

NÃO PODE FALTAR

MODELAGEM DE CASOS DE USO

Maurício Acconcia Dias

DIAGRAMA DE CASOS DE USO

Diagrama de fácil entendimento sobre o sistema, utilizado para apresentar a ideia geral do projeto aos donos do software, pois descrevem um conjunto de ações que o sistema deve executar em conjunto com os usuários externos.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

CONVITE AO ESTUDO

Bem-vindo ao início do estudo da construção e da utilização dos diagramas da linguagem UML, a qual é utilizada para a modelagem de processos de software em forma de diagramas. Nesta unidade serão mostrados diferentes tipos de diagramas UML e, para isso, será preciso tratar, basicamente, dos principais componentes do sistema e de como eles se relacionam. Por exemplo, ao ir até o caixa de um banco pagar uma conta, você seria o usuário que interage com a pessoa que manipula o caixa (atores do sistema). Em seguida, você irá pagar a conta (transação que ocorre ou um caso de uso, mais especificamente), conta esta que pode ser paga com dinheiro, com cartão ou com seu próprio saldo do banco. Se essa relação fosse modelada, todos os componentes deveriam estar presentes. Para isso, temos os diagramas de casos de uso, que apresentarão essas relações;

além disso, temos também o diagrama de classes, que apresentará as classes utilizadas para modelar os componentes desse sistema, como a conta bancária, o que será pago, a transação do pagamento, entre outros itens.

Aprender sobre os diagramas e sobre suas principais características é importante, porém outra parte fundamental do estudo de UML é saber como construir um determinado diagrama e como utilizá-lo para obter o melhor resultado no processo de desenvolvimento de sistemas. Lembre-se de que a linguagem UML é uma ferramenta para a modelagem de sistemas, uma tarefa primordial e essencial a ser realizada antes do desenvolvimento de sistemas. Sendo assim, a construção dos diagramas é imprescindível para que, ao final do processo do desenvolvimento de software, os requisitos sejam atendidos e o produto final satisfaça as necessidades do cliente.

Ao estudar o conteúdo desta unidade, você será capaz de trabalhar com três dos principais diagramas existentes na linguagem UML: o **diagrama de casos de uso**, responsável por descrever um conjunto de ações que os sistemas devem executar em conjunto com usuários externos ao sistema; o **diagrama de classes**, que será utilizado para modelar a estrutura do sistema a ser desenvolvido considerando o nível de classes e interfaces, e, por fim, o **diagrama de atividades**, que apresenta o fluxo de controle dos objetos com ênfase na sequência em que os eventos ocorrem e nas condições para que ocorram.

Esses diagramas são amplamente utilizados para a modelagem de sistemas e, ao adquirir o conhecimento de quando e como construí-los da forma correta, você perceberá que eles são de grande importância para a aplicação no mercado de trabalho. O desenvolvimento de um software, em geral, inicia-se pelo levantamento de requisitos, no qual teremos listado tudo o que o software deverá fazer, ou seja, as suas funcionalidades. Por exemplo, um software para controle de vendas deve ter requisitos sobre as operações a serem realizadas (compra, venda, troca), sobre como elas devem ser feitas e sobre as informações importantes para a empresa. Os requisitos de usuário são utilizados para a construção dos diagramas de casos de uso. Além disso, a partir dos requisitos obtidos e dos diagramas de caso de uso gerados, posteriormente, serão construídos os demais diagramas da UML. Esse diagrama é comumente utilizado pois o cliente normalmente é capaz de interpretá-lo e de identificar problemas nas relações. O diagrama de classes serve de base para todo o desenvolvimento do software, pois representa a estrutura do sistema como um todo. As classes são as unidades fundamentais do software e tudo que envolve sua implementação está representado no diagrama. Por fim, para a correta análise da relação de objetos, é necessário construir o diagrama de atividades.

Com o conhecimento apresentado nesta unidade, será possível compreender melhor a importância da linguagem UML e aprender como utilizá-la na construção de três de seus principais diagramas. Vamos lá? Bons estudos!

PRATICAR PARA APRENDER

Bem-vindo à seção de Modelagem de casos de uso. Nela você será apresentado aos conhecimentos necessários para a construção dos diagramas de casos de uso. A linguagem UML é uma ferramenta amplamente utilizada em empresas de desenvolvimento de software para a modelagem de sistemas. Ela possui quinze diagramas, que, juntos, apresentam diversas visões do software em desenvolvimento, auxiliando durante o processo e melhorando o resultado final. Além disso, os times envolvidos no desenvolvimento do software são capazes de obter as informações de forma coerente, diminuindo muito os problemas de interpretação das informações sobre o projeto.

Todo projeto de desenvolvimento de software inicia-se pela fase de obtenção dos requisitos. Os requisitos de software estão concentrados em um documento que possui tudo que o cliente espera do produto final. Nesse sentido, o primeiro diagrama UML desenvolvido é o diagrama de casos de uso, que irá modelar todas as possíveis utilizações do sistema de uma forma simples e de fácil entendimento. O diagrama de casos de uso é utilizado inclusive em reuniões com o cliente para verificação.

Sem as informações consistentes do diagrama de casos de uso fica mais complexa a tarefa de desenvolver os outros diagramas, já que é necessário revisar sempre o documento de requisitos para verificação e validação.

Com um diagrama de casos de uso bem feito, é possível obter êxito no desenvolvimento de vários outros diagramas da linguagem UML.

Nesta seção você irá aprender os principais conceitos do diagrama, como identificar seus componentes, como fazer a documentação suplementar dos casos de uso de forma correta e completa e, por fim, irá aprender a elaborar diagramas com estudos de caso de problemas reais.

Você foi contratado por uma empresa de desenvolvimento de software e, ao iniciar seus trabalhos, percebeu que a empresa passava por dificuldades com os projetos. Os prazos sempre eram perdidos, o software final possuía uma taxa de erros muito elevada e, no final das contas, as correções e manutenções do software consumiam muito tempo dos programadores.

Após perceber toda essa situação e se atentar para o fato de que a empresa não utilizava nenhuma linguagem de modelagem de software, você propôs a seus diretores uma apresentação sobre a linguagem UML, seus benefícios, seus

principais diagramas e como essa linguagem poderia solucionar diversos problemas encontrados por você no processo de desenvolvimento de software da empresa.

Os donos da companhia concordaram e então você pôde fazer uma série de apresentações. Todos da empresa gostaram da ideia e, após esse momento, a linguagem UML passou a integrar o desenvolvimento de softwares logo no primeiro projeto da empresa do qual você estava fazendo parte como desenvolvedor.

A tarefa designada ao projeto é desenvolver um sistema de gerenciamento para um novo banco, porém, como se tratava de uma instituição recém-criada, os proprietários da instituição propuseram que a empresa de desenvolvimento que você trabalha apresentasse uma primeira versão dos requisitos, a fim de que eles tivessem uma ideia das funcionalidades para, então, modificarem e acrescentarem outras necessidades de acordo com o sistema desejado.

Você achou estranho pois sabe que um documento de requisitos não seria a melhor opção de apresentação do sistema, e criar um diagrama de casos de uso sem um conjunto de requisitos seria uma tarefa complexa. Então teve a seguinte ideia: procurar por sistemas de bancos já existentes e tentar obter um diagrama de casos de uso que apresentasse ao menos as funcionalidades básicas esperadas para esse tipo de sistema.

Sendo assim, sua tarefa é apresentar um diagrama de casos de uso de um sistema que irá gerenciar um banco simples, o qual só possui contas correntes de clientes de pessoas físicas e jurídicas. Bom trabalho!

DICA

Pesquise sobre sistemas de gerenciamento de bancos e suas funcionalidades.

Pronto para conhecer melhor o diagrama de casos de uso e suas características?
Bons estudos!

CONCEITO-CHAVE

A linguagem UML é uma poderosa ferramenta de modelagem e pode ser aplicada em diversas etapas do desenvolvimento de software com o objetivo de obter um resultado melhor ao final do processo. A criação dos diagramas possibilita o detalhamento visual de diversos aspectos do software, auxiliando a equipe de

desenvolvimento e reduzindo os erros provenientes do entendimento equivocado sobre os aspectos do sistema. Com isso, o uso da UML é amplamente utilizado no processo de desenvolvimento de software de muitas empresas.

Apesar de apresentar os resultados mencionados, é importante que os diagramas sejam construídos de maneira correta para que as informações necessárias sobre o sistema estejam neles expressas. Nesta seção iremos abordar o diagrama de casos de uso, que é um diagrama simples e importante no desenvolvimento do software por apresentar o comportamento do sistema em relação a seus possíveis usuários.

■ DIAGRAMA DE CASOS DE USO

Normalmente o diagrama de casos de uso é construído após o levantamento dos requisitos dos usuários, pois nesse momento os desenvolvedores possuem todas as informações necessárias para sua elaboração. Além disso, esse diagrama é uma ferramenta visual de fácil entendimento sobre o sistema, podendo, em muitos casos, ser utilizada para apresentar a ideia geral do projeto aos donos do software.

De forma sucinta, é possível dizer que o diagrama de casos de uso captura a funcionalidade e os requisitos do sistema usando **atores**, **casos de uso** e os **relacionamentos** entre eles (RUBAUGH; JACOBSON; BOOCH, 2004).

- **Atores:** são os usuários do sistema, representam quem irá interagir com ele.
- **Casos de uso:** representam as diferentes formas através das quais o usuário pode interagir com o sistema, ou seja, os modos como cada usuário poderá utilizá-lo.
- **Relacionamentos:** representam como cada caso de uso interage com outros casos de uso ou com os atores.

Os casos de uso modelam os serviços, tarefas e funções que um sistema precisa executar. Representam, também, funcionalidades de alto nível e como um usuário manipula o sistema. Vamos agora conhecer os principais componentes desse diagrama e o que eles representam.

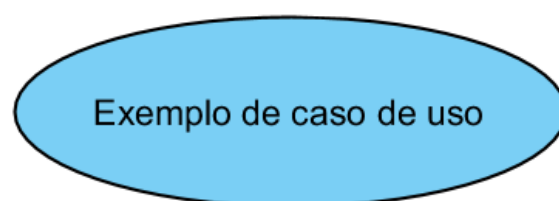
O diagrama de casos de uso tem definições complementares com relação à sua natureza. Nas definições da linguagem UML de 2.0 a 2.4, era considerado tanto um diagrama comportamental quanto estrutural (UML-DIAGRAMS, 2016). A classificação estrutural do diagrama de casos de uso era definida com base em sua relação com o diagrama de classes, que é estrutural. Já na UML 2.5, o diagrama de

casos de uso foi retirado do conjunto de diagramas comportamentais e colocado em conceitos suplementares, criando um dilema em sua classificação (UML-DIAGRAMS, 2016).

■ CASO DE USO

O primeiro componente do diagrama que iremos estudar é o “caso de uso”. Representado por uma elipse preenchida pelo nome do caso de uso, como apresentado na Figura 2.1, é usado para representar funcionalidades de alto nível e para demonstrar como o usuário manipulará o sistema. Um caso de uso representa uma funcionalidade diferente de um sistema, componente, pacote ou classe. Segundo as regras da UML, um caso de uso deve ser nomeado obrigatoriamente, porém nenhuma definição sobre como isso deve ser feito é apresentada, deixando a tarefa a cargo do criador do diagrama. Entretanto, é recomendado pelas diretrizes da linguagem que se utilize uma frase curta no formato VERBO+SUBSTANTIVO, como “sacar dinheiro”.

Figura 2.1 | Notação de caso de uso

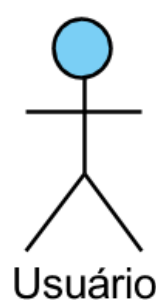


Fonte: elaborada pelo autor.

ATOR

O segundo componente do diagrama de casos de uso que iremos discutir é o ator, componente de hardware ou software, externo ao sistema, com quem interage. Esse componente é chamado de ator por representar um papel de entidade externa ao software e por interagir com o sistema. Um usuário seria o melhor exemplo de ator, sendo este uma entidade que inicia o caso de uso fora do escopo de um caso de uso. Pode ser qualquer elemento que acione uma interação com o caso de uso. Um ator pode ser associado a vários casos de uso no sistema. A notação desse elemento na UML é apresentada na Figura 2.2.

Figura 2.2 | Notação de ator



Fonte: elaborada pelo autor.

As especificações da UML, até a UML 2.5, exigem que a funcionalidade do caso de uso seja iniciada por um ator. Na UML 2.5, isso foi removido, o que significa que pode haver algumas situações em que a funcionalidade do sistema seja iniciada pelo próprio sistema, enquanto ainda fornece resultados úteis a um ator. Por exemplo, o sistema pode notificar um cliente que o pedido foi enviado, agendar limpeza e arquivamento de informações do usuário, solicitar algumas informações de outro sistema etc. (UML-DIAGRAMS, 2016).

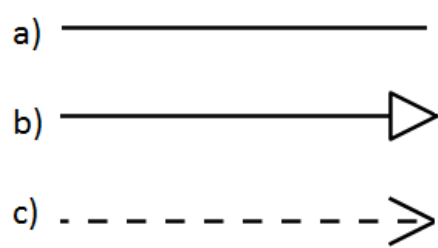
ASSIMILE

É importante destacar que o diagrama de casos de uso é o primeiro a ser construído no processo de modelagem de sistemas, e as informações para a sua construção são obtidas diretamente do documento de requisitos. Imagine um requisito de um sistema de vendas que indique que “o operador deve poder escolher entre um pagamento de cartão de crédito ou dinheiro”. Nessa circunstância será criado um caso de uso no qual o ator seja o operador do caixa, usuário do sistema, e em que existam dois casos de uso. Os nomes dos casos de uso, seguindo o padrão da UML, poderiam ser “pagamento cartão” e “pagamento dinheiro”.

RELACIONAMENTOS

Outros componentes de um diagrama de casos de uso são os seus relacionamentos. Os relacionamentos expressam o tipo de interação entre outros componentes do diagrama (atores e casos de uso). Na figura 2.3 são ilustradas as notações dos três principais tipos de interações.

Figura 2.3 | Relações do diagrama de casos de uso



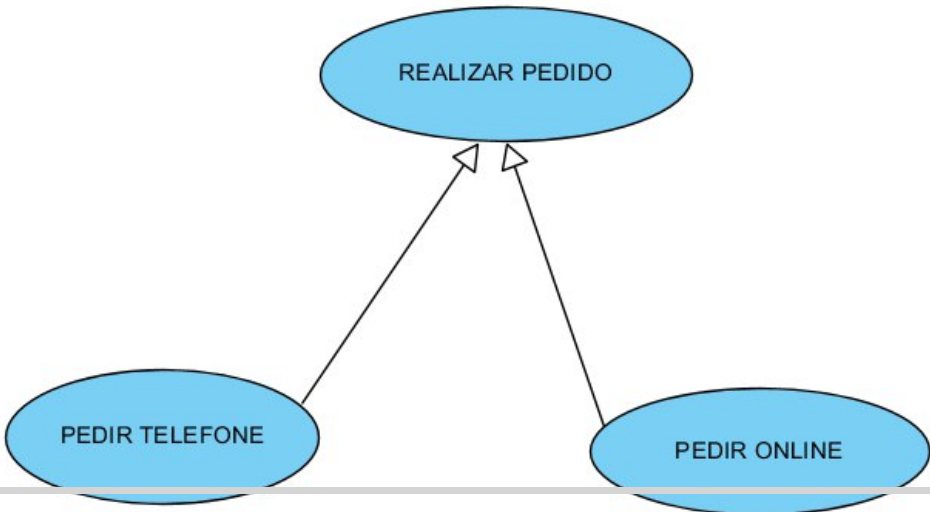
Legenda: a) associação; b) generalização; c) extensão/inclusão.

Fonte: elaborada pelo autor.

A **Associação** indica um relacionamento ou a comunicação entre um ator e um caso de uso. A notação da associação é uma linha que liga os elementos relacionados, a qual pode ser vista na Figura 2.3(a). É importante destacar que o relacionamento de associação pode ser direcionado pela inclusão de uma seta que indica a direção na linha da associação. Importa destacar que não existe no modelo um caso de uso que inicia por dois atores.

A **generalização** entre casos de uso é a mesma relação que se tem de generalização de classes. Na questão da herança de classes, a classe filha herda as características da classe pai, sendo que a classe filha possui todos os seus atributos e métodos visíveis. A representação nos casos de uso define que um caso de uso filho herda propriedades e o comportamento do caso de uso pai e deve inclusive redefinir seu comportamento. Imagine a situação de um login em um determinado sistema, ele pode ser feito de diversas formas, como: digitando o login e a senha, utilizando a função “lembrar” do navegador, ou inscrevendo-se na página. Nesse cenário teríamos um caso de uso pai, chamado de autenticação, e casos de uso filho, utilizados para redefinir a forma de autenticação do caso de uso pai (UML-DIAGRAMS, 2016). Considere o seguinte exemplo: em um sistema de pedidos, há duas maneiras de se realizar um pedido, ou pela internet ou pelo telefone. Independentemente de qual meio foi utilizado para realizar esse pedido, ele deve ser feito da mesma forma. A Figura 2.4 apresenta esse exemplo com os casos de uso e sua generalização.

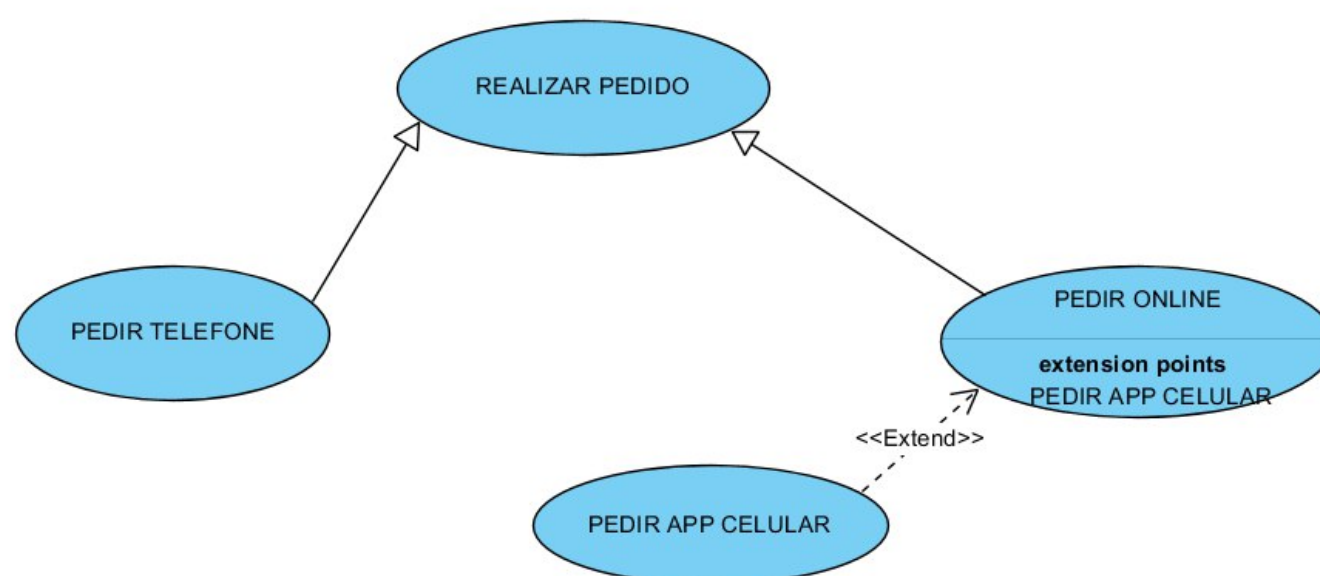
Figura 2.4 | Exemplo de generalização de casos de uso



Fonte: elaborada pelo autor.

O relacionamento do tipo **“extend”** de um caso de uso indica um comportamento opcional/adicional, ou seja, é executado em apenas determinadas situações. Esse tipo de relacionamento especifica como e quando o comportamento definido no caso de uso extensivo pode ser inserido no comportamento definido no caso de uso estendido (UML-DIAGRAMS, 2016). Trata-se de um comportamento adicional a um caso de uso, sem criação de dependência. Porém, o caso de uso extensivo não faz sentido se analisado isoladamente. Um exemplo geral é a opção de ajuda em qualquer parte do sistema: imagine um sistema que possua uma forma correta de preencher um formulário e possua um botão de ajuda no preenchimento. Esse caso de uso “ajuda no preenchimento” pode estender o caso de uso “preencher formulário”, já que não torna o caso de uso principal dependente, mas apenas complementar à sua funcionalidade. Repare que, se analisarmos o caso de uso extensivo “ajuda no preenchimento” isoladamente, ele não fará sentido. Considerando o exemplo anterior e adicionando um caso de extensão, é possível que um pedido on-line seja feito de um computador pessoal ou do aplicativo do celular. A Figura 2.5 apresenta a extensão necessária para modelar o comportamento do pedido on-line, considerando que a forma “padrão” seria pedir de um computador.

Figura 2.5 | Exemplo de relação de extensão

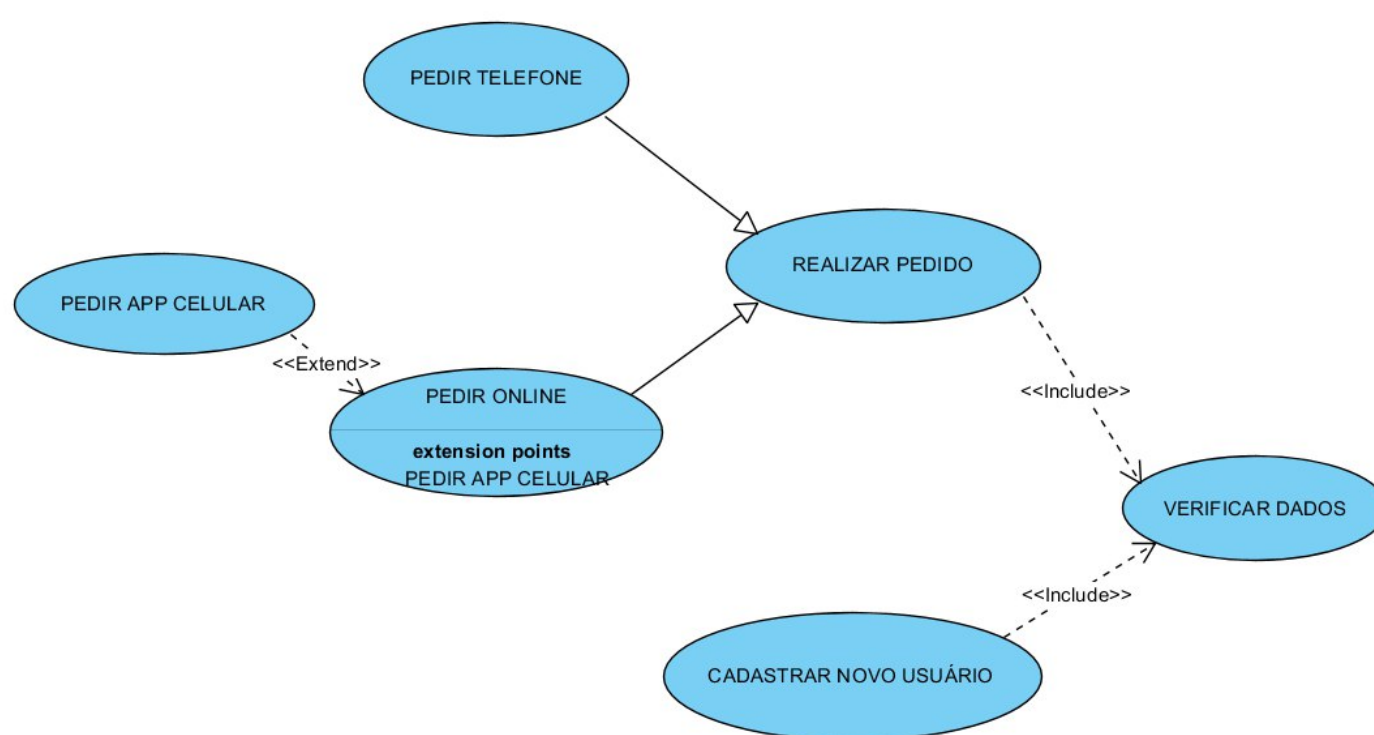


Fonte: elaborada pelo autor.

O relacionamento do tipo **“include”** (inclusão) é utilizado para representar comportamentos parciais que são comuns para outros casos de uso, ou seja, são divididos em casos de uso menores que compartilham funcionalidades. Outro exemplo de utilização é quando deseja-se decompor um caso de uso complexo. Em casos de usos menores e mais simples, esse tipo de tratativa ajuda no entendimento do caso de uso. A relação de inclusão indica obrigatoriedade, a execução de um primeiro obriga a execução de um segundo caso de uso associado.

O comportamento do caso de uso a ser incluído deve ser inserido no comportamento do caso de uso inicial. Imagine o sistema de um supermercado, no qual, tanto para pagar um produto quanto para saber seu preço, é preciso escanear o código de barras. Sendo assim, é interessante definir um caso de uso “escanear item” para ser incluído em outros dois casos de uso: “pagamento” e “consulta preço” (UML-DIAGRAMS, 2016). Considere ainda que, no exemplo do pedido realizado, exista uma funcionalidade do sistema que verifica dados; isso fará com que o pedido tenha seus dados verificados, assim como o cadastro de um novo usuário no sistema. A Figura 2.6 apresenta um diagrama com a representação desses casos de uso.

Figura 2.6 | Exemplo de relacionamento de inclusão



Fonte: elaborada pelo autor.

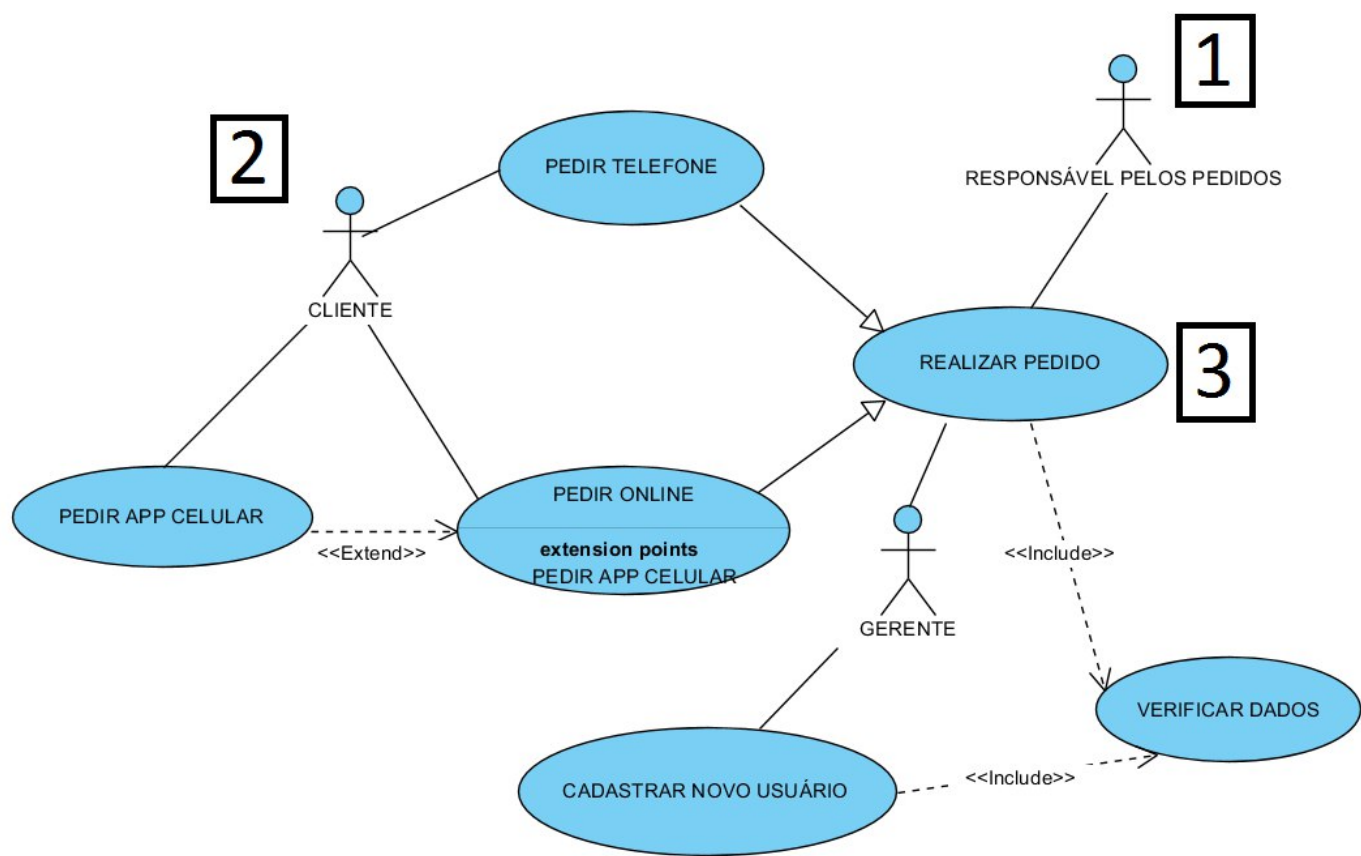
Analisando, ainda, o mesmo problema do pedido e colocando um contexto mais abrangente, foi possível identificar possíveis relacionamentos de generalização, extensão e inclusão. Após analisar os relacionamentos entre casos de uso, vamos verificar a relação entre atores e casos de uso. A associação é um relacionamento que envolve um ator e pode ocorrer de três maneiras distintas.

Considere, agora, a inclusão de três atores no diagrama da Figura 2.6 e suas relações de associação. Pode haver uma associação entre:

- Um ator e um caso de uso (caso 1 presente na Figura 2.7).
- Um ator e múltiplos casos de uso (caso 2 presente na Figura 2.7).
- Múltiplos atores e um caso de uso (caso 3 presente na Figura 2.7).

De acordo com a UML, deve-se ter no mínimo um ator associado a um caso de uso.

Figura 2.7 | Exemplos de associações com atores



Fonte: elaborada pelo autor.

Também é possível que um ator seja associado a diversos casos de uso, por exemplo, quando um usuário utiliza um programa de edição de texto e existem diversas opções de utilização. Nesse caso, o ator que representa o usuário no diagrama desse sistema vai estar associado a todos os possíveis casos de uso. Por fim, considere uma empresa que possui um sistema que classifica tipos diferentes de usuário. Todos estes tipos de usuário irão utilizar o mesmo caso de uso de login configurando a associação de múltiplos atores e um caso de uso.

O relacionamento de generalização também pode ser aplicado a atores. A generalização de um ator significa que ele pode herdar o papel de outro ator, herdando inclusive todos os casos de uso do pai. Também é possível definir mais casos de uso específicos para o ator filho.

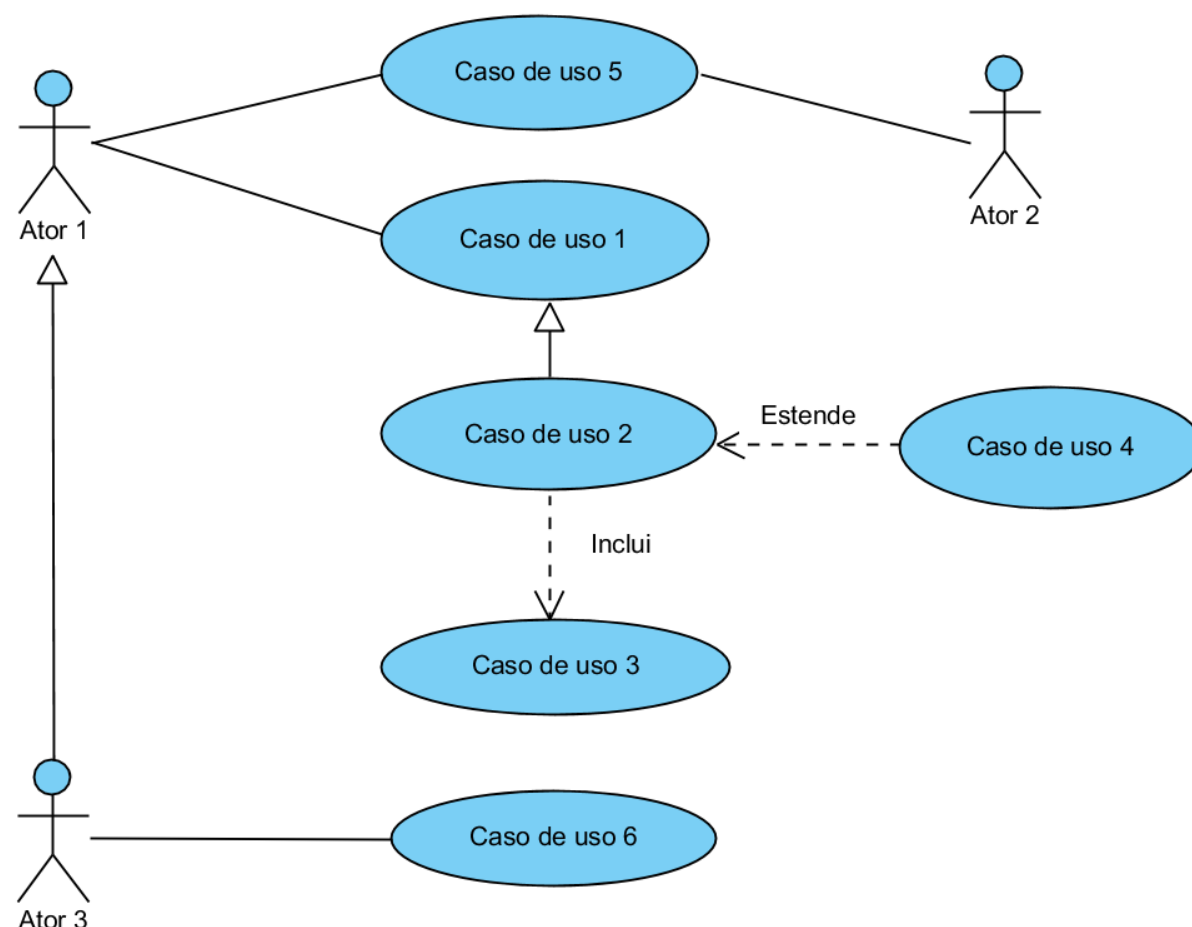
EXEMPLIFICANDO

Foram definidos até aqui os componentes básicos de um diagrama UML de casos de uso, sendo eles:

- Os atores.
- Os casos de uso.
- Os relacionamentos – associações, generalizações, inclusões e extensões.

Sendo assim, é possível construir um exemplo de diagrama de casos de uso com as relações apresentadas anteriormente.

Figura 2.8 | Exemplo de diagrama de casos de uso



Fonte: elaborada pelo autor.

Na Figura 2.8 são apresentados exemplos de todos os relacionamentos e componentes apresentados até o momento:

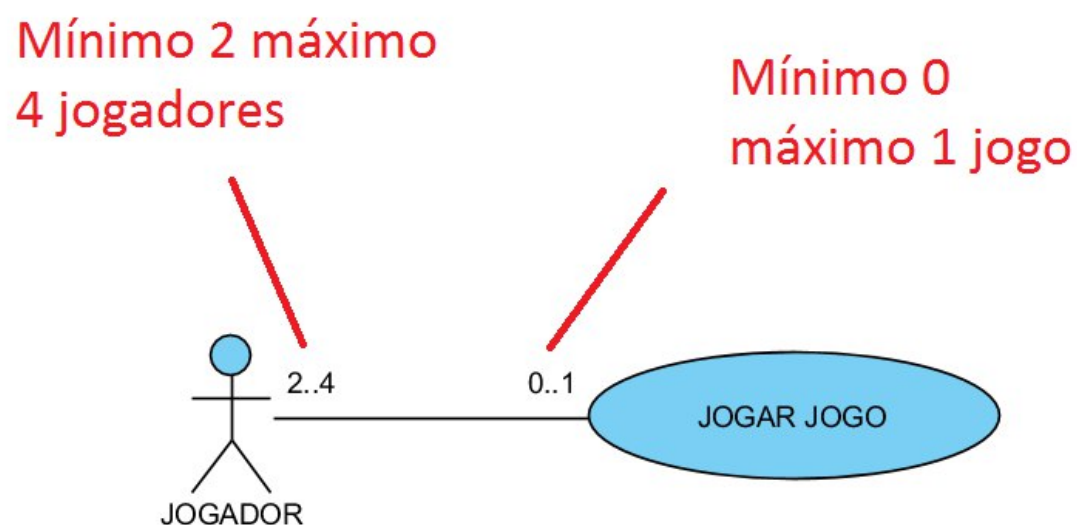
- Generalização entre os casos de uso 1 e 2.
- Generalização entre os atores 1 e 3.
- Inclusão do caso de uso 3 no caso de uso 2.
- Extensão do caso de uso 2 pelo caso de uso 4.
- Associação simples do ator 3 com o caso de uso 6.
- Associação múltipla dos atores 1 e 2 com o caso de uso 5.
- Associação múltipla do ator 1 com os casos de uso 1 e 5.

■ MULTIPLICIDADE

Os diagramas de caso de uso também podem incluir a questão da multiplicidade de associações. Nessa situação é possível indicar que mais de um ator esteja executando determinado caso de uso ao mesmo tempo. Um bom exemplo disso é a questão de jogos. Imagine o ator jogador e o caso de uso jogar um jogo. Cada jogador pode estar jogando nenhum jogo ou um jogo. Cada jogo pode ter um número determinado de jogadores. Um exemplo desse tipo de relação aplicada ao diagrama de casos de uso é ilustrado na Figura 2.9, na qual um jogo pode ser

jogado por, no mínimo, dois e, no máximo, quatro jogadores, e um jogador pode estar jogando um ou nenhum jogo. A forma correta de indicar multiplicidade está indicada na Figura 2.9.

Figura 2.9 | Exemplo de multiplicidade



Fonte: elaborada pelo autor.

Caso exista apenas uma possibilidade, ao invés de utilizar a notação “n..n”, utilize apenas um número. Por exemplo, se o jogo aceitar apenas um jogador, coloque “1” no lugar do “2..4” da Figura 2.9.

Existem basicamente dois tipos de requisitos: os **funcionais** e os **não funcionais**. Os requisitos funcionais são relativos às funcionalidades e informações do sistema, mostrando o que deve ser feito. Os requisitos não funcionais definem restrições de tempo, espaço, softwares de apoio, sistema operacional etc. (SOMMERVILLE, 2011). O diagrama de casos de uso é a melhor ferramenta para representar requisitos funcionais, porém os requisitos não funcionais são um problema. Neste caso é possível utilizar o artefato chamado documentação suplementar para os casos de uso. Essa documentação normalmente possui um formato padrão, que varia de acordo com a empresa. Um modelo desse formato pode ser encontrado em Helm ([s.d.]).

Para elaborar adequadamente um diagrama de casos de uso, é importante seguir alguns critérios básicos:

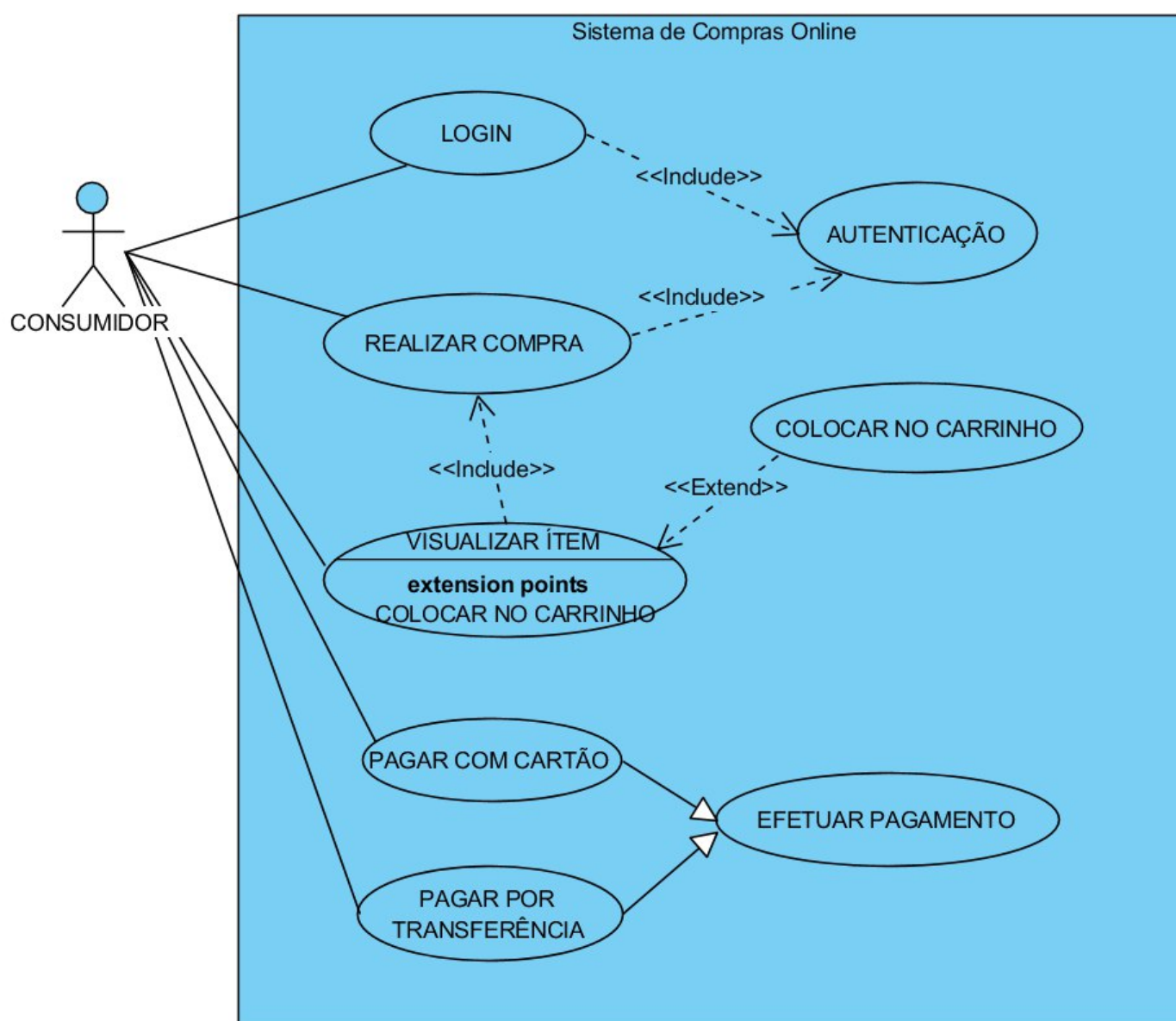
- O diagrama deve ser o mais simples possível.
- O diagrama também deve ser o mais completo possível.
- Um caso de uso deve ter todas as suas interações representadas.
- Se o diagrama é muito grande, é possível representar apenas as partes essenciais.
- Um diagrama de casos de uso deve modelar ao menos um módulo do sistema de forma completa.

Vamos agora fazer um exemplo de construção de diagramas de casos de uso. Iniciaremos com um sistema simples de biblioteca. Considere os seguintes requisitos de projeto:

- Trata-se de um sistema de compras on-line.
- Todo consumidor deve logar no sistema para fazer uma compra.
- O login passa por um sistema de autenticação.
- O consumidor pode ver os itens que deseja comprar; ao ver um item, pode comprá-lo diretamente ou colocá-lo no carrinho.
- O pagamento pode ser feito por meio de cartão ou de transferência bancária.

O conjunto de requisitos apresentados pode ser desenvolvido de várias formas possíveis. Vamos analisar a solução proposta na Figura 2.10 e entender como os relacionamentos, atores e casos de uso alocados atendem aos requisitos do projeto. O diagrama da Figura 2.10 foi construído com a ferramenta *Visual Paradigm*.

Figura 2.10 | Possível solução para o problema da compra on-line



Fonte: elaborada pelo autor.

Os casos de uso apresentados na Figura 2.10 atendem aos requisitos do projeto. A relação de inclusão do sistema de autenticação, nos casos login e realizar compra, garantem que o consumidor esteja logado. Ao ver um item, o consumidor pode