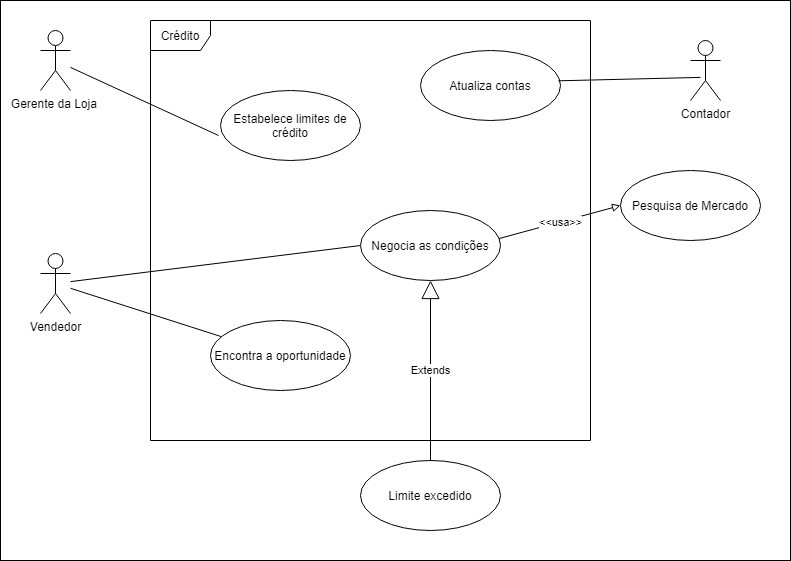
uml – casos de uso

Frank Coelho de Alcantara

# Casos de uso

Um caso de uso é a documentação de um conjunto de interações de um ator com o sistema. Este conjunto de interações deve prover um resultado para o ator na forma de valor, ele consegue algo do sistema ou consegue que o sistema faça algo por ele, ou na forma de suporte para outra atividade que irá gerar valor para o ator. Destaque-se que os casos de uso não explicitam como o sistema irá realizar este conjunto de atividades. Um caso de uso é representado, no diagrama de casos de uso por uma elipse, como pode ser visto na Figura 1.

Figura 1: Diagramas de Caso de Uso



Fonte: o autor (2020) Diagrama de caso de uso – exemplo

Na Figura 1 temos três atores e cinco casos de uso. Por exemplo, o ator *gerente da loja* interage com o sistema *estabelecendo limites de crédito.*

Os casos de uso estão fortemente relacionados com as atividades da engenharia de requisitos. A análise de requerimentos deve incluir uma descrição dos processos de interação entre atores e sistemas, explicitando o que o ator espera de cada interação e, dessa forma, além de registrar estas interações, a análise de requisito determina o valor esperado em cada interação. Casos de uso, não são ferramentas exclusivas da programação orientada a objetos ou das tecnologias de desenvolvimento ágil. Casos de uso são apenas a descrição escrita de uma, ou mais interações com o sistema. Se a amável leitora fosse criar um software para um jogo de dados o caso de uso poderia ser:

O jogador joga os dados, se o resultado for 7 ele ganha o jogo.

Neste exemplo temos um ator, o jogador, uma interação: joga os dos e um valor: ganha o jogo. Há ainda uma condição: se o resultado for 7. Esta frase, apenas uma interação com o sistema possui uma descrição completa da interação para a construção do programa e não explicita nada sobre a construção deste sistema.

A forma mais eficiente que temos de encontrar os atores é examinar a lista de atores que foram definidos durante a análise de requisitos, as mesmas entrevistas, os mesmos questionários e os mesmos eventos que usamos para entender os requisitos do sistema podem ser utilizados para a definição dos atores e seus casos de uso. Começamos, pela lista de atores principais vamos detalhando esta lista ao longo do processo.

A documentação dos casos de uso começa na análise de requisitos. Use, durante as entrevistas um modelo de documentação onde seja possível registrar:

1. **o nome do Caso de uso:** que seja uma frase verbal e imperativa;
2. **um resumo:** contendo nome, atores envolvidos e resultado esperado de forma simples, no máximo duas linhas de texto;
3. **uma descrição do caso de uso:** para que ele serve, qual o seu objetivo principal, que ator, ou atores estão envolvidos;
4. **o fluxo de informação para o sucesso:** qual ação deve ser executada e o que se espera em retorno.
5. **exceções:** uma descrição do quê pode dar errado, problemas de comportamento do usuário e possível falhas.
6. **Interface:** uma descrição da interface esperada para este determinado caso de uso.
7. **Prioridade:** que deve ser definida de comum acordo entre você, que está encarregado da análise do sistema, os desenvolvedores e os *stakeholders*.

Você verá, ao longo deste capítulo que este processo é interativo e incremental. Então, escolha uma ferramenta flexível, que lhe permita registrar os casos de uso a medida que vai progredindo no processo de análise. O sistema que você escolher para registro deve incluir, por si só, sem a sua intervenção, informações de histórico, recuperação de versões anteriores e autoria. Além disso, a plataforma escolhida para este registro deve se capaz de permitir a inclusão de material de referência, como legislação, trabalhos anteriores e exemplos.

O processo de documentação dos casos de uso não é padronizado o que permite que você escolha um processo de documentação que seja adequado as suas necessidades para cada projeto. Esta falta de padronização na documentação torna difícil definir o que é um bom caso de uso. Também não existe uma granulação padronizada, ou seja, você irá encontrar, ao longo da sua vida profissional, casos de uso de uma linha ou várias páginas. Esta última possibilidade não é desejada. Os casos de uso devem ser de entendimento fácil para todos os envolvidos nos processos de análise, desenvolvimento e aceitação do sistema.

Usar apenas uma frase também não é uma boa opção. Casos de uso devem ser descritivos e excetuando-se as interações que são comuns a todos os sistemas, uma frase não será suficiente para descrever a interação, o ator, as exceções e os objetivos que devem ser alcançados. Por fim, lembre-se, não importa quão boa seja a sua documentação. Casos de uso não são ferramentas de desenvolvimento. O conjunto de casos de uso comporá a ferramenta que você irá usar para modelar o sistema e, depois disso, desenvolver o código, interfaces, testes e ferramentas de entrega.

## Sintaxe dos casos de uso

A UML é uma linguagem e, como tal, possui uma sintaxe que precisa se obedecida para que a semântica possa ser percebida. Devemos representar os casos de uso obedecendo as regras da UML 2.51 se queremos que outros analistas e desenvolvedores possam entender o que foi documentado. Ainda que não seja restritivo, a UML preconiza o uso dos elementos de notação apresentados na Figura 2.

## Escrevendo casos de uso efetivos

O conceito de casos de uso foi definido em 1986, por Ivar Jacobson, mas foi Alistair Cockburn, em seu livro *Writing Effective Use Cases[[1]](#footnote-1)* quem popularizou o uso desta ferramenta. Efetivo é aquilo que produz resultados, aquilo que é definitivo. O objetivo de Cockburn ao escrever seu livro foi melhorar a qualidade dos produtos de software por meio do entendimento dos requisitos. Assim, em seu livro ele destaca uma sequência de quatro níveis de precisão que devem ser seguidas durante o processo de escrita destes casos e você deve começar da precisão mais baixa:

**Nível de precisão 1:** identifique o ator primário e seu objetivo;

**Nível de precisão 2:** descreva o caso de sucesso principal desta interação;

**Nível de precisão 3:** analise as possíveis extensões deste caso;

**Nível de precisão 4:** descreva os passos para o tratamento das extensões.

Esta é uma forma estruturada de explicar o processo. Durante a redação dos casos de uso responderemos a um conjunto de perguntas e, todas as vezes que conseguirmos responder uma destas perguntas progrediremos na redação dos casos de uso. Cada uma destas perguntas, permitirá que a leitora reveja seu trabalho e inclua novos detalhes na criação de casos de uso. A primeira pergunta que deve ser respondida, antes do processo de detalhamento dos casos de uso é: qual é o escopo?

### Qual é o Escopo?

Precisamos responder esta pergunta entendendo qual é o objetivo geral do sistema que será desenvolvido, determinar quais são os objetivos que precisam ser satisfeitos para atingir o objetivo geral do sistema, identificar um objetivo principal para cada interação com o sistema e identificá-lo com uma sentença verbal, imperativa. Logo em seguida, devem ser determinados os níveis de cada ator envolvido (primário, secundário) e suas características. Esta informação precisa ser validada com os *stakeholders*.

Na prática, esta deve ser um processo ágil, interativo e incremental. Você irá precisar de um, e somente um, caso de uso bem documentado relativo a cada objetivo principal do sistema para dar continuidade ao processo.

Para determinar o objetivo geral do sistema, precisamos saber qual o problema que ele irá resolver, como se espera que o sistema resolva este problema e, principalmente, por que esta forma de solução é adequada. Você pode recolher estas informações durante o processo de elicitação de requisitos, mas aqui, na definição dos casos de uso, esta informação deve estar claramente explicitada em um parágrafo:

A empresa **FraldasVerdes**, uma fabricante de fraldas ecológicas precisa que todas as suas vendas sejam controladas pelo sistema **FraldasVerdes Vende**, um sistema de vendas ao consumidor final, online, móvel e acessível. Este sistema irá permitir que o usuário final da fralda ecológica entenda o processo de fabricação e possa fazer seus pedidos sem a necessidade de sair de casa.

Neste caso, temos a empresa, seu objetivo, a explicitação do ator principal e os objetivos do sistema e do ator principal. Uma vez que este objetivo principal tenha sido determinado precisamos listar os casos de uso que irão permitir que o sistema atinja os objetivos determinados. Neste momento precisaremos identificar os atores e seus casos de uso.

Vimos anteriormente que atores são pessoas, ou dispositivos, que interagem com o sistema, também vimos que cada ator representa uma função que estas pessoas, ou dispositivos, terão no sistema. No caso das **FraldasVerdes**, teremos pelo menos um ator, que poderemos chamar de cliente. Esta é a função das pessoas que vão acessar o sistema **FraldasVerdes Vende** para comprar o produto do nosso cliente. Este é o ator que tem o objetivo de comprar fraldas e podemos criar o caso de uso usando uma frase verbal: *Comprar Fraldas*.

Para cada objetivo de um ator existem dois atributos importantes, o nível do objetivo e sua categoria, ou tipo. Vamos classificar os objetivos em três níveis: sistema, interno e contexto.

Os objetivos, representados por casos de uso, em nível de sistema, são aqueles em que podem ser satisfeitos por uma interação direta com o sistema, como por exemplo, *imprimir o Extrato*, *Procurar Fraldas*, ou *Autorizar Pagamento*.

Os casos de uso que fazem parte do sistema, mas não estão diretamente relacionados com o objetivo principal, são pequenos objetivos que são usados para construir, o processo, ou para informar o sistema são classificados como Internos. Neste caso estão, por exemplo, *Logar no Sistema* e *Esvaziar o Carrinho de Compras*.

Por fim, se tivermos objetivos que envolvam outros sistemas, vamos classifica-los com contextuais. Nesta classe estão, por exemplo: *Autorizar a Compra no Cartão* e *Buscar Endereço pelo CEP*.

Em resumo se o caso de uso diz respeito a uma ação direta para um objetivo com valor definido, classificamos como **caso de uso de sistema**. Se a interação é incremental para atingir o objetivo, classificamos como **caso de uso interno** e, por fim, se a interação é entre sistemas classificamos com **caso de uso contextual**.

Quanto aos tipos, neste momento, devemos simplificar os casos de uso em três tipos: sistema, sumário, abuso e *templates*.

Os casos do tipo sistema, complementam os casos de uso do nível sistema. Estão neste tipo os casos de uso que incluem uma interação direta com o sistema e um objetivo. Os casos de uso do tipo sumário são dizem respeito a descrição de um objetivo que contém um conjunto de outros objetivos. Podemos utilizar estes casos do tipo sumário para organizar um conjunto de outros casos, focando em um objetivo principal para o conjunto e mantendo uma ordem coerente para os casos de uso internos deste objetivo. Muitas vezes a descrição destes casos de uso do tipo sumário começa com a palavra gerenciar. Por exemplo: *Gerenciar Contas de Usuário* que pode conter os casos *Criar Conta de Usuário*, *Modificar Conta de Usuário* e *Remover Conta de Usuário*.

Os casos do tipo Abuso, estão relacionados com os objetivos que não queremos no nosso sistema. Estes são os casos de uso que estarão relacionados com as interações de atores que podem prejudicar, de alguma forma, o funcionamento do sistema. Estes atores, devem ser listados e suas ações precisam ser contempladas. Neste tipo estão os casos de uso: *Roubar dados de Usuários* e *Comprar como outro Cliente*. Definir estes atores e estes casos de uso é uma boa prática de análise de sistemas.

De forma prática, em um ambiente de práticas ágeis, partimos de um caso de uso do tipo sumário e detalhamos todos os seus casos de uso, para encontrar os que são de sistema, internos e contextuais. Depois analisamos os casos de uso do tipo abuso.

### Quem quer o quê?

Esta é a pergunta que nos levará ao nível de precisão 1. As respostas para esta pergunta são originadas tanto da análise de requisitos quanto da análise de *stakeholders*. O conhecimento dos *stakeholders*, seus objetivos e anseios permite a identificação dos atores primários e formata os casos de uso de sistema tanto do tipo sumário quanto do tipo sistema. Ou seja, a partir do conhecimento dos *stakeholders* podemos definir os atores e sua interação com o sistema.

Na definição dos casos de uso precisarmos conhecer a condição do ator, antes da interação com o sistema, estas condições são chamadas de condições prévias. O sistema deve verificar estas condições antes do caso de uso ser iniciado. Em casos de uso de interação direta homem-máquina, estas condições prévias podem ser definidas no processo de interface e não precisam ser explicitadas no sistema, um bom exemplo é a condição prévia *uso do mouse*. Contudo, em casos de uso onde os atores são outros sistemas, estas condições prévias podem, ou não, ser motivo de verificação e validação, e, neste caso, o sistema precisa certificar as condições prévias antes de iniciar o caso de uso.

O conhecimento dos *stakeholders* também irá determinar o disparo, em inglês *trigger*, do caso de uso. Este disparo especifica a ação que deve ser executada para que o caso de uso seja iniciado. Pode ser um clique, a digitação de uma informação e até mesmo a ocorrência de um evento temporal como um dia, ou hora específicos.

### Como será realizado?

Até agora respondemos perguntas analisando *stakeholders* e requisitos, pensamos em atores e sua interação com o sistema. Mesmo que estes atores representem funções que serão realizadas por outros sistemas, tudo o que fizemos foi determinar o porquê das interações e quais interações são necessárias. Agora precisamos nos preocupar um pouco com o como estas interações serão realizadas. Importante, este como não tem relação com desenvolvimento, ou design de interação, ou qualquer outra função diretamente relacionada com a construção do sistema. Ao responder esta pergunta, estaremos no nível de precisão 2 e vamos começar a dar forma aos casos de uso. Desta vez, a leitora terá que interagir aumentar suas entrevistas e incluir desenvolvedores e testadores.

Ao responder esta pregunta você precisa documentar o sucesso do caso de uso. Escolha, neste momento, o caso de uso mais importante, e documente o sucesso, documente os passos que precisam ser seguidos para que a interação do ator com o sistema produza o valor desejado. E, depois de documentar, valide sua concepção com os *stakeholders*, desenvolvedores e testadores.

O sucesso de um caso de uso ocorre quando a interação entre ator e sistema produz o valor desejado para o ator. Este ó o cenário onde o mundo é rosa, o céu é azul e tudo dá certo. Não é raro que um caso de uso contenha vários cenários, mas existe apenas um cenário principal, um cenário onde o valor é produzido e entregue ao ator. Todos os outros cenários que possam existir em um determinado caso de uso serão documentados, mas não neste momento. Neste momento estamos preocupados com o principal. Nesta documentação a leitora precisará ser clara, positiva e simples. No momento, certo, veremos algumas dicas que podem ser usadas para a documentação de casos de uso.

### O que mais pode acontecer?

Você respondeu a pergunta anterior e neste processo documentou o cenário de sucesso dos seus casos de uso. Uma vez que esta documentação tenha sido validada com *stakeholders*, desenvolvedores e testadores, podemos progredir para o nível de precisão 3. A pergunta que responderá será a base do detalhamento da estória que estamos escrevendo para nosso caso de uso. Especificamente vamos olhar para coisas que podem acontecer para nos impedir de terminar este caso de uso, ou que precisam acontecer para que ele termine. A estas condições, chamamos de extensões.

Para responder esta pergunta vamos considerar, o que acontece quando o caso de uso depende de algum processo de validação, cartão de crédito, por exemplo, e esta validação falha. Ou ainda, precisamos destacar áreas onde a eficiência do sistema é crítica para o processo. Para cada caso de uso, precisamos observar quais são os limites aceitáveis de eficiência. Estes limites de eficiência irão determinar a possibilidade ou não do caso de uso chegar ao sucesso. Na internet, por exemplo, um tempo de carga de 10 segundos seria considerado excelente nos anos 2000, no final dos anos 2010, uma página web com tempo de carga maior que 2s seria penalizada pelos sites de busca. No começo dos anos 2020, todos estão buscando tempos de carga inferiores a 1s. Observe que um sistema perfeito para 2009 não seria aceitável em 2015.

Outro fator importante que precisa ser considerado neste momento são os caminhos, *paths* em inglês, alternativos para o sucesso do caso de uso. Este cenário ocorre quando o ator precisa escolher uma de várias opções durante a interação com o sistema. Estas opções devem ser documentadas como extensões. Lembre-se nós já determinamos qual era o caminho de sucesso para este caso de uso, quando fizemos isso, documentamos o principal caminho para o sucesso. Todos os outros são complementares e devem ser documentados na forma de extensões.

O ator pode não se comportar da forma desejada e, neste caso, este é um cenário que também precisa ser considerado e, se for o caso, documentado. Por exemplo, o ator, cliente, das FraldasVerdes, pode abandonar o carrinho de compras. Ou pode clicar no lugar errado.

Por fim, precisamos considerar as condições de falha no sistema. Estes são cenários importantes que na maior parte dos casos de uso são negligenciados. Aqui, há uma crítica implícita as práticas ágeis. Nenhum dos casos de uso que analisei, durante a construção dos sistemas desde 2005 consideram as condições de falha. Existe nas práticas ágeis, um conhecimento implícito, que segurança e continuidade são atribuições da equipe de desenvolvimento. Não são! São atribuições do analista de sistemas, cabe a este desenhar os casos de uso considerando possíveis falhas e indicando os cenários que devem ser obedecidos.

### Como lidamos com os casos de uso?

Esta é a última pergunta que deve ser respondida, quando a leitora chegar neste ponto, estaremos no nível de precisão 4, o mais fino de todos. Aqui, vamos trabalhar a documentação das extensões. Cada extensão realiza uma função. Estas funções podem ser simples e complexas, mas são, com certeza, indispensáveis para a conclusão do caso de uso e, consequentemente, para a geração de valor. Lembre-se que verificamos a existência de extensões a medida que fomos aumentando a precisão dos nossos casos de uso. Agora precisamos rever cada uma e, se for o caso, removê-la ou documentá-la.

Em alguns casos, vamos perceber que as interações que denominamos de extensão são, na verdade, casos de uso que não havíamos percebido. Em todos os casos, cabe a você que está fazendo a análise do sistema considerar cada uma, sua função, necessidade e interação dentro do caso de uso. Caso você se isente desta responsabilidade caberá ao desenvolvedor tomar as decisões necessárias para fazer o sistema funcionar.

Os estadunidenses têm um dito popular muito interessante: não existem almoços de graça. Com isso eles querem dizer que sempre há um preço que precisa ser pago, ainda que não o estejamos vendo. Em práticas ágeis, prega-se o uso de estórias de usuário. Basicamente um caso de uso descrito apenas com o cenário de maior sucesso. Esta prática diminui o tempo de análise e inicia o processo de desenvolvimento em um tempo mais curto. Isto é uma troca de precisão por velocidade no desenvolvimento. Ou seja, estamos trocando a possibilidade de ter o sistema imediatamente, pela possibilidade de termos falhas e erros neste sistema. Trata-se de um compromisso baseado na confiança. Confiança no *stakeholder* e deste na equipe de desenvolvimento. Se a leitora usar este processo de descrição cuidadosamente irá limitar os erros que o sistema terá, mas infelizmente irá demorar mais para colocar o sistema em produção.

# Obras Citadas

BAUSOLA, D. **Activity Diagram. zeroinfluence**, 2012. Disponivel em: http: zeroinfluence.wordpress.com/uml. Acesso em: 04 Ago. 2020.

DENNIS, A.; WIXOM, B. H.; ROTH, R. M. System Analysis and Design. Danvers, MA. USA: John Wiley & Sons, Inc., 2009.

OBJECT MANAGEMENT GROUP. **Unified Modeling Language**. Object Management Group, 2017. Disponivel em: [https://www.omg.org/spec/UML/2.5.1/PDF.](https://www.omg.org/spec/UML/2.5.1/PDF) Acesso em: 04 ago. 2020.

SOMMERVILLE, I. **Engenharia de software**. São Paulo, SP. Brasil: Pearson , 2012.

UNHELKAR, B. **Software Engineering with UML**. Boca Raton, FL. USA: Taylor & Francis Group, LLC, 2018.

WIKIMEDIA COMMONS, THE FREE MEDIA REPOSITORY. File:ClassDiagramInitial.jpg. Wikipedia, 2016. Disponivel em:[https://commons.wikimedia.org/w/index.php?title=File:ClassDiagramInitial.jpg&oldid=407262714.](https://commons.wikimedia.org/w/index.php?title=File:ClassDiagramInitial.jpg&oldid=407262714) Acesso em: 04 ago 2020.

WIKIPEDIA CONTRIBUTORS. Ivar Jacobson. Wikipedia, 2020. Disponível em: [https:](https://en.wikipedia.org/w/index.php?title=Ivar_Jacobson&oldid=944058726) [//en.wikipedia.org/w/index.php?title=Ivar\_Jacobson&oldid=944058726.](https://en.wikipedia.org/w/index.php?title=Ivar_Jacobson&oldid=944058726)Acesso em: 4 ago. 2020.

1. Em tradução livre: Escrevendo casos de uso efetivos. [↑](#footnote-ref-1)