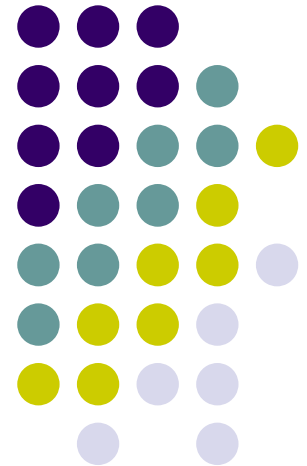


Aula 7:

Modelagem de Estados

(Notação e Aplicação em Nível de Análise)

Prof. Fabiano Cutigi Ferrari
2º semestre de 2022



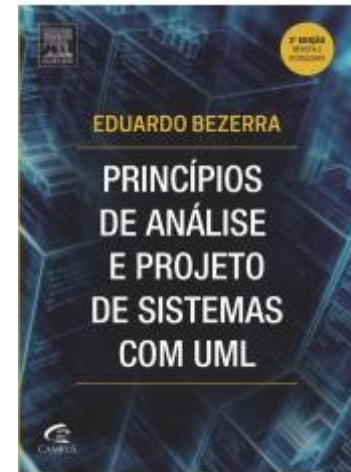
Recados Iniciais



- Mantenham seus e-mails atualizados no Moodle.

Notas Iniciais

- Preparado com base nos materiais a seguir*:
 - Slides disponibilizados em conjunto com o livro
 - Eduardo BEZERRA: Princípios de Análise e Projeto de Sistemas com UML, 3ª ed., Campus/Elsevier (2015).
 - Notas de aula e slides elaborados pelo professor, e outros materiais disponíveis na Web

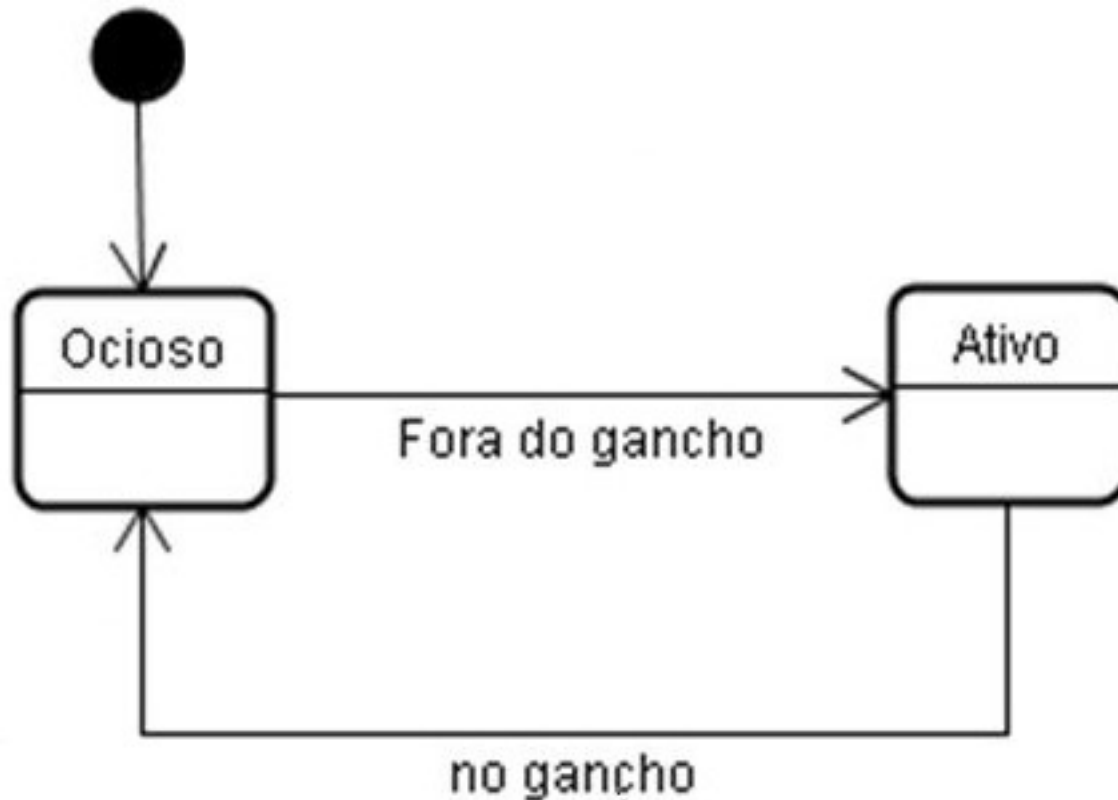


* Notas de rodapé ajudam a identificar os slides produzidos por Bezerra (2015).

- Introdução
- Diagramas de transição de estados
- Identificação dos elementos de um diagrama de estados
- Modelagem de estados no processo de desenvolvimento

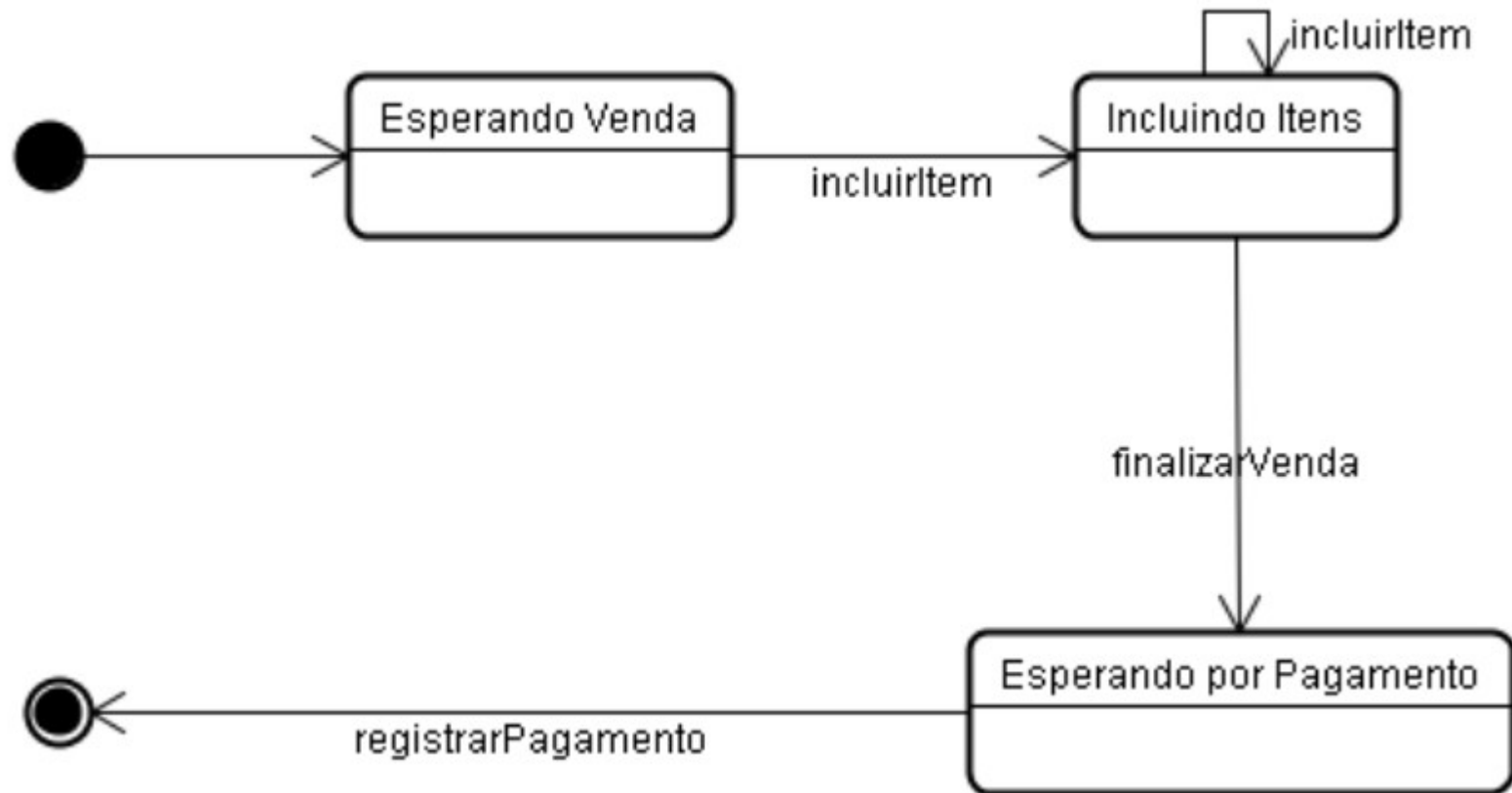
Exemplos Iniciais

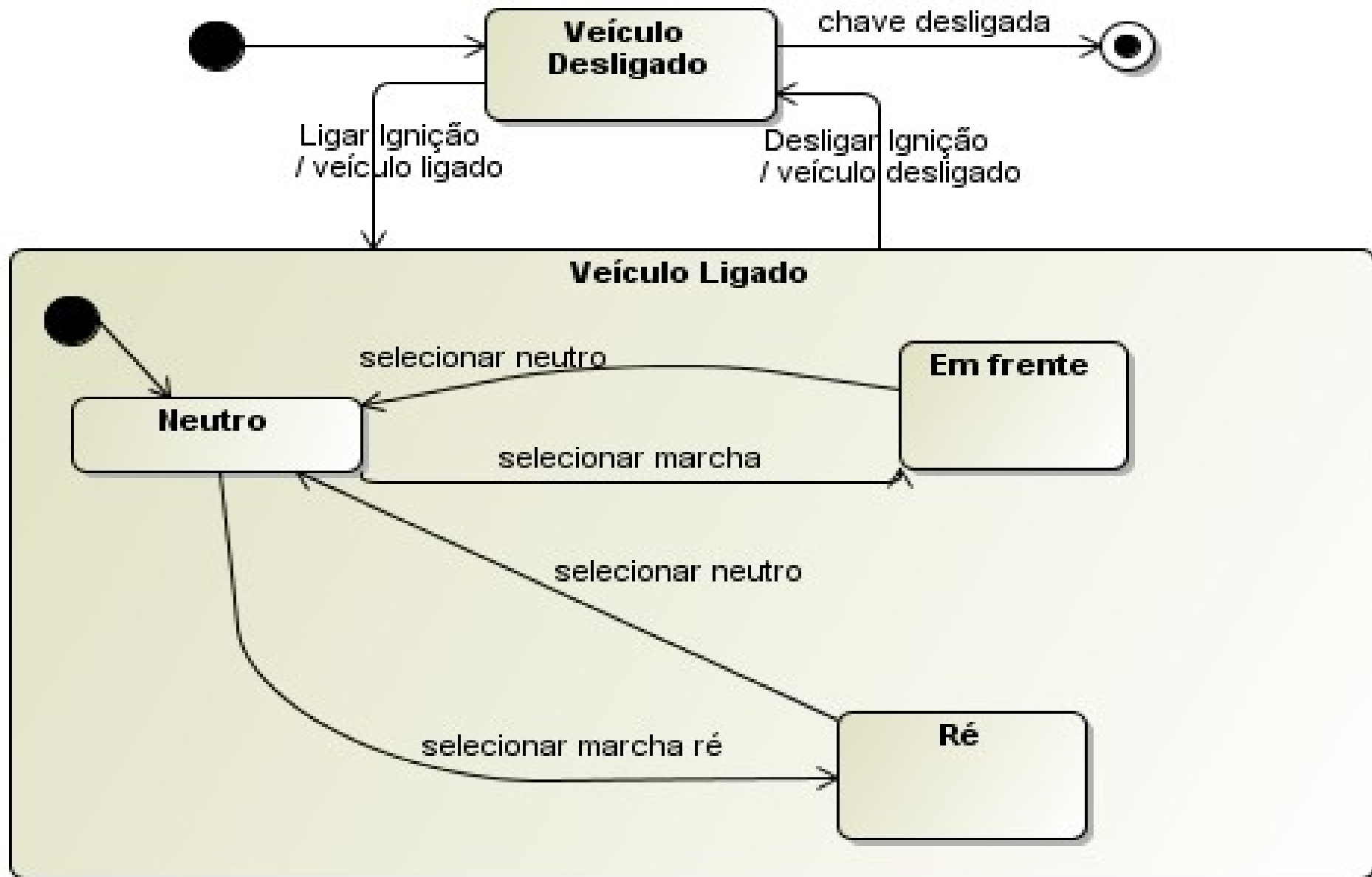
O que queremos representar?



Exemplos Iniciais

O que queremos representar?





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.
- Durante a transição de um estado para outro, um objeto realiza determinadas ações dentro do sistema.
- Quando um objeto transita de um estado para outro, significa que o sistema no qual ele está inserido também está mudando de estado.

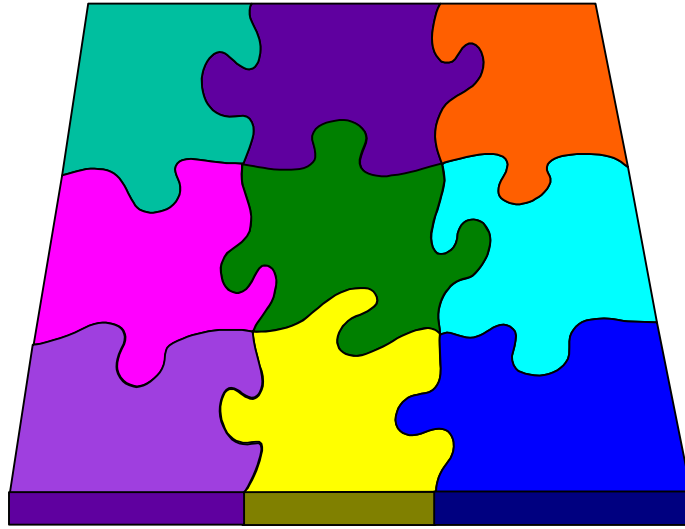


Diagrama de transição de estados

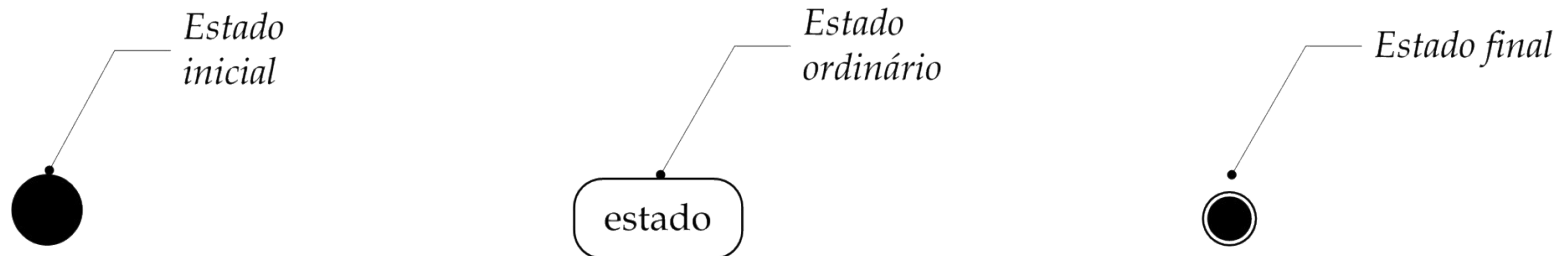
Diagrama de transição de estado

- Através da análise das transições entre estados dos objetos de um sistema de software, podem-se prever todas as possíveis operações realizadas, em função de eventos que possam ocorrer.
- O diagrama da UML que é utilizado para realizar esta análise é o diagrama de transição de estado (DTE).
- A UML tem um conjunto rico de notações para desenhar um DTE.
 - Estados
 - Transições
 - Evento
 - Ação
 - Atividade
 - Estados aninhados (ou subestados, ou estados compostos)
 - Estados concorrentes

- Situação na vida de um objeto em que ele satisfaz a alguma condição ou realiza alguma atividade. Caracteriza-se pelos valores dos atributos e (ou) pelas ligações com outros objetos.
 - O atributo *reservado* deste objeto livro tem valor verdadeiro.
 - Uma conta bancária passa para o *vermelho* quando o seu saldo fica *negativo*.
 - Um professor está *licenciado* quando não está ministrando curso algum durante o semestre.
 - Um tanque está *na reserva* quando nível de óleo está abaixo de 20%.
 - Um pedido está *atendido* quando todos os seus itens estão atendidos.
- Estados podem ser vistos como uma abstração dos atributos e associações de um objeto.

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 indica o fim do ciclo de vida de um objeto.
 - É opcional e pode haver mais de um estado final em um DTE.
- Notação da UML para estados:



Transições

- Os estados estão associados a outros pelas transições.
- Uma transição é mostrada como uma linha conectando estados, com uma seta apontando para um dos estados.
 - Auto-transição é um transição que conecta um estado a ele próprio.
- 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] / ação

- Uma transição possui um evento associado.
- Um evento é algo que acontece em algum ponto no tempo e que pode modificar o estado de um objeto:
 - Pedido realizado
 - Fatura paga
 - Cheque devolvido
- Os eventos relevantes a um sistema de software podem ser classificados em nos seguintes tipos:
 - **Evento de chamada:** recebimento de uma mensagem de outro objeto.
 - **Evento temporal:** passagem de um intervalo de tempo predefinido.
 - **Evento de mudança:** uma condição que se torna verdadeira.

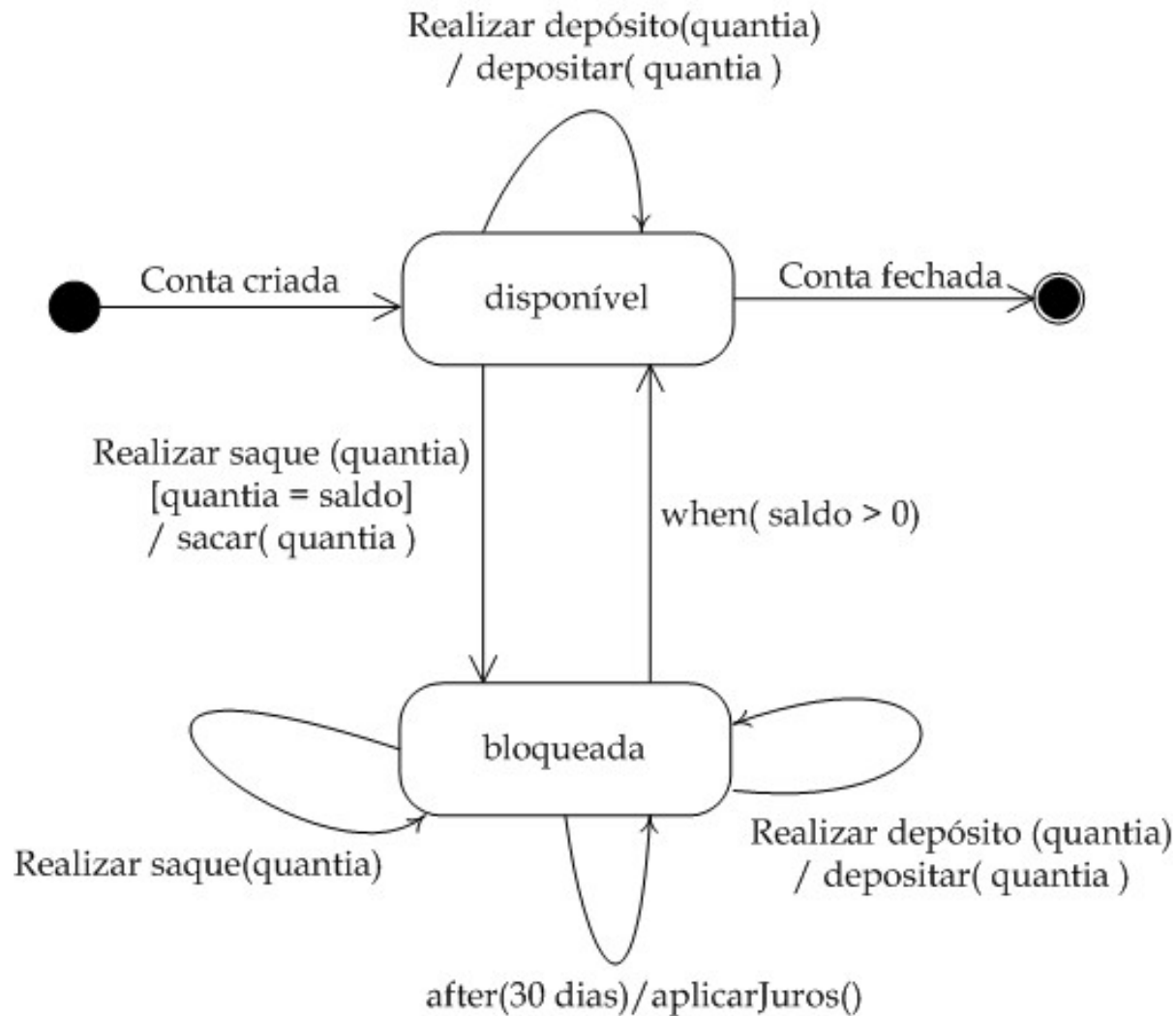
Tipos Mais Comuns de Eventos

- Evento de chamada
 - Corresponde ao envio de uma mensagem ao objeto (invocação de um método).
- Evento temporal
 - Corresponde à passagem de um intervalo de tempo predefinido.
 - O objeto pode interpretar a passagem de um certo intervalo de tempo como sendo um evento.
 - É especificado com a cláusula **after** seguida de um parâmetro que especifica um intervalo de tempo.
 - *after(30 segundos)*: indica que a transição será disparada 30 segundos após o objeto ter entrado no estado atual.

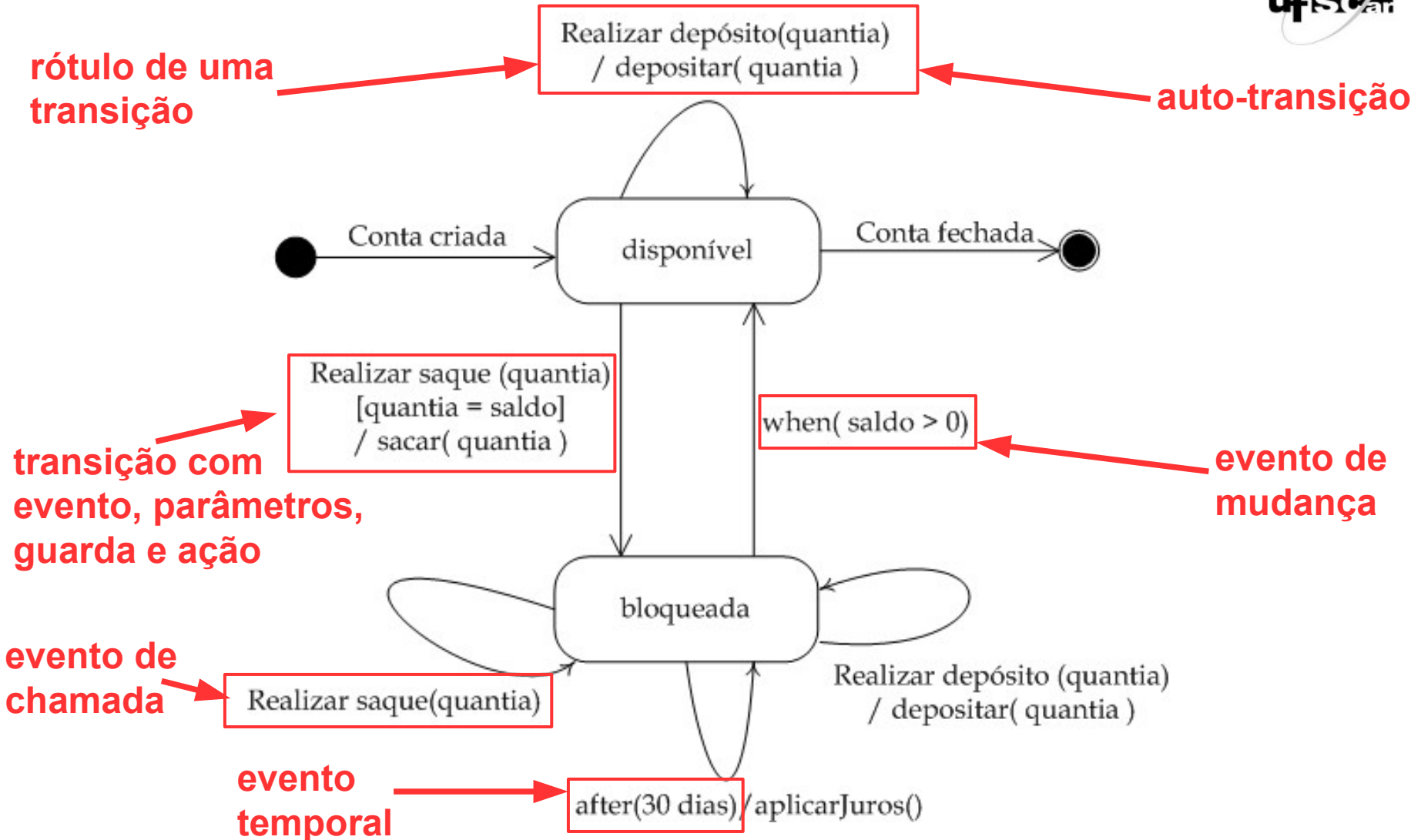
Tipos Mais Comuns de Eventos (cont.)

- Evento de mudança
 - Corresponde a uma condição que se torna verdadeira.
 - É representado por um predicado utilizando-se a cláusula **when**.
 - *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)*

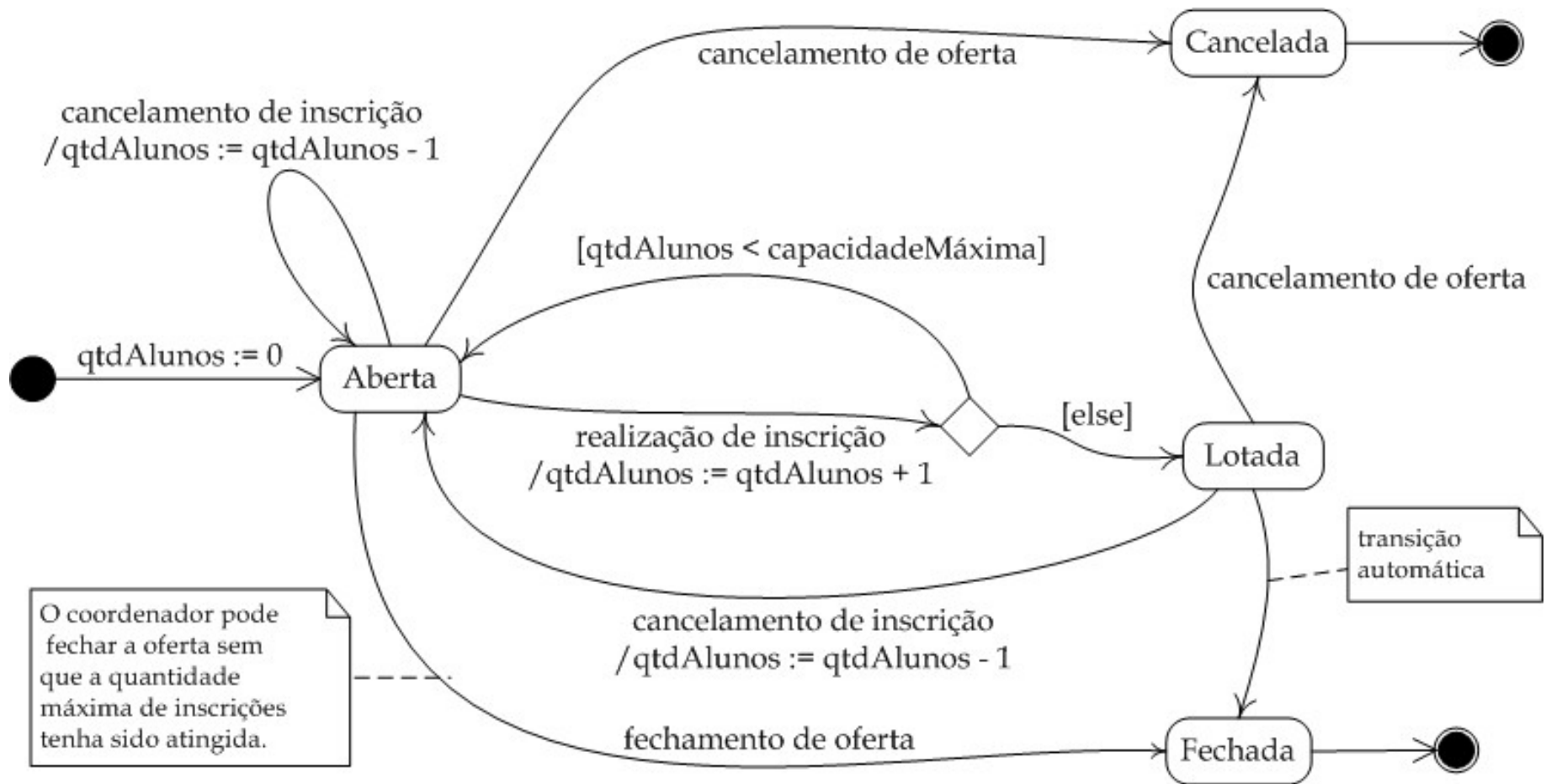
Exemplo (Conta Bancária)



Exemplo (Conta Bancária)



Exemplo (Oferta de Disciplina)



Guarda (ou Condição de Guarda)

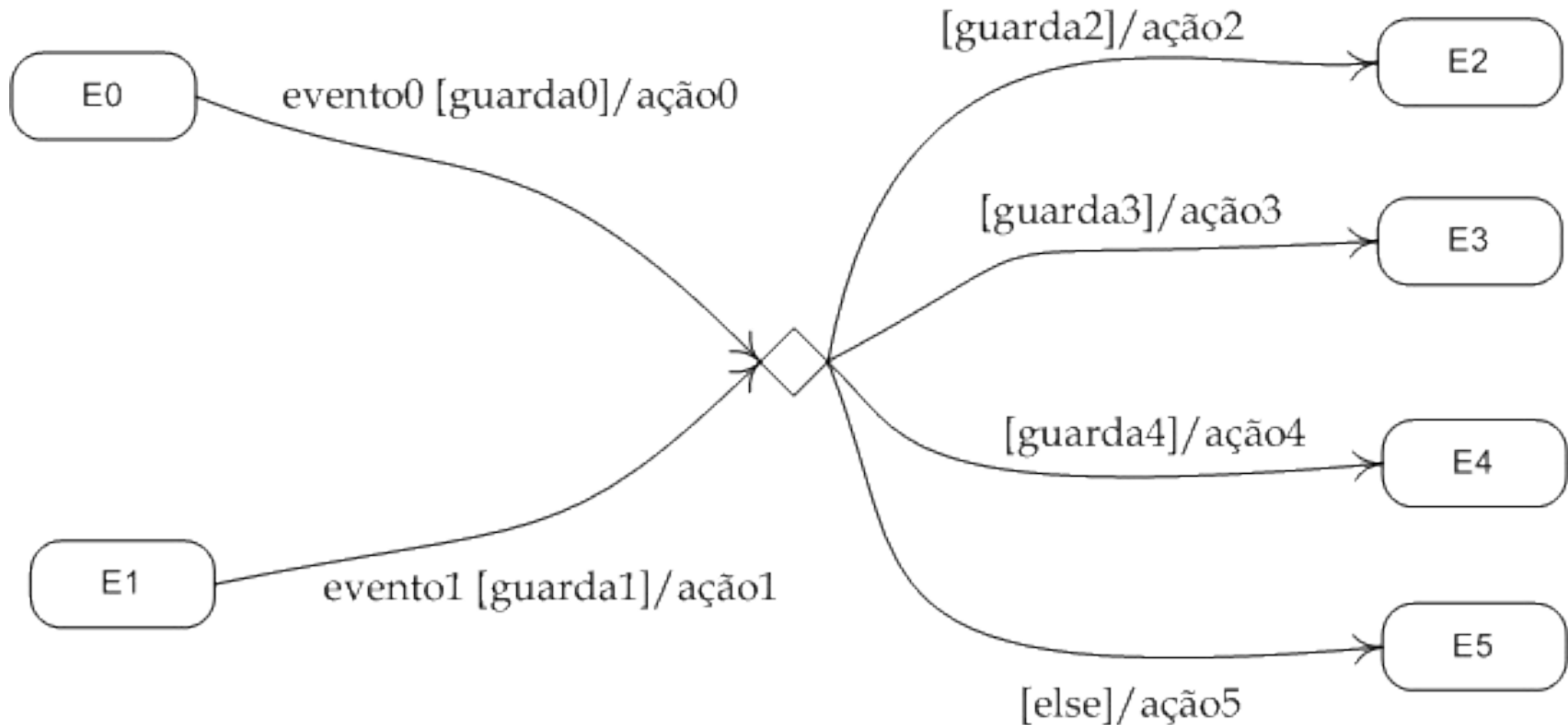
- É uma expressão de valor lógico (predicado) que condiciona o disparo de uma transição.
- A transição correspondente é disparada se e somente se o evento associado ocorre e a condição de guarda é verdadeira.
 - Uma transição que não possui condição de guarda é sempre disparada quando o evento ocorre.
- A condição de guarda pode ser definida utilizando-se parâmetros passados no evento e também atributos e referências a ligações da classe em questão.

- Ao transitar de um estado para outro, um objeto pode realizar uma ou mais ações.
- Uma ação é uma expressão definida em termo dos atributos, operações, associações da classe. Parâmetros do evento também podem ser utilizados.
- A ação associada a uma transição é executada se e somente se a transição for disparada.

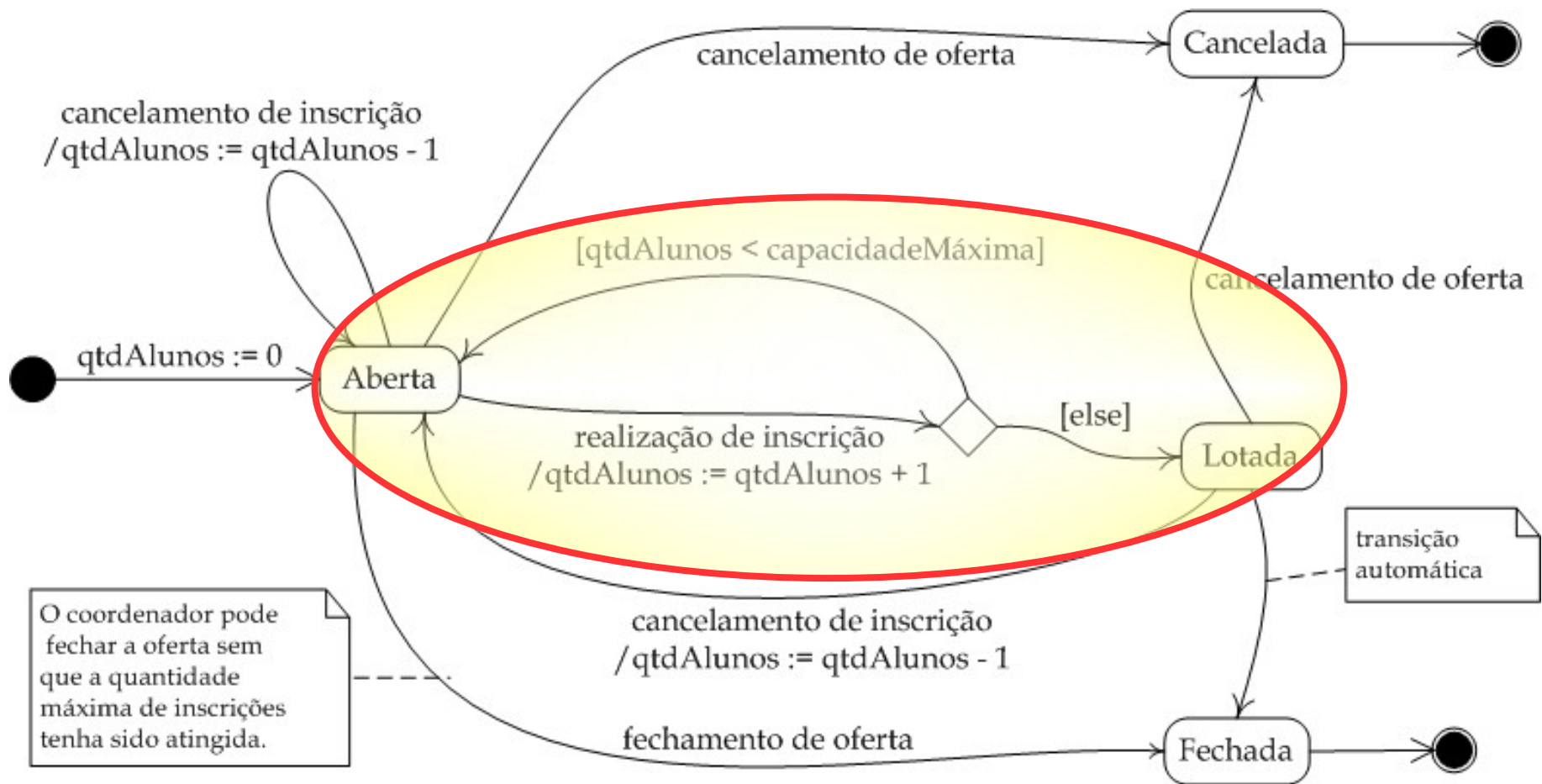
Ponto de Junção

- 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 (else), 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.
- De uma forma geral, pode haver um número ilimitado de transições saindo de um ponto de junção.

Ponto de Junção - Caso Genérico



Exemplo (Oferta de Disciplina)



- No compartimento adicional de um retângulo de estado podem-se especificar ações ou atividades a serem executadas.
- Sintaxe geral:

evento / [ação | atividade]

- Há três cláusulas predefinidas: *entry*, *exit*, *do*

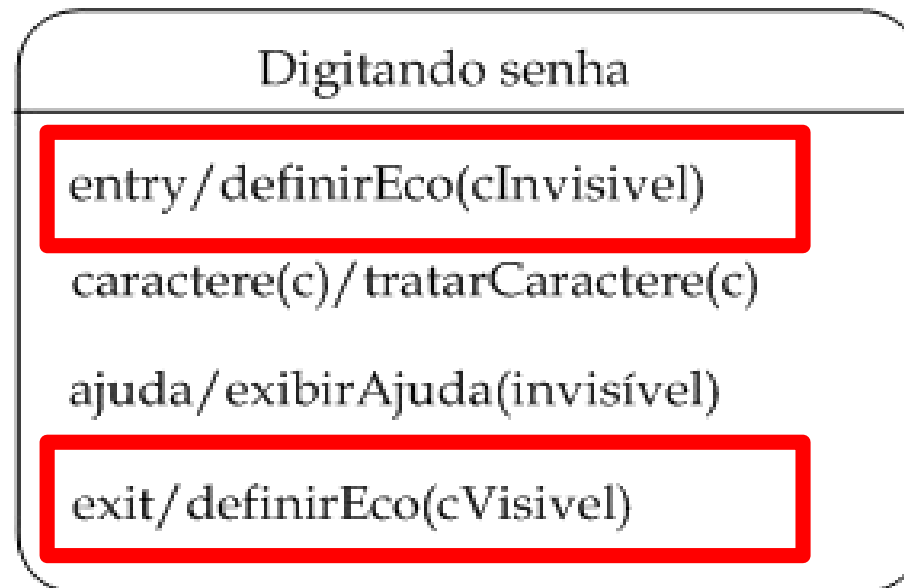
Cláusula *entry*

- Especifica 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.
- 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.

Exemplo: Cláusulas *entry* / *exit*

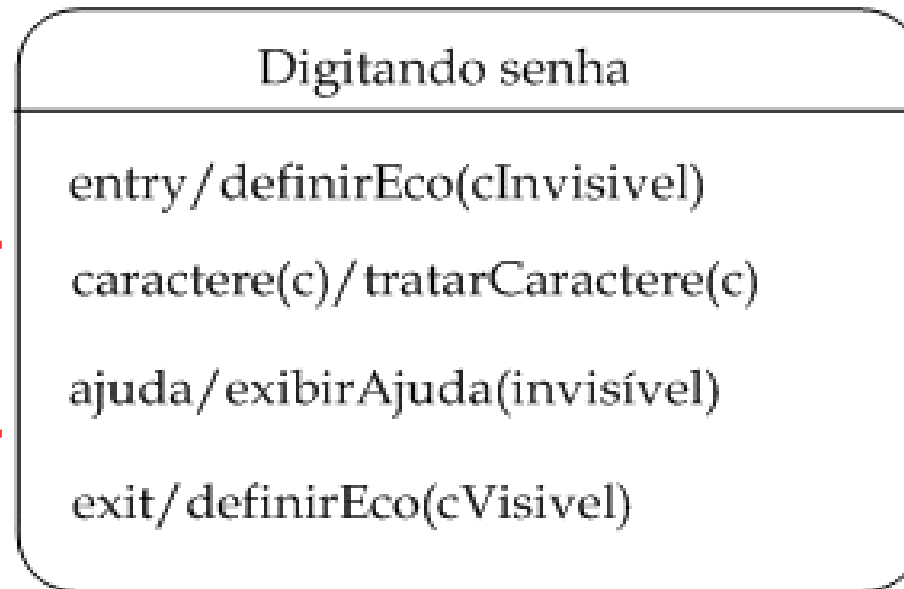


Transições Internas

- Representam um tipo de transição que não dispara ações de entrada ou de saída.

Exemplo: Transição Interna

transições
internas



- Semelhante a uma ação, uma atividade é algo que deve ser executado.
- No entanto, uma atividade pode ser interrompida (uma ação não pode).
 - Por exemplo, enquanto a atividade estiver em execução, pode acontecer um evento que a interrompa.
- Outra diferença: uma atividade sempre está associada a um estado (enquanto uma ação está associada a uma transição).

Cláusula do

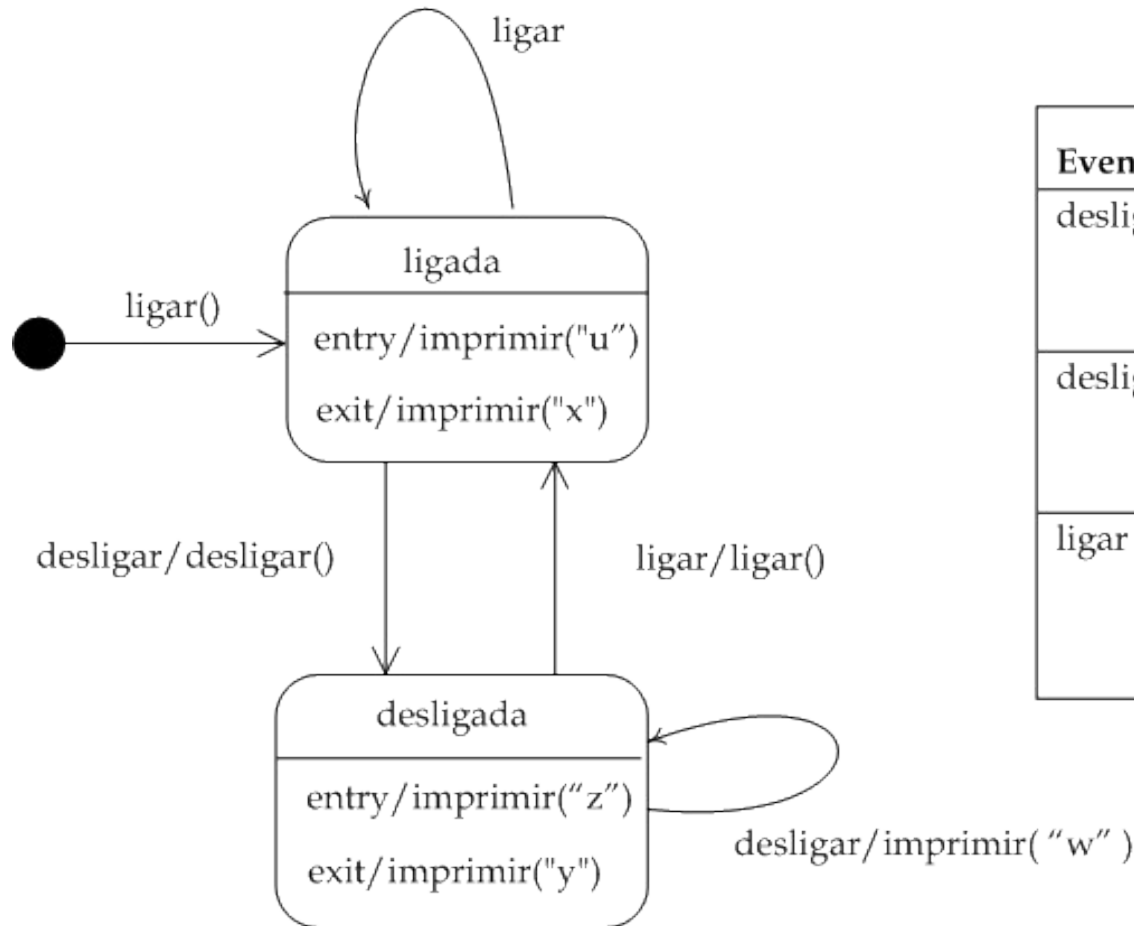
- Usada para definir alguma atividade a ser executada quando o objeto passa para um determinado estado.
- Ao contrário da cláusula entry, serve para especificar uma atividade, em vez de uma ação.
- É útil para disparar transições de forma automática.
 - Nesse caso, a transição não tem um evento nem uma guarda associada.
 - Quando a atividade é concluída, a transição ocorre automaticamente.

Exemplo: Cláusula *do*, Atividade e Transição automática



OBS: a inexistência de um evento na transição Estado2 -> Estado3 indica que a transição ocorrerá automaticamente quando a atividade2 for finalizada

Outros Exemplos

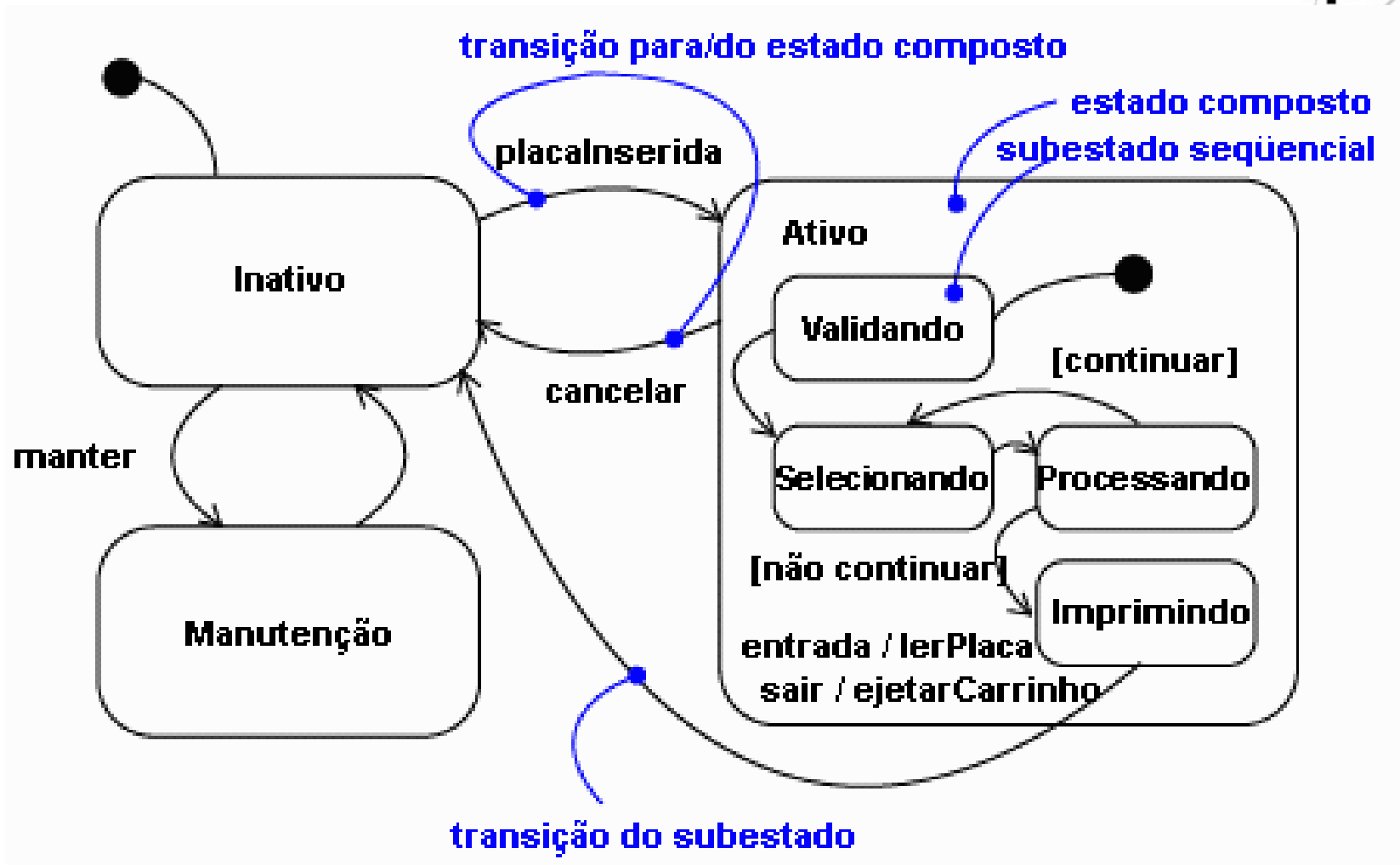


Evento	Ações executadas
desligar	imprimir("x") desligar() imprimir("z")
desligar	imprimir("y") imprimir("w") imprimir("z")
ligar	imprimir("y") ligar() imprimir("u")

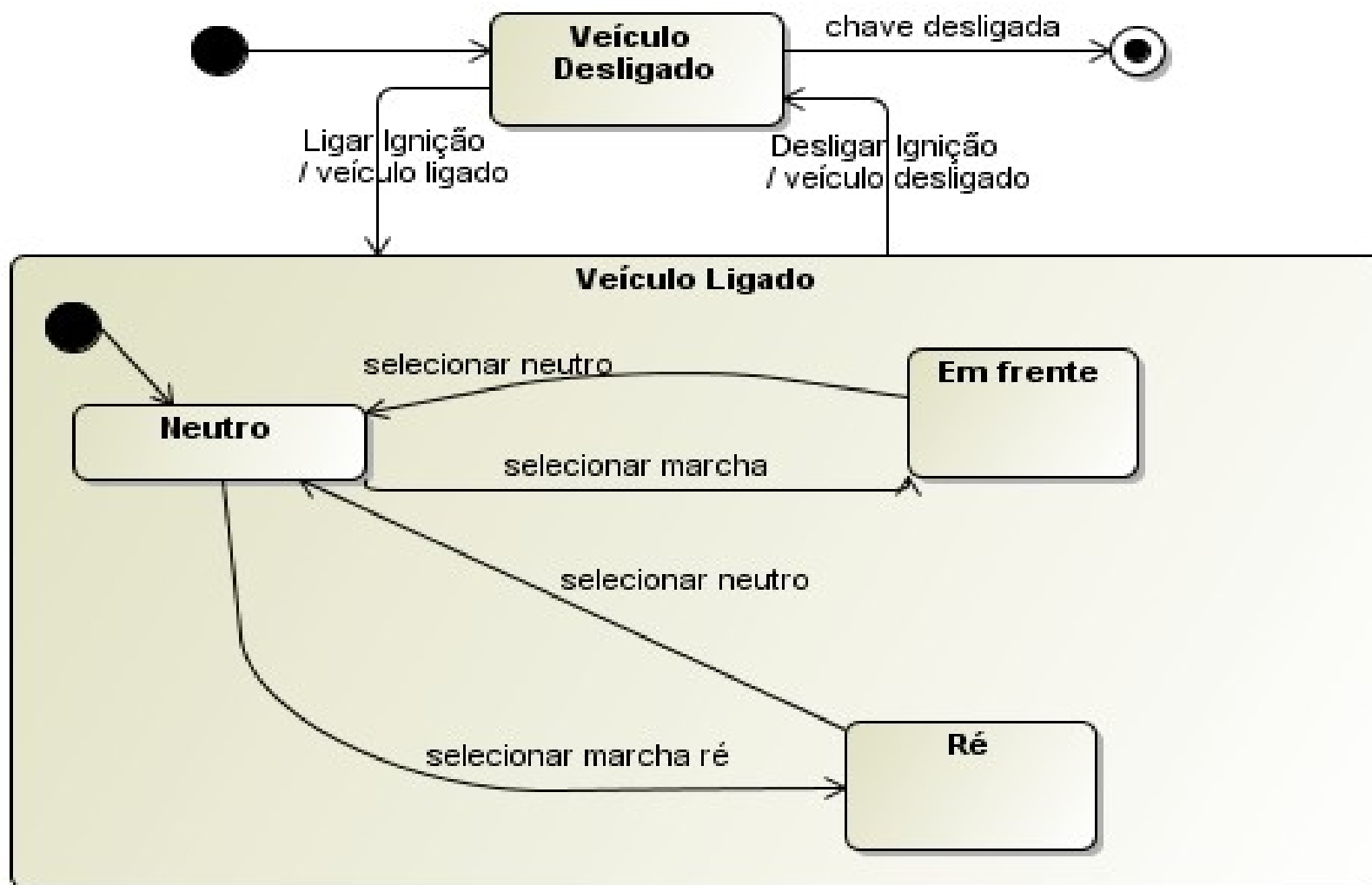
Estados Aninhados (ou Subestados)

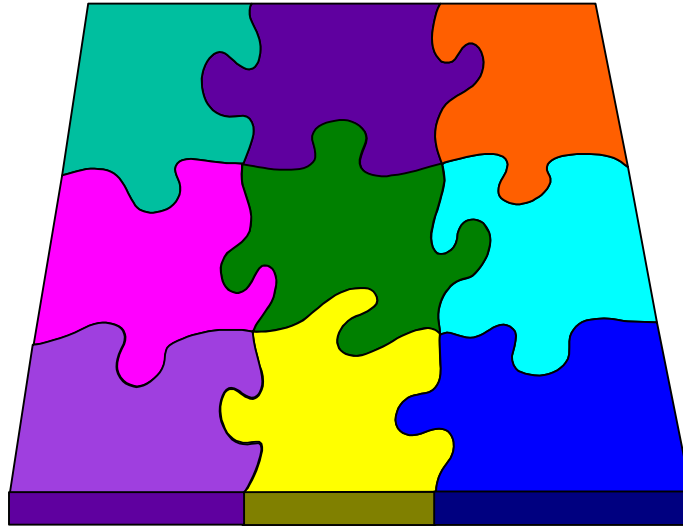
- São partes de um estado composto.
- São usados para simplificar máquinas complexas de estados simples.
 - Mostram que alguns estados são possíveis apenas dentro de um determinado contexto (o estado confinado).
- Podem ser aninhados em qualquer nível.

Estados Aninhados: Exemplos



Estados Aninhados





**Identificação dos elementos de um
diagrama de estados**

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.
- De uma forma geral, cada operação com visibilidade pública de uma classe pode ser vista como um evento em potencial.

Identificação de elementos do DTE

- 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".

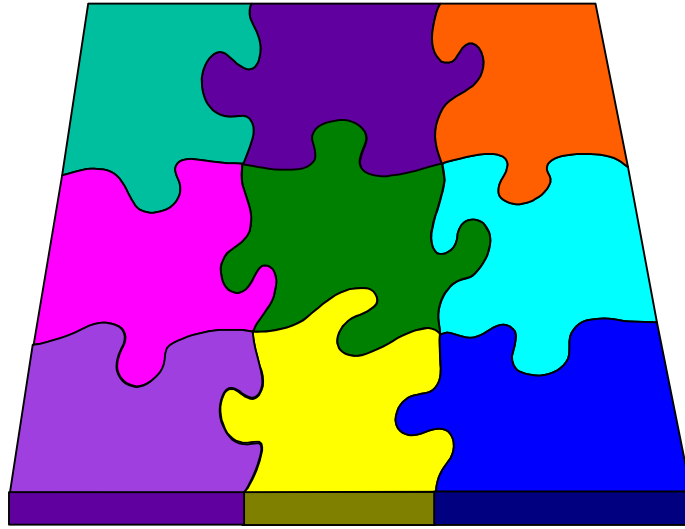
"O número máximo de alunos por curso é igual a 30".

Para quais classes construímos o DTE?

- Os 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.

Procedimento para construção de um DTE

- 1) Identifique os estados relevantes para a classe.
- 2) Identifique os eventos relevantes. Para cada evento, identifique qual a transição que ele ocasiona.
- 3) Para cada estado: identifique as transições possíveis quando um evento ocorre.
- 4) Para cada estado, identifique os eventos internos e ações correspondentes.
- 5) Para cada transição, verifique se há fatores que influenciam no seu disparo (definição de condições de guarda e ações).
- 6) Para cada condição de guarda e para cada ação, identifique os atributos e ligações que estão envolvidos.
- 7) Defina o estado inicial e os eventuais estados finais.
- 8) Desenhe o DTE.



Modelagem de estados no processo de desenvolvimento

Modelagem de Estados no PDS



- A construção de um DTE frequentemente leva à descoberta de novos atributos para uma classe
 - principalmente atributos para servirem de abstrações para estados.
- Além disso, este processo de construção permite identificar novas operações na classe.
 - Os objetos precisam reagir aos eventos que eles recebem.
- Essas novas propriedades devem ser adicionadas ao modelo de classes.

Referências

- BEZERRA, E.: Princípios de Análise e Projeto de Sistemas com UML, 3ª edição, Campus - Elsevier, 2015.
- http://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/statechart_diagram_640B5D0B.html#Internal%20Transitions