

Projeto de Software

# Diagrama de Transição de Estados

**Lesandro Ponciano**

2024

# Objetivos

- Contextualizar o uso do Diagrama de Transição de Estados (DTE) no Projeto de Software
- Apresentar os principais componentes do diagrama
- Discutir exemplos

# 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*)
- Costuma ser chamado de
  - Diagrama de estados
  - Diagrama de máquina de estados

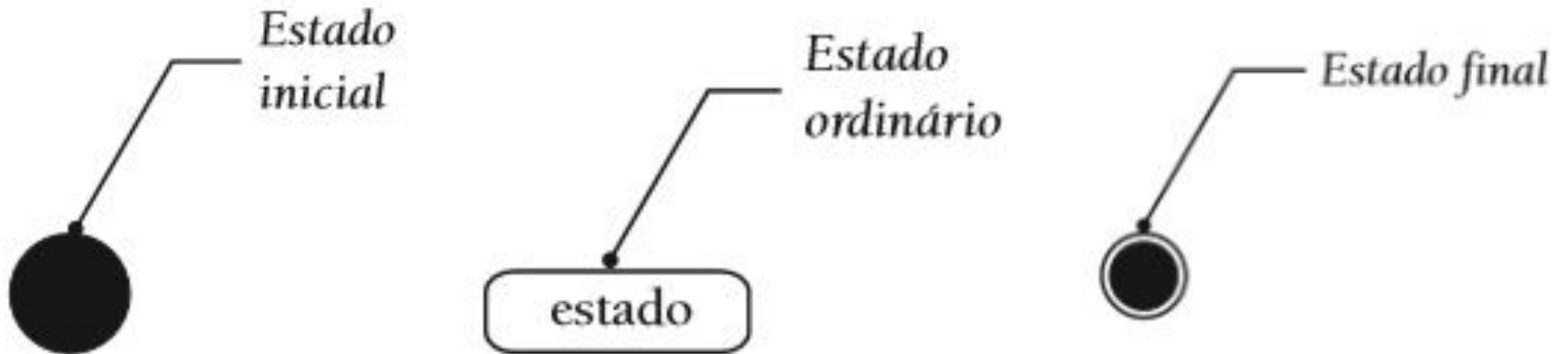
# Elementos Básicos

- 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

# Representações de Estados no DTE





# 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	1 Pedido realizado
2 Disco inserido	2 Fatura paga
	3 Cheque devolvido

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

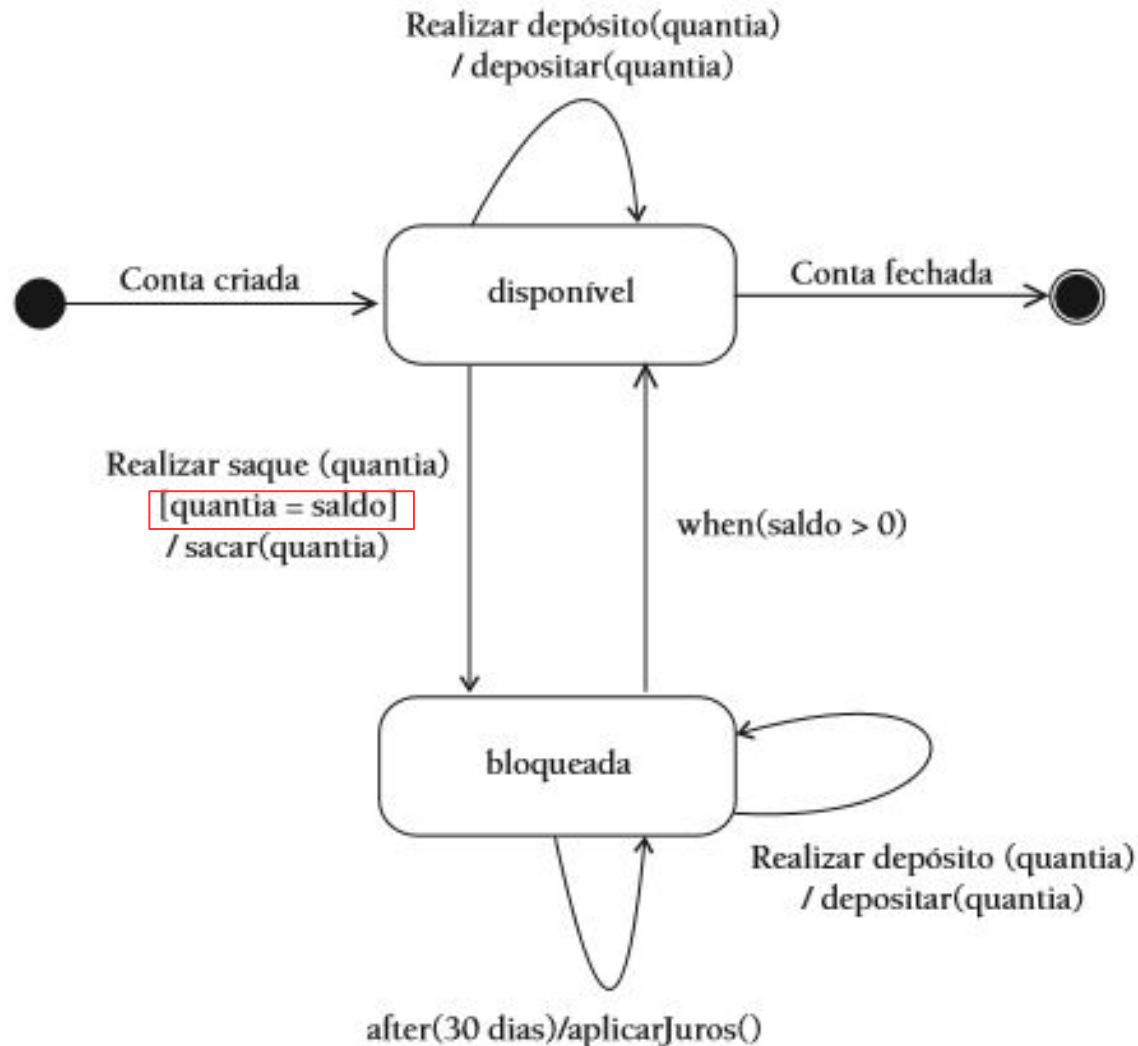
# Tipos Comuns de Eventos

- 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

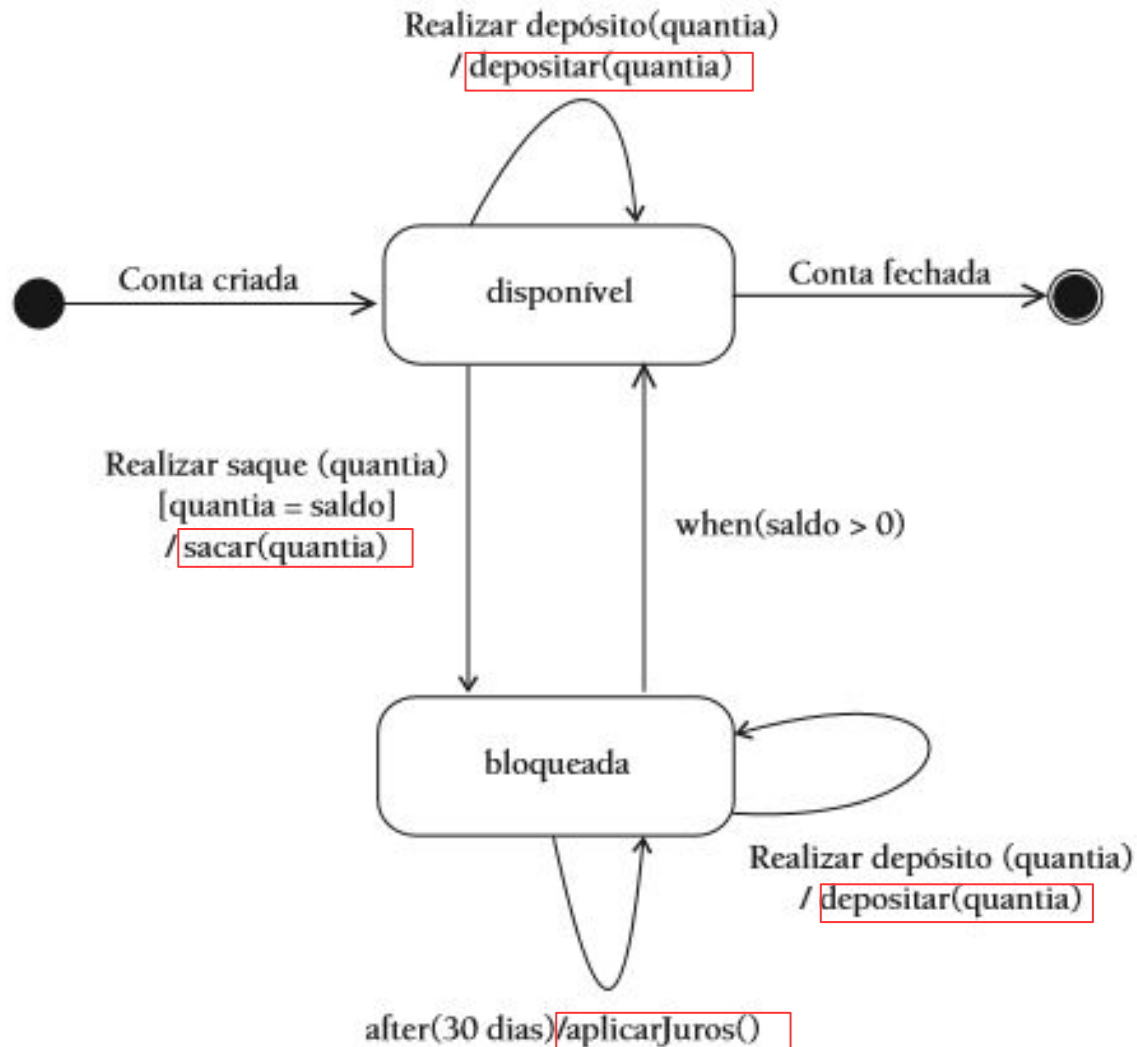
# Exemplo de **Condição de Guarda**



# 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

# Exemplos de Ações

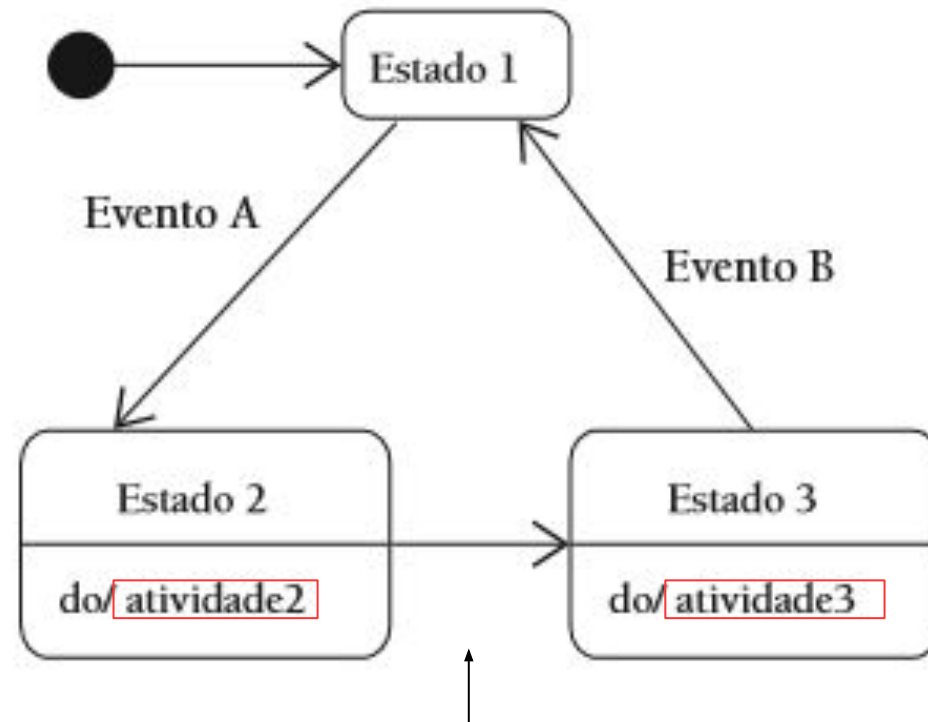


# Atividade

- 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



# Exemplos de Atividade

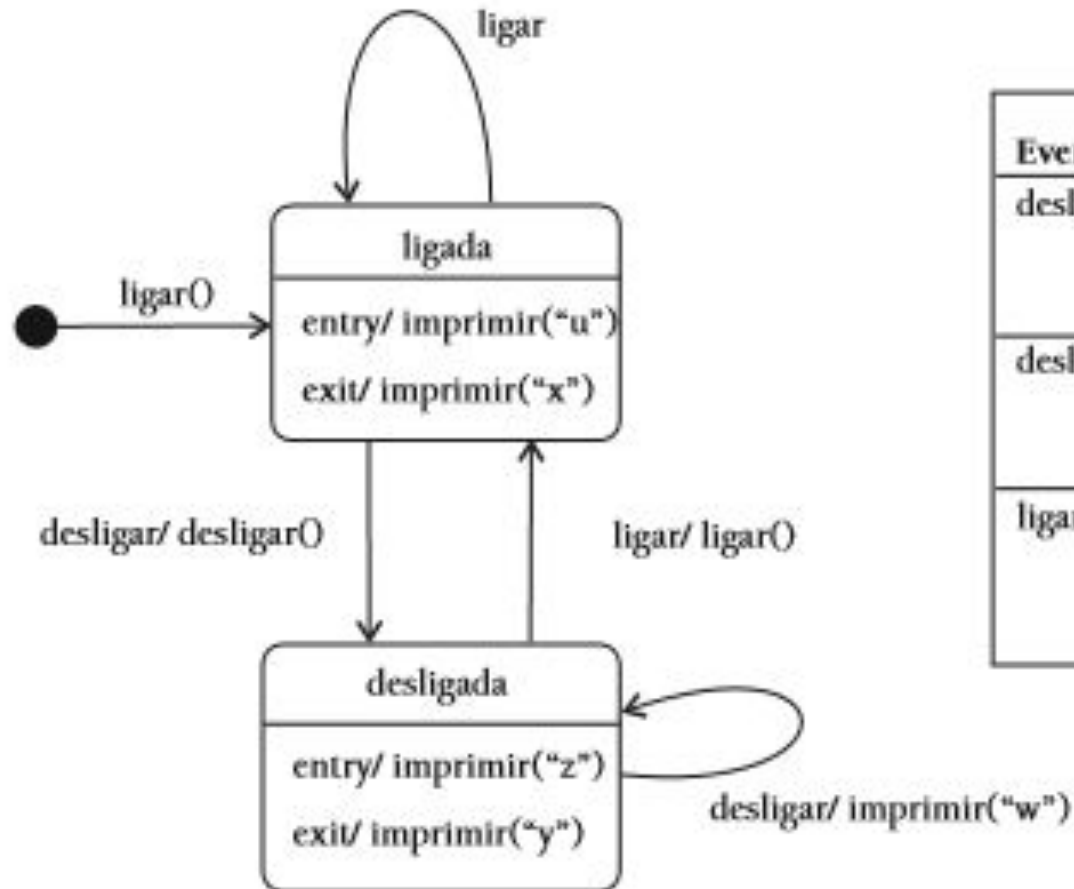


Note a relação entre atividade e evento

# Cláusulas entry, exit, do

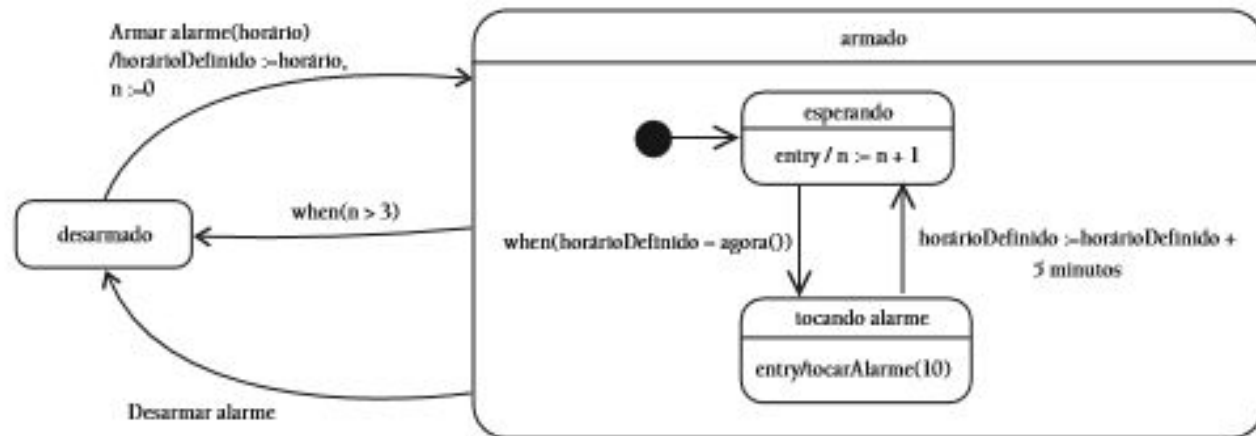
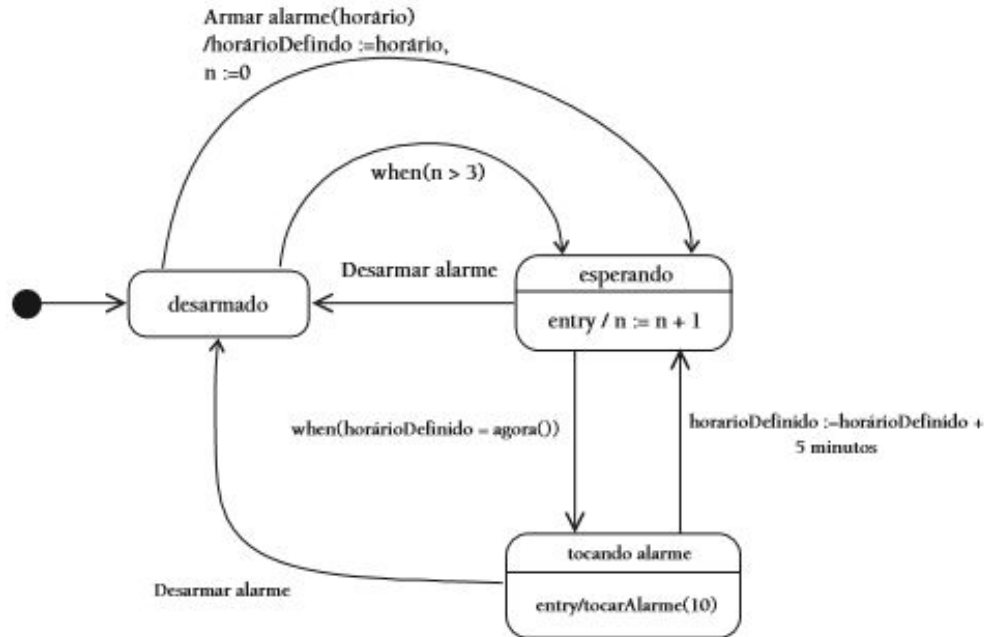
- Entry
  - usada para especificar uma **ação** a ser realizada no momento em que o objeto entra em um estado
- Exit
  - Serve para declarar ações que são executadas sempre que o objeto sai de um estado
- Do
  - Serve para definir alguma **atividade** a ser executada quando o objeto passa para um determinado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

# Exemplos de Entry e Exit

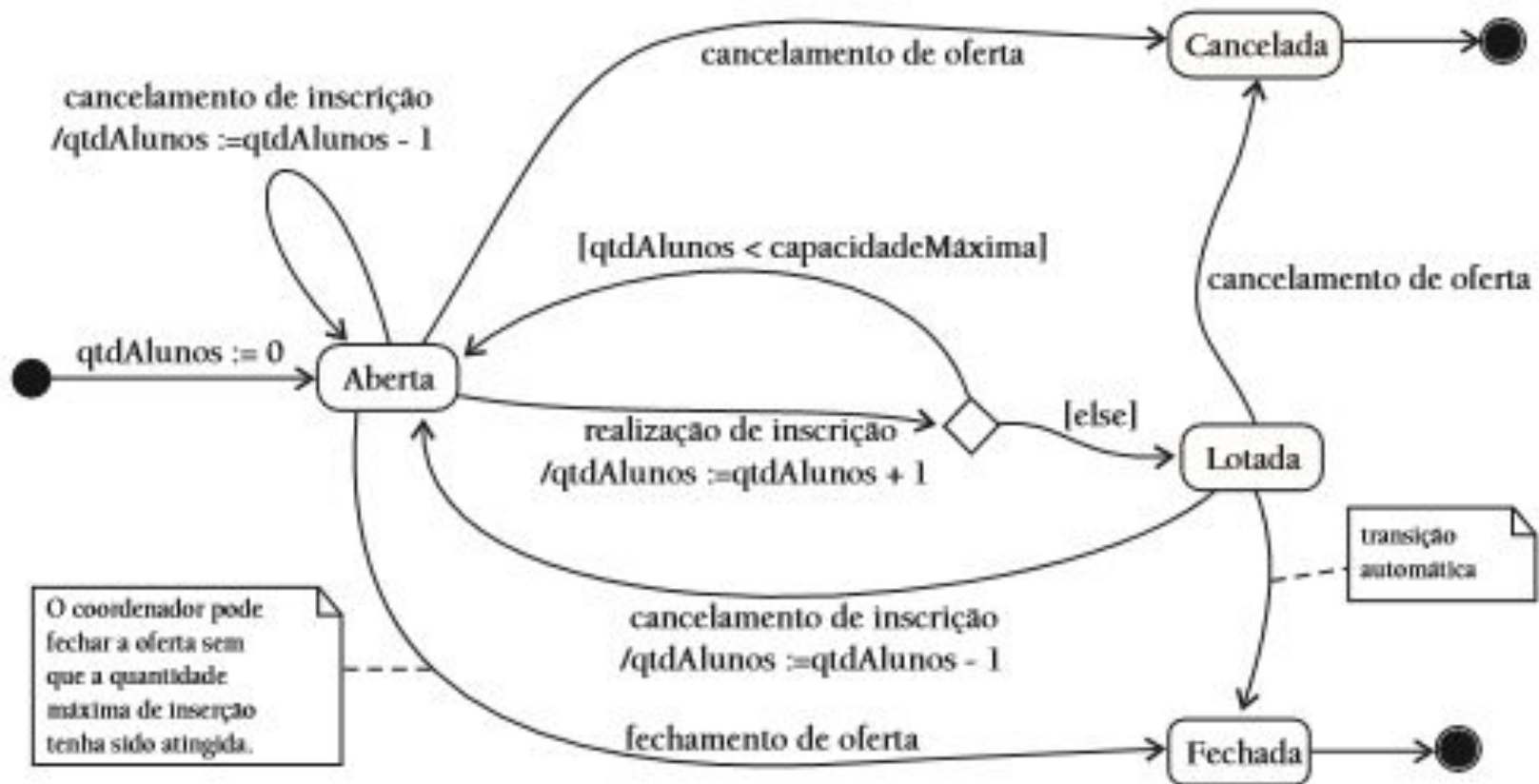


Evento	Ações executadas
desligar	imprimir("x") desligarO imprimir("z")
desligar	imprimir("y") imprimir("w") imprimir("z")
ligar	imprimir("y") ligarO imprimir("u")

# Estados Compostos e Transições Internas



# Pseudo-estado de Escolha



## 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.

# Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson 548 ISBN 9788579361081 (Capítulo 5)
- GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec, c2011. 484 p. ISBN 9788575222812 (Capítulo 9)
- Bezerra, Eduardo. Princípios de Análise e Projeto de Sistema com UML. Vol. 3. Elsevier Brasil, 2007. (Capítulo 9)
- Os diagramas foram feitos usando a ferramenta Astah <<http://astah.net/editions>>

Projeto de Software

**Prof. Dr. Lesandro Ponciano**

<https://orcid.org/0000-0002-5724-0094>