

# Engenharia de Requisitos de Software

Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# Modelagem Dinâmica

- ▶ O esquema comportamental de um sistema visa especificar as ações realizadas por um sistema de informação.
- ▶ O efeito de uma ação pode ser uma alteração em sua base de informação e/ou a comunicação de alguma informação ou comando para um ou mais destinatários.
- ▶ Uma parte importante do modelo comportamental de um sistema é o modelo de casos de uso, o qual fornece uma visão das funcionalidades que o sistema deve prover.

# Modelagem Dinâmica

- ▶ Durante a realização de um caso de uso, atores geram eventos de requisição de ações para o sistema, solicitando a execução de alguma ação.
- ▶ O sistema realiza ações e ele próprio pode gerar outras requisições de ação.
- ▶ É necessário então modelar essas **requisições de ações**, as correspondentes **ações a serem realizadas pelo sistema e seus efeitos**.
- ▶ **Este é o propósito da modelagem dinâmica.**



# Modelagem Dinâmica

- ▶ Em uma abordagem orientada a objetos, requisições de ação correspondem a mensagens trocadas entre objetos.
- ▶ As ações propriamente ditas e seus efeitos são tratados pelas operações das classes.
- ▶ Assim, a modelagem dinâmica está relacionada com as trocas de mensagens entre objetos e a modelagem das operações das classes.



# Modelagem Dinâmica

- ▶ Diagramas de classes representam apenas os elementos estáticos de um modelo de análise orientada a objetos.
- ▶ Precisamos modelar o comportamento dinâmico da aplicação, representando o comportamento do sistema como uma função do tempo e de eventos específicos.
- ▶ Um modelo de dinâmico indica **como** o sistema irá responder a eventos ou estímulos externos e auxilia o processo de descoberta das operações das classes do sistema.

# Modelagem Dinâmica

- ▶ Para apoiar a modelagem da dinâmica de sistemas, a UML oferece três tipos de diagramas:
- ▶ • **Diagrama de Gráfico de Estados:** mostra uma máquina de estados que consiste dos estados pelos quais objetos de uma particular classe podem passar ao longo de seu ciclo de vida e as transições possíveis entre esses estados, as quais são resultados de eventos que atingem esses objetos.
- ▶ Diagramas de gráfico de estados (ou diagramas de transição de estados) são usados principalmente para modelar o comportamento de uma classe, dando ênfase ao comportamento específico de seus objetos.

# Modelagem Dinâmica

- ▶ • **Diagrama de Interação:** descreve como grupos de objetos colaboram em certo comportamento.
- ▶ Diagramas de interação podem ser de dois tipos: diagramas de comunicação e diagramas de sequência.
- ▶ Um **diagrama de sequência** é um diagrama de interação que dá ênfase à ordenação temporal das mensagens trocadas por objetos.
- ▶ Já um **diagrama de comunicação** dá ênfase à organização estrutural dos objetos que enviam e recebem mensagens.



# Modelagem Dinâmica

- ▶ • **Diagrama de Atividades:** mostra o fluxo de uma atividade para outra em um sistema, incluindo sequências e ramificações de fluxo, subatividades e objetos que realizam e sofrem ações.
- ▶ Diagramas de atividades são usados principalmente para a modelagem das funções de um sistema, dando ênfase ao fluxo de controle na execução de um comportamento.
- ▶ Diagramas de estados focalizam o comportamento de objetos de uma classe específica. Um diagrama de interação observa os objetos que passam mensagens; um diagrama de atividades focaliza as atividades e suas entradas e saídas, i.e., objetos passados de uma atividade para outra.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Um sistema de informação mantém uma representação do estado do domínio em sua base de informações.
- ▶ Esse estado de coisas do domínio em um dado ponto no tempo corresponde ao conjunto de instâncias dos tipos de entidades (classes) e de relacionamentos (associações) relevantes que existem no domínio naquele momento.
- ▶ O esquema estrutural se preocupa com essa perspectiva.
- ▶ Entretanto, o sistema também realiza ações.

# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ O efeito de uma ação pode ser uma alteração em sua base de informação e/ou a comunicação de alguma informação ou comando para um ou mais destinatários.
- ▶ Um evento de requisição de ação (ou simplesmente uma requisição) é uma solicitação para que o sistema realize uma ação.
- ▶ Na análise de requisitos, assume-se que a tecnologia é perfeita e, por conseguinte, que o sistema executa as ações requisitadas instantaneamente.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Dependendo de como são iniciadas, requisições podem ser **explícitas**, **temporais** ou **geradas**.
- ▶ Uma requisição **explícita** é iniciada explicitamente por um ator (requisição externa) ou por uma outra ação (requisição induzida), como parte de seu efeito.
- ▶ Uma requisição **temporal** é iniciada pela passagem do tempo, ocorrendo independentemente do sistema.
- ▶ Por fim, uma requisição **gerada** é iniciada quando uma condição de geração da requisição é satisfeita. O sistema detecta que a condição foi satisfeita e gera a correspondente requisição.

# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Por exemplo, em um sistema de controle de estoque, uma requisição de compra pode ser gerada quando a quantidade mínima de um produto for atingida.
- ▶ A maioria das requisições é externa.
- ▶ Dois importantes tipos de requisições externas são: notificações de eventos de domínio e consultas.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Uma consulta é uma requisição externa que provê alguma informação para o ator que iniciou a requisição.
- ▶ Consultas não alteram a base de informações do sistema.
- ▶ Uma notificação de evento de domínio é uma requisição externa cujo efeito é uma mudança na base de informações do sistema, correspondendo a um evento de domínio.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Nem todas as mudanças na base de informações de um sistema são admissíveis.
- ▶ Os fatos nessa base mudam ao longo do tempo, mas não de maneira arbitrária.
- ▶ Os eventos de domínio definem, exatamente, as mudanças admissíveis.
- ▶ Por meio de notificações de eventos de domínio, atores dizem para o sistema que um evento de domínio ocorreu.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Por exemplo, quando ocorre no mundo real o evento de reserva de um carro em uma locadora de automóveis, um usuário, ao realizar o caso de uso correspondente (p.ex., Reservar Carro), está notificando o sistema que esse evento ocorreu, o que inicia uma sequência de ações (os passos do caso de uso), por meio da qual o sistema sabe que o evento ocorreu no domínio.
- ▶ Um evento de domínio corresponde a um conjunto não vazio de eventos estruturais, percebido ou considerado como uma alteração única no domínio.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Um evento estrutural, por sua vez, é uma ação elementar que insere ou remove um fato na base de informações do sistema.
- ▶ Há quatro tipos básicos de eventos estruturais: inserção de entidade, remoção de entidade, inserção de relacionamento e remoção de relacionamento.
- ▶ Esses eventos são ditos estruturais, porque eles são completamente determinados pelo esquema conceitual estrutural e não são explicitamente mostrados no esquema comportamental.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

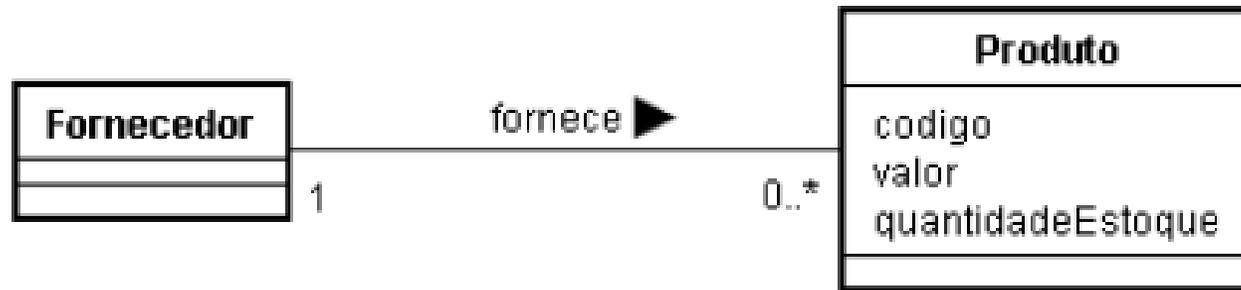
- ▶ No desenvolvimento orientado a objetos, os eventos estruturais correspondem a operações básicas das classes.
- ▶ Assim, toda classe tem, implicitamente, operações para:
  - ▶ criar objetos da classe (evento estrutural de inserção de entidade), dita operação construtora da classe;
  - ▶ eliminar objetos (evento estrutural de remoção de entidade), dita operação destruidora da classe;
  - ▶ estabelecer ligações e atribuir valores para atributos (eventos estruturais de inserção de relacionamentos); e
  - ▶ remover ligações ou excluir valores de atributos (eventos estruturais de remoção de relacionamentos).



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Seja o exemplo de um sistema de controle de produtos:



- ▶ Nesse exemplo, quando a companhia começa a trabalhar com um novo produto (p.ex., Prod1), o estado do domínio se altera, caracterizando um evento de domínio.

# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Esse evento de domínio corresponde a cinco eventos estruturais:
- ▶ (1) a criação do objeto Prod1,
- ▶ (2) a atribuição de um valor para o atributo codigo,
- ▶ (3) a atribuição de um valor para o atributo valor,
- ▶ (4) a atribuição de um valor para o atributo quantidadeEstoque e
- ▶ (5) o estabelecimento da associação fornece com seu fornecedor.
  
- ▶ Esse novo produto é considerado um evento de domínio, porque o conjunto de cinco eventos estruturais que o compõe é visto como um evento único.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ Cada evento de domínio possui um conjunto de eventos estruturais, chamado de efeito do evento.
- ▶ A correspondência entre eventos e seus efeitos é dada por uma expressão de mapeamento.
- ▶ O efeito do evento pode ser definido segundo duas abordagens: a abordagem de pós-condição e a abordagem procedimental.
- ▶ Na abordagem procedimental, quando o paradigma orientado a objetos é adotado, o efeito de um evento é definido como uma operação de uma classe.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ O corpo dessa operação deve ser tal que sua execução produza o conjunto de eventos estruturais que compõe o evento de domínio.
- ▶ A execução dessa operação vai deixar a base de informações em um novo estado, o qual deve satisfazer a todas as restrições estáticas (definidas no modelo estrutural).
- ▶ Assim, durante a definição da operação correspondente ao efeito de um evento de domínio, devem-se levar em conta as restrições capturadas no modelo estrutural e garantir que o novo estado da base de informações do sistema vai satisfazer a todas elas.



# Modelagem Dinâmica

## Tipos de Requisições de Ação

- ▶ No que concerne a consultas, na modelagem conceitual define-se apenas o conteúdo de informação das respostas, abstraindo-se detalhes que dizem respeito ao formato e a características de dispositivos de saída.
- ▶ Para que as informações de atributos e associações possam ser recuperadas para serem mostradas como parte das respostas a consultas, as classes precisam prover operações básicas para obter essas informações.
- ▶ Assim como as demais operações básicas, assume-se que toda classe possui implicitamente operações para se obter os valores correntes de seus atributos e associações.



# Modelagem Dinâmica

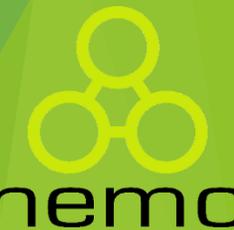
## Tipos de Requisições de Ação

- ▶ Durante a realização de um caso de uso, atores geram requisições para o sistema, solicitando a execução de ações.
- ▶ O sistema realiza ações e ele próprio pode gerar outras requisições de ação.
- ▶ O conjunto de casos de uso tem de ser consistente com o conjunto de requisições definidas no esquema comportamental do sistema.



# Diagramas de Gráfico de Estados

- ▶ Todo objeto tem um tempo de vida.
- ▶ Na criação, o objeto nasce; na destruição, ele deixa de existir.
- ▶ Entre esses dois momentos, um objeto poderá interagir com outros objetos, enviando e recebendo mensagens.
- ▶ Essas interações representam o comportamento do objeto e ele pode ser variável ao longo do ciclo de vida do objeto.
- ▶ Ou seja, muitas vezes, o comportamento dos objetos de uma classe depende do estado em que o objeto se encontra em um dado momento.
- ▶ Nestes casos, é útil especificar o comportamento usando uma máquina de estados.



# Diagramas de Gráfico de Estados

- ▶ Classes com estados (ou classes modais) são classes cujas instâncias podem mudar de um estado para outro ao longo de sua existência, mudando possivelmente sua estrutura, seus valores de atributos ou comportamento dos métodos.
- ▶ Classes modais podem ser modeladas como máquinas de estados finitos.
- ▶ Uma máquina de estados finitos é uma máquina que, em um dado momento, está em um e somente um de um número finito de estados.



# Diagramas de Gráfico de Estados

- ▶ Os estados de uma máquina de estados de uma classe modal correspondem às situações relevantes em que as instâncias dessa classe podem estar durante sua existência.
- ▶ Um estado é considerado relevante quando ele ajuda a definir restrições ou efeitos dos eventos.
- ▶ Em qualquer estado, uma máquina de estados pode receber estímulos.
- ▶ Quando a máquina recebe um estímulo, ela pode realizar uma transição de seu estado corrente (dito estado origem) para outro estado (dito estado destino), sendo que se assume que as transições são instantâneas.

# Diagramas de Gráfico de Estados

- ▶ A definição do estado destino depende do estado origem e do estímulo recebido.
- ▶ Além disso, os estados origem e destino em uma transição podem ser o mesmo.
- ▶ Diagramas de Transições de Estados são usados para modelar o comportamento de instâncias de uma classe modal na forma de uma máquina de estados.
- ▶ Todas as instâncias da classe comportam-se da mesma maneira.

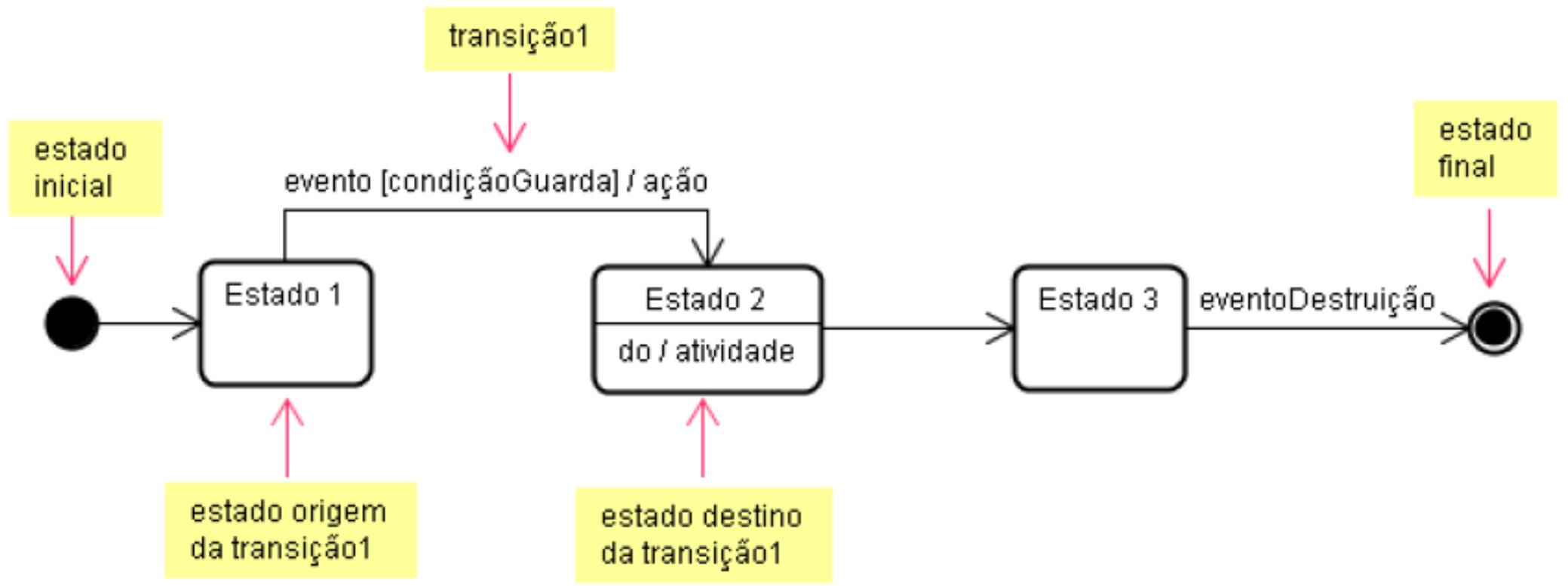


# Diagramas de Gráfico de Estados

- ▶ Em outras palavras, cada diagrama de estados é construído para uma única classe, com o objetivo de mostrar o comportamento ao longo do tempo de vida de seus objetos.
- ▶ 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.
- ▶ Uma máquina de estado especifica a ordem válida dos estados pelos quais os objetos da classe podem passar ao longo de seu ciclo de vida.



# Diagramas de Gráfico de Estados



# Diagramas de Gráfico de Estados

- ▶ 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.
- ▶ Quando um objeto fica realizando uma atividade durante todo o tempo em que permanece em um estado, deve-se indicar essa atividade no compartimento de ações do respectivo estado.
- ▶ É importante realçar que uma atividade tem duração significativa e, quando concluída, tipicamente a conclusão provoca uma transição para um novo estado.
- ▶ A notação da UML para representar atividades de um estado é: do / <<nomeAtividade>>.

# Diagramas de Gráfico de Estados

- ▶ Uma transição envolve um estado origem, um estado destino e normalmente um evento, dito o gatilho da transição.
- ▶ Quando a máquina de estados se encontra no estado origem e recebe o evento gatilho, então o evento dispara a transição e a máquina de estados vai para o estado destino.
- ▶ Se uma máquina recebe um evento que não é um gatilho para nenhuma transição, então ela não é afetada pelo evento.
- ▶ Uma transição pode ter uma condição de guarda associada.

# Diagramas de Gráfico de Estados

- ▶ Às vezes, há duas ou mais transições com o mesmo estado origem e o mesmo evento gatilho, mas com condições de guarda diferentes.
- ▶ Neste caso, a transição é disparada somente quando o evento gatilho ocorre e a condição de guarda é verdadeira.
- ▶ Quando uma transição não possuir uma condição de guarda associada, então ela ocorrerá sempre que o evento ocorrer.
- ▶ Por fim, quando uma transição é disparada, uma ação instantânea pode ser realizada. Assim, o rótulo de uma transição pode ter até três partes, todas elas opcionais:
  - ▶ **evento [condiçãoGuarda] / ação**

# Diagramas de Gráfico de Estados

- ▶ O fato de uma transição não possuir um evento associado normalmente aponta para a existência de um evento implícito.
- ▶ Isso tipicamente ocorre em três situações:
  - ▶ (i) o evento implícito é a conclusão da atividade do estado origem e a transição ocorrerá tão logo a atividade associada ao estado origem tiver sido concluída;
  - ▶ (ii) o evento implícito é temporal, sendo disparado pela passagem do tempo;
  - ▶ (iii) o evento implícito torna a condição de guarda verdadeira na base de informações do sistema, mas o evento em si não é modelado.



# Diagramas de Gráfico de Estados

- ▶ Embora ambos os termos ação e atividade denotem processos, eles não devem ser confundidos.
  - ▶ Ações são consideradas processos instantâneos;
  - ▶ Atividades, por sua vez, estão sempre associadas a estados e têm duração no tempo.
- ▶ Vale a pena observar que, no mundo real, não há processos efetivamente instantâneos.



# Diagramas de Gráfico de Estados

- ▶ Por mais rápida que seja, uma ação ocorrerá sempre em um intervalo de tempo.
- ▶ Esta simplificação de se considerar ações instantâneas no modelo conceitual pode ser associada à ideia de que a ação ocorre tão rapidamente que não é possível interrompê-la.
- ▶ Em contraste, uma atividade é passível de interrupção, sendo possível, por exemplo, que um evento ocorra, interrompa a atividade e provoque uma mudança no estado do objeto antes da conclusão da atividade.



# Diagramas de Gráfico de Estados

- ▶ Às vezes quer se modelar situações em que uma ação instantânea é realizada quando se entra ou sai de um estado, qualquer que seja a transição que o leve ou o retire desse estado.
- ▶ Seja o exemplo de um elevador:
- ▶ Neste contexto, ao parar em um andar, o elevador abre a porta.
- ▶ Suponha que a abertura da porta do elevador seja um processo que não possa ser interrompido e, portanto, que se opte por modelá-lo como uma ação.



# Diagramas de Gráfico de Estados

- ▶ Essa ação deverá ocorrer sempre que o elevador entrar no estado “Parado” e deve ser indicada no compartimento de ações desse estado como sendo uma ação de entrada no estado.
- ▶ A notação da UML para representar ações de entrada em um estado é: entry / <<nomeAção>>.
- ▶ Para representar ações de saída de um estado a notação é: exit / <<nomeAção>>.



# Diagramas de Gráfico de Estados

- ▶ Restam ainda dois tipos especiais de estados: os ditos estados inicial e final.
- ▶ Ao ser instanciado, um objeto precisa estar em algum estado. O estado inicial é precisamente esse estado.
- ▶ Seu significado é o seguinte: quando o objeto é criado, ele é colocado no estado inicial e sua transição de saída é automaticamente disparada, movendo o objeto para um dos estados da máquina de estados.
- ▶ Toda máquina de estados tem de ter um (e somente um) estado inicial.



# Diagramas de Gráfico de Estados

- ▶ Note que o estado inicial não se comporta como um estado normal, uma vez que objetos não se mantêm nele por um período de tempo.
- ▶ Ao contrário, uma vez que eles entram no estado inicial, sua transição de saída é imediatamente disparada e o estado inicial é abandonado.
- ▶ A transição de saída do estado inicial tem como evento gatilho implícito o evento responsável pela criação do objeto e, na UML, esse evento não é explicitamente representado.
- ▶ Estados iniciais têm apenas transições de saída.



# Diagramas de Gráfico de Estados

- ▶ As transições de saída de um estado inicial podem ter condições de guarda e/ou ações associadas.
- ▶ Quando houver condições de guarda, deve-se garantir que sempre pelo menos uma das transições de saída poderá ser disparada.
- ▶ Quando um objeto deixa de existir, obviamente ele deixa de estar em qualquer um dos estados.
- ▶ Isso pode ser dito no diagrama por meio de uma transição para o estado final.
- ▶ O estado final indica, na verdade, que o objeto deixou de existir.



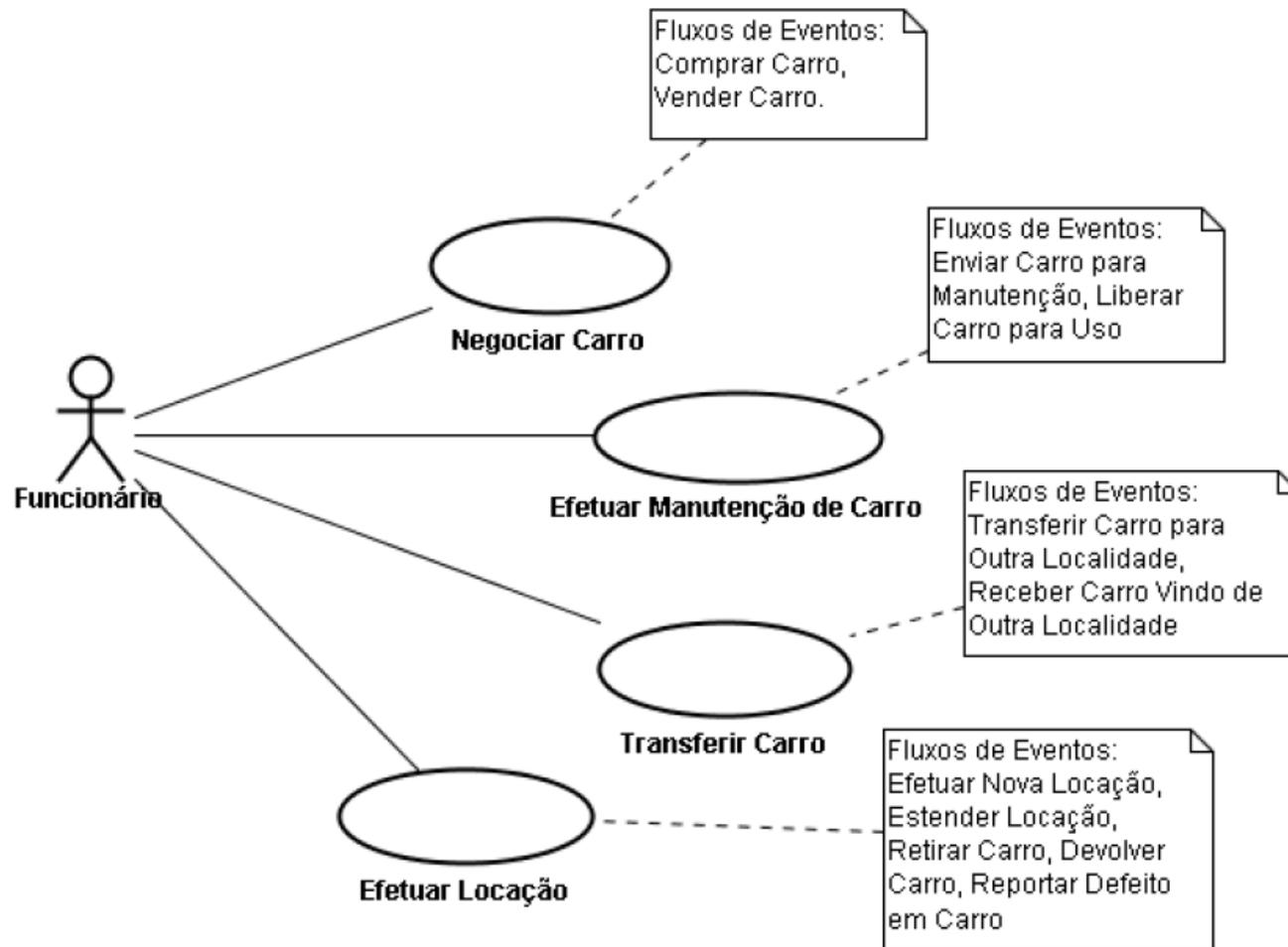
# Diagramas de Gráfico de Estados

- ▶ As transições para o estado final definem os estados em que é possível excluir o objeto.
- ▶ Classes cujos objetos não podem ser excluídos, portanto, não possuem um estado final.
- ▶ Assim como o estado inicial, o estado final não se comporta como um estado normal, uma vez que o objeto também não permanece nesse estado (já que o objeto não existe mais).
- ▶ Ao contrário do estado inicial, contudo, uma máquina de estados pode ter vários estados finais. Além disso, deve-se representar o evento que elimina o objeto.



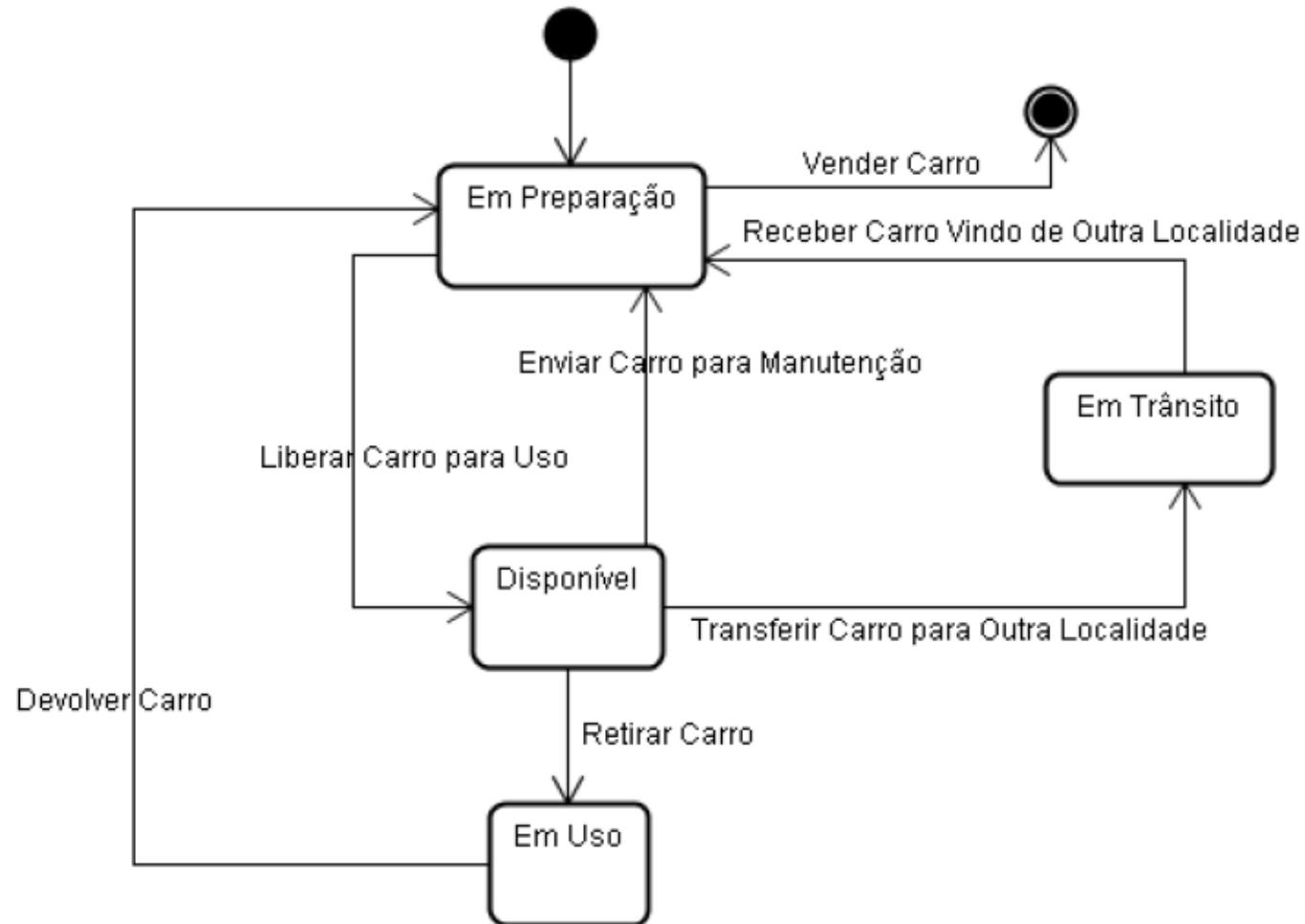
# Diagramas de Gráfico de Estados

- ▶ Seja o exemplo de uma locadora de automóveis, que possua, dentre outros, os casos de uso seguintes:



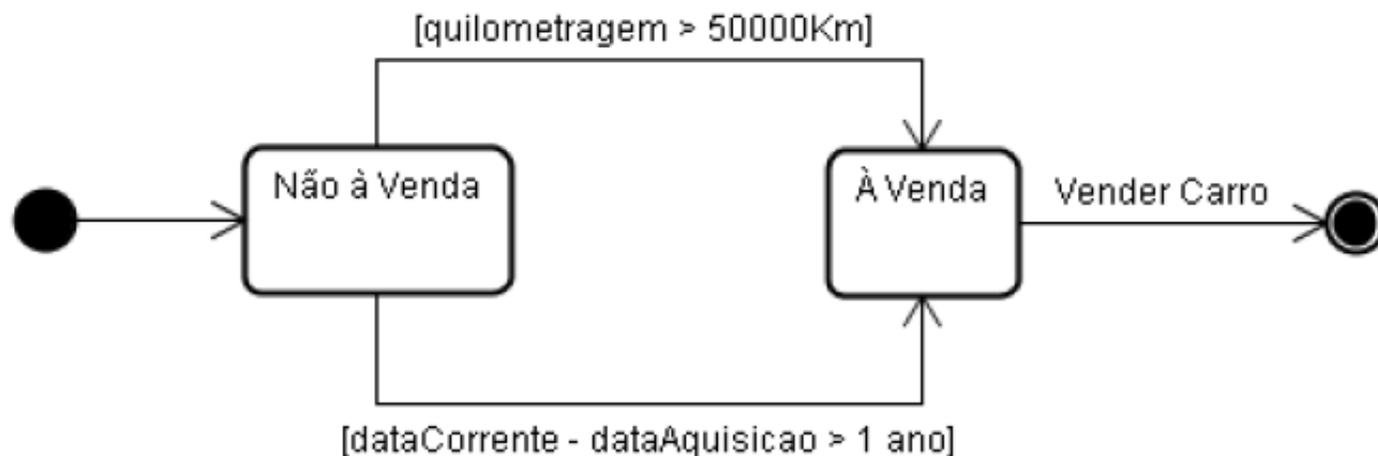
# Diagramas de Gráfico de Estados

- ▶ A classe Carro tem o seu comportamento definido pela máquina de estados a seguir:



# Diagramas de Gráfico de Estados

- ▶ Para algumas classes, pode ser útil desenvolver mais do que um diagrama de estados, cada um deles modelando o comportamento dos objetos da classe por uma perspectiva diferente.
- ▶ O diagrama anterior mostra os possíveis estados de um carro segundo um ponto de vista de disponibilidade. Entretanto, independentemente da disponibilidade, do ponto de vista de possibilidade de negociação, um carro pode estar em dois estados (Não à Venda, À Venda).



# Diagramas de Gráfico de Estados

- ▶ Todos os estados mostrados até então são estados simples, i.e., estados que não possuem subestados.
- ▶ Entretanto, há também estados compostos, os quais podem ser decompostos em um conjunto de subestados disjuntos e mutuamente exclusivos e um conjunto de transições.
- ▶ Um subestado é um estado aninhado em outro estado.
- ▶ O uso de estados compostos e subestados é bastante útil para simplificar a modelagem de comportamentos complexos.





# Diagramas de Gráfico de Estados

- ▶ O estado de um objeto deve ser mapeado no modelo estrutural.
- ▶ De maneira geral, o estado pode ser modelado por meio de um atributo.
- ▶ Esse atributo deve ser monovalorado e obrigatório.
- ▶ O conjunto de valores possíveis do atributo é o conjunto dos estados possíveis, conforme descrito pela máquina de estados.



# Diagramas de Gráfico de Estados

- ▶ Assim, é bastante natural que o tipo de dados desse atributo seja definido como um tipo de dados enumerado.
- ▶ Um nome adequado para esse atributo é “estado”. Contudo, outros nomes mais significativos para o domínio podem ser atribuídos.
- ▶ Em especial, quando uma classe possuir mais do que uma máquina de estado e, por conseguinte, mais do que um atributo de estado for necessário, o nome do atributo de estado deve indicar a perspectiva capturada pela correspondente máquina de estados.

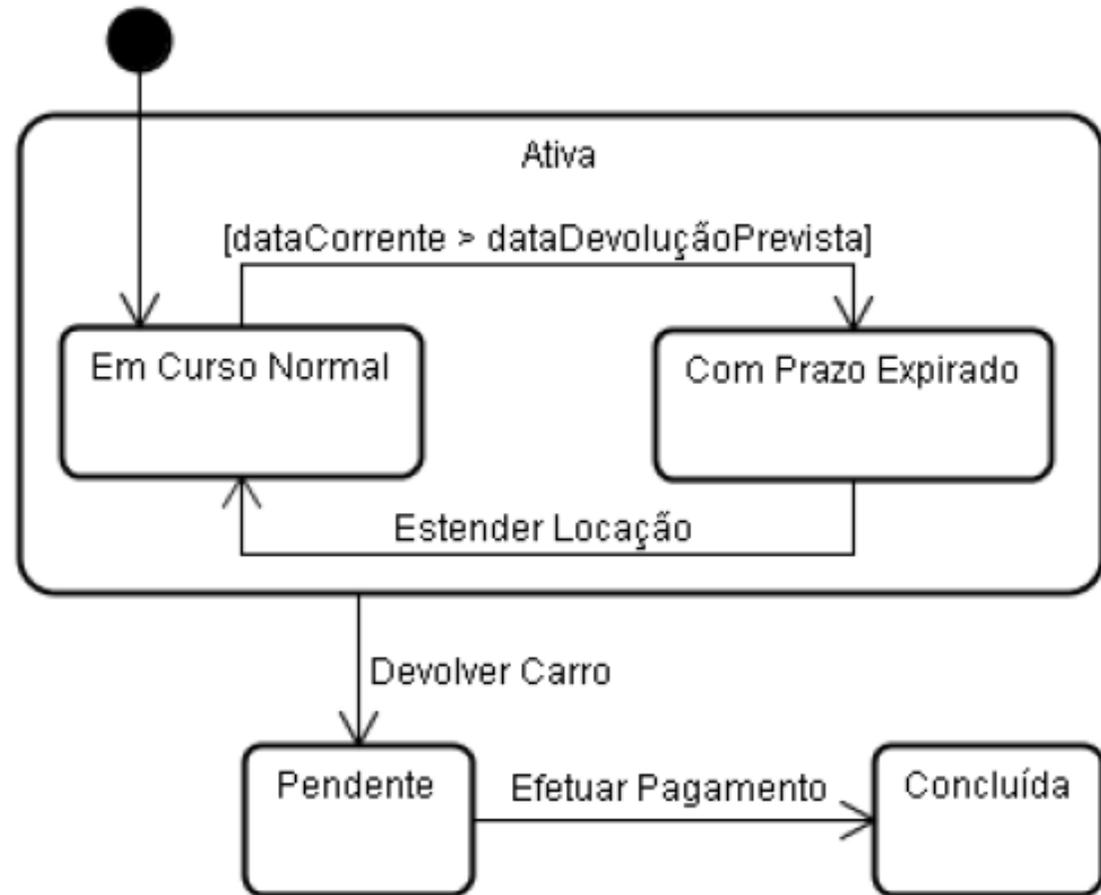


# Diagramas de Gráfico de Estados

- ▶ É interessante observar que algumas transições podem mudar a estrutura da classe.
- ▶ Quando os diferentes estados de um objeto não afetam a sua estrutura, mas apenas, possivelmente, os valores de seus atributos e associações, diz-se que a transição é estável e os diferentes estados podem ser mapeados para um simples atributo.
- ▶ Entretanto, há situações em que, conforme um objeto vai passando de um estado para outro, ele vai ganhando novos atributos ou associações, ou seja, há uma mudança na estrutura da classe.

# Diagramas de Gráfico de Estados

- ▶ Seja o exemplo de uma locação de carro.
- ▶ Quando uma locação é criada, ela está ativa, em curso normal.
- ▶ Quando o carro não é devolvido até a data de devolução prevista, a locação passa a ativa com prazo expirado.
- ▶ Se a locação é estendida, ela volta a ficar em curso normal.
- ▶ Quando o carro é devolvido, a locação fica pendente.
- ▶ Finalmente, quando o pagamento é efetuado, a locação é concluída.



# Diagramas de Gráfico de Estados

- ▶ Locações ativas (e em seus subestados, obviamente) têm como atributos: data de locação, data de devolução prevista, valor devido e caução.
- ▶ Quando uma locação vai para o estado pendente, é necessário registrar a data de devolução efetiva e os problemas observados no carro devolvido.
- ▶ Finalmente, quando o pagamento é efetuado, é preciso registrar a data do pagamento, o valor e a forma de pagamento.



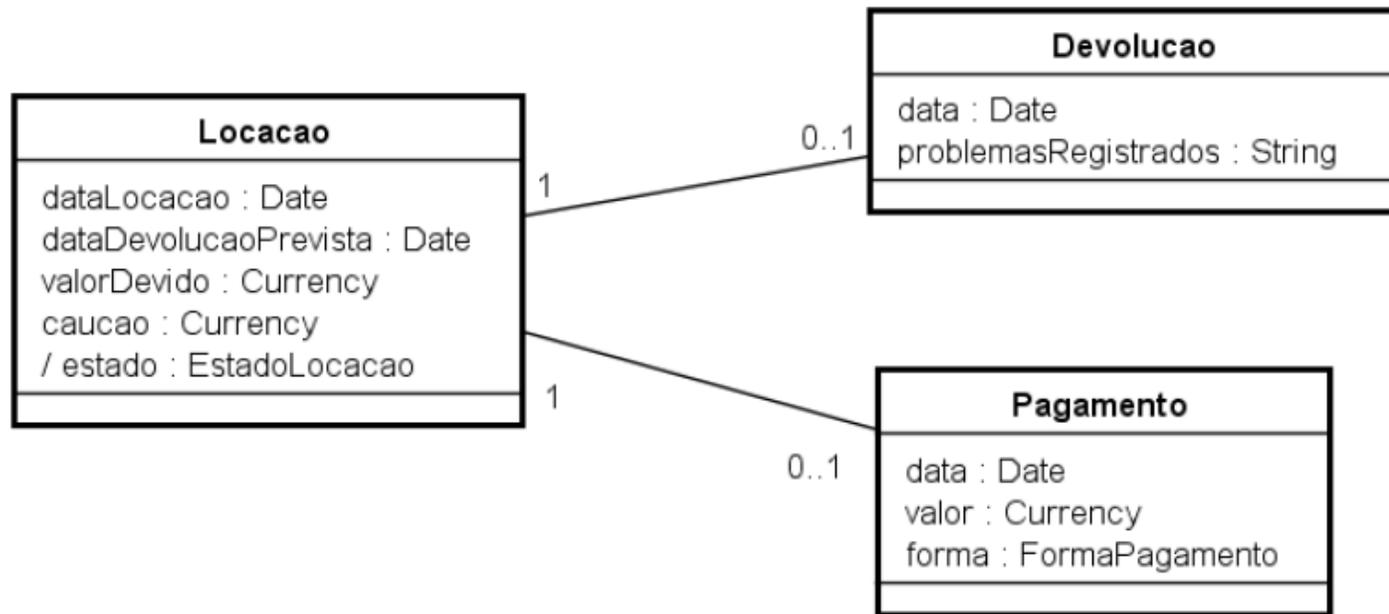
# Diagramas de Gráfico de Estados

- Uma solução frequentemente usada para capturar essa situação no modelo conceitual estrutural consiste em criar uma única classe (Locacao) e fazer com que certos atributos sejam nulos até que o objeto mude de estado.



# Diagramas de Gráfico de Estados

- ▶ É possível modelar essa situação desdobrando o conceito original em três: um representando a locação efetivamente, outro representando a devolução e outro representando o pagamento.
- ▶ Desta forma, capturam-se claramente os eventos de locação, devolução e pagamento, colocando as informações de cada evento na classe correspondente.



# Diagramas de Gráfico de Estados

- ▶ Estados de uma classe modal podem ser tratados por meio de operações ao invés de atributos.
- ▶ Seja o exemplo anterior de locações de carros.
- ▶ O estado de uma locação pode ser computado a partir dos atributos e associações da classe Locacao, sem haver a necessidade de um atributo estado.

Que operações podem definir estados da classe  
Locação?



# Diagramas de Gráfico de Estados

- ▶ Se uma locação não tem uma devolução associada, então ela está ativa.
- ▶ Estando ativa, se a data corrente é menor ou igual à data de devolução prevista, então a locação está em curso normal; caso contrário, ela está com prazo expirado.
- ▶ Se uma locação possui uma devolução, mas não possui um pagamento associado, então ela está pendente.
- ▶ Finalmente, se a locação possui um pagamento associado, então ela está concluída.
- ▶ Em casos como este, pode-se optar por tratar estado como uma operação e não como um atributo.
- ▶ Opcionalmente, pode-se utilizar a operação para calcular o valor de um atributo derivado estado.
- ▶ Atributos derivados são representados na UML precedidos por uma barra (no exemplo, /estado).

That's all Folks!



nemo