ele está inserido também muda de estado**.
- Os estados e as transições de estado de um objeto **constituem o seu ciclo de vida**.
  - Cada objeto pode passar por um **número finito de estados** durante a sua vida.

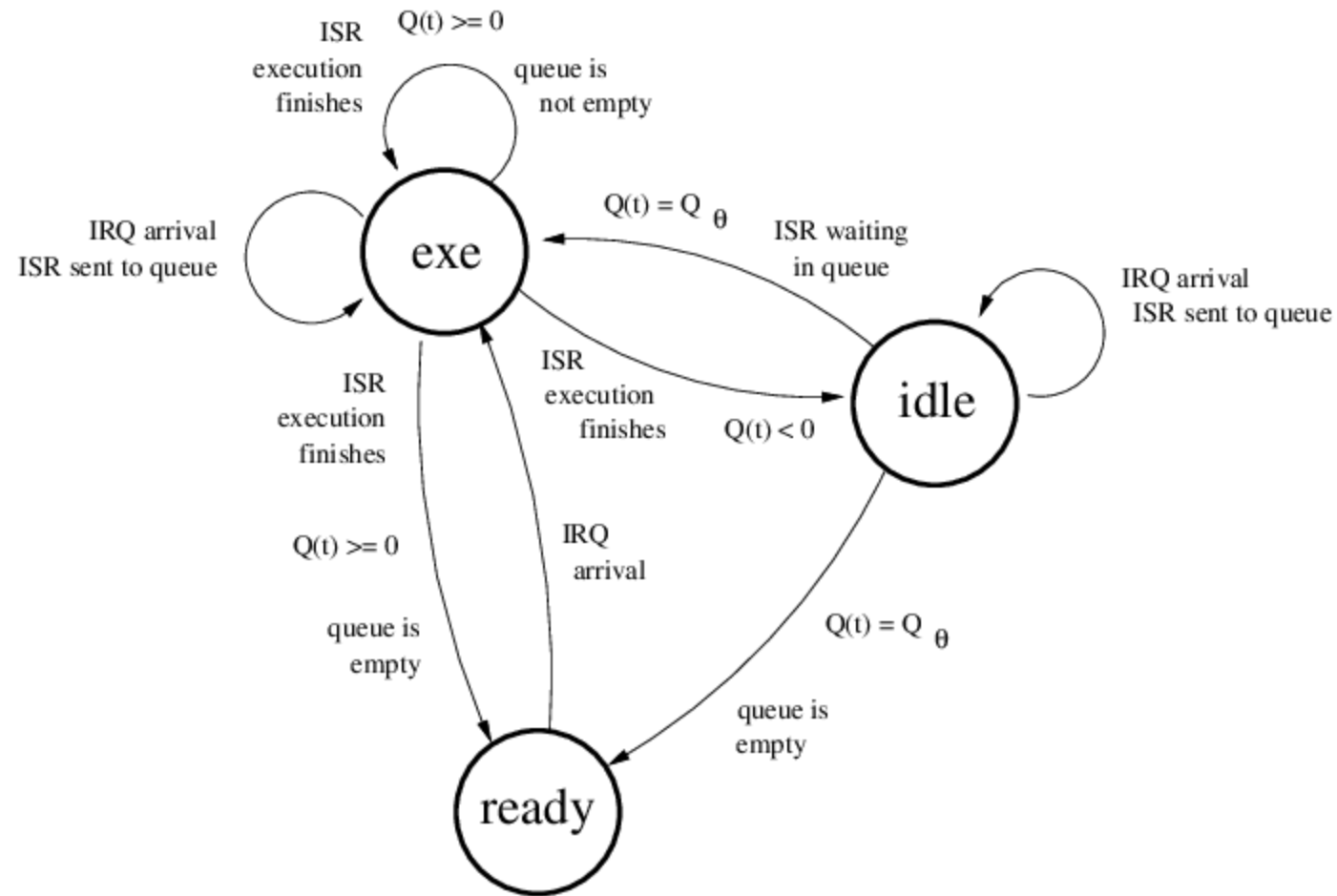
# Diagrama de Transição de Estados (DTE)

- Descreve
  - o ciclo de vida de objetos de uma classe
  - os eventos que causam a transição de um estado para outro
  - a realização de operações resultantes da transição
- O diagrama é baseado na ideia de máquina de estados finitos (*finite state machines*)

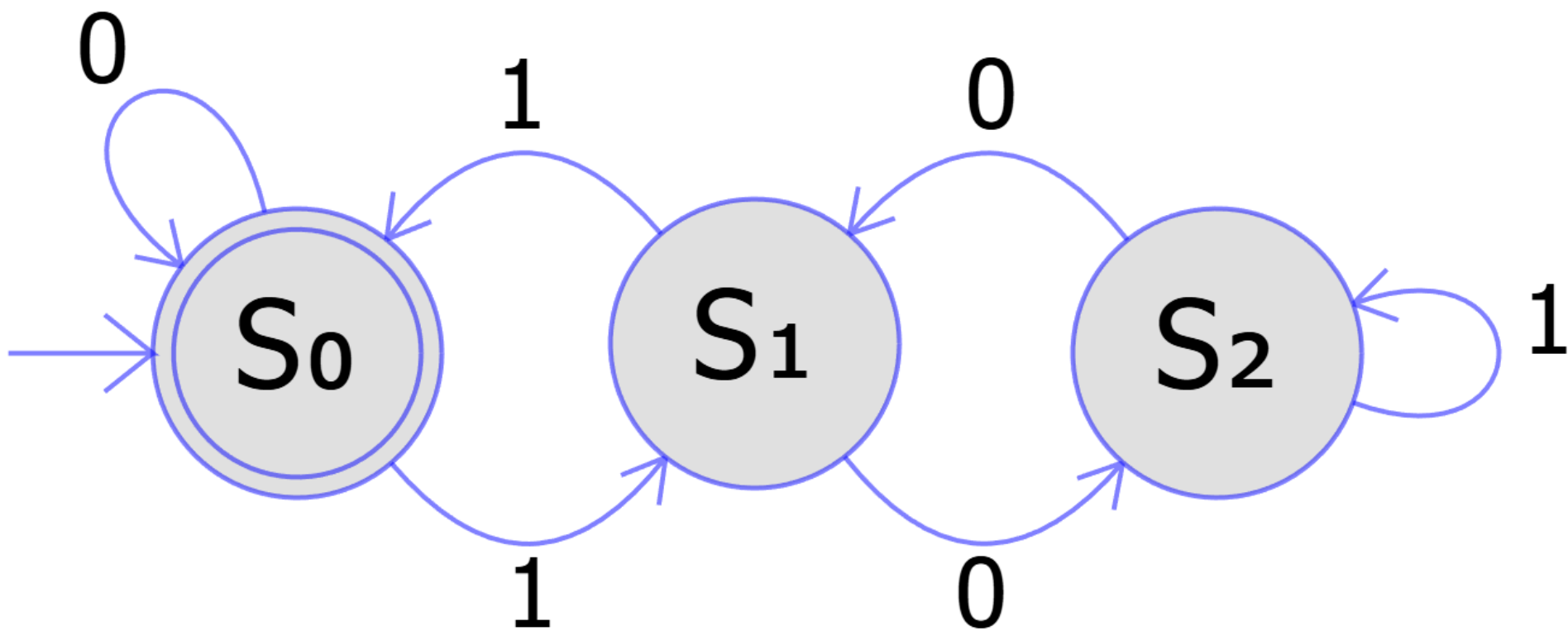
# Máquina de estados finitos



# Máquina de estados finitos



# Autômatos Finitos



# DTE – Elementos Básicos

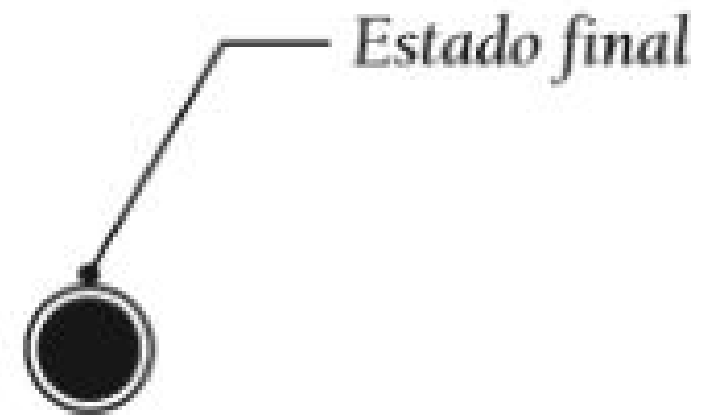
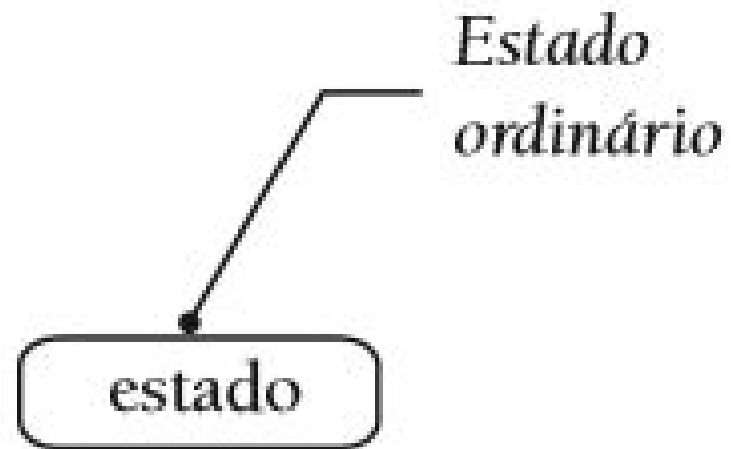
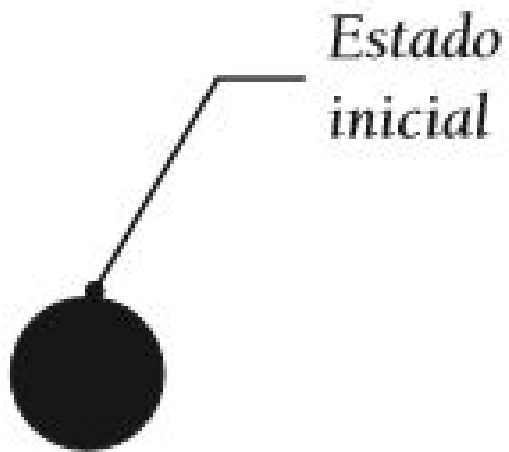
- Conceitos Básicos:
  - **Estados**
  - **Transições**
- Conceitos Associados
  - **Eventos**
  - **Ação**
  - **Atividade**



# Estados

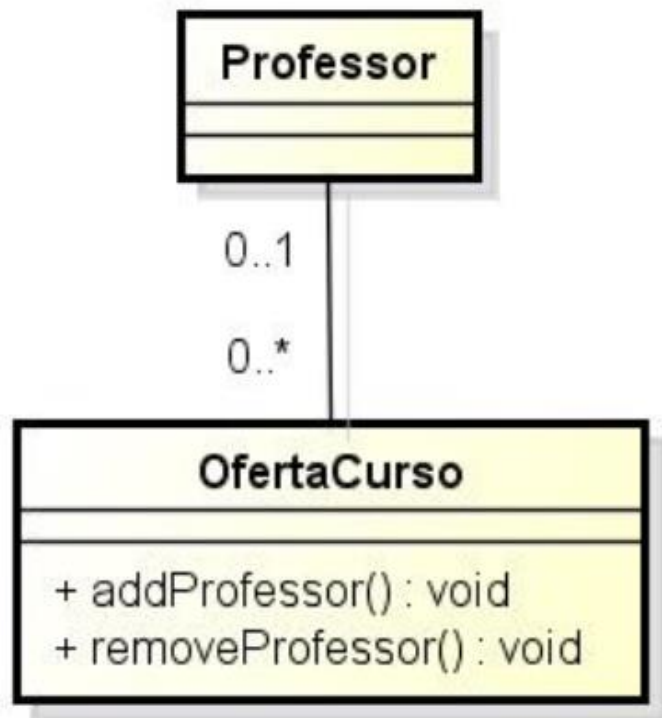
- **Um estado é uma situação na vida de um objeto**
  - Em um estado, o objeto satisfaz alguma condição ou realiza alguma atividade
- Estados geralmente consistem em **atributos que assumem determinado valor.**
  - “o atributo *reservado* deste objeto livro tem valor *verdadeiro*“
  - “Uma conta bancária passa para o *vermelho* quando o seu saldo fica *negativo*“
- Pode-se definir um estado **inicial** e **final** para um objeto

# Representação de Estados

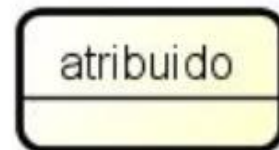


# Estados

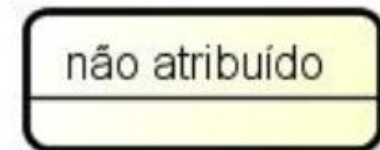
- Os estados podem ser distinguidos pelas associações



Associação com o professor existe

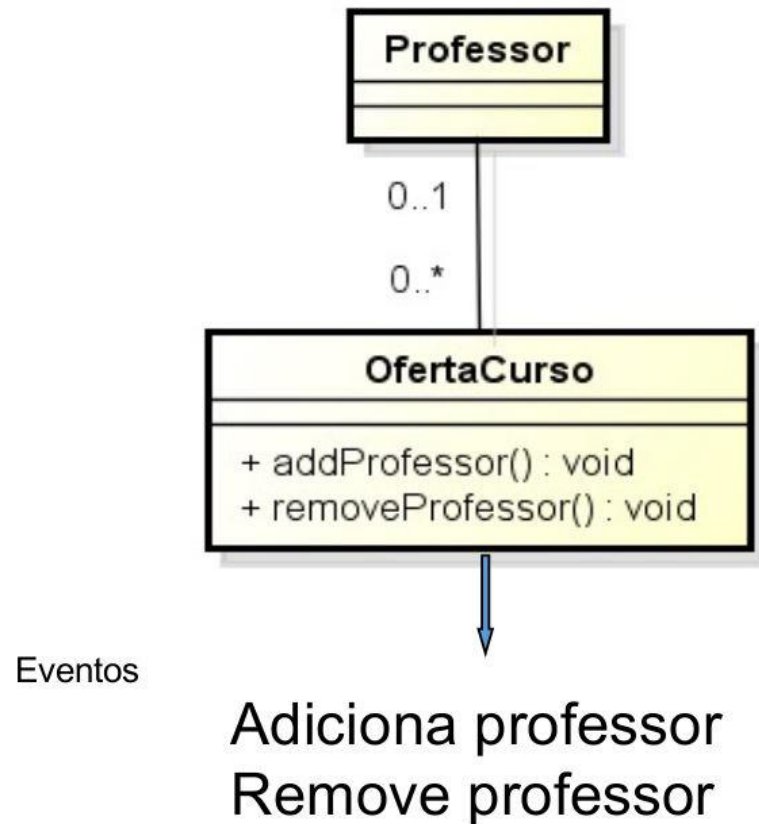


Associação com o professor não existe



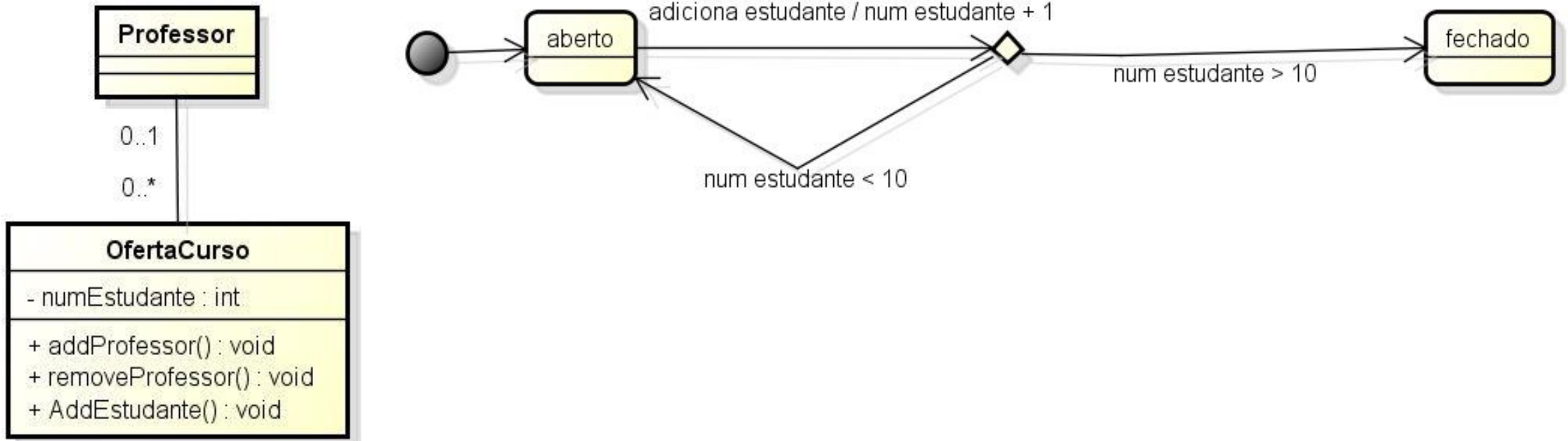
# Estados

- Os estados podem ser distinguidos pelas associações



# Estados

- Eventos podem ser mapeados para operações
- Estados são muitas vezes representados usando atributos



# Transições

- Os estados estão associados a outros pelas transições
  - Pode existir uma transição de um estado para ele mesmo
  - Quando uma transição ocorre, diz-se que ela foi *disparada*



- Uma transição pode ser rotulada com uma expressão, cuja forma geral é:

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

# Evento

- Um evento é **algo** que acontece em algum ponto no tempo e que **pode modificar o estado de um objeto**

Exemplos de eventos:

| Em sistemas de software                 | Em processos de negócio                                   |
|---|---|
| 1 Mouse pressionado<br>2 Disco inserido | 1 Pedido realizado<br>2 Fatura paga<br>3 Cheque devolvido |

- Um evento pode conter uma **lista de parâmetros** para fornecer **informações úteis** ao objeto **receptor do evento**.

# Tipos Comuns de Eventos

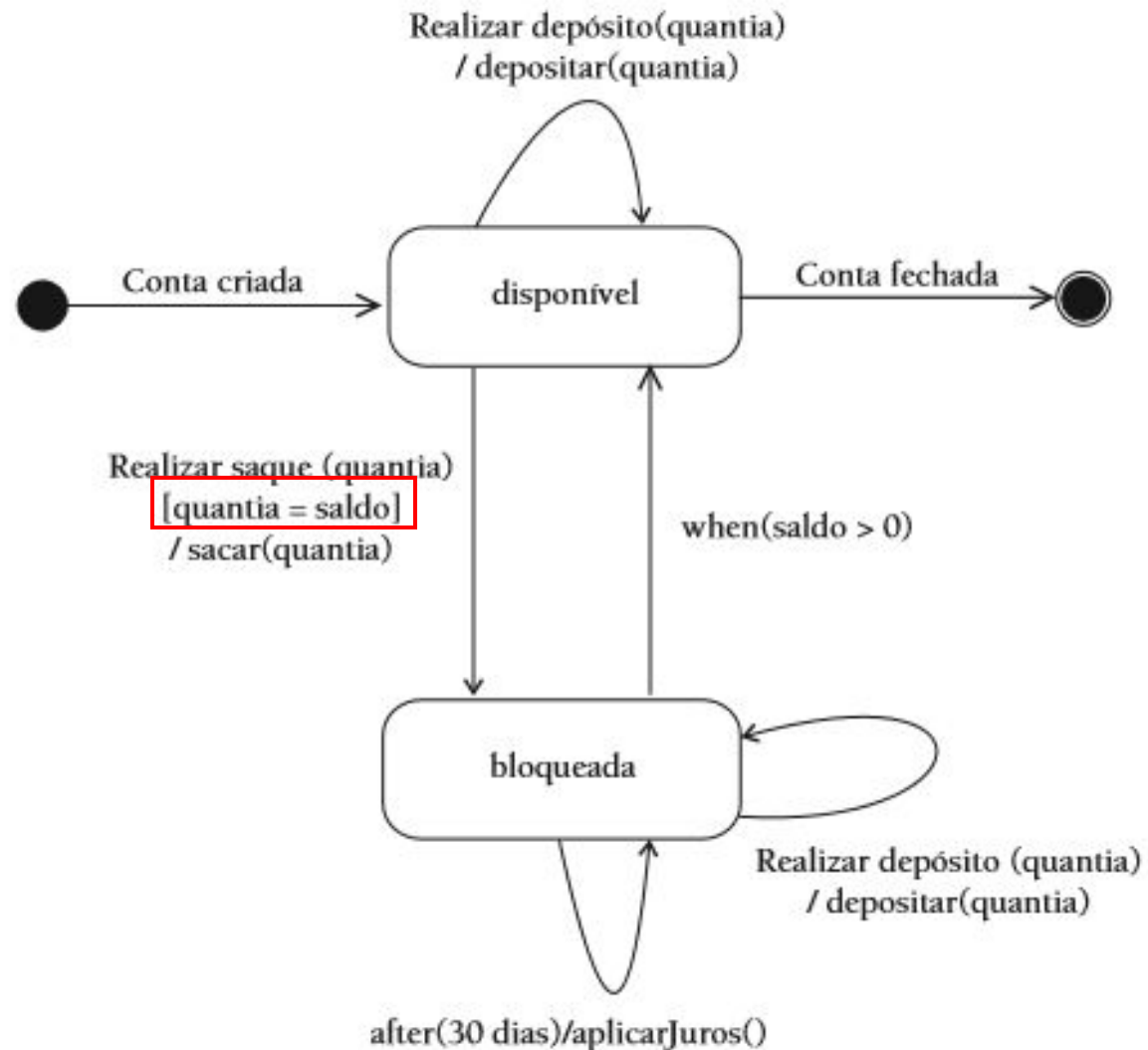
- Evento de chamada
  - Recebimento de mensagem de outro objeto
  - Exemplo: *período letivo finalizado*
- Evento temporal
  - Passagem de um intervalo de tempo predefinido
  - Exemplo: *After(30 segundos)*
- Evento de mudança
  - Uma condição se torna verdadeira
  - Exemplo: *when(saldo>0)*



# Condição de Guarda

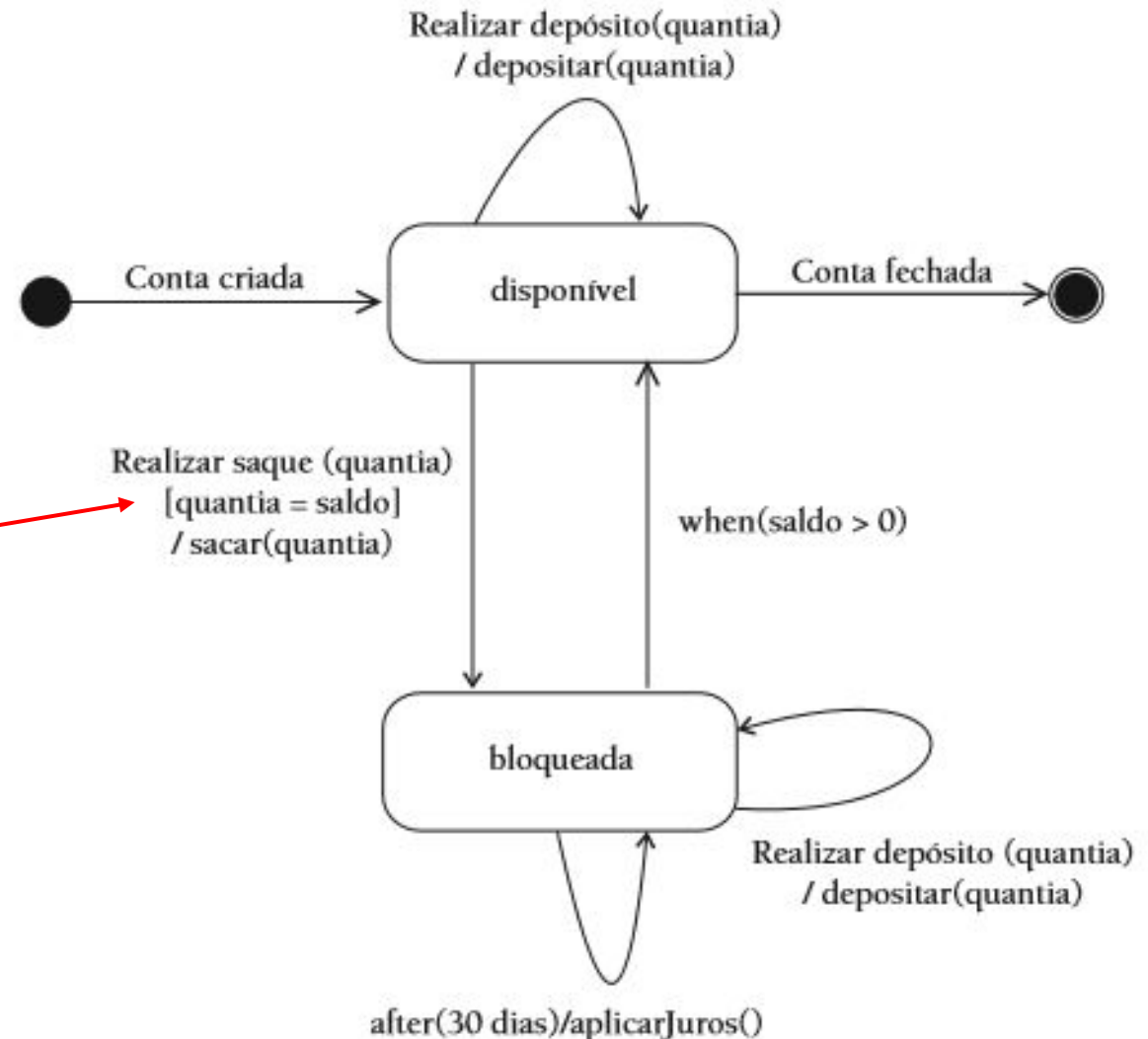
- Uma condição de guarda (ou apenas *guarda*) é uma expressão de valor lógico.
- Uma transição na qual foi definida uma condição de guarda é disparada somente se os dois critérios abaixo forem satisfeitos
  - O evento associado ocorre
  - A condição de guarda é verdadeira

# Exemplo de Condição de Guarda



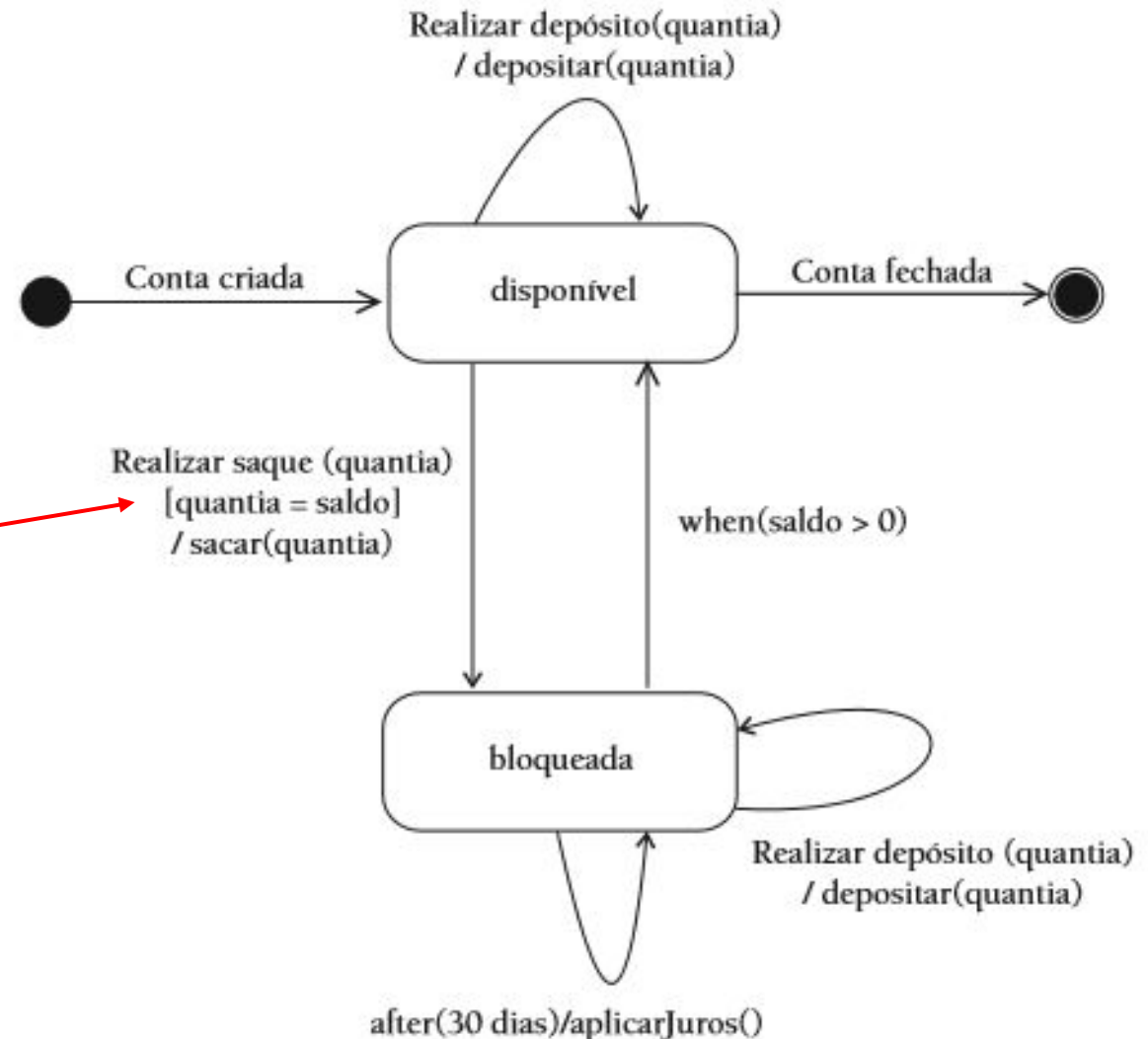
# Exemplo de Condição de Guarda

- **O Evento (Realizar saque (quantia)):** O usuário tenta fazer um saque.
- **A Guarda/Condição ([quantia = saldo]):** Esta é a parte crucial. A transição só ocorre se o valor que o usuário deseja sacar for exatamente igual ao saldo atual da conta.
- **A Ação (/ sacar(quantia)):** Se a condição for verdadeira, o sistema executa o saque.



# Exemplo de Condição de Guarda

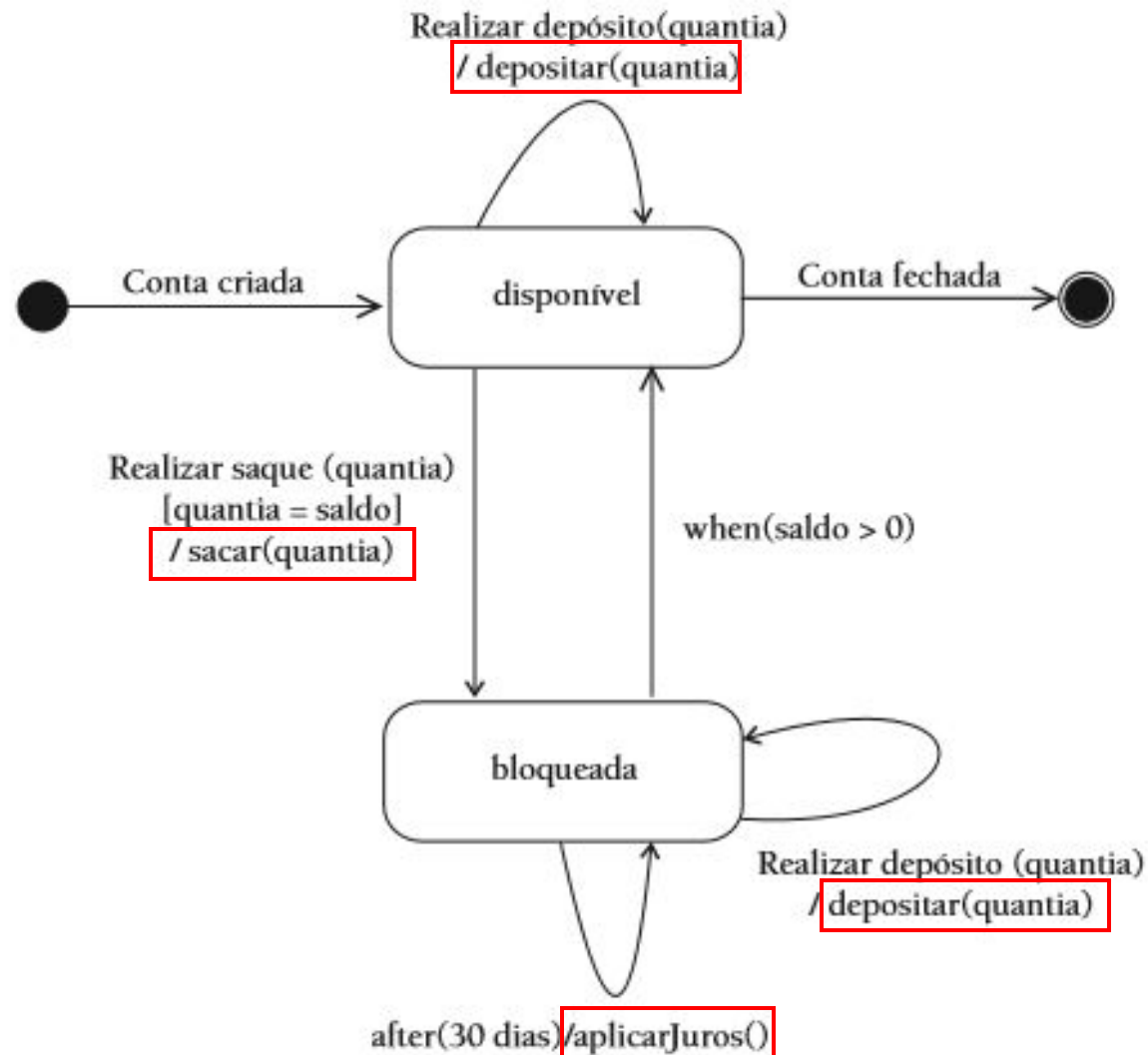
- **Saque Total:** Sacar o valor total move a conta para **bloqueada**.



# Ação

- Ao transitar de um estado para outro, um objeto pode realizar **uma ou mais ações**.
- Uma ação pode 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 “/”)
  - A ação é executada somente se a transição for disparada

# Exemplo de Ação



# Atividade

- Semelhante a uma ação, uma atividade é algo que é executado pelo objeto, mas
  - enquanto que ações não podem ser interrompidas, atividades podem
  - enquanto que uma ação está associada a uma transição, uma atividade está associada a um estado

# Atividade

- Semelhante a uma ação, uma atividade é algo que é executado pelo objeto, mas
  - enquanto que ações **não podem ser interrompidas**, **atividades podem**
  - enquanto que uma ação está associada a uma transição, uma **atividade está associada a um estado**



# Atividade x Ações

- **Ações:** São operações consideradas **instantâneas** ou **atômicas**. Na teoria, elas levam tempo zero para acontecer. Ou acontecem totalmente, ou não acontecem.
- **Por que não podem ser interrompidas?** Como elas são "instantâneas", não existe um "meio do caminho" onde você possa parar. Assim que o gatilho dispara, a ação executa até o fim imediatamente.
- **Onde vivem:** Normalmente nas **setas (transições)**.

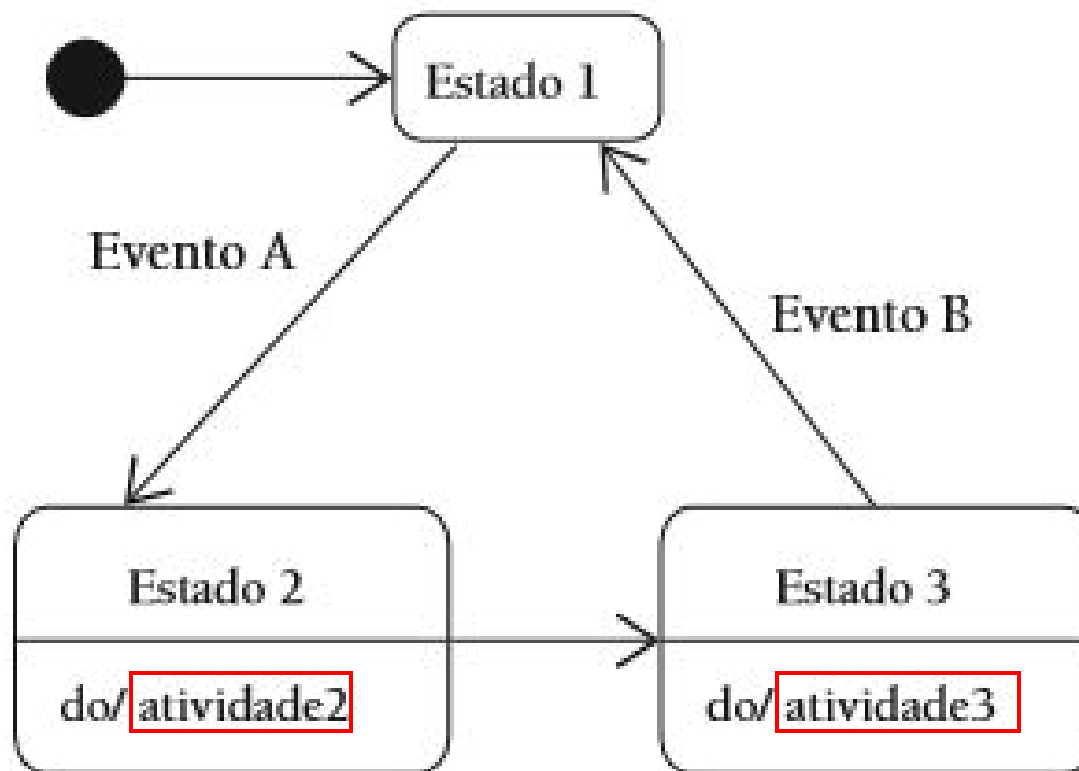
# Atividade x Ações

- **Atividades:** São operações que **levam tempo** para serem concluídas (segundos, horas, dias). É um processo contínuo que ocorre *enquanto* o objeto permanece naquele estado.
- **Por que podem ser interrompidas?** Se um evento acontecer enquanto a atividade está rodando (por exemplo, alguém aperta um botão "Cancelar" ou ocorre um erro), a máquina sai daquele estado imediatamente e a atividade é cortada no meio.
- **Onde vivem:** Dentro das **caixas (estados)**, geralmente marcadas com a etiqueta *do/* (do inglês *doing*, fazendo).

# Atividade x Ações

| <b>Característica</b> | <b>Ação (Action)</b>                 | <b>Atividade (Activity)</b>  |
|-----------------------|--------------------------------------|------------------------------|
| <b>Duração</b>        | Instantânea (tempo insignificante)   | Duradoura (leva tempo)       |
| <b>Interrupção</b>    | Não (Atômica)                        | Sim (pode parar no meio)     |
| <b>Onde fica?</b>     | Na seta (Transição) ou Entrada/Saída | Dentro do retângulo (Estado) |
| <b>Sintaxe UML</b>    | /nomeDaAção()                        | do / nomeDaAtividade         |

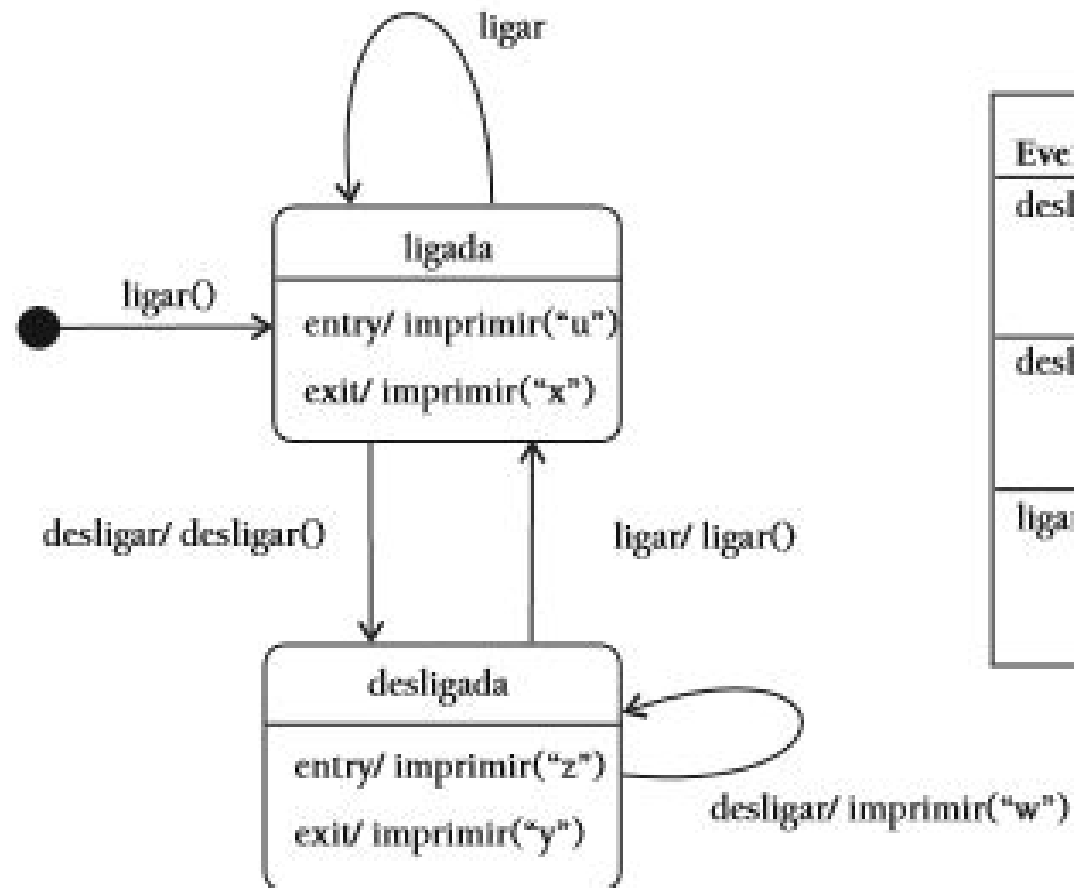
# Atividade



# Cláusulas entry, exit, do

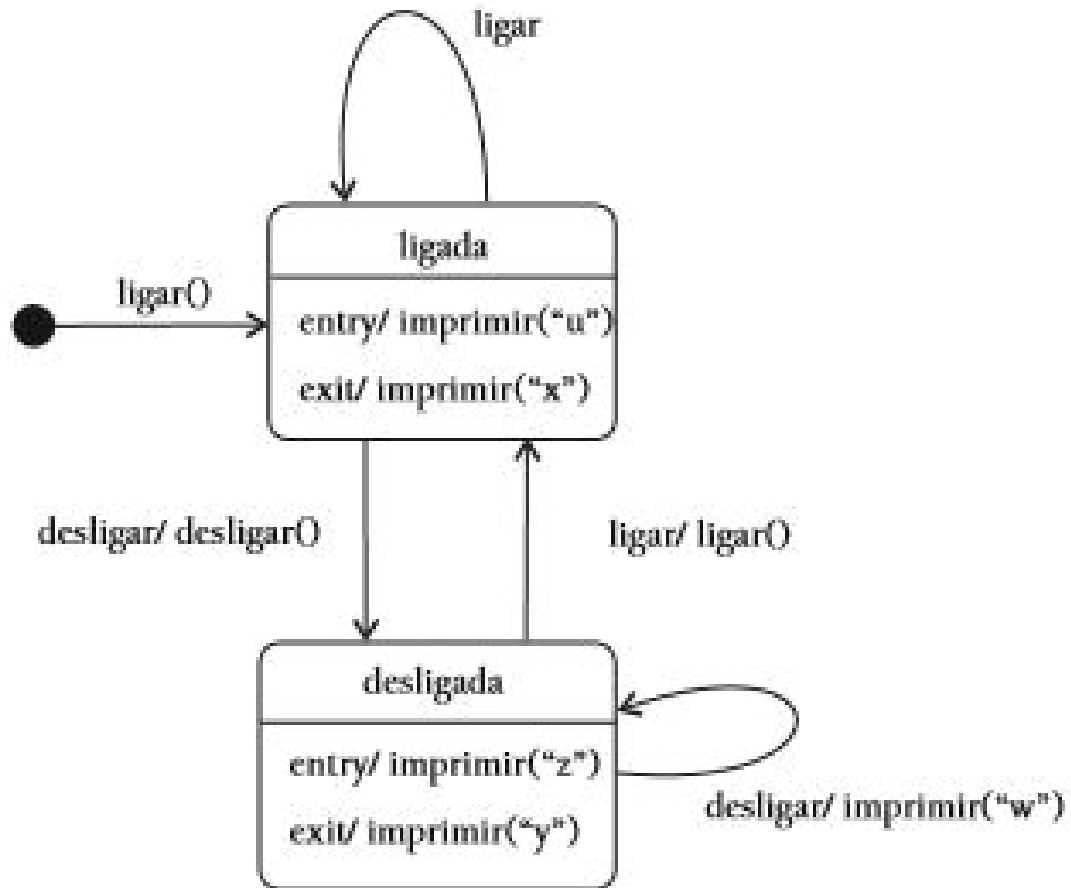
- Entry
  - usada para especificar uma **ação** a ser realizada no momento em que o objeto entra em um estado
- Exit
  - Serve para declarar ações que são executadas sempre que o objeto sai de um estado
- Do
  - Serve para definir alguma **atividade** a ser executada quando o objeto passa para um determinado estado

# Exemplos de entry e exit

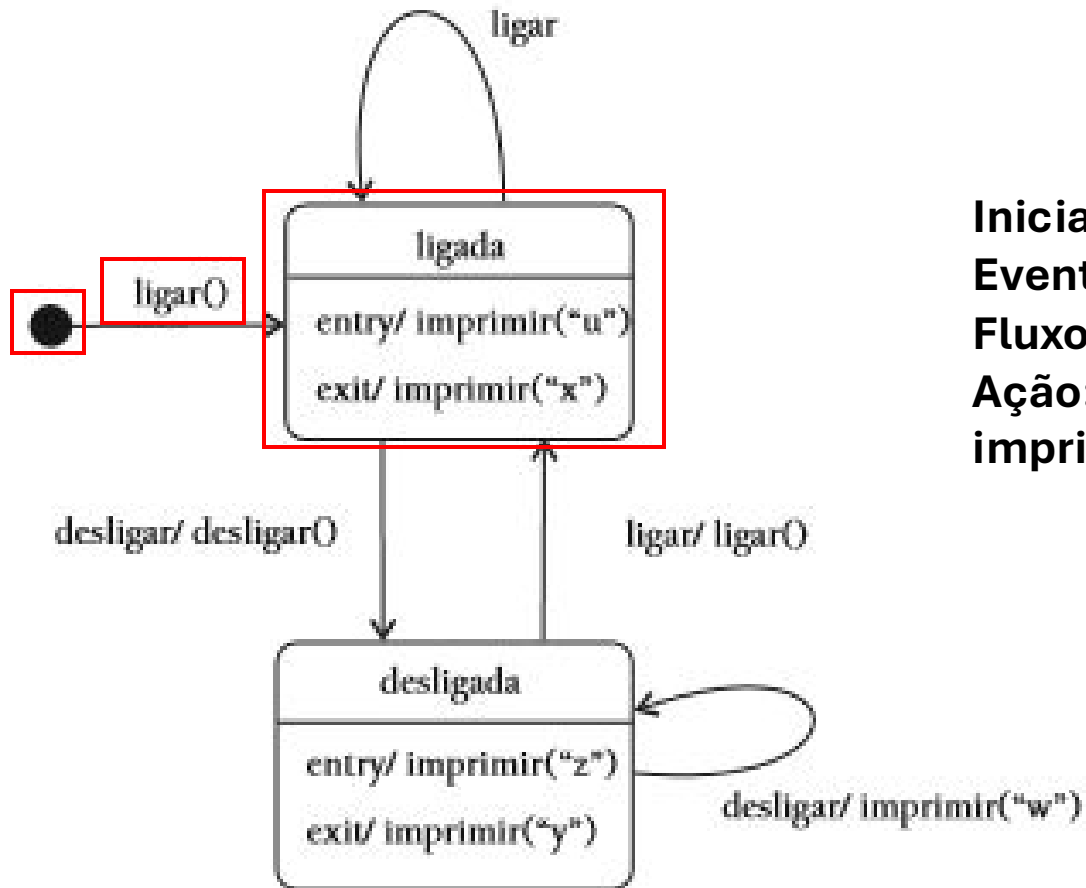


| Evento   | Ações executadas                                |
|----------|---|
| desligar | imprimir("x")<br>desligar()<br>imprimir("z")    |
| desligar | imprimir("y")<br>imprimir("w")<br>imprimir("z") |
| ligar    | imprimir("y")<br>ligar()<br>imprimir("u")       |

# Exemplos de entry e exit



# Exemplos de entry e exit



## Inicialização (Começando o sistema)

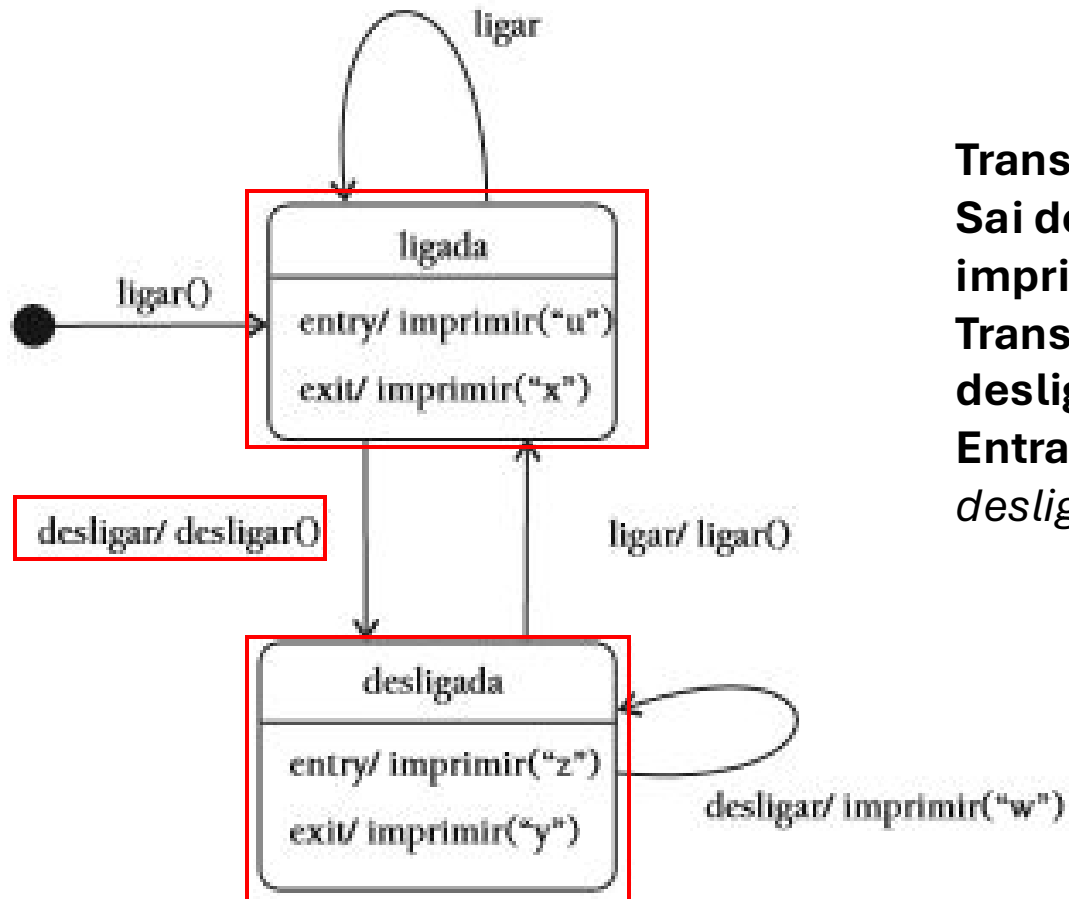
**Evento:** O sistema inicia com o evento **ligar()**.

**Fluxo:** O sistema entra no estado **ligada**.

**Ação:** Como entrou no estado, executa o entry: **imprimir("u")**.



# Exemplos de entry e exit



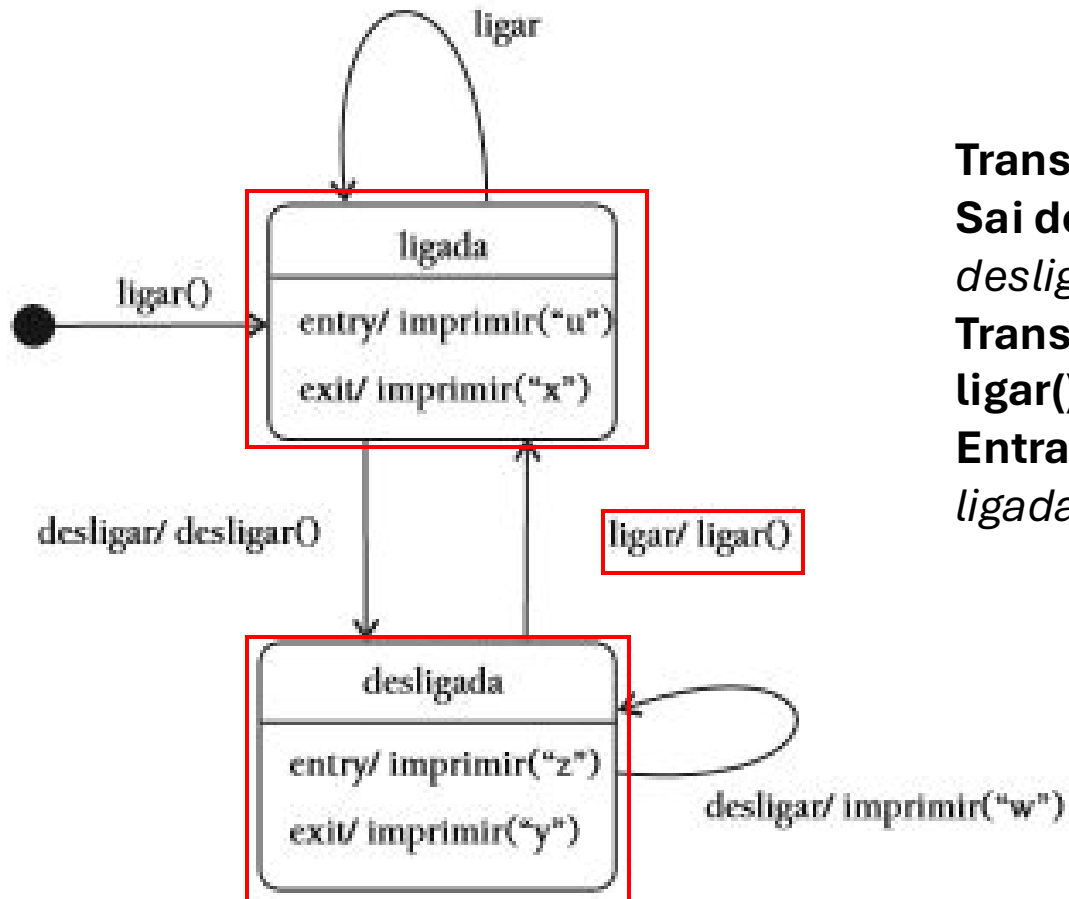
**Transição: De "Ligada" para "Desligada"**

**Sai do estado atual:** Executa o exit de *ligada* -> `imprimir("x")`.

**Transição:** Executa a ação escrita na seta -> `desligar()`.

**Entra no novo estado:** Executa o entry de *desligada* -> `imprimir("z")`.

# Exemplos de entry e exit



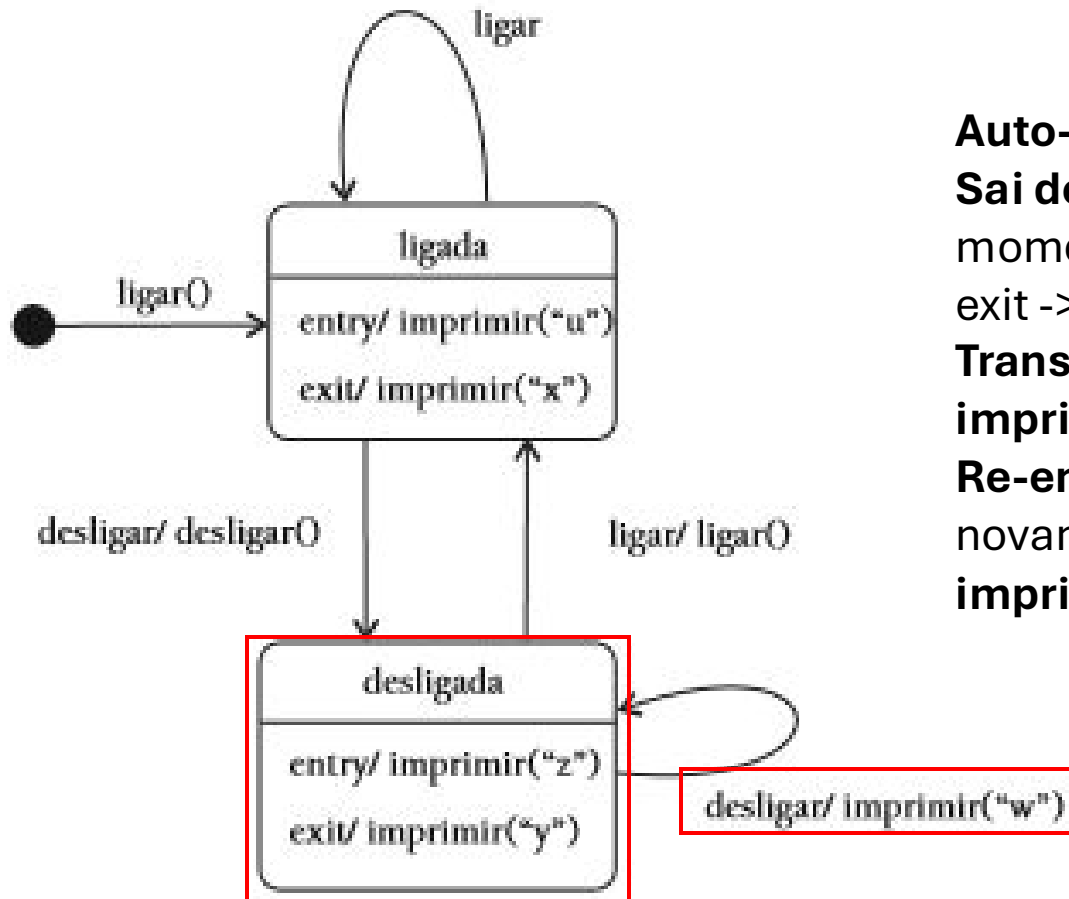
**Transição: De "Desligada" para "Ligada":**

**Sai do estado atual:** Executa o exit de *desligada* -> **imprimir("y")**.

**Transição:** Executa a ação escrita na seta -> **ligar()**.

**Entra no novo estado:** Executa o entry de *ligada* -> **imprimir("u")**.

# Exemplos de entry e exit



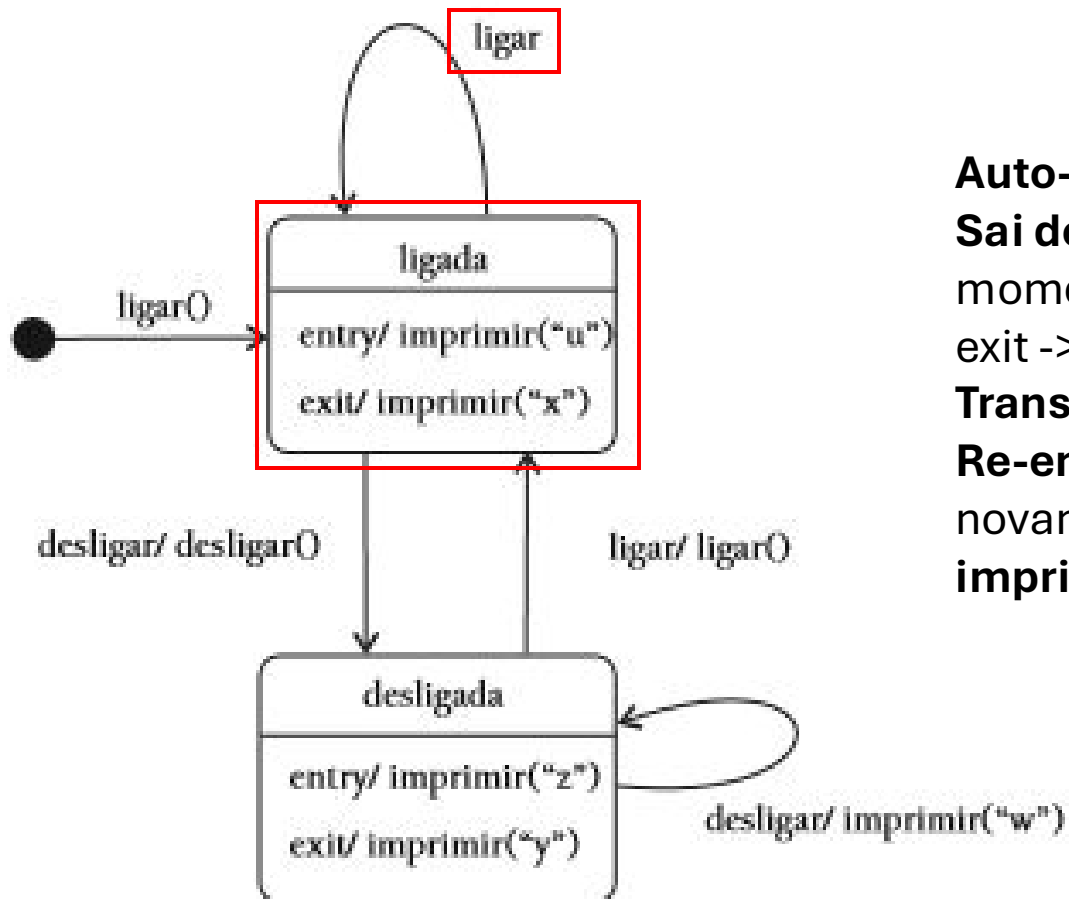
**Auto-transição (Loop) no estado "Desligada"**

**Sai do estado:** O sistema sai momentaneamente do estado, disparando o exit -> **imprimir("y")**.

**Transição:** Executa a ação na seta de loop -> **imprimir("w")**.

**Re-entra no estado:** O sistema entra novamente no estado, disparando o entry -> **imprimir("z")**.

# Exemplos de entry e exit



**Auto-transição (Loop) no estado "Ligada"**

**Sai do estado:** O sistema sai momentaneamente do estado, disparando o exit -> **imprimir("x")**.

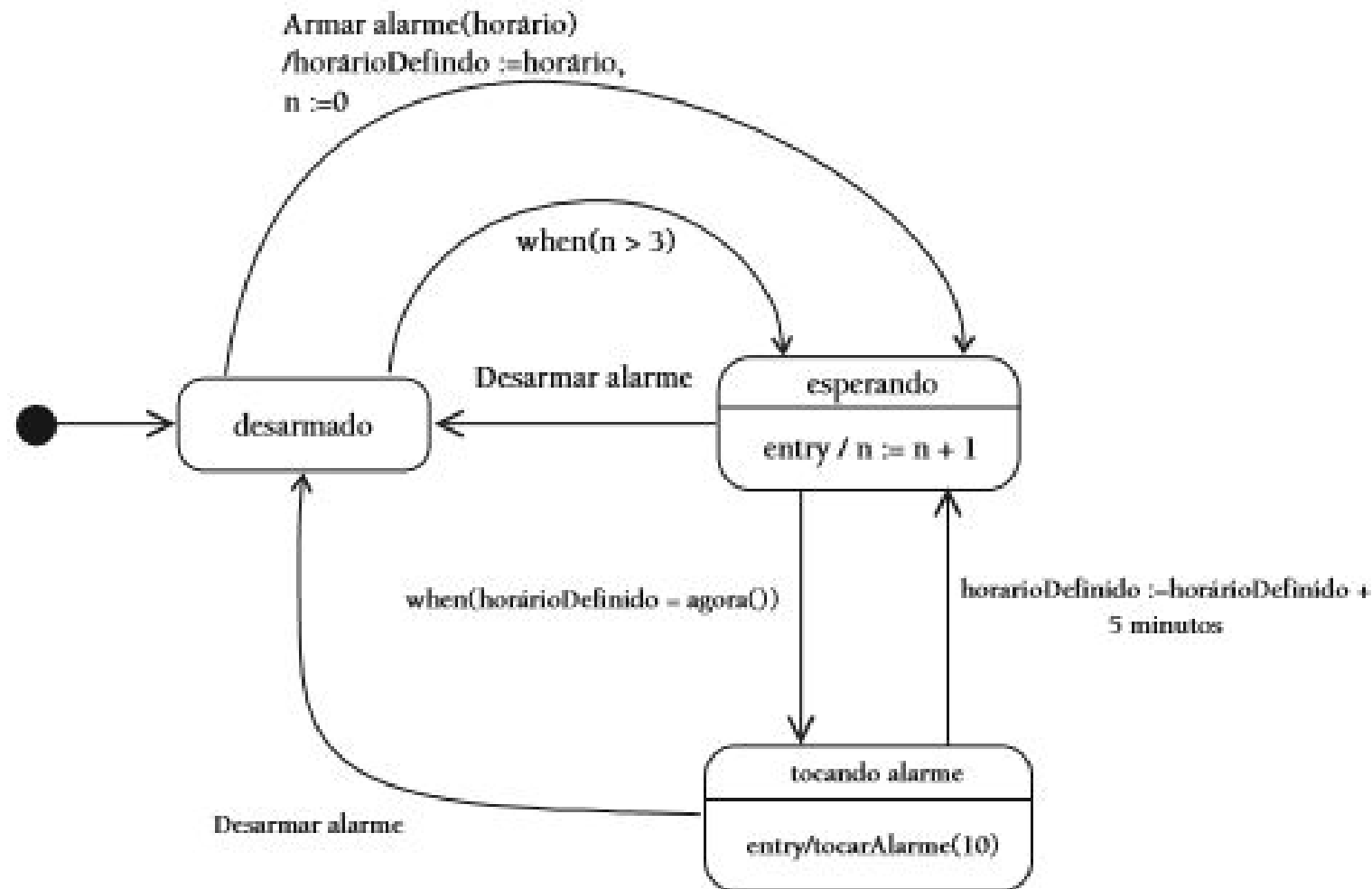
**Transição:** Não realiza nada

**Re-entra no estado:** O sistema entra novamente no estado, disparando o entry -> **imprimir("u")**.

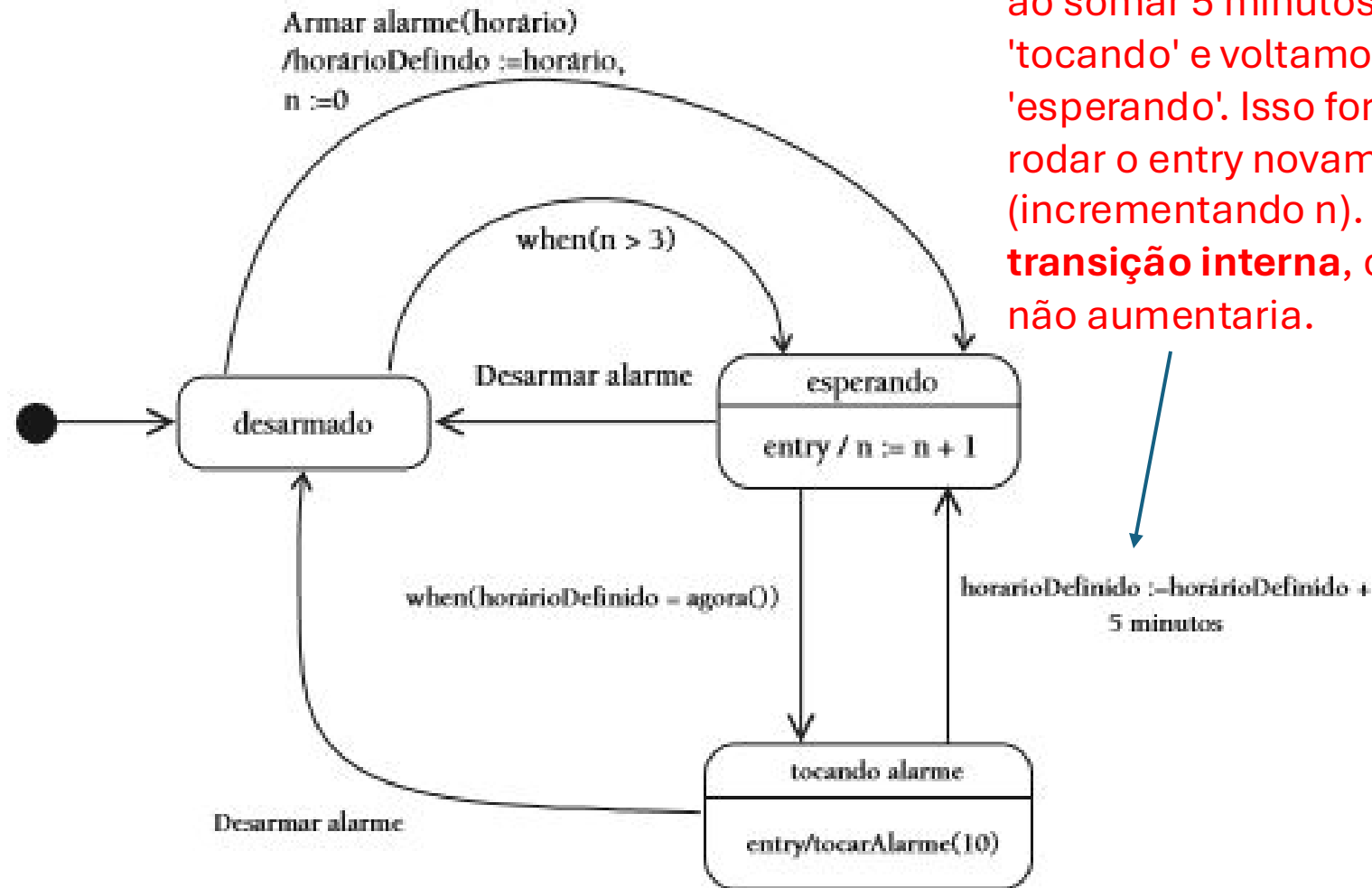
# Transições Internas

- **Definição:** Eventos tratados dentro do próprio estado, sem causar uma mudança de estado (transição).
- **Características Principais:**
  - **Sem Saída/Retorno:** O objeto permanece no mesmo estado durante e após a execução.
  - **Regra de Ouro:** Diferente de uma auto-transição (loop externo), a transição interna **NÃO** dispara as ações de entry (entrada) nem de exit (saída).
  - **Sintaxe UML:** nomeDoEvento / ação (escrito dentro da caixa do estado).

# Transições Internas



# Transições Internas



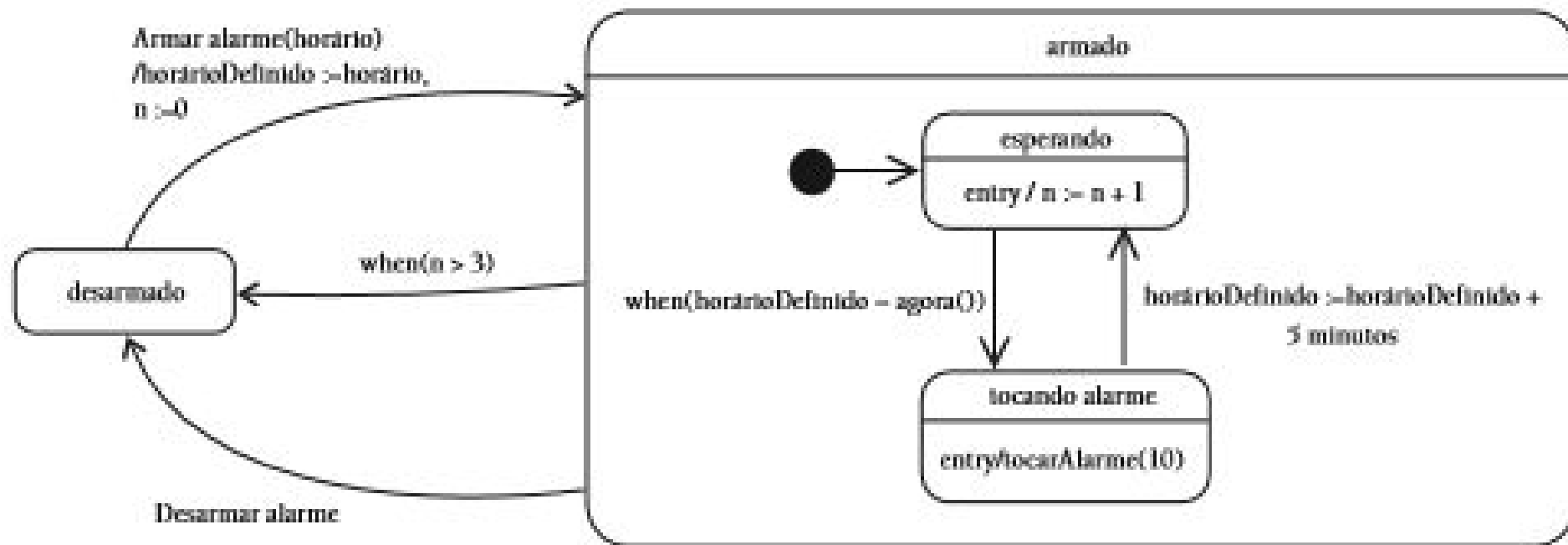
ao somar 5 minutos, saímos de 'tocando' e voltamos para 'esperando'. Isso força o sistema a rodar o entry novamente (incrementando n). Se fosse uma **transição interna**, o contador n não aumentaria.

# Estados Compostos

- Ajuda a simplificar a modelagem de comportamentos complexos
  - É composto por vários estados que representam detalhadamente as etapas do super estado

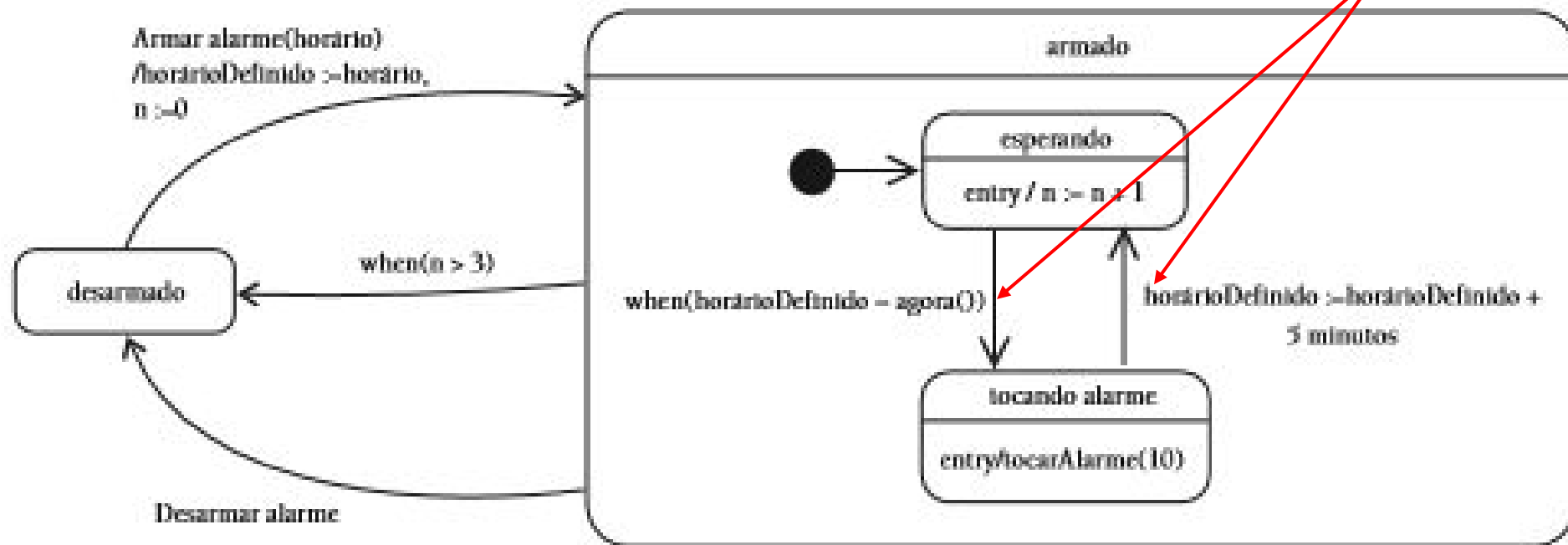


# Estados Compostos



# Estados Compostos

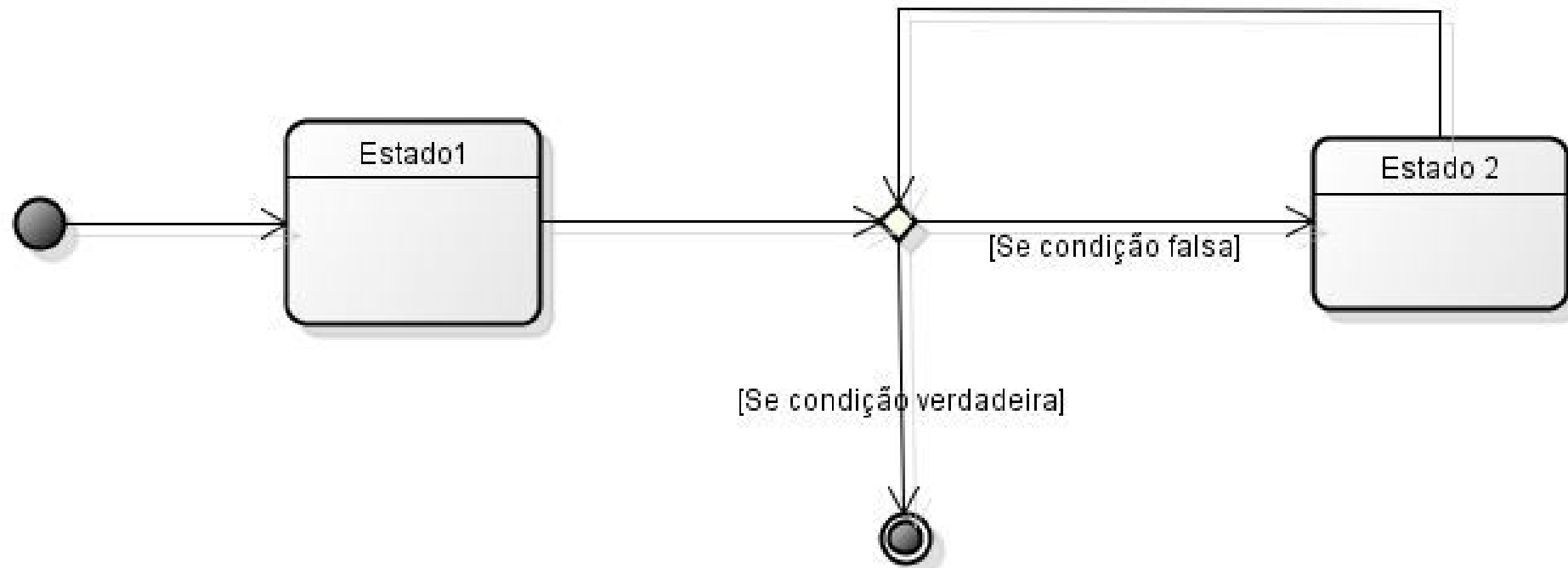
Muda-se de subestado, mas não de estado



# Pseudo Estado de Escolha

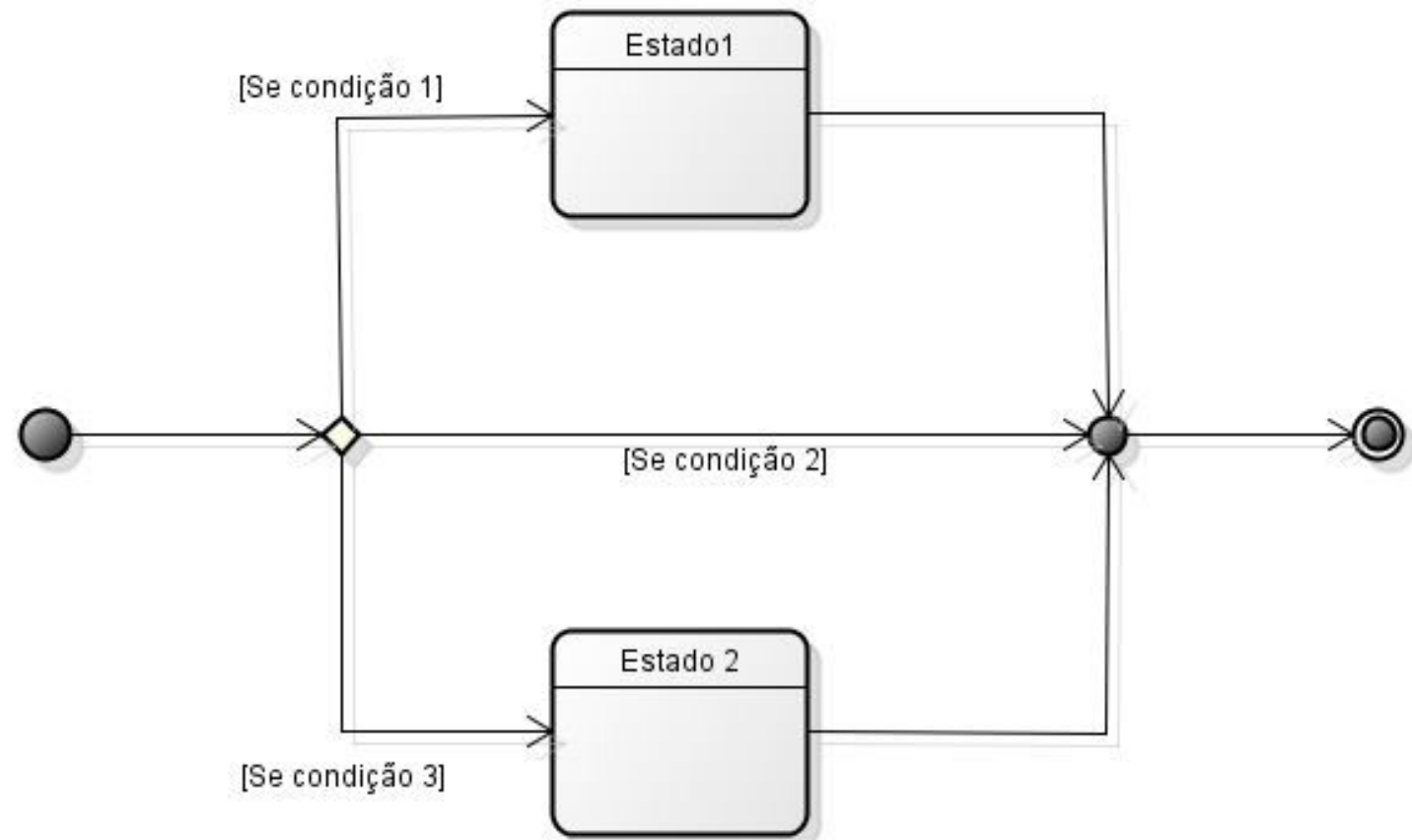
- Quando se tem **mais de uma possibilidade** de ação e deve-se tomar uma **decisão**
  - As condições de guarda atuam na escolha do "caminho" a ser seguido.
  - É representado por um losango de onde partem as opções.

# Pseudo Estado de Escolha



# Pseudo Estado de Junção

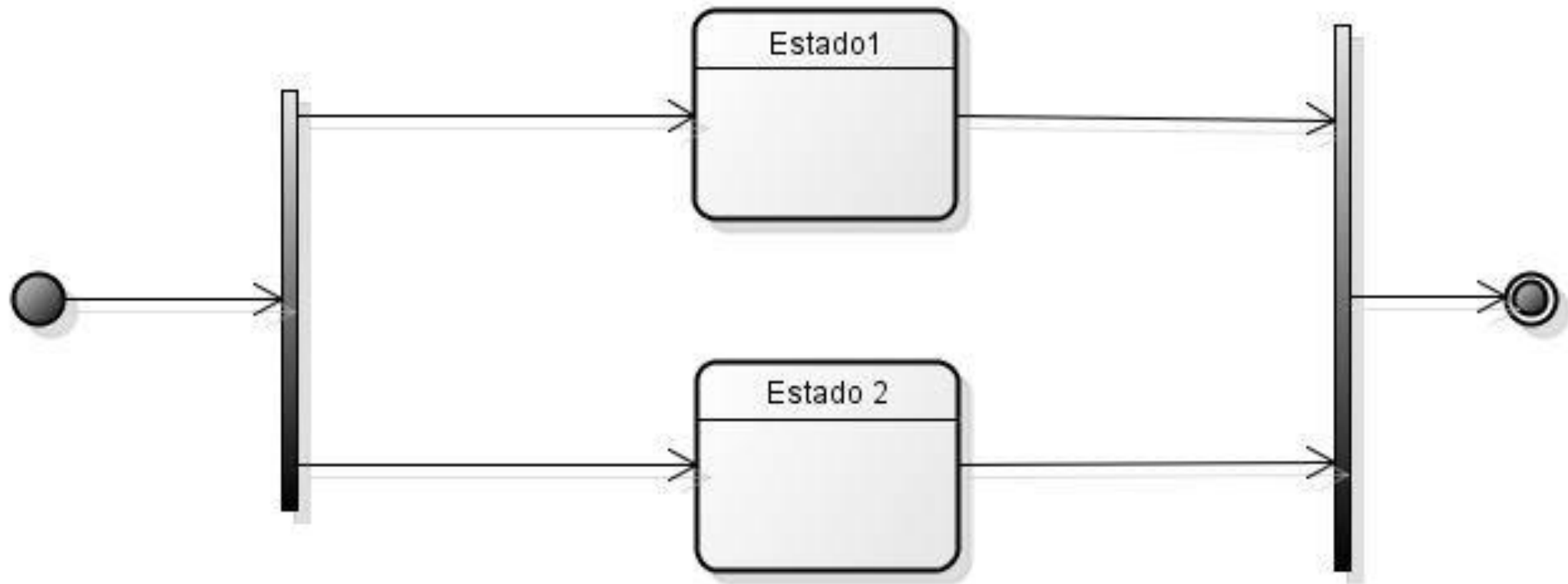
- Possibilita a **junção de vários caminhos em um único**
  - Pode-se utilizar condições de guarda como auxílio



# Barra de Bifurcação e União

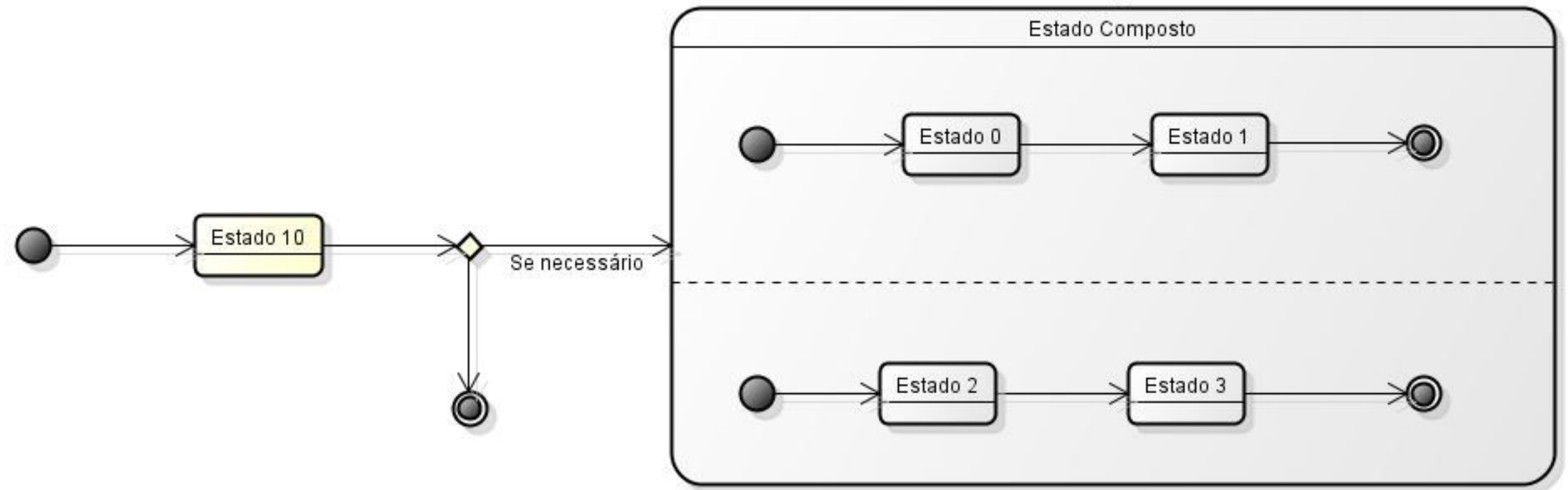
- Utilizada quando se tem estados operando em paralelo (transições concorrentes).
  - Representada por barras

# Barra de Bifurcação e União



# Estados Compostos Ortogonais

- O super estado (ou estado composto) apresenta mais de uma região e, em cada uma delas, os estados atuam paralelamente





# Identificação de Elementos do DTE

- Um bom ponto de partida para identificar eventos é a descrição dos **casos de uso**
- Os eventos encontrados na descrição dos casos de uso são externos ao sistema
- Contudo, uma transição pode também ser disparada por um evento interno ao sistema

# Identificação de Elementos do DTE

- De uma forma geral, **cada operação com visibilidade pública** de uma classe pode ser vista como um evento em potencial
- Uma outra fonte para identificação de eventos associados a transições é analisar as regras de negócio
  - “*Um cliente do banco não pode retirar mais de R\$ 1.000 por dia de sua conta*”
  - “*Os pedidos para um cliente não especial devem ser pagos antecipadamente*”
  - “*O número máximo de alunos por curso é igual a 30*”

# Um DTE para uma classe

- diagramas de estados são desenhados por classe
  - **Desvantagem:** dificuldade na visualização do estado do sistema como um todo
  - Essa desvantagem é parcialmente compensada pelos diagramas de interação
- Nem todas as classes de um sistema precisam de um DTE
  - Somente classes que exibem um **comportamento dinâmico relevante**
  - Objetos cujo **histórico precisa ser rastreado pelo sistema** são típicos para se construir um diagrama de estados

# Exercício de Fixação

Na natureza a água pode ser encontrada em quatro formas - líquido, gasoso, sólido ou plasma. Se a água está no formato líquido, ela pode congelar e passar ao formato sólido, ou ela pode evaporar e passar ao formato gasoso. Se a água está no formato sólido, ela pode descongelar e passar ao formato líquido, ou ela pode sublimar e passar ao formato gasoso. Se a água está no formato gasoso, ela pode condensar e passar ao formato líquido, ela pode depositar e passar ao formato sólido, ou ionizar e passar ao formato de plasma. Se a água está no formato de plasma, ela pode desionizar e passar ao formato gasoso.