

# Análise e Programação Orientada a Objetos

Sistemas de Informação

Unidade II – Parte IV  
Modelagem de Estados  
*Projetando o Sistema*

Prof. Marciel de Liz Santos

# Introdução

- ♦ Objetos do mundo real se encontram em estados particulares a cada momento.
  - uma jarra está cheia de líquido
  - uma pessoa está cansada.
- ♦ Da mesma forma, cada objeto participante de um sistema de software orientado a objetos se encontra em um estado particular.
- ♦ Um objeto muda de estado quando acontece algum evento interno ou externo ao sistema.

# Introdução

- ♦ Ciclo de vida de um objeto: as sequências de estados que pode passar durante a sua vida em resposta a eventos, em conjunto com as respostas a esses eventos, ou seja:
  - os estados possíveis (exemplo - estado civil: solteiro, casado, ...)
  - as transições de estado possíveis (exemplo: pode passar de solteiro para casado, mas não o contrário ☺)
  - os eventos que causam essas transições (exemplo: o casamento implica a passagem ao estado de casado)
  - as ações do objeto em resposta a esses eventos (ex: despedida de solteiro)

# Introdução

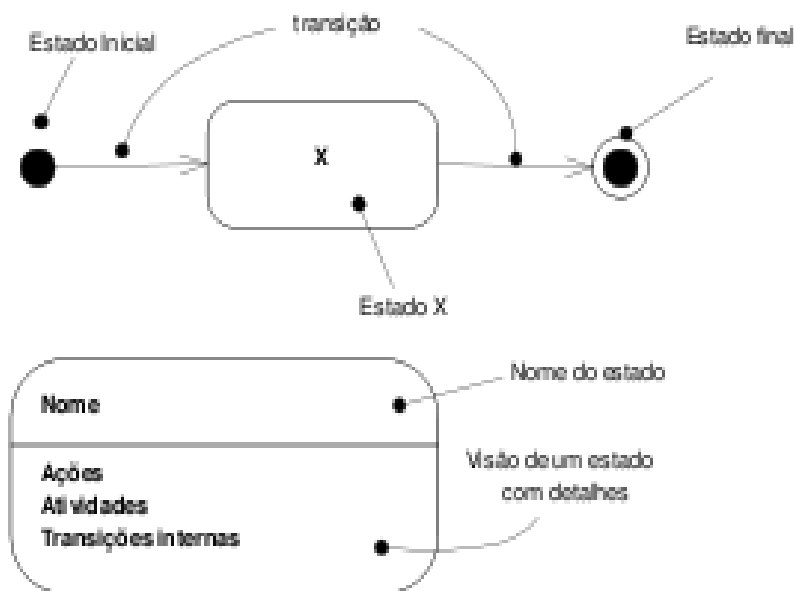
- ♦ O diagrama da UML que permite observar quais os possíveis estados dos objetos de uma classe, os eventos que causa a transição de um estado para o outro e a realização de operações resultantes é o **diagrama de transição de estado (DTE)**. (statechart)
- ♦ Portanto, um DTE é útil para modelar o comportamento e o ciclo de vida de um objeto
- ♦ Objeto como **máquina de estados**



- ♦ Objetos da mesma classe têm o mesmo ciclo de vida, basta construir um diagrama de estados por **classe relevante**

# Diagrama de transição de estado

- ◆ A UML tem um conjunto rico de notações para desenhar um DTE.
  - Estados
  - Transições
  - Eventos
  - Ações
  - Atividades
  - Transições internas
  - Estados aninhados
  - Estados concorrentes



# Estado

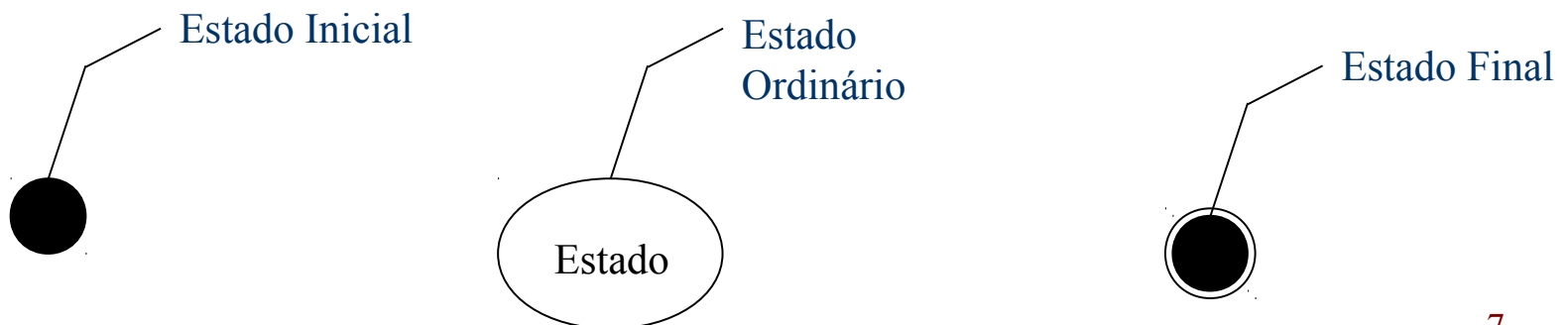
- ♦ Um **estado** é uma condição ou situação na vida de um objeto, durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou espera por algum evento
- ♦ Exemplo (relativamente ao estado civil de uma pessoa): solteira, casada, ...
- ♦ um telefone está no estado “ocioso”, após o fone ter sido colocado no gancho e até que o mesmo seja retirado novamente do gancho
- ♦ Cada estado de um objeto é determinado pelos **valores dos seus atributos** e (ou) pelas suas **ligações com outros objetos**.

Exemplos:

- “o atributo reservado deste objeto livro tem valor verdadeiro”.
- “uma conta bancária passa para o vermelho quando o seu saldo fica negativo”.
- ♦ Representado graficamente por um retângulo com cantos arredondados.

# Estado inicial e final

- ♦ O estado inicial indica o estado de um objeto quando ele é criado. Só pode haver um estado inicial em um DTE.
  - Essa restrição serve para definir a partir de que ponto um DTE deve começar a ser lido.
- ♦ O estado final é representado como um círculo “eclipsado” e indica o fim do ciclo de vida de um objeto.
  - Este estado é opcional e pode haver mais de um estado final em um DTE.



# Transições

- ♦ Uma **transição** é um relacionamento entre dois estados, indicando que, quando um evento ocorre o objeto passa do estado anterior para o estado subsequente.
  - Ex.: quando ocorre o evento “tira do gancho”, o telefone sofre uma **transição** do estado “ocioso” para o estado “ativo”.
- ♦ quando uma transição entre estados ocorre, diz-se que a **transição foi disparada**
- ♦ Uma transição pode ser rotulada com uma expressão da seguinte forma:

evento (lista-parâmetros) [guarda] /
- ♦ Guarda - expressão de valor lógico que condiciona o disparo de uma transição.
  - Uma transição que não possui condição de guarda é sempre disparada quando o evento ocorre.



# Transições

- ♦ Uma transição pode ter múltiplos estados-origem e estados-destino, mas não é comum este tipo de utilização.
- ♦ **Transições internas**
  - Eventos que exigem uma resposta do objeto mas não provocam uma mudança de estado.
    - Ex: `help` / `mostrarHelp()`
- ♦ Auto-Transições ou Transições Reflexivas 2.
- ♦ Evento que não provoca mudança de estado, mas provoca a interrupção do estado corrente, obrigando à sua reentrada.

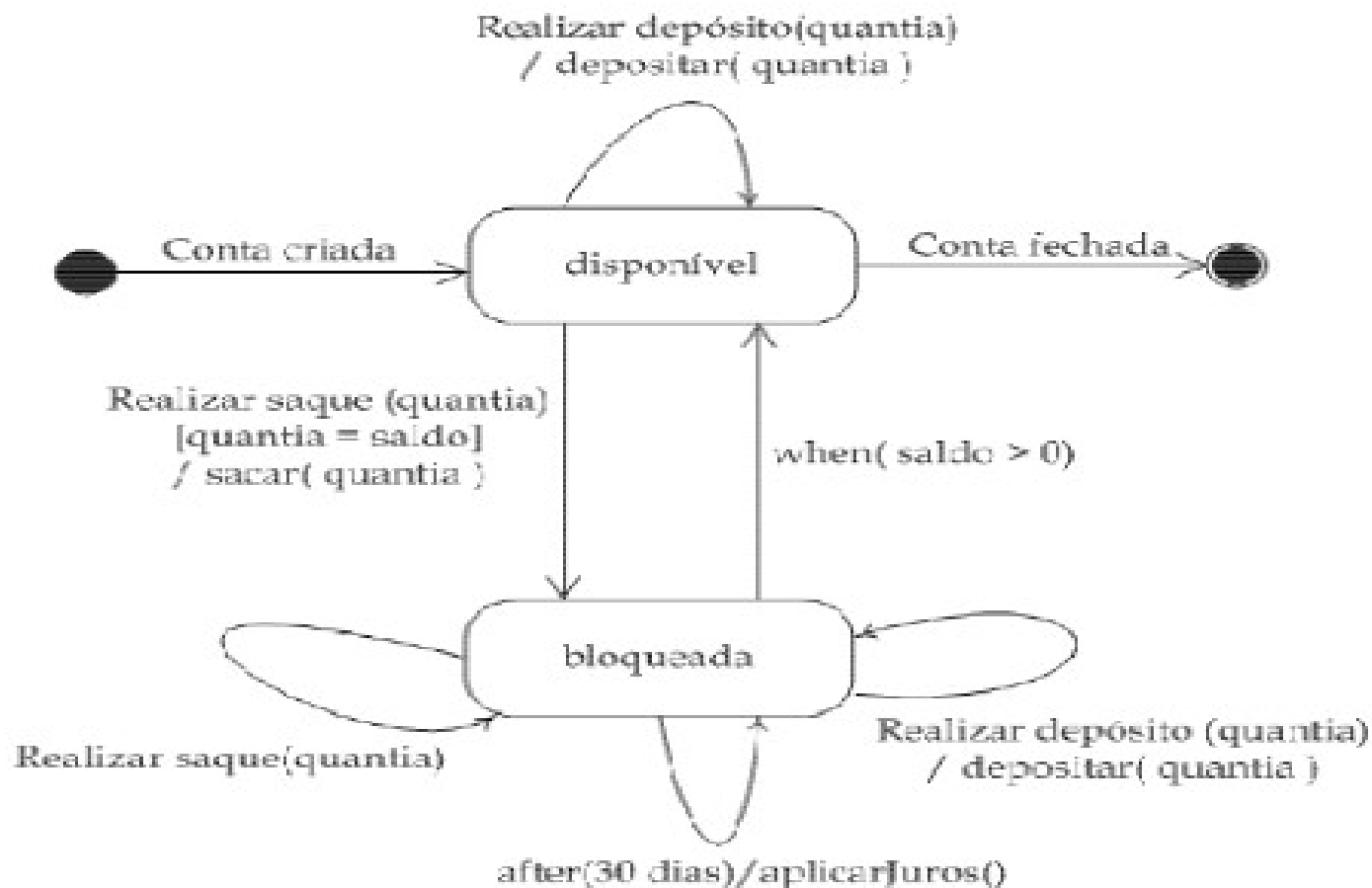
# Eventos

- ♦ Um evento é ocorrência de um estímulo em algum ponto no tempo e que pode corresponder a uma transição de estado:
  - aparelho telefônico é tirado do gancho
  - pedido foi realizado
  - mouse pressionado / cd inserido no drive
- ♦ Os eventos relevantes a um sistema de software podem ser classificados em quatro tipos:
  - 1. Evento de chamada: recebimento de uma mensagem de outro objeto (solicitação de serviço de um objeto a outro).
  - 2. Evento de sinal: recebimento de um sinal. A diferença básica entre o evento de sinal e o evento de chamada é que neste último o objeto que envia a mensagem fica esperando a execução da mesma

# Eventos

- ♦ 3. **Evento temporal**: passagem de um intervalo de tempo predefinido.
  - Um evento temporal é especificado com a cláusula **after** seguida de um parâmetro que especifica um intervalo de tempo.
  - Ex: **after(30 segundos)**: indica que a transição correspondente será disparada 30 segundos após o objeto ter entrado no estado atual.
- ♦ 4. **Evento de mudança**: uma condição que se torna verdadeira. É representado por uma expressão de valor lógico (verdadeiro ou falso) e é especificado utilizando-se a cláusula **when**.
  - Ex: **when(saldo > 0)**: significa que a transição é disparada quando o valor do atributo saldo for positivo.
  - Eventos temporais também podem ser definidos utilizando-se a cláusula **when**.
  - **when(data = 13/07/2002)**
  - **when(horário = 00:00h)**

# Exemplo (Conta Bancária)



# Ações

- ♦ Ao transitar de um estado para outro, um objeto pode realizar uma ou mais **ações**.
- ♦ Computação atômica – execução não interrompível.
  - Podem ser associadas a estados e transições
  - Num estado
    - Ações de entrada: **entry** / **ação**
      - ♦ equivale a associar a ação a cada transição que entra no estado
    - Ações de saída : **exit** / **ação**
      - ♦ equivale a associar a ação a cada transição que sai do estado
    - Outras ações internas: **evento** / **ação**

# Ações (Exemplo)

Digitando senha

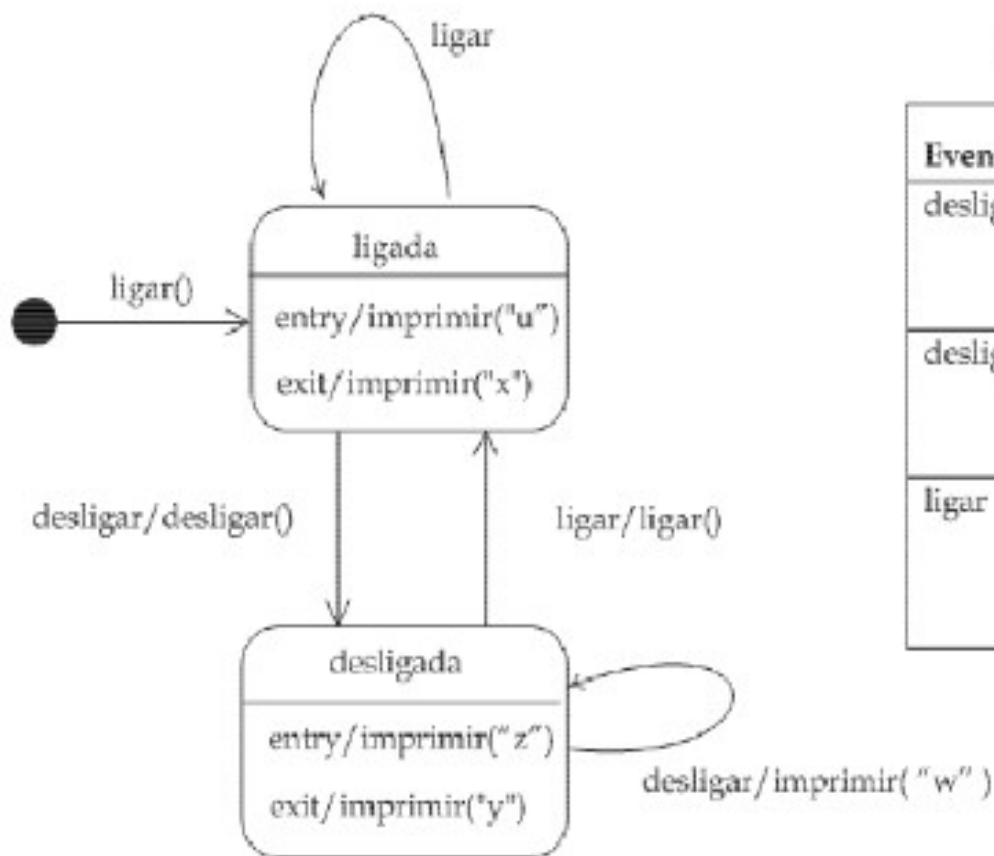
entry/definirEco(cInvisivel)

caractere(c)/tratarCaractere(c)

ajuda/exibir Ajuda(invisivel)

exit/definirEco(cVisivel)

# Ações (Exemplo)

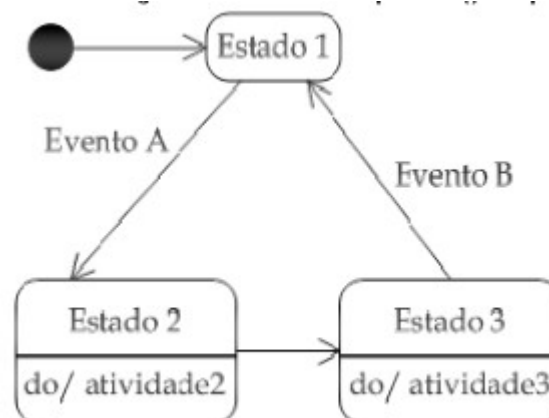


Estado inicial = ligada

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

# Atividades

- ◆ Semelhantes a ações, atividades são algo que deve ser executado
  - ◆ Computação não atômica – execução interrompível
- Uma atividade só pode estar associada a um estado
- ◆ Especificadas pela cláusula “do”
    - Sequência de ações : “do / oper1(); oper2();...”

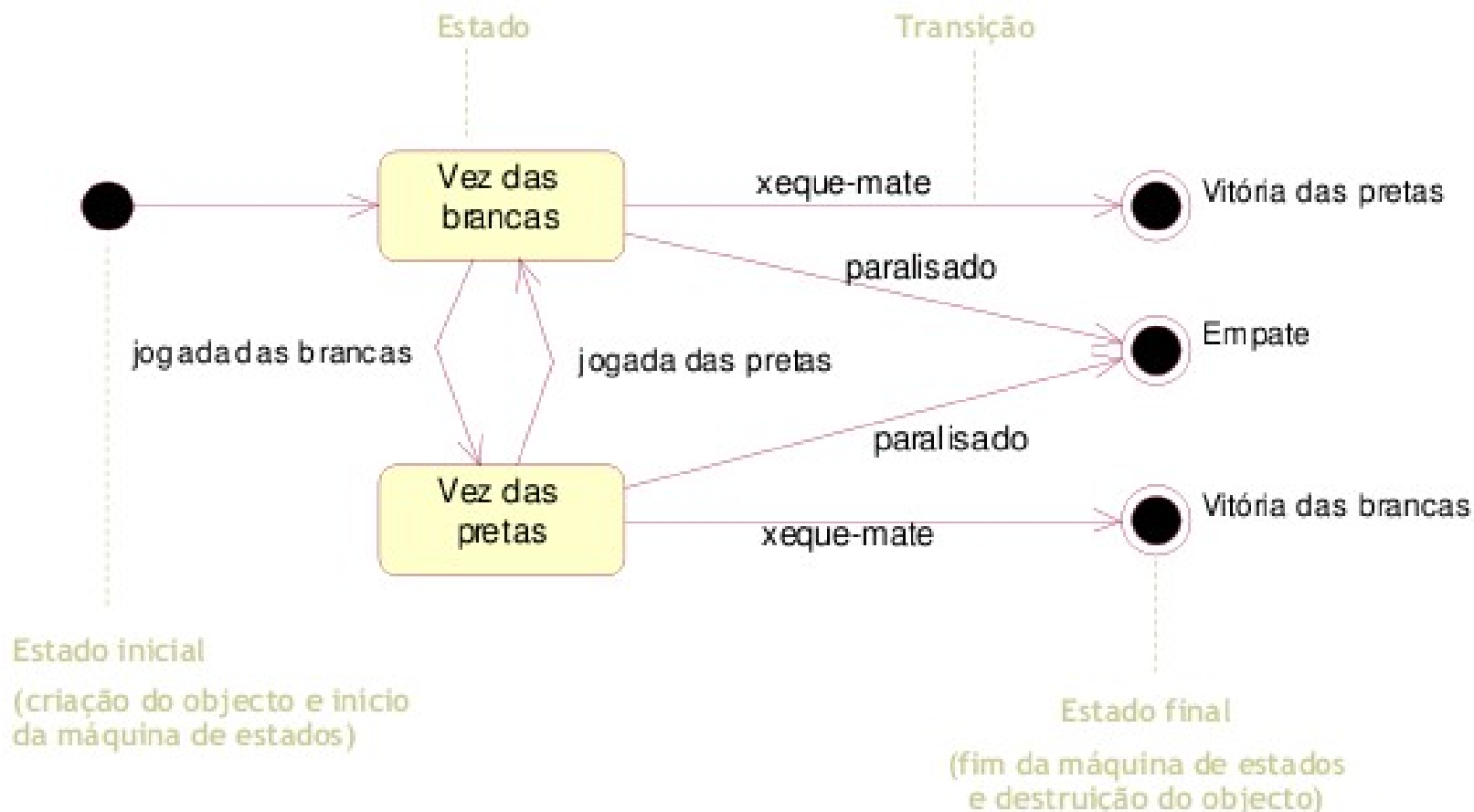




# Notação Básica (Resumo)

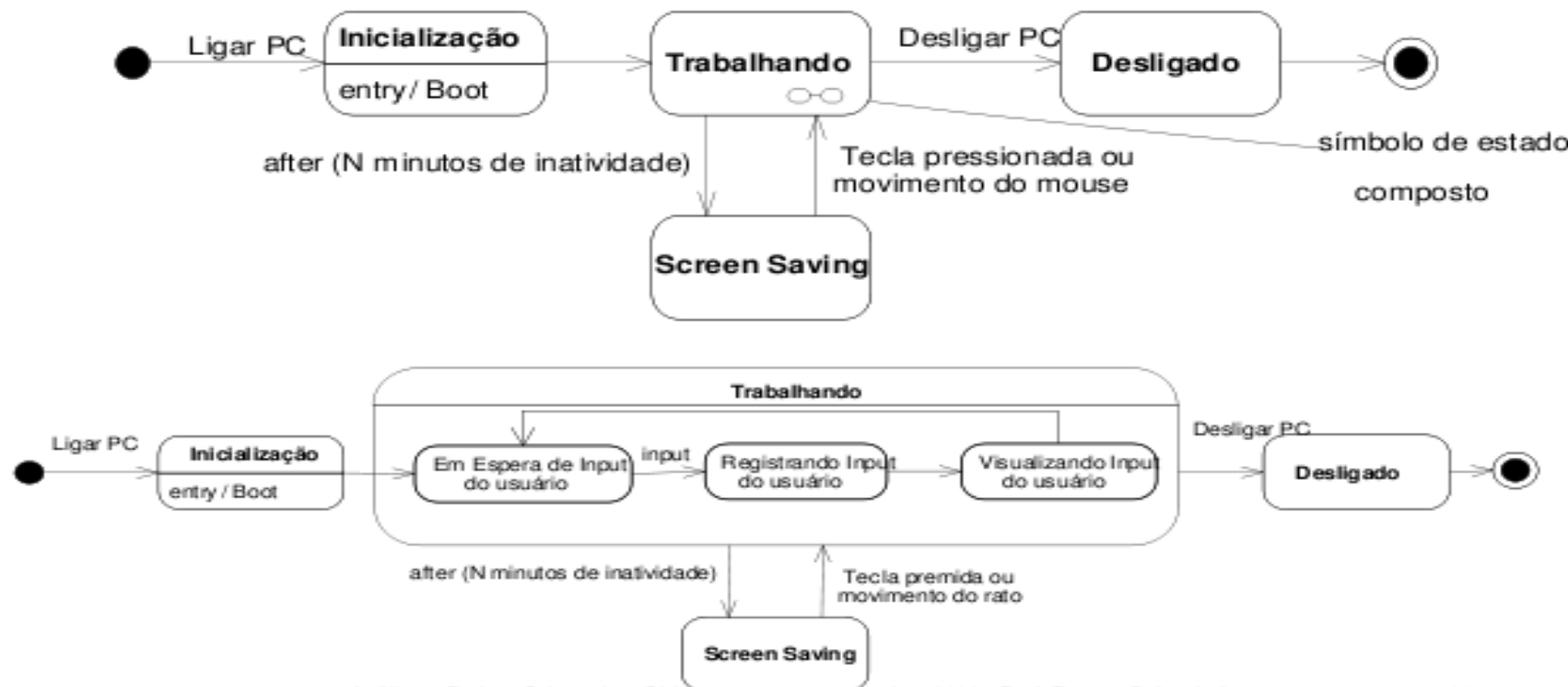


# Exemplo (Jogo Xadrez)

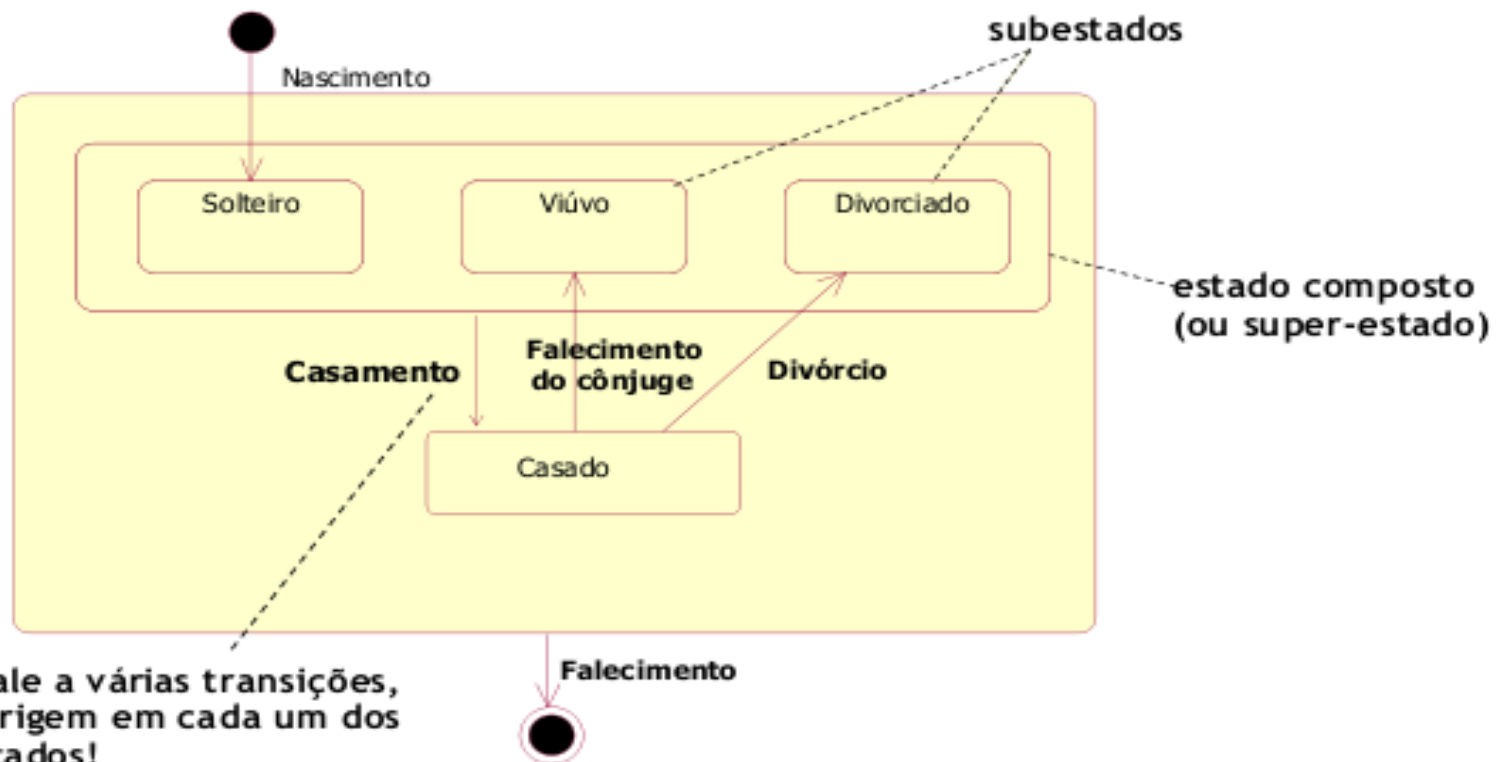


# Estado Aninhados

- Vários estados de um diagrama (e transições entre esses estados) podem ser agrupados num único **estado composto**



# Exemplo (Estado Civil)

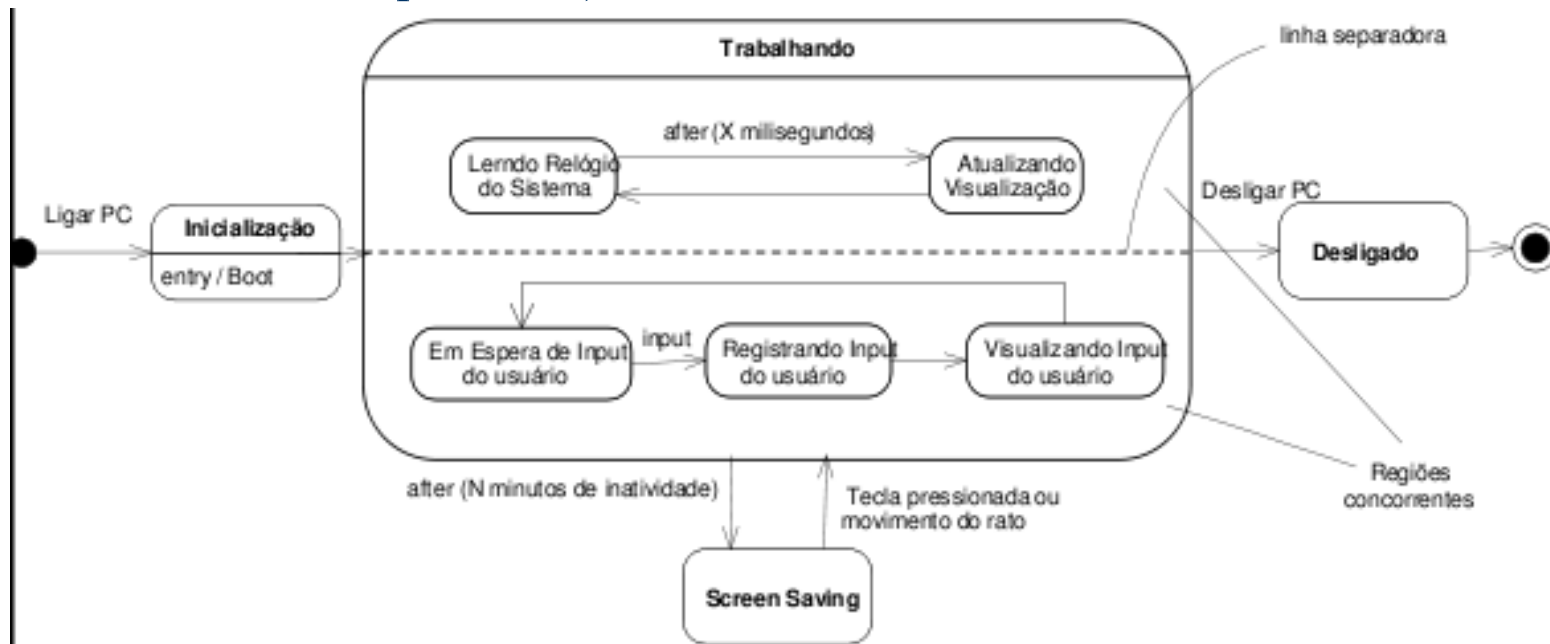


Equivale a várias transições,  
com origem em cada um dos  
subestados!

Evita explosão combinatória de  
transições!

# Estados Concorrentes

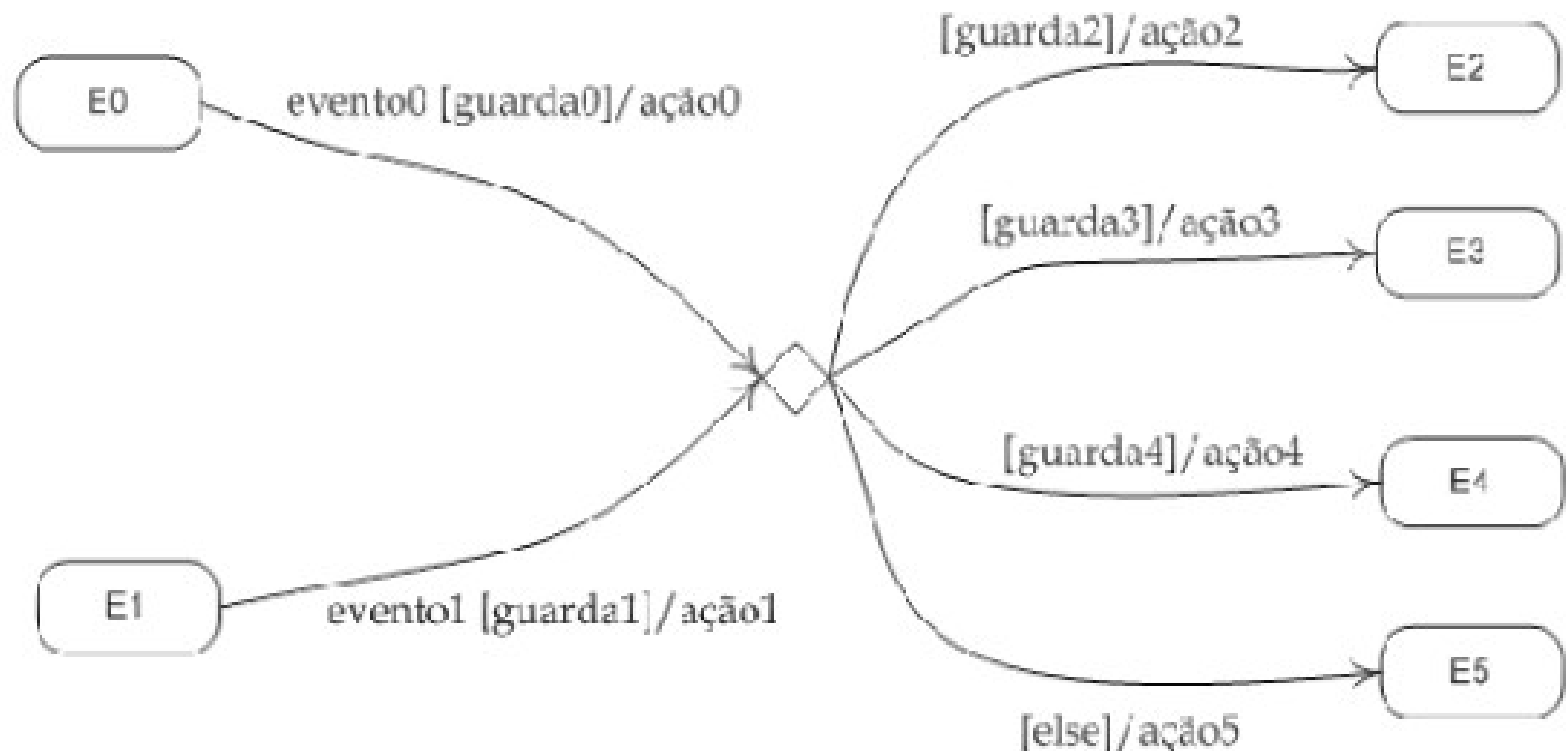
- Um estado (**estado composto**) pode ser dividido em duas ou mais regiões concorrentes, separadas por linhas a tracejadas, representando **subestados concorrentes** (que, por sua vez, têm normalmente subestados sequenciais)



# Ponto de Junção

- ♦ Pode ser que o próximo estado de um objeto varie de acordo com uma condição.
  - Se o valor da condição for **verdadeiro**, o objeto vai para um estado E1; se o valor for **falso**, o objeto vai para outro estado E2.
  - É como se a transição tivesse bifurcações, e cada transição de saída da bifurcação tivesse uma condição de guarda.
- ♦ Essa situação pode ser representada em um DTE através de um **ponto de junção**
- ♦ Pontos de junção permitem que duas ou mais transições compartilhem uma “trajetória de transições”.

# Ponto de Junção



# Exemplo (estados de uma publicação)

