

## 1 Engenharia de Requisitos

### 1.1 Requisitos de Sistema

Requisitos são características do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir seus objetivos. Eles também podem ser entendidos como uma condição ou capacidade que deve ser alcançada ou estar presente em um sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto.

Os maiores desafios no desenvolvimento de sistemas, é a compreensão do domínio do problema, comunicação efetiva com os usuários reais e a evolução contínua dos requisitos do sistema.

## 2 Diagramas de caso de uso e UML

A UML é uma linguagem visual para especificar, construir e documentar os artefatos do sistema. Um apoio ao processo de desenvolvimento de forma visual. Com a adoção do paradigma OOP, surgiram diversas formas de modelar um sistema, com isso, a UML surgiu. Ela permite que cada elemento gráfico possua uma sintaxe e uma semântica que define o significado do elemento e para que ele deve ser usado.

A criação do UML e da linguagem de modelagem envolve a criação de diversos documentos, sendo alguns deles.

- Documentos textuais e gráficos
- Documentos de artefato de software
- Artefatos que compõem as visões do sistema

Os modelos de caso de uso representam as funcionalidades de forma a ser observada externamente o sistema, e a forma como ele interage com si mesmo.

### 2.1 Caso de Uso

O modelo de caso de uso possibilita a apresentação compreensível que pode ser utilizada para aprimorar a comunicação, especialmente entre projetistas da aplicação e clientes. Eles são úteis para fases posteriores do ciclo de vida, ajudando na identificação dos objetos, desenvolvimento de planos de teste e documentação.

Os casos de uso representam quem faz o que com o sistema, sem considerar o comportamento o comportamento interno do sistema, e esses comportamentos podem ser descritos de diversas maneiras, como por meio de descrições: Narrativas; Contínuas; Numeradas e Cliente/Sistema.

Também pode ser tomado em conta o nível de detalhamento a ser utilizado na descrição de um Caso de Uso, podendo ser sucinta, ou expandida.

Os casos de uso possuem três elementos principais dentro deles, que são os casos de uso, os atores e os relacionamentos entre os elementos anteriores.

## 2.2 Atores

Os atores são elementos externos que interagem com o sistema, e eles podem ser categorizados de diversas formas, como por exemplo: Pessoas; Organizações; Outros sistemas e Equipamentos. Um ator corresponde a um papel representado em relação ao sistema.

Os atores podem ser tanto **primários**, que é quando eles iniciam a sequência de interações de um caso de uso, como podem ser **secundários**, que é quando eles operam, mantêm ou auxiliam na utilização do sistema.

## 2.3 Relacionamentos

Os relacionamentos dentro da UML podem ser divididos em:

- Comunicação ou associação
- Inclusão
- Extensão
- Generalização

A **inclusão** ocorre quando dois ou mais casos de uso incluem uma sequência de interações comum, esta sequência comum pode ser descrita e um outro caso de uso.

A **extensão** por outro lado, utiliza a ideia de diferentes sequências de interações podem ser inseridas em um caso de uso. Por exemplo, uma determinada ação como editar um documento, pode ter além dessa atribuição, a possibilidade de corrigir a ortografia automaticamente, isso diz respeito a extensão dos elementos serem estendidos.

A **generalização** é quando o reuso é extremamente evidente, e ele pertence que um caso de uso herde características de outro caso de uso, porém mais genérico.

## 2.4 Documentação dos atores dos casos de uso

Uma breve descrição para cada ator deve ser adicionada ao modelo de caso de uso. O nome do ator deve lembrar o papel desempenhado pelo sistema.

## 2.5 Regras de Negócio

São políticas, condições ou restrições que devem ser consideradas na execução dos processos existentes em uma organização. Eles descrevem a maneira pela qual a organização funciona. Estas regras são identificadas e documentadas no modelo de regras do negócio

## 3 Diagrama de Classes

Os diagramas de classes nos ajudam a representar coisas no mundo real ou imaginário que podemos facilmente identificar. E isso é feito por meio do paradigma de orientação a objetos. Um **Objeto** pode ser entendido como um elemento, que possui atributos e propriedades dentro de si. As classes são representadas por retângulos com três divisões dentro de si, onde cada uma dessas divisões possui uma finalidade dentro de si. Sendo o nome, atributos e métodos respectivamente.

Os **Relacionamentos** possibilitam compartilhar informações e colaborar com a execução dos processos do sistema. Os relacionamentos podem ser de cinco tipos diferentes, sendo eles:

- Associação
- Agregação
- Composição
- Especialização
- Dependência

### 3.1 Multiplicidade

A multiplicidade diz respeito a quantidade de elementos que devem ser relacionados nas formas de relacionamentos presentes dentro da engenharia de software, e possuem os seguintes tipos com os seguintes significados:

Multiplicidade	Significado
0..1	No mínimo zero e no máximo um, devem se relacionar com no máximo um elemento
1..1	Um e somente um
0..*	No mínimo nenhum e no máximo muitos
*	Muitos
1..*	No mínimo um no máximo muitos
3..5	No mínimo 3 no máximo 5

### 3.2 Associação

Descreve um conjunto de vínculos entre elementos de modelo. Possui um relacionamento estrutural que especifica objetos de um item conectados a objetos de outro item. A associação ainda pode ser dividida em **binária** e **unária**, que é quando há duas classes envolvidas na associação de forma direta de uma para outra, e quando há um relacionamento de classe consigo mesma, respectivamente.

A simbologia é representada por uma linha preta completa, sem nenhum tipo de elemento em seu final.

### 3.3 Agregação

É um tipo especial de associação que tenta demonstrar que as informações de um objeto como um todo precisam ser complementadas pelas informações contidas em um objeto parte. A existência do objeto-parte faz sentido mesmo não existindo o objeto-todo.

Sua notação no diagrama de classes, é feita por meio de uma linha que possui um diamante vazio na ponta do elemento que é o objeto-parte

### 3.4 Composição