

## Requisitos e Engenharia de Requisitos

A **engenharia de requisitos** (ER) é uma subárea fundamental da **engenharia de software** cujo objetivo principal é obter uma **especificação correta e completa** dos requisitos de um sistema de software. Ela visa garantir que os requisitos sejam bem definidos, para que o sistema atenda às necessidades dos stakeholders e cumpra seus objetivos de maneira eficiente.

### A Importância da Engenharia de Requisitos

De acordo com **Pressman (2010)**, a engenharia de requisitos começa durante a **especificação** e segue até a fase de **modelagem**, sendo uma ação contínua no processo de software. Para o sucesso de um projeto, é fundamental que haja uma compreensão clara dos requisitos de software, com o desenvolvedor e o cliente atuando como parceiros para traduzi-los e refiná-los.

### O Que São Requisitos?

**Sommerville (2007)** define requisitos como as condições ou capacidades que o sistema deve ter para atender aos interesses dos usuários ou outros stakeholders. Os requisitos especificam **o que o sistema deve fazer** e as **restrições** sobre sua operação e implementação.

#### Objetivos dos Requisitos:

- **Estabelecer e manter** concordância entre os stakeholders sobre o que o sistema deve fazer.
- Fornecer uma **compreensão melhor** do sistema aos desenvolvedores.
- **Delimitar** o escopo do sistema e planejar seu desenvolvimento.
- Servir como **base** para estimar **custos** e **prazos** de desenvolvimento.

### Tipos de Requisitos

**Sommerville (2007)** classifica os requisitos em diferentes categorias, com algumas delas tendo subclasses. Abaixo, estão as principais categorias:

#### 8.1 Requisitos Funcionais

São os requisitos que **descrevem as funcionalidades** ou serviços que o sistema deve fornecer. Exemplos:

- Cadastro de cliente.

- Emissão de nota fiscal.
- Consulta ao estoque.
- Geração de pedido.

## 8.2 Requisitos Não Funcionais

Estes requisitos não estão diretamente relacionados às funcionalidades do sistema, mas sim às **propriedades e restrições** do sistema, como usabilidade, desempenho e segurança. Podem ser classificados em:

### 1. Requisitos de Produto

- **Facilidade de uso (usabilidade):** Esforço para aprender ou utilizar o sistema.
- **Confiabilidade:** Frequência de falhas, recuperabilidade após falhas, entre outros.
- **Eficiência:** Desempenho do sistema, uso de recursos e tempo de resposta.
- **Portabilidade:** Capacidade de transferir o produto para outros ambientes.

### 2. Requisitos Organizacionais

Estão relacionados a **políticas organizacionais** do cliente ou desenvolvedor. Exemplos incluem:

- **Requisitos de implementação:** Restrições sobre a codificação e os ambientes de execução.
- **Requisitos de padrões:** Definição de linguagens de programação e normas a seguir.
- **Requisitos de entrega:** Como o sistema será implantado.

### 3. Requisitos Externos

Estes requisitos vêm de fatores externos ao sistema e ao processo de desenvolvimento, como:

- **Requisitos éticos.**
- **Requisitos legais** (ex: políticas de privacidade, direitos autorais).
- **Requisitos de interoperabilidade.**

### 8.3 Requisitos de Domínio

Derivam do **domínio da aplicação** do sistema, como normas ou regras de negócios específicas de uma área. Exemplo: requisitos para garantir conformidade com uma norma de intercâmbio de dados, como a **Z39.50**.

### 8.4 Requisitos do Usuário

Refletem as necessidades do usuário, sendo **funcionais e não funcionais**. Muitas vezes, apresentam problemas como falta de clareza e confusão, devido à forma como são expressos.

### 8.5 Requisitos do Sistema

São **descrições detalhadas** dos requisitos do usuário e fornecem a base para o projeto do sistema. Esses requisitos são utilizados para criar contratos de implementação e garantir que o sistema seja desenvolvido conforme o especificado.

## Padrões para Elaboração de Requisitos

O **IEEE** (Instituto de Engenheiros Eletricistas e Eletrônicos) desenvolveu o **IEEE Recommended Practice for Software Requirements Specification**, que fornece padrões e práticas recomendadas para a elaboração do **DERS (Documento de Especificação de Requisitos de Software)**.

## Benefícios de um Bom DERS

Um **bom DERS** deve ser:

- **Completo e não ambíguo.**
- **Clareza** na comunicação entre o cliente e os desenvolvedores.
- Auxiliar na **redução de esforço** de desenvolvimento.
- Servir como base para **estimativas de custo e prazo**.
- Fornecer **fundamento** para validação e verificação do sistema final.
- Facilitar a **manutenção** do software.

## Stakeholders (Influenciadores/envolvidos)

Os **stakeholders** são as partes interessadas no projeto, ou seja, as pessoas ou organizações que possuem **interesses** que podem ser afetados pelo projeto e podem exercer influência sobre seus resultados.

## Conclusão

A engenharia de requisitos é essencial para o sucesso do desenvolvimento de software, pois garante que todas as partes envolvidas compartilhem uma **visão comum** do sistema. Além disso, a definição de requisitos bem estruturados é fundamental para planejar, implementar, validar e manter o software de forma eficaz.