

Análise Orientada a Objetos



Modelagem Dinâmica

Introdução

- ❑ Modelos de casos de uso fornecem uma visão das funcionalidades que o sistema deve prover, ou seja uma visão comportamental do sistema.
- ❑ O modelo conceitual estrutural (diagramas de classe) definem os tipos de entidades e relacionamentos que o sistema deve representar para prover as funcionalidades previstas nos C.U.

Comportamento dinâmico

- ❑ Os diagramas de classes representam apenas elementos estáticos, dados;
- ❑ É preciso, ainda, representar o comportamento da aplicação em função do tempo e de eventos específicos;
- ❑ Modelar o comportamento:
 - Indica como o sistema irá responder a eventos ou estímulos externos;
 - Auxilia o processo de descoberta das operações das classes do sistema.

Tipos de Requisições

- Uma requisição é uma solicitação para que um sistema realize uma ação. A representação de como essas ações afetam o estado do domínio é o foco da modelagem dinâmica.
- **Requisição explícita:** é iniciada explicitamente por um ator (externa) ou por outra ação (induzida).
- **Requisição temporal:** é iniciada pela passagem do tempo.
- **Requisição gerada:** é iniciada quando uma condição de geração é satisfeita.

Tipos de Requisições

- ❑ A maioria das requisições são externas.
- ❑ Requisições externas são classificadas em 2 tipos:
 - **Consulta:** requisição externa que provê alguma informação para o ator e não alteram a base de informações do sistema.
 - **Notificação de evento de domínio:** requisição externa cujo efeito é uma mudança na base de informações do sistema, correspondendo **a um evento de domínio**

Modelos da UML

- ❑ Diagrama de gráficos de Estados:
 - Descrevem os estados possíveis pelos quais um particular objeto pode passar e suas transições, estímulos e atividades.
- ❑ Diagrama de Atividades
 - Mostra o fluxo de uma atividade para outra em um sistema.
- ❑ Diagramas de Interação:
 - Descrevem como grupos de objetos colaboram entre si em um certo comportamento.

Análise Orientada a Objetos



Modelagem Dinâmica
Diagramas de Estados

Introdução

- ❑ Um estado é uma situação na vida de um objeto durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou aguarda a ocorrência de um evento
- ❑ Classes com estados (ou classes modais) são classes cujas instâncias podem mudar de um estado para outro ao longo de sua existência, mudando sua estrutura, seus valores de atributos ou comportamento dos métodos.
- ❑ Diagramas de estados são utilizados para especificar o comportamento de classes modais

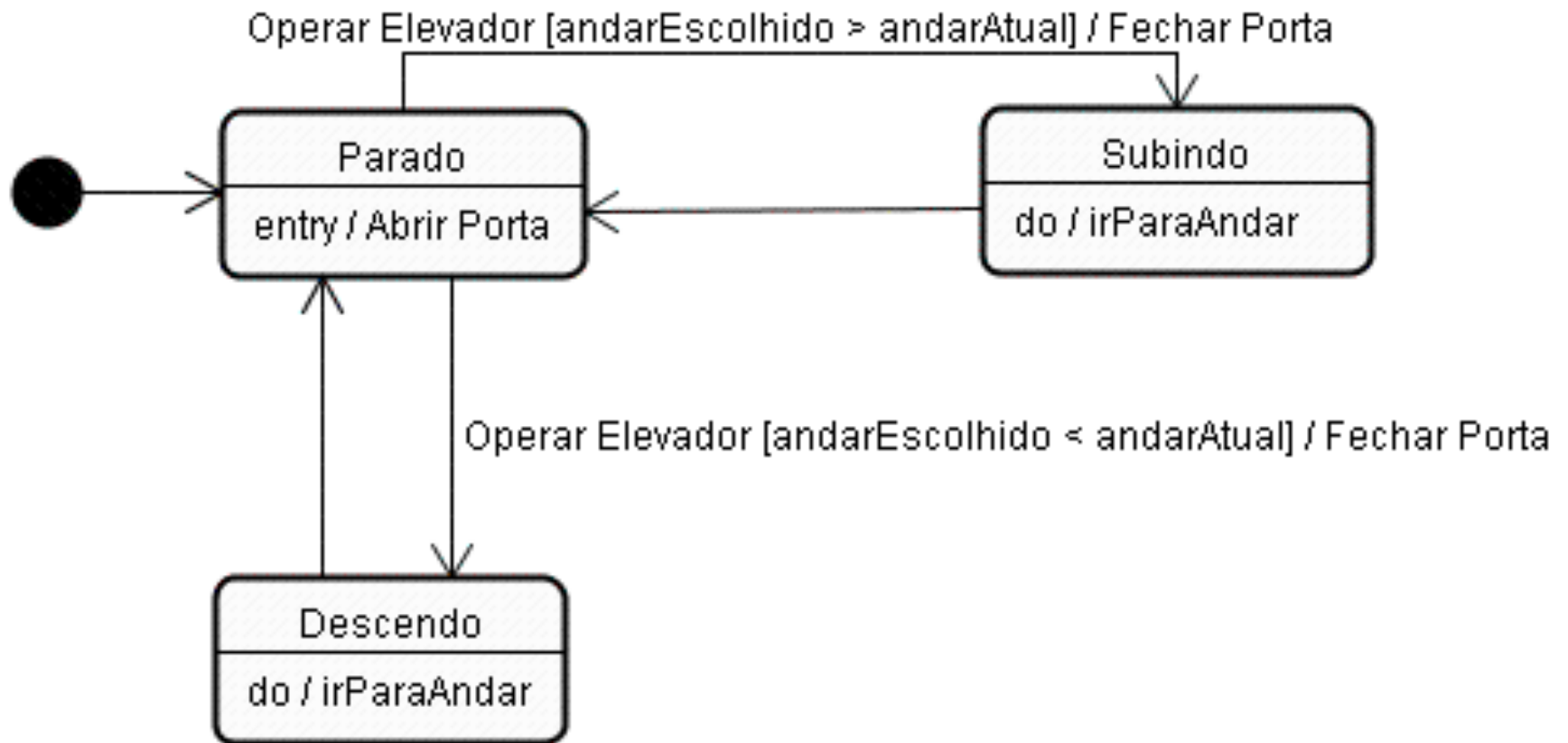
Conceitos básicos

- ❑ Todas as instâncias de uma classe modal se comportam da mesma maneira;
- ❑ Um diagrama de estados é um construído para uma classe modal;
- ❑ Cada instancia da classe estará em um único estado em um dado momento.
- ❑ Um estado é relevante quando ajuda a definir restrições ou efeitos de eventos;

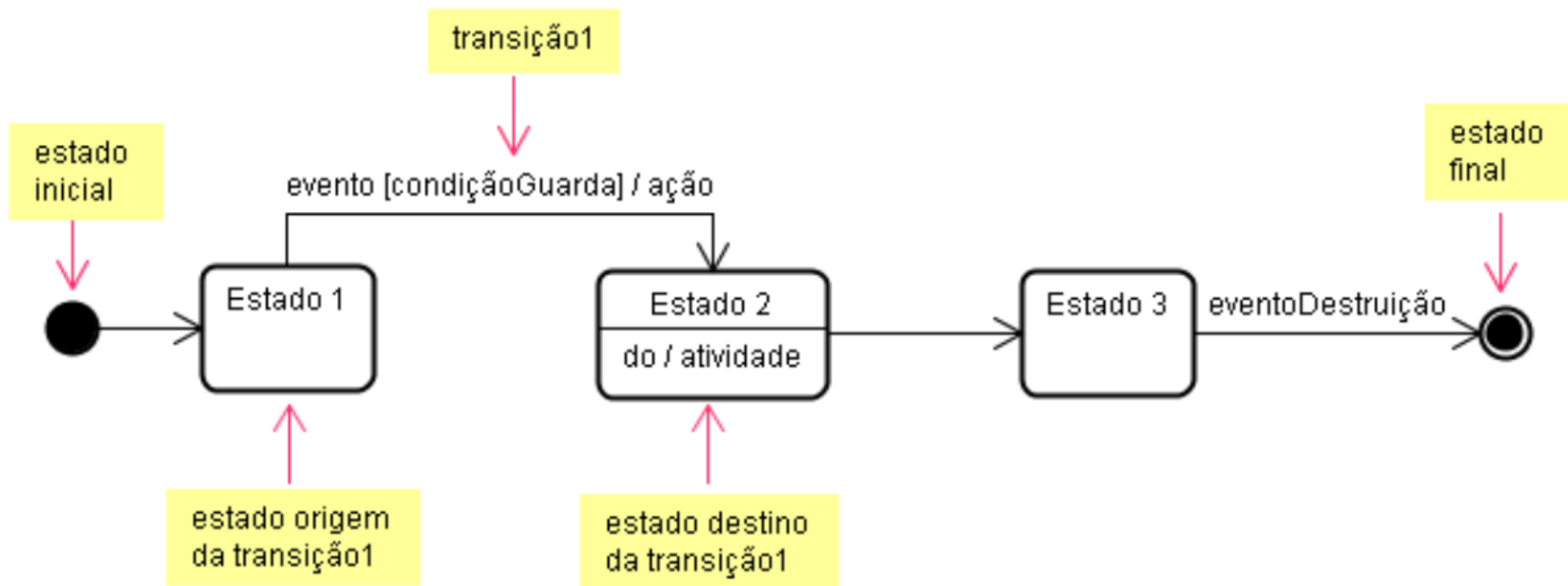
Conceitos básicos

- ❑ Diagramas de estados descrevem os possíveis estados pelos quais objetos da classe podem passar e as alterações dos estados como resultado de eventos (estímulos) que atingem esses objetos;
- ❑ Diagramas de estados especificam a ordem válida dos estados pelos quais os objetos de uma classe podem passar, bem como os eventos que alteram os estados

Exemplo

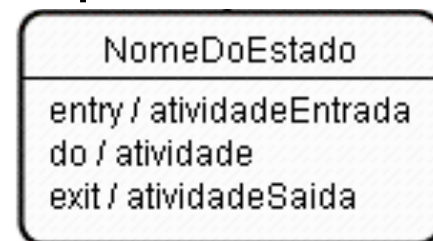


Notação

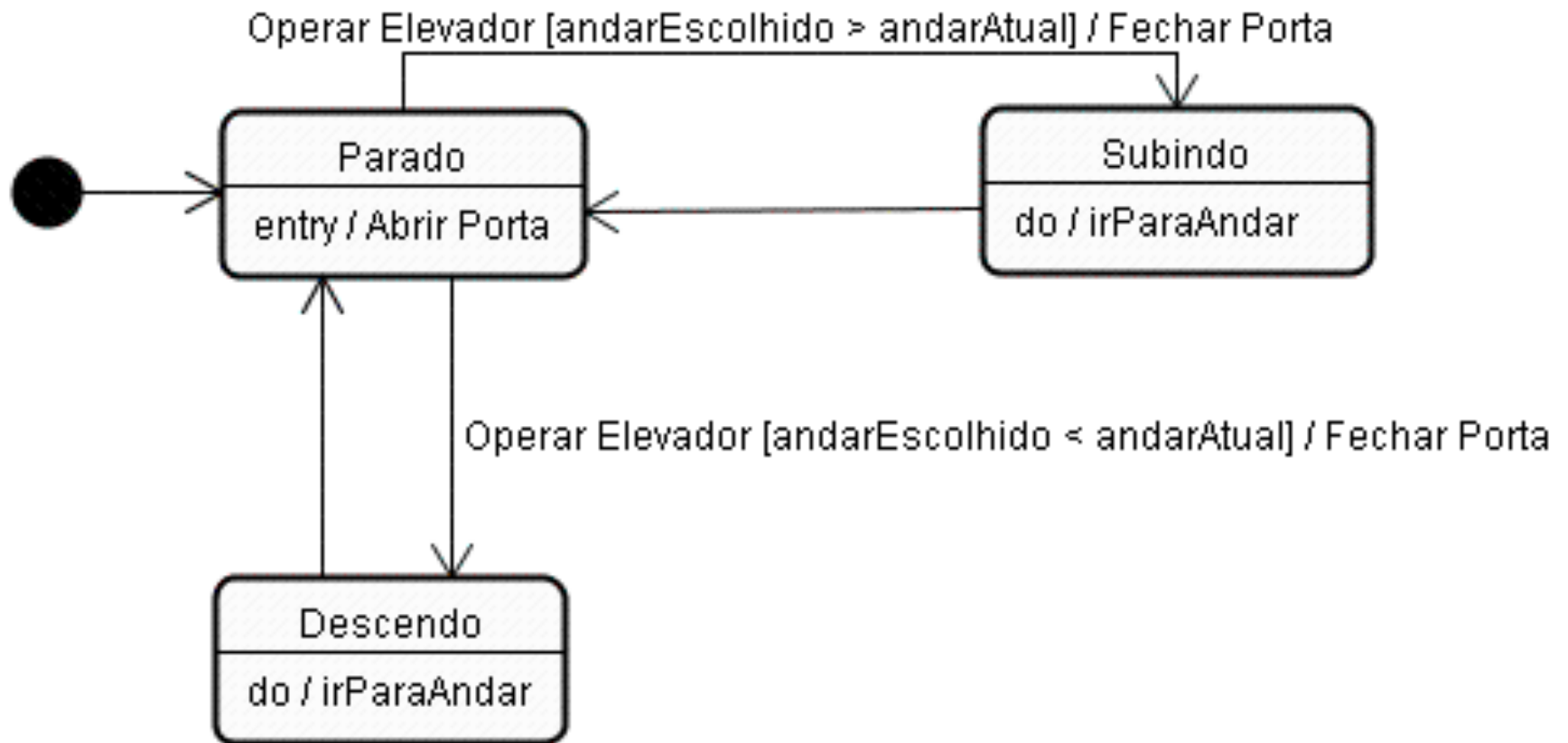


Representando Estados

- ❑ Nome: deve descrever claramente o estado e deve ser único em um diagrama;
 - *"o <<objeto>> está <<nome do estado>>"*
 - *"o <<objeto>> está no estado <<nome do estado>>"*
- ❑ Atividades:
 - De entrada: ocorrem ao entrarmos no estado;
 - De saída: ocorrem ao saírmos do estado;
 - Convencional (*do*): ocorrem enquanto o objeto estiver naquele estado.

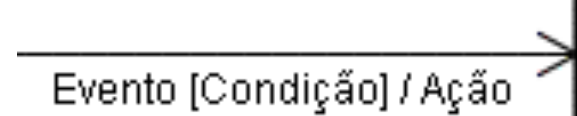


Exemplo



Representando Transições

- Indicam a possibilidade de ir de um estado a outro;



 - Evento: evento externo que motivou a transição;
 - Condição de guarda: condição necessária para efetuar a transição (além do evento);
 - Ação: ação realizada durante a transição.
- Quando o *evento* ocorre, se a *condição de guarda* é verdadeira, a *transição* dispara e a *ação* é realizada instantaneamente. O objeto muda de estado. Se o estado destino possuir uma *atividade* a ser realizada, ela é iniciada.

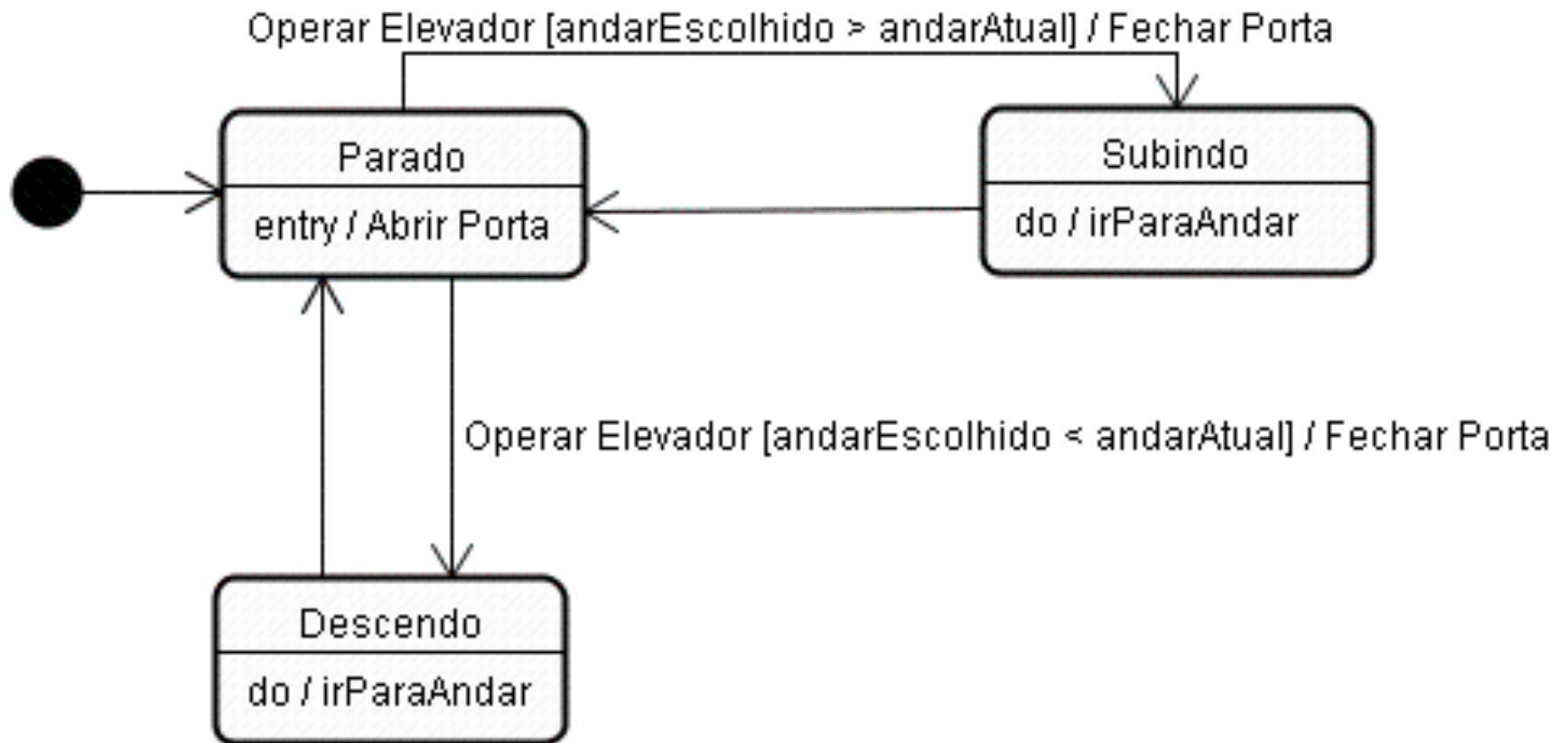
Representando Transições

- ❑ Evento implícito: transição não tem evento associado. Ocorre quando:
 - O evento implícito é a conclusão da atividade do estado de origem (ex.: transição de “subindo” para “parado” no exemplo do elevador).
 - O evento implícito é temporal (ex.: após três dias na fase pendente de pagamento o pedido é cancelado)
 - O evento implícito torna a condição de guarda verdadeira (ex.: modela apenas a condição de guarda “estoque atual < estoque mínimo” sem deixar explícito o evento que reduz o estoque atual).


Ação x Atividade

- ❑ Ação x atividade convencional:
 - Uma ação ocorre durante a transição (ela é atômica);
 - Uma atividade ocorre enquanto o objeto permanece no estado (quando ela acabar, ele muda de estado);
- ❑ Ação x atividades de entrada/saída:
 - Opta-se por representar como entrada/saída quando é executado sempre, independente da transição
 - Caso só haja uma transição de entrada/saída, não há diferença na prática.

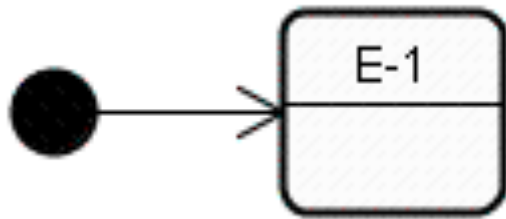
Exemplo



Estado inicial

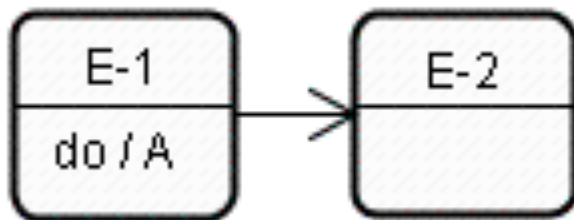
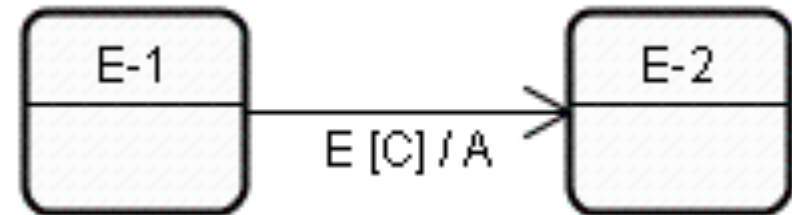
- ❑ Estado inicial é representado por 
- ❑ Deve haver apenas um estado inicial
- ❑ Os estados ligados ao estado inicial são aqueles nos quais o objeto pode estar logo quando for construído;
- ❑ Pode ter condições de guarda nas transições de saída de um estado inicial mas deve-se garantir que ao menos uma das condições de saída será disparada. (ex.: vendas abaixo de R\$1000 são criadas como “pendente de entrega” enquanto vendas maiores “pendente de aprovação”).

Exemplos




Ao ser criado, o objeto da classe em questão encontra-se no estado E-1.

Se o objeto estiver no estado E-1 e ocorrer o evento E, se a condição C = true, ocorre a ação A e o objeto passa para o estado E-2.



Enquanto estiver no estado E-1, o objeto realiza a atividade A. Ao terminar a atividade A, automaticamente ele muda para o estado E-2.

Estados final

- ❑ O estado final é utilizado para representar que o objeto deixou de existir
- ❑ Representado por 
- ❑ Não é obrigatório ter e pode ter mais de um;
- ❑ As transições para o estado final definem em quais estados é possível excluir um objeto.

Como identificar estados

- ❑ Somente para algumas classes:
 - Que possuam comportamento variável no tempo – em cada estado se comportam de maneira diferente;
 - Nas quais seja possível identificar ao menos três estados distintos;
 - Cujo passado influencia no seu comportamento atual;

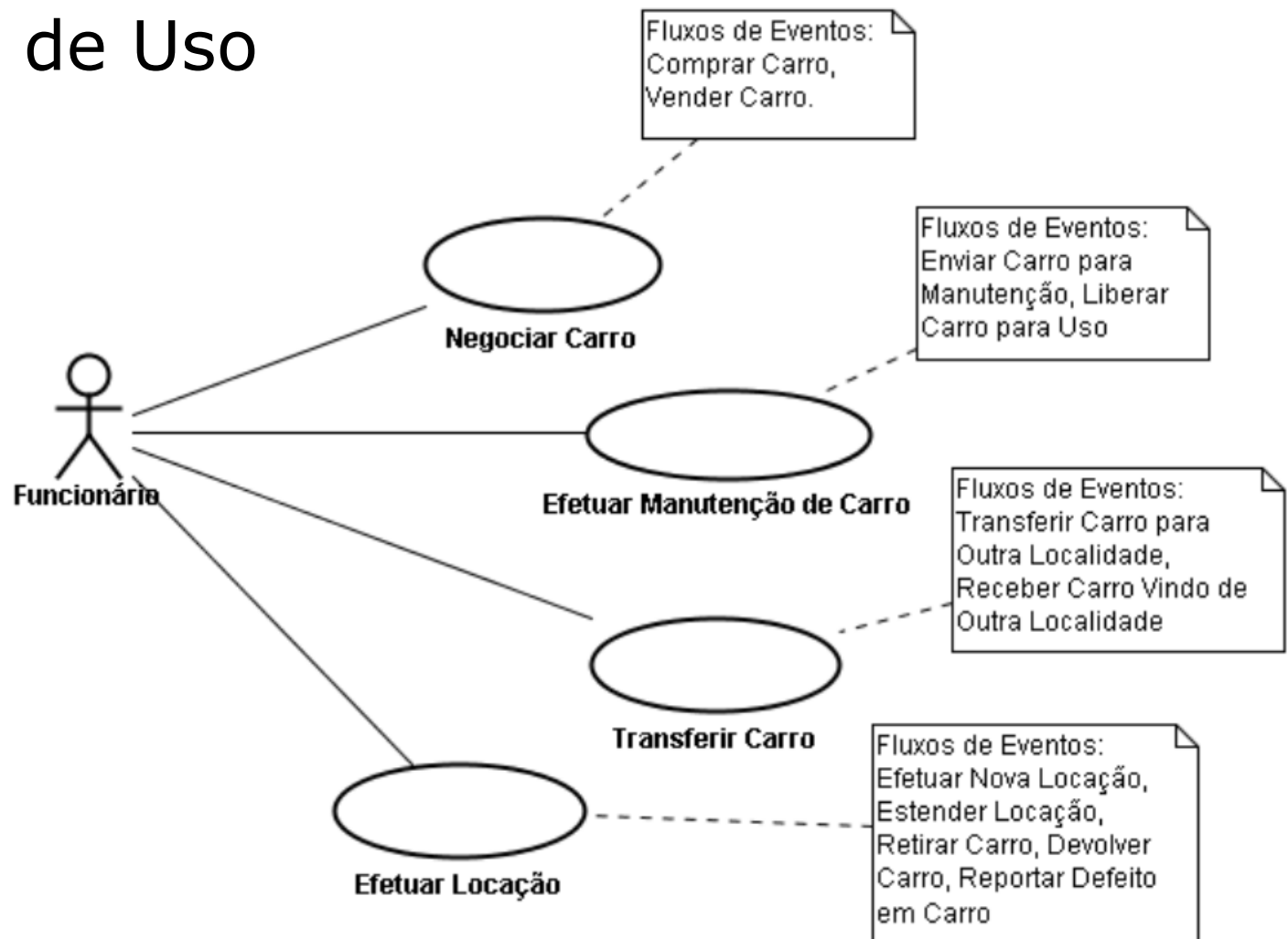
Um estado é uma situação durante a vida de um objeto durante a qual ele satisfaz alguma condição, realiza alguma atividade ou aguarda algum evento.

Como identificar transições

- Dado um estado, verificamos para quais outros estados ele pode passar e o evento em que isso ocorre;
- **Na grande maioria das vezes, os casos de uso (ou fluxos de caso de uso) são os eventos que acionam as transições;**
- Após identificadas, avalie condições de guarda e ações.

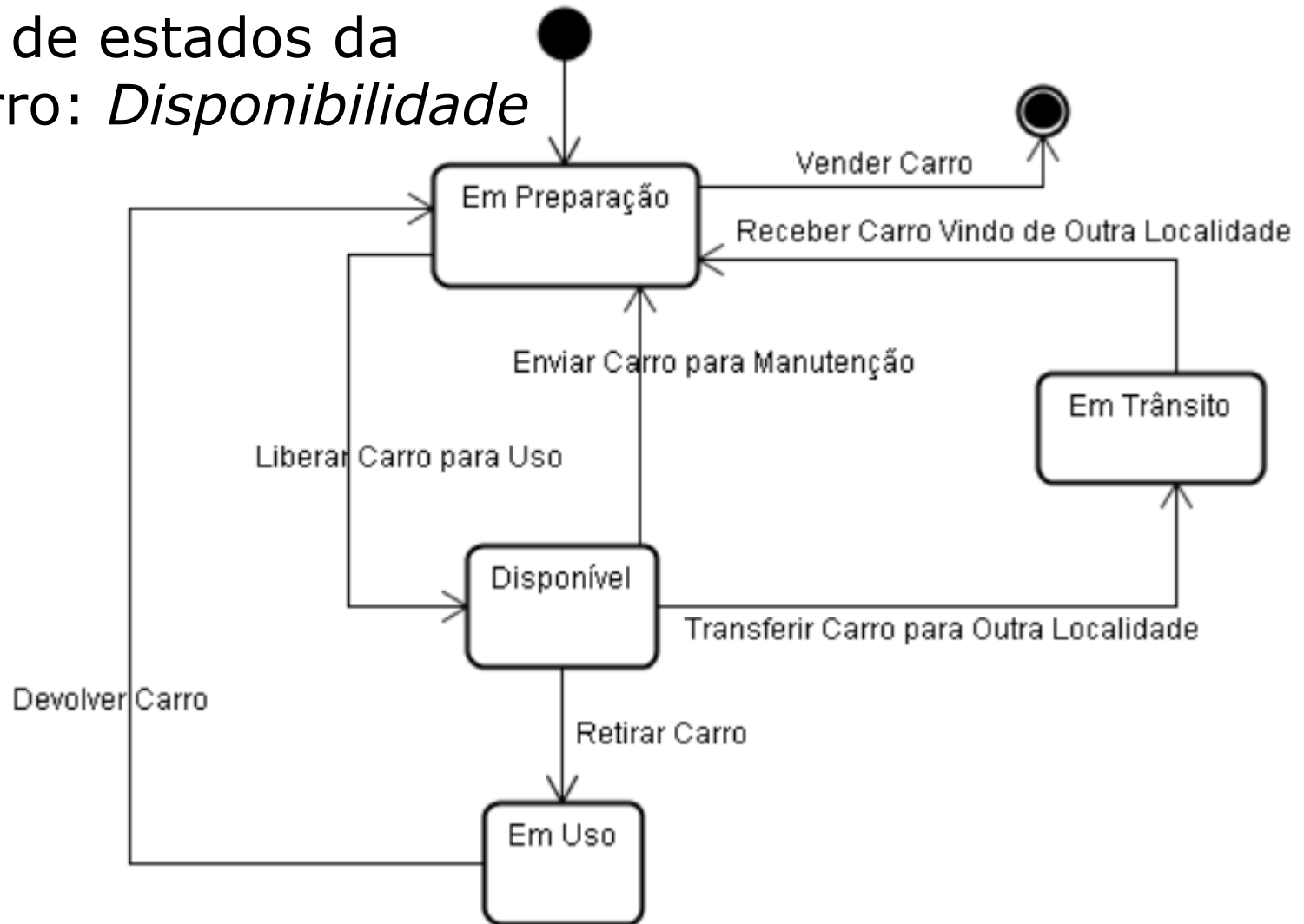
Exemplo: Locadora de Automóveis

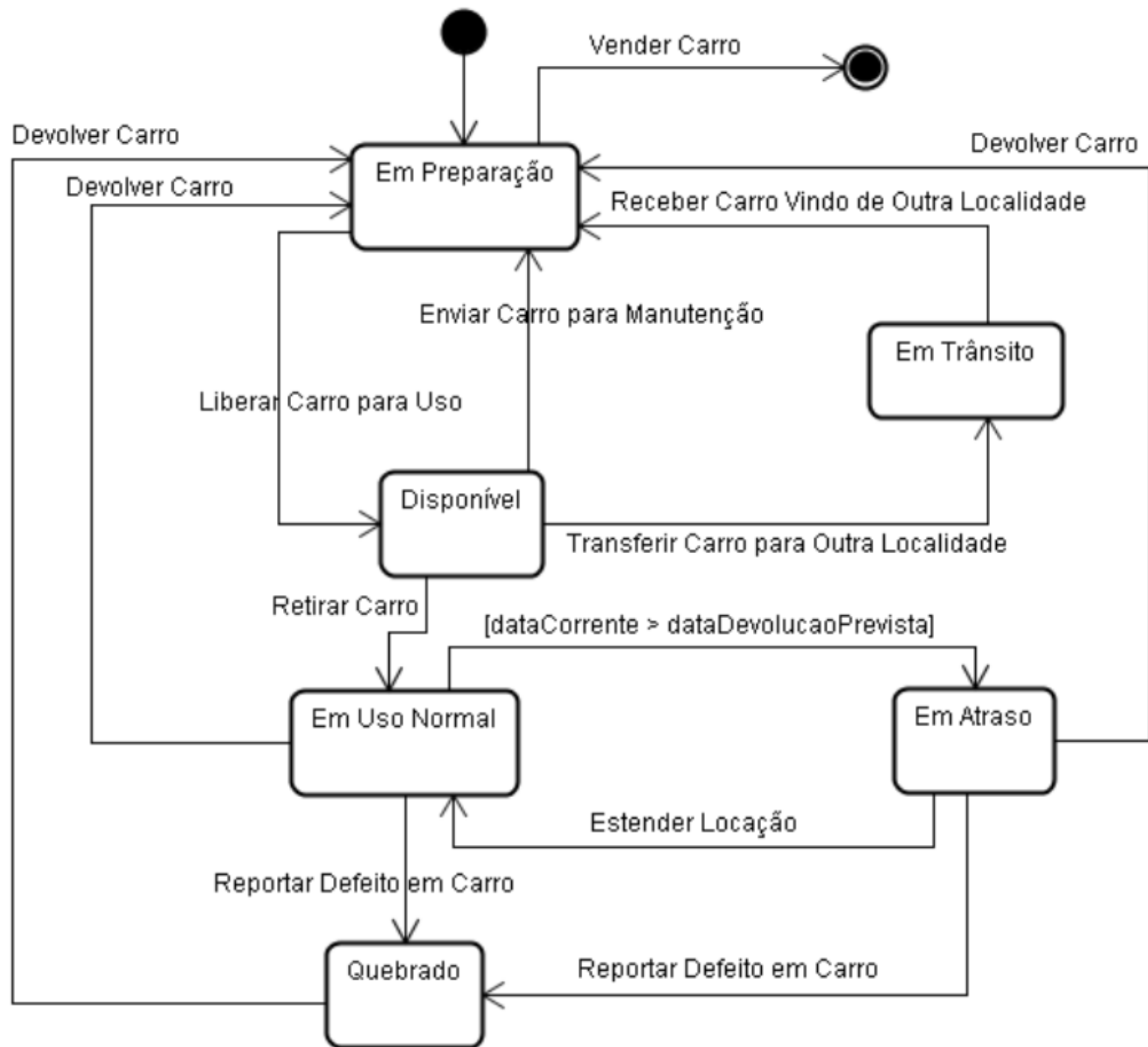
■ Casos de Uso



Exemplo: Locadora de Automóveis

Diagrama de estados da classe Carro: *Disponibilidade*

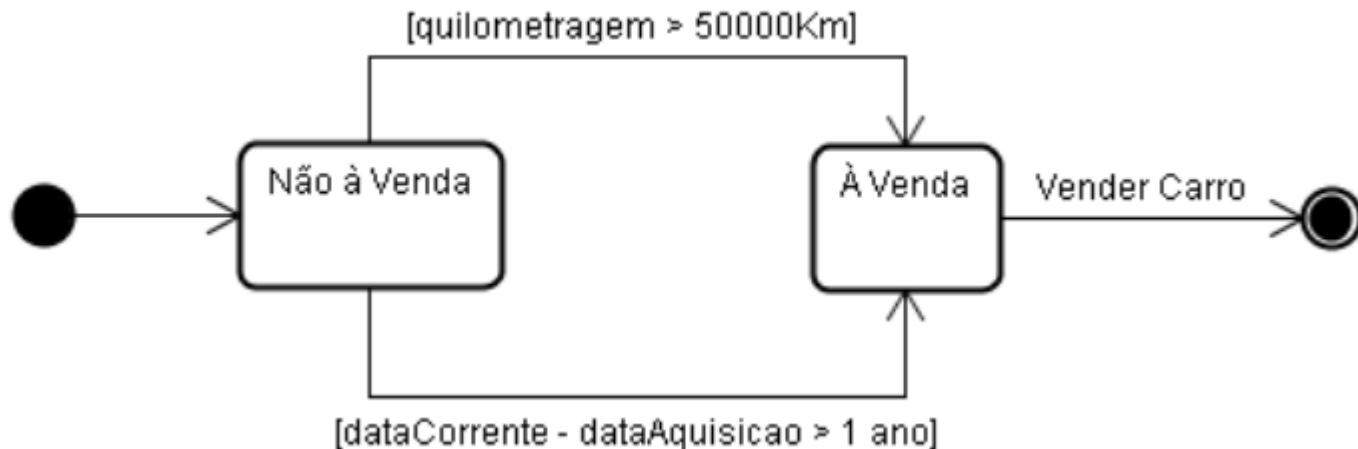




Exemplo: Locadora de Automóveis

Diagrama de estados da classe Carro:

Possibilidade de Venda

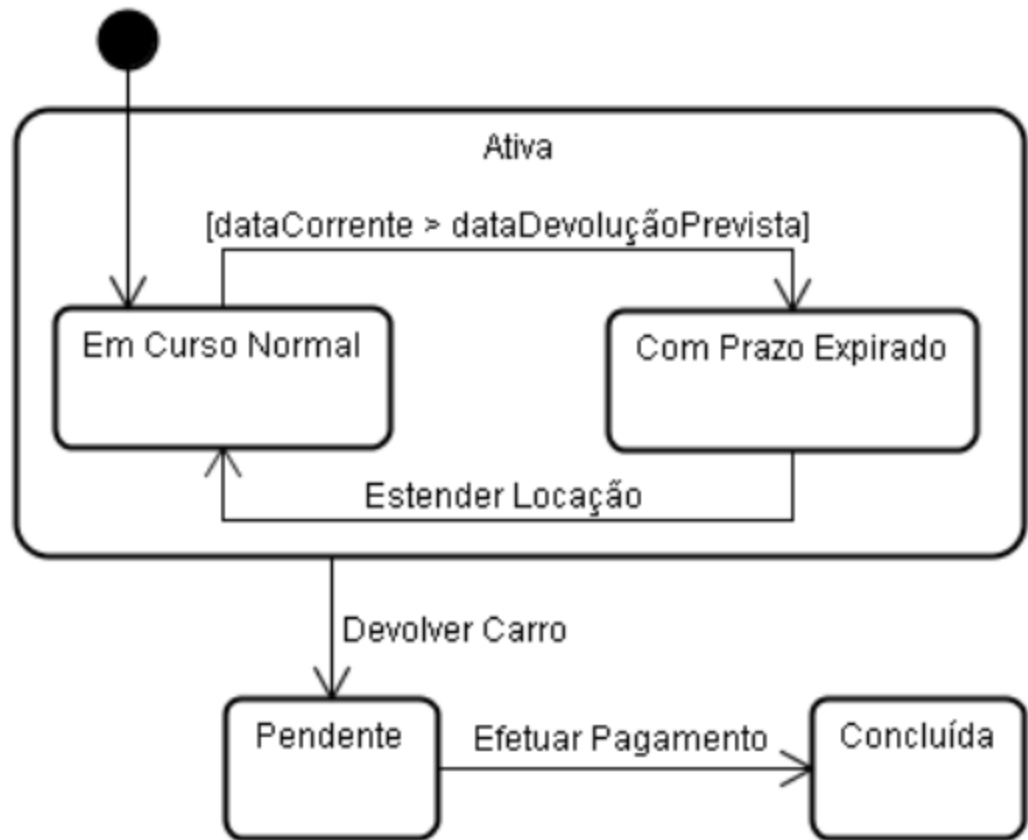


- Eventos que alteram a quilometragem estão implícitos (fluxo *Devolver Carro* e fluxo *Receber Carro vindo de outra Localidade*)

Exemplo: Locadora de Automóveis

Diagrama de estados da classe Locação

Uma locação no estado *Pendente* tem atributos a mais em relação a uma Locação *Ativa*...



Exemplo: Locadora de Automóveis

Classe locação com atributos inerentes a diferentes estados: classe complexa com várias regras de consistência



Atributo *estado* é derivado (seu valor é computável).
Poderia ser substituído por uma operação.
Atributos nulos dependendo do estado.

Exemplo: Locadora de Automóveis

Classe locação: Dividindo responsabilidades

