

MODELAGEM DE ESTADOS

Iolanda Cláudia Sanches Catarino

0

Ver anotações

DIAGRAMA DE MÁQUINA DE ESTADOS

O diagrama de máquina de estados permite descrever o ciclo de vida de objetos de uma classe, os eventos que causam a transição entre os estados e a realização de operações resultantes.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

CONVITE AO ESTUDO

Olá! Seja bem-vindo a esta unidade do livro de Análise Orientada a Objetos.

Independentemente do modelo de processo de engenharia de software a ser adotado no desenvolvimento de um sistema, cada empresa ou gerente de projeto define a sua melhor metodologia de desenvolvimento de sistemas em consonância com as boas práticas da engenharia de software, abrangendo todo o processo de software, os métodos e as ferramentas, a fim de atenderem aos diferentes domínios e complexidades de sistemas de softwares.

As atividades iniciais de análise de requisitos e análise de um processo de desenvolvimento de software exigem maiores esforços, comunicação e interação do analista de sistemas com os usuários responsáveis pelas atividades dos

processos de negócio que envolvem a área da solução a ser desenvolvida.
Conhecer e compreender o domínio do sistema, bem como delimitar o escopo do projeto de software, são ações extremamente importantes para cumprir com sucesso todo o processo de desenvolvimento de um sistema, conforme previsto.

É importante ressaltar que a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*) apoia o desenvolvimento incremental do software orientado a objetos a partir de modelos que podem evoluir com a inclusão de novos detalhes, enfatizando a descrição de um sistema nas perspectivas estrutural e comportamental, ou seja, a visão estática e dinâmica do sistema especificada em diferentes técnicas de modelagem. Assim, a adoção da UML não está vinculada exclusivamente a uma atividade ou a uma fase do processo de desenvolvimento de software, definindo quem deve fazer o que, quando e como.

Para modelagem da atividade ou fase de análise com a UML, é usual especificar, primeiramente, a modelagem dos casos de uso correspondentes aos requisitos funcionais do sistema, o modelo de classes e a modelagem de atividades que demonstram o fluxo de controle das funcionalidades do sistema, ressaltando os aspectos dinâmicos do sistema. Na sequência, recomenda-se documentar o comportamento dos objetos das classes, descrevendo como um sistema responde aos eventos internos e externos que influenciam na mudança dos estados de um objeto. Assim, na primeira seção desta unidade, será apresentado o Diagrama de Máquina de Estados, que tem como objetivo demonstrar o comportamento de um elemento através de um conjunto de transições de estados realizadas entre os estados dos objetos de uma classe, de um caso de uso ou mesmo de um sistema completo. Na segunda seção, continuaremos com a modelagem dinâmica do sistema sob a perspectiva de interação de como os objetos do sistema agem internamente para apoiarem a realização das funcionalidades representadas pelos casos de uso, trabalhando com o Diagrama de Sequência. Por fim, na terceira seção, exploraremos os conceitos, componentes e exemplos dos demais diagramas de interação da UML: Diagrama de Comunicação, Diagrama de Visão Geral de Interação e Diagrama de Tempo.

PRATICAR PARA APRENDER

Caro aluno, seja bem-vindo à seção sobre a modelagem de estados dos objetos.

Avançando na modelagem de análise, é fundamental definir o comportamento dos objetos na perspectiva dos estados que eles assumem durante a execução das funcionalidades do sistema. Todo objeto do mundo real ou do mundo computacional assume diferentes estados durante a sua existência. Por exemplo, no mundo real, um avião está em repouso, decolando, voando ou pousando; uma pessoa está parada, caminhando, correndo, etc. Da mesma forma, cada objeto, no mundo computacional, se encontra em um estado particular, decorrente de algum

acontecimento que ocasiona as mudanças de estado, a fim de atender a uma regra de negócio do contexto do sistema. Por isso, compreender a modelagem dos estados dos objetos e suas transições é essencial para documentar de forma consistente um software orientado a objetos.

Durante a execução de uma funcionalidade do sistema, um objeto muda de estado quando acontece algum evento interno ou externo ao sistema, provocando uma transição entre os estados do objeto e, com isso, ele realiza determinadas ações responsáveis pela consistência e integridade dos dados do sistema. Como exemplo, imagine você efetuando uma compra on-line em uma loja de produtos esportivos. Ao você escolher e confirmar os produtos que deseja comprar, o seu pedido de compra assume o estado de “Pedido recebido”; posteriormente, o sistema de venda da loja integrado ao sistema de faturamento confere se o pagamento foi aprovado e, uma vez que isso acontece, ele altera o estado do mesmo objeto pedido para “Pagamento aprovado”; na sequência do processo de compra do cliente, o setor de estoque e distribuição de produtos da loja separa os produtos do pedido e encaminha-os para o setor de logística, o qual providencia o transporte do pedido ao destinatário. A partir desse momento, o sistema atualiza a situação do pedido para “Transporte em andamento” e, assim que o cliente recebe o pedido, o sistema atualiza, por último, para “Entrega realizada”. No contexto desse exemplo, observa-se que o mesmo objeto – pedido de compra do cliente – transitou por diferentes estados durante um período de tempo de execução do sistema, portanto cada objeto pode transitar por um número finito de estados durante a sua existência.

Nesta seção, você compreenderá os conceitos e os componentes que integram a técnica de modelagem comportamental denominada Diagrama de Máquina de Estados, que representa o ciclo de vida dos objetos com a especificação dos estados e suas transições. Também, conhecerá a notação gráfica de cada elemento do diagrama, conforme a UML, para aplicá-la corretamente na elaboração do Diagrama de Máquina de Estados. Para reforçar o seu aprendizado, em sua atuação como analista de sistemas, você terá como desafio a abstração e identificação dos objetos que possuem estados relevantes a serem documentados, a definição dos estados e suas transições e a modelagem dos Diagramas de Máquina de Estados para um sistema de hotelaria, referente ao módulo recepção.

Você, como analista de sistemas em uma empresa de desenvolvimento e soluções de software e na sua atuação de responsável pelo projeto de um sistema de hotelaria – módulo recepção, considerando que toda a modelagem da análise de requisitos e a modelagem estática da atividade de análise do sistema já foram especificadas, principalmente o Diagrama de Classes, terá que definir para quais

objetos das classes do sistema deve ser modelado o Diagrama de Máquina de Estados, a partir das regras de negócio estabelecidas na modelagem dos requisitos funcionais do sistema.

Para reforçar a sua aprendizagem, considere que você faz parte de uma equipe de desenvolvedores de uma empresa de desenvolvimento de sistemas de software e que um dos projetos em andamento é referente ao módulo recepção de um sistema de hotelaria. A descrição a seguir relata o contexto das principais atividades da rotina da recepção do hotel, incluindo o controle de reservas, com suas regras de negócio.

DESCRIÇÃO DO MÓDULO RECEPÇÃO

O módulo recepção é responsável por controlar as reservas, os cadastros dos hóspedes e de empresas e o check-in (entrada do hóspede) e check-out (saída do hotel) de um hóspede.

É necessário manter um cadastro dos apartamentos do hotel, no qual deve constar o número, o tipo (padrão solteiro, padrão casal, padrão conjugado, luxo solteiro, etc.), a situação (disponível, ocupado, em arrumação, em manutenção ou desativado) e o valor da diária. Cada apartamento é classificado com um tipo apenas.

Um hóspede deve ser cadastrado com: CPF, RG, nome, endereço completo, data de nascimento, telefone residencial, celular, sexo, filiação, escolaridade, estado civil, nacionalidade, endereço eletrônico, número do passaporte (se for hóspede estrangeiro), situação (normal, preferencial, inadimplente, encerrado), placa do carro e nome da empresa que trabalha. Para um hóspede que se hospeda com sua família, é necessário fazer o cadastro de todos os membros da família hospedados. Um hóspede pode se hospedar em um hotel vinculado à empresa que trabalha ou de uma empresa provedora de gerenciamento global de viagens corporativas, sendo necessário ter o cadastro prévio dessas empresas. Uma empresa conveniada com o hotel deve ser cadastrada com: CNPJ, nome fantasia, razão social, inscrição estadual, ramo de atividade, endereço completo, telefone, endereço eletrônico, celular, e-mail e nome de uma pessoa para contato.

Um hóspede, para se hospedar no hotel, tem que ter um cadastro com seus dados pessoais efetuado previamente e uma reserva prévia realizada por ele (por telefone, por aplicativo ou pela web) ou pela empresa que tem vínculo. Sendo a empresa responsável pela estada do hóspede, ela é a encarregada de custear integralmente as despesas referentes às diárias, e quanto às refeições, ela pode custear ou não. Uma reserva é mantida por: nome da(s) pessoa(s) que irá(ão) se hospedar em um apartamento, data que efetuou a reserva, data de entrada, hora prevista de entrada, data de saída, quantidade de adultos, quantidade de crianças, idade das crianças e nome da empresa que trabalha (se a empresa for a responsável pela estada do hóspede). Um hóspede ou uma empresa pode realizar várias reservas.

Um hóspede, ao chegar ao hotel, informa seu nome para resgatar a reserva, confere seu cadastro pessoal e o atualiza, caso seja necessário. O check-in é registrado com: número do apartamento, dados do hóspede, data de entrada, hora de entrada, data prevista de saída, nome do acompanhante (também cadastrado como hóspede), se tiver, placa do carro e observação.

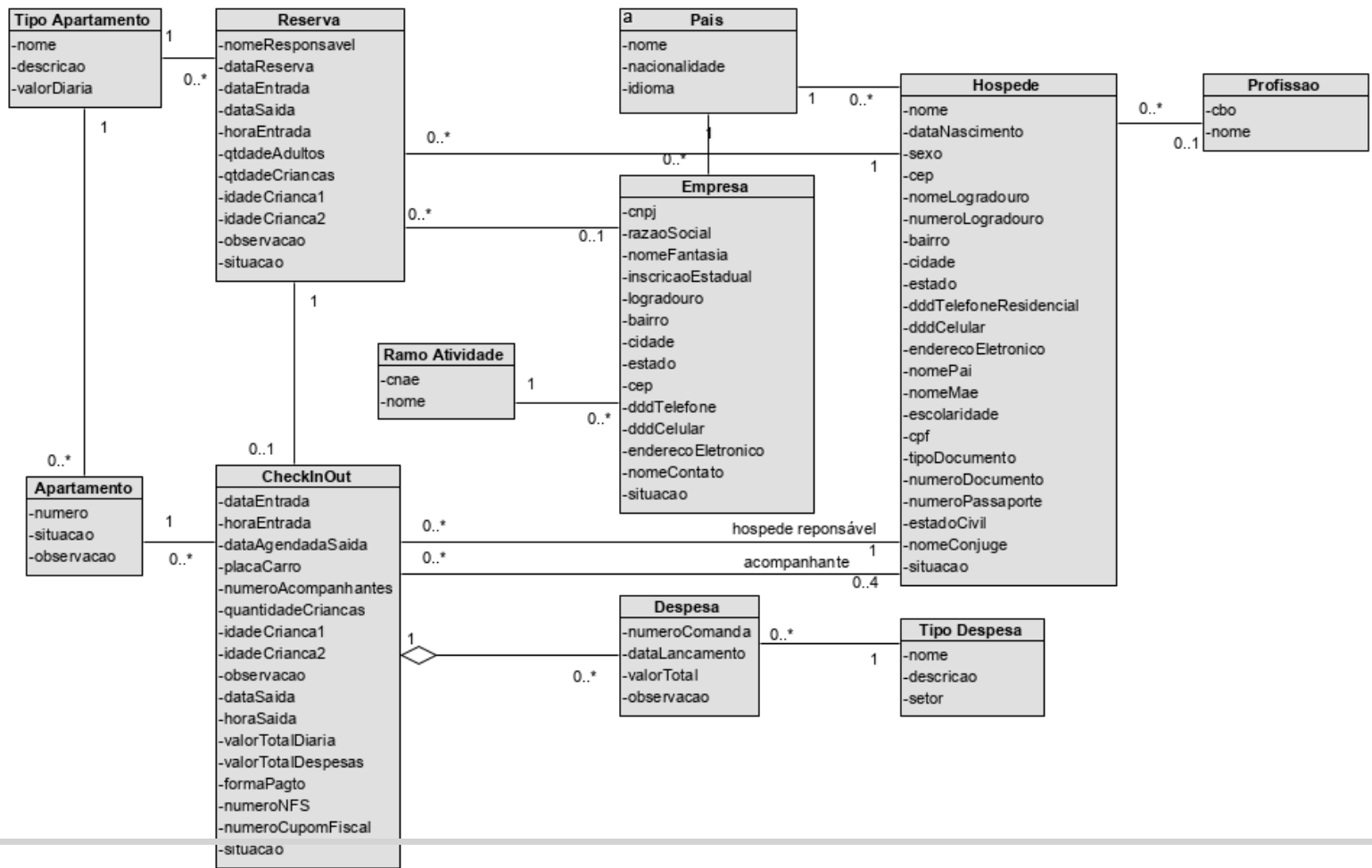
Durante a estada do hóspede no hotel, todos os alimentos consumidos ou serviços realizados para o hóspede são lançados como despesas, registrando o número do apartamento, a data, o tipo de despesa (serviço de lavanderia, refeição no restaurante, lanche, etc.), a descrição da despesa, a quantidade e o valor.

Na saída do hóspede é realizado o check-out, registrando: a data e a hora de saída, o valor total das despesas e das diárias, totalizando o valor do check-out, e a forma de pagamento (dinheiro ou cartão). Assim que o hóspede confere todas as despesas lançadas e confirma o seu check-out, são emitidos a nota fiscal de saída, referente às diárias, e o cupom fiscal, referente às despesas.

Considerando que a modelagem dos casos de uso foi especificada e analisando o Diagrama de Classes de análise (apenas com os atributos) a seguir, para sua melhor familiarização com o contexto do sistema, faça:

Elabore o Diagrama de Máquina de Estados correspondente à classe de objetos “Reserva” com os estados Realizada, Confirmada, Cancelada ou Efetivada. Defina as transições de estados com regras que considere pertinente ao contexto do sistema.

Figura 3.1 | Diagrama de Classes



Bom trabalho e ótimos estudos!

CONCEITO-CHAVE

As técnicas de modelagem comportamentais da UML representam o comportamento e a interação entre os elementos do sistema orientado a objetos, colaborando para modelagem da visão dinâmica do sistema. Geralmente, inicia-se a modelagem comportamental do sistema a partir da especificação do Diagrama de Casos de Uso e do Diagrama de Atividades e, na sequência, enfatiza-se a elaboração do Diagrama de Máquina de Estados correspondente aos objetos das classes com estados relevantes.

Você já sabe que todas as técnicas de modelagem da UML abrangem um conjunto de elementos que se vinculam aos conceitos básicos e aos princípios do paradigma orientado a objetos. Dessa forma, é fundamental lembrarmos os conceitos de estado, transição de estado e evento que sustentam o Diagrama de Máquina de Estados.

CONCEITOS BÁSICOS E PRINCÍPIOS DO PARADIGMA ORIENTADA A OBJETOS

- **Eventos:** representam os acontecimentos que provocam a mudança de estado dos objetos, podendo ser uma ação interna ou externa do objeto. Um evento, ao ser disparado em um instante de tempo de execução do sistema, invoca uma operação específica dos objetos de uma classe, a partir do envio de uma mensagem ao objeto, provocando uma mudança de estado dele. Na definição de Rumbaugh *et al.* (1997), um evento é uma ocorrência única de um estímulo individual de um objeto para outro que acontece em certo momento, disparando uma transmissão ou informação unidirecional entre os objetos.
- **Estados:** representam a abstração de uma forma de apresentação dos objetos em um instante de tempo de execução do sistema com uma duração finita, o qual demonstra a reação de um objeto em resposta a um evento. Na definição de Booch, Jacobson e Rumbaugh (2006, p. 290), um estado é “uma condição ou situação na vida de um objeto durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou aguarda um evento. Um objeto permanece em um estado por uma quantidade finita de tempo”. Considere como exemplo um objeto do tipo “Automóvel,” o qual, num instante de tempo (t_1), se encontra no estado “desligado” e, a partir de um evento disparado, por

exemplo, o acionamento da chave na ignição do veículo, uma ação humana, o objeto reage e muda o seu estado para “ligado”, diante da partida do motor, no instante de tempo seguinte (t_2).

- **Transição de estado:** representa a mudança de estado de um objeto em resposta a um evento disparado. Para Booch, Jacobson e Rumbaugh (2006), uma transição é um relacionamento entre dois estados, indicando que um objeto no primeiro estado realiza alguma ação e assume outro estado quando um evento é disparado e condições são satisfeitas.

DIAGRAMA DE MÁQUINA DE ESTADOS

A técnica de modelagem comportamental – Diagrama de Máquina de Estados – demonstra o comportamento dinâmico de um elemento por meio de um conjunto de estados relevantes e das transições entre os estados finitos dos objetos de uma classe, de casos de uso ou até mesmo do sistema como um todo. É considerado um estado relevante ou importante ao contexto do sistema o estado de um objeto que implica em ações a serem consistidas durante a execução do sistema, por exemplo, um objeto “funcionário”, que se encontra no estado de “afastado por licença médica” em determinado período, não pode ser permitido trabalhar durante o período estipulado do afastamento. Uma forma de consistir isso via sistema é não permitir que o usuário funcionário tenha acesso aos sistemas da empresa; ou um objeto “produto”, que se encontra no estado de “esgotado”, não pode ter a sua venda permitida. Dessa forma, na prática, o Diagrama de Máquina de Estados é recomendado, principalmente, para modelar os estados dos objetos das classes.



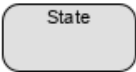
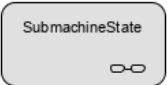
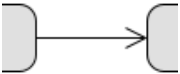

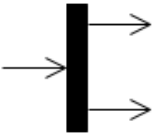
Segundo Guedes (2018), o Diagrama de Máquina de Estados demonstra o comportamento de um elemento, geralmente uma instância de uma classe, a partir de um conjunto finito de transições de estado. No entanto, pode-se usá-lo para modelar também o comportamento de um caso de uso. Conforme Bezerra (2014), esse diagrama permite descrever o ciclo de vida de objetos de uma classe com a indicação dos eventos que causam as transições entre os estados, a partir da realização das operações dos objetos.

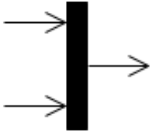
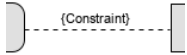






ELEMENTOS DO DIAGRAMA DE MÁQUINA DE ESTADOS


Um Diagrama de Máquina de Estados é representado, basicamente, pelos elementos: estado inicial, estados, transições de estados e estado final, sendo que este não é obrigatório e, também, pode conter vários estados finais no mesmo

diagrama. A notação gráfica de todos os elementos que podem compor o Diagrama de Máquina de Estados e uma breve descrição de cada elemento são apresentadas no Quadro 3.1 a seguir.

Quadro 3.1 | Elementos do Diagrama de Máquina de Estados

Notação	Elemento
	Estado Inicial (<i>Inicial State</i>) representa o estado de um objeto quando ele é criado, indicando o estado padrão que ele assumirá. Só pode haver um estado inicial na máquina de estados.
	Estado Final (<i>Final State</i>) representa o fim do ciclo de vida de um objeto. Este estado é opcional e pode haver mais de um na máquina de estados.
	Estado (<i>State</i>) representa uma situação de existência dos objetos de uma classe, durante a qual ele satisfaz alguma condição ou realiza alguma atividade.
	Estado de Submáquina ou Estado Composto (<i>Submachine State</i>) indica que um estado contém internamente dois ou mais estados com suas transições, gerados independentes ou não. É uma forma de simplificar a representação da máquina de estados a partir do detalhamento de um estado principal.
	Transição (<i>Transition</i>) representa um relacionamento entre dois estados, indicando a mudança de estado a partir da ocorrência de um evento.
	Escolha (<i>Choice</i>) representa um ponto na transição de estados de um objeto em que deve ser tomada uma decisão. As transições de um estado de escolha devem ser indicadas com uma condição de guarda.
	Barra de Sincronização com Bifurcação (<i>Fork</i>) representa a ocorrência de estados paralelos, causados por transições concorrentes. Esta barra indica que dois ou mais processos paralelos estão sincronizados em um determinado momento do processo.

Notação	Elemento
	Barra de Sincronização com Junção (<i>Join</i>) representa a ocorrência de estados paralelos, causados por transições concorrentes. Esta barra indica o momento em que dois ou mais subprocessos se uniram em um único processo.
	Condição (<i>Constraint</i>) indica uma restrição representada em linguagem natural no formato de texto, com o propósito de declarar parte da semântica de um elemento.
	Ponto de Entrada (<i>Entry Point</i>) indica a entrada de uma máquina de estado ou estado composto.
	Ponto de Saída (<i>Exit Point</i>) indica a saída de uma máquina de estado ou estado composto.
	Vértices de Junção (<i>Junction</i>) são usados para encadear várias transições, indicando a construção de caminhos de transição compostos entre estados.
	Histórico Superficial (<i>Shallow History</i>) representa uma transição para um estado histórico superficial de estados compostos, chamando o estado mais recente que estava ativo, ou seja, o histórico superficial guarda informações sobre o estado atual, mas não sobre os seus subestados.
	Estado Profundo (<i>Deep History</i>) representa a configuração ativa mais recente do estado composto que contém diretamente esse pseudoestado, por exemplo, a configuração de estado que estava ativa quando o estado composto foi encerrado pela última vez, ou seja, o histórico profundo guarda informações sobre o estado atual e os seus subestados.
	Terminador indica que a execução da máquina de estado por meio de seu objeto está terminada, ou seja, equivale à invocação do método de destruição o objeto.

Notação	Elemento
	Nota ou comentário é utilizado para descrever observações aos elementos da máquina de estados, contudo não expressa força semântica específica aos elementos da máquina de estado, mas pode conter informações úteis para os desenvolvedores.

Fonte: adaptado de <https://bit.ly/2QOqeq6>. Acesso em: 15 maio 2020.

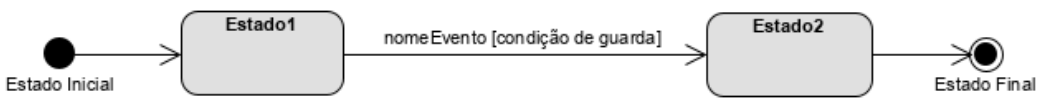
Ver anotações

O Quadro 3.1 ilustra os elementos básicos do Diagrama de Máquina de Estados na sua forma mais simplificada, ou seja, a representação dos estados apenas com seu nome e a indicação das transições de estados e dos estados inicial e final. O nome de um estado deve ser único no diagrama, iniciando com letra maiúscula. Tipicamente, usa-se um verbo no gerúndio, por exemplo, “Ativando”, “Cancelando”, “Enviando”, etc.

ATENÇÃO

É importante lembrar que, os estados e as transições de estado de um objeto constituem o seu ciclo de vida, assim, cada diagrama tem apenas um único estado inicial e pode ter nenhum, um ou vários estados finais. Cada objeto passa por um número finito de estados durante a sua existência, realizando determinadas ações durante a execução do sistema, sendo que, quando não é indicado um estado final no ciclo de vida de um objeto, isso indica que ele é contínuo.

Figura 3.2 | Elementos do Diagrama de Máquina de Estados



Fonte: elaborada pela autora.

ASSIMILE

O Diagrama de Máquina de Estados é adotado, principalmente, para modelar o comportamento dos objetos das classes, contudo nem todas as classes especificadas no Diagrama de Classes precisam de uma máquina de estados correspondente. O Diagrama de Máquina de Estados é elaborado apenas para as classes de objetos que possuem um comportamento dinâmico relevante e específico ao contexto do sistema modelado.

Na elaboração do Diagrama de Máquina de Estados, é fundamental identificar as regras de negócio aplicadas ao contexto dos objetos, para auxiliar na definição dos estados e das transições. Para identificar os possíveis estados de um objeto, deve-se analisar os valores de seus atributos, simulando a instanciação dos objetos, a partir da execução das funcionalidades do sistema, já especificadas pelos casos de uso, bem como compreender a troca de mensagens entre os objetos.

Para definir as transições entre os estados, deve-se identificar os eventos internos e externos aos objetos da classe e analisar se há algum fator que condicione a transição de estado, nesse caso, deve-se representar por meio da indicação de condições de guarda.

A elaboração do Diagrama de Máquina de Estados pode consistir na simples representação dos estados e nas transições entre eles, assim como em uma representação mais detalhada dos estados dos objetos com a indicação das atividades internas, também denominadas de ações de estado, e ainda apresentar as transições internas dos estados. De acordo com Guedes (2018), uma ação está associada a uma transição, ou seja, quando o objeto assume um novo estado ou está mudando de estado, ocasionando uma simples atribuição de um valor a um atributo. Uma atividade interna está sempre associada ao estado que o objeto assumiu, ou seja, corresponde aos métodos executados pelo objeto, porém não causam alteração na situação do estado. As ações de estado são representadas pelas cláusulas predefinidas “*entry*”, “*exit*” e “*do*” no interior do retângulo do estado, sendo:

- **Do:** representa uma atividade realizada durante o tempo em que o objeto se encontra no estado.
- **Entry:** representa as ações realizadas no momento em que o objeto assume o novo estado.
- **Exit:** representa as ações executadas quando o objeto está mudando de estado.

Os seguintes passos são recomendados para construir um Diagrama de Máquina de Estados, a partir da identificação e definição das classes (BEZERRA, 2014):

1. Identificar os estados relevantes para os objetos da classe.
 2. Identificar os eventos e as transições de estados que ele ocasiona.
 3. Identificar para cada estado as transições possíveis quando um evento ocorre, respeitando as regras de negócio do contexto do sistema.
 4. Identificar para cada estado as atividades internas relevantes.
 5. Verificar se há fatores que influenciam nos eventos que ocasionam a transição entre os estados. Se for o caso, uma condição de guarda deve ser definida para a transição, e verificar se alguma ação deve ser executada quando uma transição é disparada, indicando, assim, as ações internas do estado.
 6. Identificar os atributos que estão envolvidos na indicação de cada condição de guarda das transições de estado. Pode ser que esses atributos não foram previamente identificados. Nesse caso, deve-se adicioná-los às classes correspondentes.
-

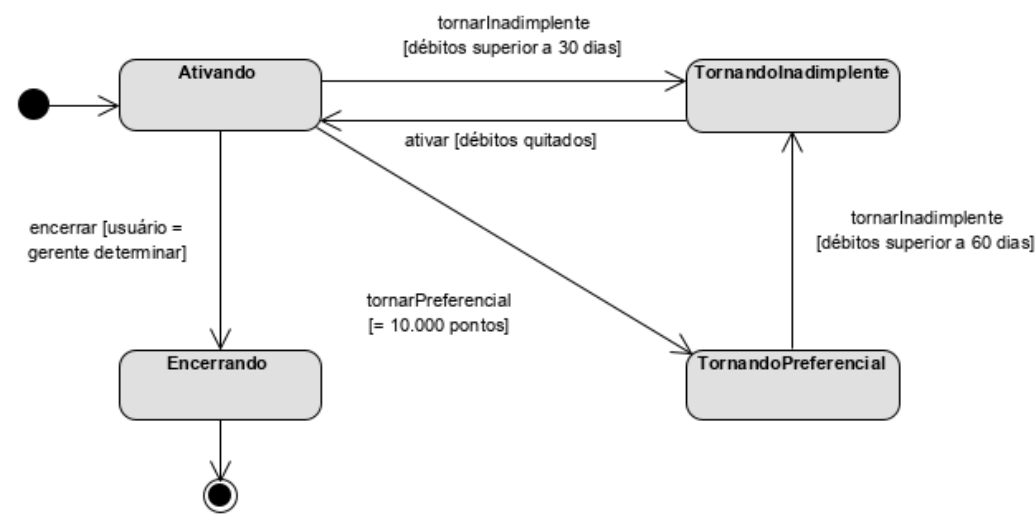
7. Definir o estado inicial e os eventuais estados finais.
8. Desenhar o Diagrama de Máquina de Estados, dispondo os estados de tal forma que o ciclo de vida do objeto possa ser visualizado de cima para baixo e da esquerda para a direita.

REFLITA

Sobre a principal indicação da modelagem do Diagrama de Máquina de Estados, ou seja, os objetos das classes que possuem estados relevantes, quem define quais estados dos objetos são relevantes ao contexto do domínio do sistema?

Vamos conferir a representação simplificada de um Diagrama de Máquina de Estados, ilustrado na Figura 3.3, correspondente aos objetos da classe “Cliente”, referentes a um sistema de uma loja virtual. Considerando as regras de negócio do contexto do sistema, foram definidos os estados Ativando, TornandoPreferencial, TornandoInadimplente e Encerrando. O diagrama representa estados simples apenas com o nome e suas transições de estados. Observe que, nas transições de estados, junto ao nome dos eventos, foi indicada uma condição de guarda específica às regras e particularidades do contexto do sistema, a qual indica que cada objeto só assumirá o novo estado se a condição de guarda for verdadeira.

Figura 3.3 | Diagrama de Máquina de Estados da classe Cliente

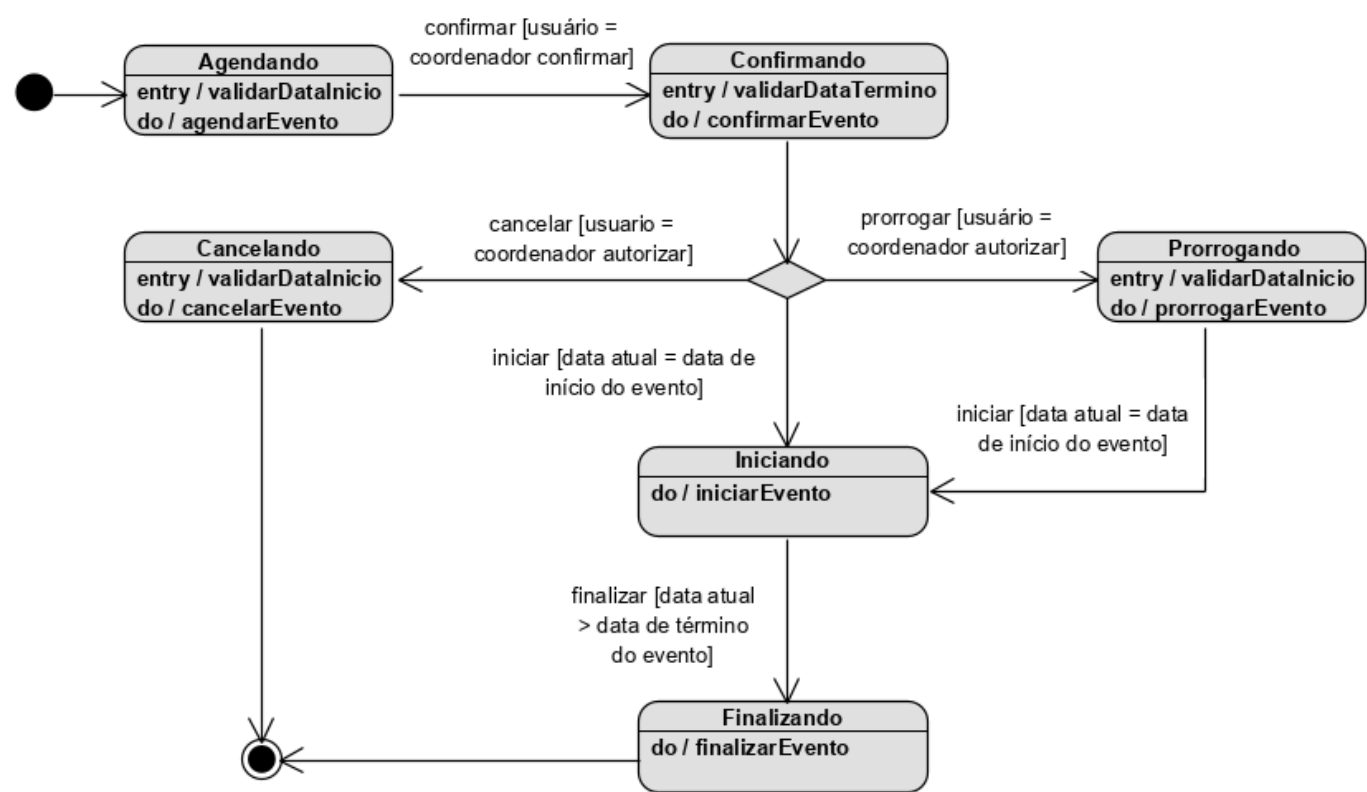


Fonte: elaborada pela autora.

Agora, confira, na Figura 3.4, o Diagrama de Máquina de Estados correspondente aos objetos da classe “Evento” com os estados “Agendando”, “Confirmando”, “Iniciando”, “Cancelando”, “Prorrogando” e “Finalizando”, referentes a um sistema de controle de eventos científicos. Observe que todos os estados possuem a ação de estado indicada pela cláusula “Do”, que corresponde à operação nominada ao estado que o objeto assumirá durante a execução do sistema. Cada ação de estado do tipo “Do” deve ser listada como uma operação na classe especificada na versão final do Diagrama de Classes, geralmente, o da atividade de projeto. As condições de guarda indicadas nas transições de estados estão de acordo com regras do contexto do sistema, as quais devem ser implementadas posteriormente, a fim de

assegurarem a consistência dos objetos do sistema. O estado inicial “Agendando” possui a ação de entrada (cláusula “Entry”) “validarDataInicio”, usada se no momento da instanciação de um novo evento a data inicial indicada é futura à data atual do sistema. A mesma consistência deve ser realizada quando o objeto assumir os estados “Confirmando”, “Cancelando” ou “Prorrogando”, a partir das transições de estados estabelecidas sob a condição de autorização do coordenador do evento, ou seja, as transições serão realizadas mediante um evento disparado por um ator (coordenador) que interage com o sistema. Por fim, uma vez que um objeto assumir os estados “Cancelando” ou “Finalizando”, encerra-se o ciclo da máquina de estados dos objetos da classe “Evento”.

Figura 3.4 | Diagrama de Máquina de Estados da classe Evento



Fonte: elaborada pela autora.

EXEMPLIFICANDO

Na modelagem de sistemas de informação para comércio eletrônico (e-commerce) de diferentes segmentos de lojas virtuais, bem como de sistemas de gerenciamento de logística para o monitoramento de entregas, é imprescindível a adoção do Diagrama de Máquina de Estados para especificar fielmente os estados e as transições de estados dos objetos e, posteriormente, implementar todas as operações correspondentes às ações de estados. Na perspectiva da visão dos usuários de um sistema em execução, a mudança de estado dos objetos reflete no gerenciamento da “situação” das transações realizadas, principalmente de forma on-line e remota. No exemplo citado de monitoramento de entregas, o próprio cliente consegue acompanhar o rastreamento de uma entrega pela sua situação, geralmente sendo: encaminhado para o centro de distribuição,

em processo de coleta, recebido no centro de distribuição, transferência para a cidade destino, recebido no centro de distribuição da cidade destino, separado para o roteiro de entrega e processo de entrega e entrega realizada. Esses são os estados de um objeto.

Considerando as técnicas de modelagem comportamentais da UML, o Diagrama de Máquina de Estados é fundamental para demonstrar o comportamento do conjunto de estados e transições de estados dos objetos das classes especificados no Diagrama de Classes, para assim compor a documentação da atividade de análise de um sistema orientado a objetos condizente com o que será implementado.

REFERÊNCIAS

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2014.

BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Campus, 2006.

GUEDES, G. T. A. **UML: uma abordagem prática**. 3. ed. São Paulo: Novatec, 2018.

PRESSMAN, R.; MAXIM, B. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2018.

RUMBAUGH, J. *et al.* **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1997.