

IDENTIFICAR REQUISITOS FUNCIONAIS DE UM SISTEMA COM USO DE DIAGRAMA DE CASOS DE USO

REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

O desenvolvimento de um sistema começa pelo entendimento do problema de negócio e o levantamento de quais requisitos esse sistema de software deve satisfazer para resolvê-lo. Não existe uma definição padronizada para requisito na literatura de Engenharia de Software. A norma ISO/IEC/IEEE (IEEE, 1990) apresenta os seguintes enunciados para explicar o que é requisito:

Uma condição ou capacidade do sistema solicitada por um usuário, para resolver um problema ou atingir um objetivo.

Uma condição ou capacidade que deve ser atendida por uma solução para satisfazer um contrato, uma especificação, um padrão ou quaisquer outros documentos formais impostos.

Documentação da representação das condições ou capacidades apresentadas nos dois anteriores.

Necessidades quantificadas e documentadas, desejos e expectativas do patrocinador, clientes e outras partes interessadas.

Os requisitos são o ponto de partida para todo o processo de desenvolvimento, sendo a base para as atividades de análise e projeto, bem como implementação e testes. De acordo com Sommerville (2011), a área de Engenharia de Software faz uma distinção, em termos de nível de detalhamento de requisitos, entre requisitos de usuário e requisitos de sistema. O primeiro expressa os requisitos abstratos de alto nível, e o segundo revela uma descrição detalhada do que o sistema deve fazer.

Requisitos de usuário são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.

SOMMERVILLE, 2011

Requisitos de sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software.

SOMMERVILLE, 2011

Em uma perspectiva mais técnica, os requisitos de software são normalmente classificados como requisitos funcionais e requisitos não funcionais.

Requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações.

SOMMERVILLE, 2011

Atenção

Requisito funcional é uma funcionalidade do sistema, é algo que o sistema deve realizar para prover um resultado para o usuário. Requisitos funcionais pressupõem interação do sistema com usuários.

Veja uma lista com alguns exemplos de enunciados dos chamados requisitos funcionais:

O usuário deve ser capaz de pesquisar por nome na base de dados disponível.

O sistema deve designar um identificador único (NUMERO_PEDIDO) para cada pedido do usuário.

O sistema deve apresentar o documento selecionado para leitura do usuário.

Perceba que, em todos os exemplos, estamos definindo atividades que o sistema deve realizar, e, em todas essas atividades, o usuário está envolvido, seja fornecendo entradas ou recebendo saídas (resultados).

A especificação de um sistema precisa considerar também como esses requisitos serão entregues. Ou seja, quais características de qualidade e quais restrições desejamos impor ao sistema. Assim, definimos os chamados requisitos não funcionais.

Requisitos não funcionais são restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

SOMMERVILLE, 2011

Requisitos não funcionais (RNF) são frequentemente mais críticos do que os requisitos funcionais (RF). Deixar de atender a um RNF pode significar a inutilização de todo o sistema. Os RNF podem afetar a arquitetura geral de um sistema em vez de apenas componentes individuais. No caso de um website, um único RNF pode originar uma série de RF relacionados, que, por sua vez, definem os serviços necessários.

Existem diversas classificações que visam a organizar e prover melhor entendimento sobre os RNF. Um exemplo desse tipo de classificação é apresentado na lista a seguir.

1-Requisitos de produto: usabilidade, eficiência, desempenho, confiança, proteção.

2-Requisitos organizacionais: ambientais, operacionais, processo de desenvolvimento.

3-Requisitos externos: reguladores, éticos, legais, contábeis.

Exemplo

Aqui estão alguns exemplos de enunciados de requisitos não funcionais:

- O sistema deve estar disponível durante todas as horas normais de trabalho (2ª a 6ª, 8h30 às 17h). Períodos de não operação dentro desses intervalos não podem ultrapassar 2 segundos em um dia.
- O sistema deve requerer autenticação dos usuários para permitir acesso.
- O sistema deve implementar disposições de privacidade dos usuários de acordo com a Lei Geral de Proteção de Dados.

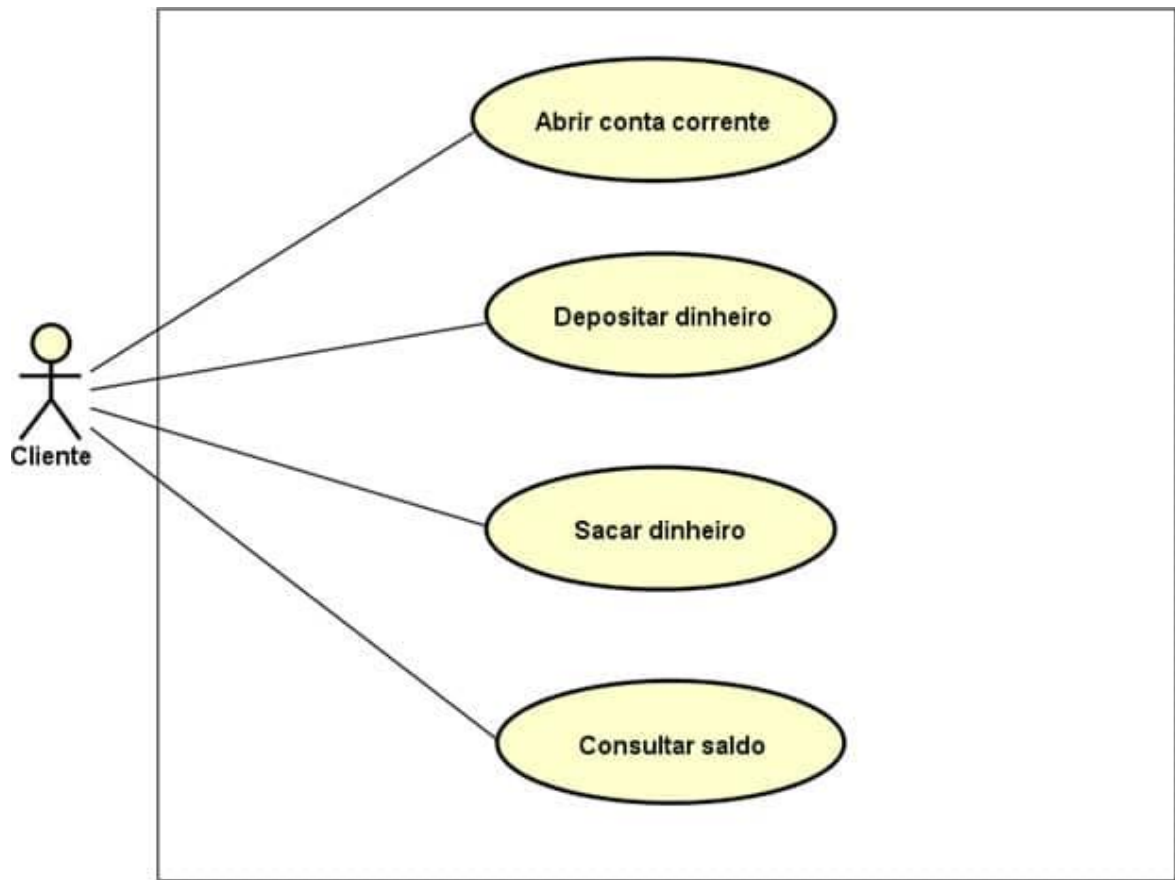
Você pode perceber, nesses exemplos, que esses requisitos não indicam o que o sistema realiza, mas como ele entrega uma ou mais de suas funcionalidades.

DIAGRAMA DE CASOS DE USO

A visão de casos de uso da UML apresenta o conjunto de funções (ou usos) que o sistema deve executar para atender às necessidades dos usuários. No modelo de casos de uso, o sistema é visto sob a perspectiva do usuário. A documentação dos requisitos funcionais em um Diagrama de Casos de Uso dá início ao processo de análise do sistema e permite entender sua abrangência e complexidade.

O objetivo do Diagrama de Casos de Uso é apresentar uma visão geral dos requisitos funcionais do sistema e, portanto, ele é o diagrama mais abstrato, flexível e simples da UML.

Esse diagrama é composto pelos seguintes elementos: **casos de uso**, **atores** e **relacionamentos**. A figura 1 mostra um exemplo de Diagrama de Casos de Uso para um sistema bancário.



powered by Astah

Os possíveis usos de um sistema são representados pelos casos de uso no diagrama. Um **Caso de Uso** pode estar diretamente associado a um ou mais requisitos funcionais, identificados na etapa de levantamento de requisitos do processo de desenvolvimento.

Conforme observado na figura 1, o caso de uso assume a forma gráfica de uma elipse contendo o seu nome no interior; nela, podemos observar quatro casos de usos representados:

Abrir conta corrente

Depositar dinheiro

Sacar dinheiro

Consultar saldo

O caso de uso é o elemento fundamental do Modelo de Casos de Uso.

Atenção

Embora não exista um padrão para nomear casos de uso, a boa prática recomenda utilizar uma frase verbal iniciando com verbo no infinitivo, denotando a ação prevista na funcionalidade representada pelo Caso de Uso.

O segundo elemento do Diagrama de Casos de Uso é o **Ator**, que é representado no Diagrama de Casos de Uso por um boneco palito, com o nome na parte inferior.

O ator é quem utiliza ou interage em um caso de uso, ou seja, é uma entidade externa que interage com o sistema, mas que não faz parte dele.

Interagir significa trocar informações com o sistema, seja fornecendo ou recebendo essas informações. Na figura 1, temos a representação do ator **Cliente**, que interage com os quatro casos de uso. Por exemplo, no caso de uso **Consultar Saldo**, o ator Cliente informa os dados de sua conta e recebe como resposta do sistema o valor de seu saldo.

O terceiro elemento do Diagrama de Casos de Uso são os **Relacionamentos**, que permitem associar:

- Casos de uso com seus atores
- Casos de uso com outros casos de uso
- Atores com outros atores

Saiba mais

A forma mais comum e obrigatória dos relacionamentos é entre atores e casos de uso, que assume a representação gráfica de segmento de reta (uma linha cheia) ligando o ator ao Caso de Uso no qual ele interage.

Na figura 1, observamos que o ator Cliente interage com os quatro casos de uso presentes nesse modelo, portanto, esses elementos gráficos estão ligados por uma reta.

A seguir, vamos detalhar cada um desses elementos.

ATORES

Medeiros (2004) explica que atores atuam em casos de uso realizando atividades. O ator não é uma entidade específica, mas representa um papel ou aspecto particular de uma entidade. O ator pode representar papéis executados por usuários humanos, um componente de hardware, outros sistemas com os quais o sistema em desenvolvimento se comunica. Desse modo, o nome do ator deve expressar esse papel de forma clara. No exemplo da figura 1, o ator é o papel de cliente do banco. Esse papel pode ser assumido pelos diversos clientes do banco.

Resumindo

Durante a execução de uma funcionalidade do sistema (um ou mais requisitos funcionais), o ator precisa prover entradas (informações) para o sistema e deve receber respostas dele.

O ator não é uma entidade específica, mas representa um papel ou aspecto particular de uma entidade. O ator pode representar papéis executados por usuários humanos, um componente de hardware, outros sistemas com os quais o sistema em desenvolvimento se comunica. Desse modo, o nome do ator deve expressar esse papel de forma clara. No

exemplo da figura 1, o ator é o papel de cliente do banco. Esse papel pode ser assumido pelos diversos clientes do banco.

Tipos de atores que geralmente aparecem nos sistemas são:

Cargos

Por exemplo: empregado, cliente, gerente, vendedor, comprador etc.

Organizações e suas divisões

Por exemplo: fornecedor, agência reguladora, administradora de cartões, setor de vendas etc.

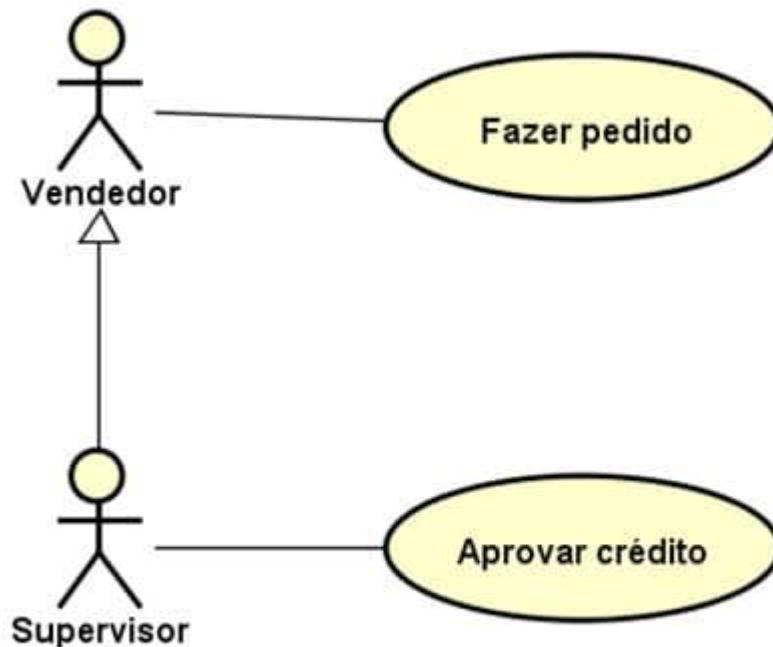
Outros sistemas

Por exemplo: sistema de vendas, sistema de cobrança, sistema de pagamento etc.

Hardware

Por exemplo: sensor de temperatura, leitora de código de barras etc.

O relacionamento de generalização entre atores indica que uma instância do primeiro ator interage com os mesmos tipos de instância de casos de uso do segundo. Porém, o contrário não é verdade. Os casos de uso associados ao ator mais específico nesse relacionamento são exclusivos desse ator, que herda os casos de uso associados ao ator mais genérico. A figura 2 mostra um exemplo desse tipo de relacionamento, no qual percebemos que tanto o ator Vendedor quanto o ator Supervisor interagem com o caso de uso Fazer pedido. No entanto, apenas o Supervisor participa do caso de uso Aprovar crédito.



powered by Astah

Figura 2 – Exemplo do relacionamento de generalização entre atores, elaborado com ferramenta ASTAH (2021).
Imagem: Flavia Maria Santoro.

Normalmente, o ator inicia a sequência de interações em um caso de uso. Portanto, a forma mais fácil de começar a construir um modelo de casos de uso é identificar todos os atores.

O analista de sistemas pode observar as fontes que devem fornecer informações a serem processadas pelo sistema e para quem os resultados do processamento devam ser fornecidos.

Bezerra (2015) sugere algumas perguntas úteis para guiar os analistas de sistemas:

- 1. Que órgãos, empresas ou pessoas utilizarão o sistema?**
- 2. Que sistemas ou equipamentos vão se comunicar com o sistema a ser construído?**
- 3. Alguém deve ser informado de alguma ocorrência no sistema?**
- 4. Quem está interessado em certo requisito funcional do sistema?**

A partir daí, durante a identificação dos casos de uso, o analista poderá perceber a existência de outros atores não identificados nessa etapa.

CASO DE USO

Caso de Uso é a descrição de uma sequência de interações entre um sistema e um ou mais atores externos a esse sistema. De acordo com Bezerra (2015), os casos de usos permitem a um observador externo ao sistema entender quais são as funcionalidades fornecidas pelo sistema e quais são os resultados externos produzidos por elas. Porém, não é possível saber como o sistema trabalha internamente para produzir esses resultados. Um modelo de casos de uso pode conter vários casos de uso, dependendo do tamanho e da complexidade do sistema.

Atenção

Casos de uso não se limitam à sua representação gráfica; tão importante quanto isso é a sua descrição textual. As descrições devem explicar de forma clara, simples e de fácil entendimento, o que o sistema faz, sem entrar nos detalhes de como ele deve fazê-lo. Casos de uso facilitam o processo de comunicação com usuários e são utilizados para documentar a interação do sistema com eles. Desse modo, casos de uso são úteis, tanto para os profissionais envolvidos no desenvolvimento do sistema (analistas, projetistas, programadores, testadores) quanto para seus usuários e clientes. A descrição textual de um caso de uso é uma espécie de história por meio da qual contamos como um usuário realiza uma ação utilizando o sistema.

Casos de uso podem relatar as diversas formas de utilização de uma funcionalidade do sistema. Assim, podemos definir cada uma dessas formas por meio de cenários.

Cenário pode ser entendido como a instância de execução de um caso de uso. Portanto, diversos cenários para o mesmo caso de uso podem ser escritos. Os cenários podem ser usados para a especificação dos casos de testes do sistema.

Para identificar os casos de uso, podemos pensar em dois tipos que sempre estarão presentes nos sistemas: primário e secundário.

Clique nas barras para ver as informações.

CASOS DE USO PRIMÁRIOS

Estão relacionados diretamente aos objetivos dos atores, ou seja, são as funcionalidades que agregam valor aos usuários. Bezerra (2015) sugere uma lista de perguntas para auxiliar os analistas de sistemas na descoberta dos casos de uso primários de um sistema:

1. Quais são as necessidades e os objetivos de cada ator em relação ao sistema?
2. Que informações o sistema deve produzir?
3. O sistema deve realizar alguma ação que ocorre regularmente no tempo?
4. Para cada requisito funcional, existe um caso de uso (ou mais) para atendê-lo?

Alguns casos de uso são iniciados não por um ator, mas por um evento temporal. Desse modo, normalmente o ator é o tipo de entidade que recebe a informação resultante (por exemplo, um cliente que receba uma notificação do sistema).

CASOS DE USO SECUNDÁRIOS

Esses casos não apresentam uma relação direta de valor para os atores, mas são imprescindíveis para o funcionamento do sistema. Por exemplo, a manutenção de cadastros (inclusão, alteração, consulta e exclusão das diversas informações que

alimentam o sistema). Um sistema bancário precisa cadastrar as informações sobre os clientes.

RELACIONAMENTOS

Os relacionamentos são responsáveis por estabelecer as associações entre atores e casos de uso.

Todos os atores devem estar associados explicitamente aos casos de uso com os quais interagem. Além disso, podemos estabelecer relacionamentos entre os casos de uso ou entre os atores do sistema. O Modelo de Casos de uso permite os seguintes tipos de relacionamentos: comunicação, inclusão, extensão e generalização.

Relacionamento de comunicação

É uma linha simples que mostra a que caso de uso determinado ator está associado, ou seja, esse ator interage com o sistema por meio do caso de uso. Um ator pode se relacionar com mais de um caso de uso, e todos os atores devem estar associados a pelo menos um caso de uso. Esse relacionamento foi mostrado na figura 1.

Atenção

Um erro comum na elaboração de diagramas de casos de uso é estabelecer relacionamentos de comunicação entre casos de uso. Não existe troca de informações entre casos de uso, uma vez que são funcionalidades independentes entre si.

Os relacionamentos de inclusão (include), extensão (extend) e generalização são estabelecidos entre casos de uso.

Relacionamento de inclusão

Tem como objetivo principal a reutilização. Suponha que uma parte dos passos de execução de um caso de uso se repita em diversos outros casos de uso. Podemos criar um caso de uso separado e “incluí-lo” nos demais. A inclusão funciona de forma semelhante à chamada de uma função em um programa. A representação gráfica do relacionamento de inclusão é uma linha pontilhada com o estereótipo <<include>> apontando do caso de uso inclusor para o incluído.

A figura 3 mostra um exemplo desse tipo de relacionamento.



powered by Astah

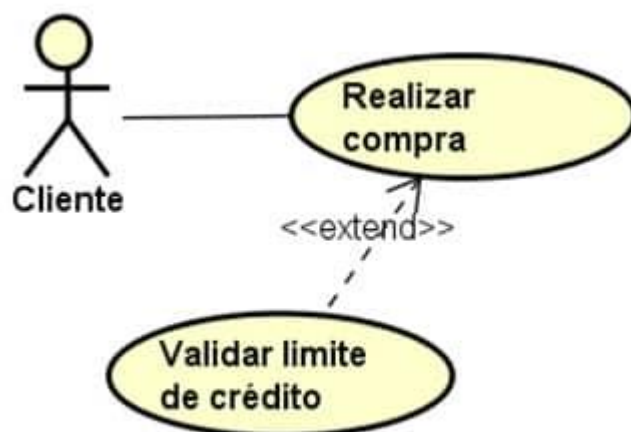
Figura 3 – Exemplo do relacionamento de inclusão entre casos de uso, elaborado com ferramenta ASTAH (2021)

No exemplo da figura 3, o ator Médico participa de dois casos de uso: Fazer pedido de exame e Fazer pedido de consulta. Em ambos os casos de uso, é necessário realizar os passos do caso de uso Solicitar prontuário. Portanto, este caso de uso é incluído nos dois primeiros.

Relacionamento de extensão

Tem como objetivo representar situações em que diferentes sequências de passos possam ser inseridas no mesmo caso de uso. Ou seja, dependendo de uma condição, o caso de uso segue um caminho diferente. Desse modo, um caso de uso estende outro, permitindo um comportamento eventual. Note que a existência do caso de uso estendido deve ser independente da existência de quaisquer casos de uso que o estendam.

A figura 4 mostra um exemplo do relacionamento de extensão.



powered by Astah

Figura 4 – Exemplo do relacionamento de extensão entre casos de uso, elaborado com ferramenta ASTAH (2021)

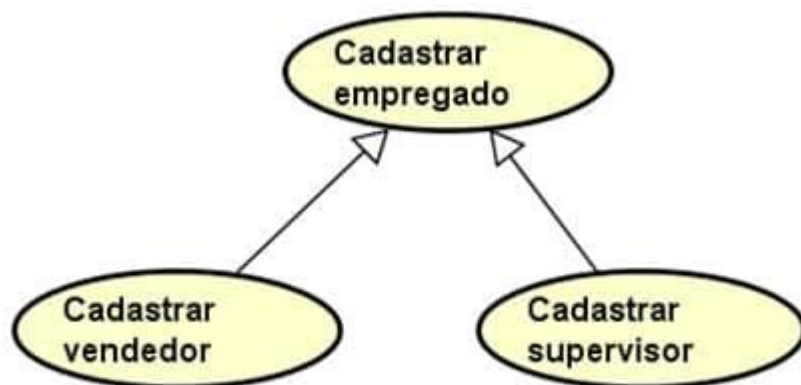
Observe no exemplo da figura 4, em que o ator Cliente interage com o caso de uso Realizar compra. O caso de uso Validar limite de crédito pode ser necessário nesse contexto, portanto, é estendido para Realizar compra. Ambos os casos de uso representam funcionalidades independentes no sistema.

Relacionamento de generalização

Pode existir entre dois casos de uso ou entre dois atores (conforme vimos em Atores). Esse relacionamento permite que um caso de uso (mais específico) herde características de outro (mais genérico).

Na generalização entre casos de uso, quando um caso de uso herda de outro, significa que as sequências de passos de execução do caso de uso mais genérico (pai) valem também para o mais específico (filho). Além disso, o caso de uso herdeiro participa de qualquer associação do qual o caso de uso mais genérico participe. Isso significa que um ator que interage com o caso de uso pai pode também interagir em qualquer caso de uso filho.

A figura 5 mostra um exemplo do relacionamento de generalização:



powered by Astah

Figura 5 – Exemplo do relacionamento de generalização entre casos de uso, elaborado com ferramenta ASTAH (2021)

Como podemos observar na figura 5, o caso de uso Cadastrar empregado executa uma sequência de passos de cadastro para qualquer funcionário. Já os casos de uso mais específicos (Cadastrar vendedor e Cadastrar supervisor) herdam esses passos e ainda possuem passos específicos para os tipos de funcionários Vendedor e Supervisor, respectivamente.

Os tipos de relacionamento entre casos de uso abordados permitem manter a descrição dos casos de uso mais simples e sem repetições.

A inclusão deve ser usada quando um comportamento se repetir em mais de um caso de uso.

A extensão é normalmente usada quando um comportamento que não é executado em todas as instâncias de um caso de uso deva ser descrito.

A generalização deve ser aplicada em uma situação em que dois ou mais casos de uso possuam comportamentos comuns.