

Tipos de Requisitos

Objetivo da Aula

Conhecer os tipos de requisitos e como eles são classificados em um projeto de software.

Apresentação

Os requisitos são as características que um software deve apresentar para atender as necessidades de um cliente. Eles podem ser classificados a partir de suas características.

A classificação de requisitos funcionais e não funcionais são as mais comuns, mas há outras formas de definir um requisito e o que ele deve garantir em um projeto de software.

Nesta aula aprenderemos a realizar essa classificação, abordando quais são as necessidades que tais requisitos devem cumprir.

1. Tipos de Requisitos

O entendimento dos requisitos é essencial para a construção de um software com qualidade. Basicamente os requisitos são divididos em Funcionais e Não-Funcionais.

Os Requisitos Funcionais são aqueles que dizem respeito às funcionalidades do sistema, ou seja, aquilo que é executável. Uma declaração de requisitos pode trazer também a identificação do que o sistema NÃO deve fazer, para que isso fique explícito para os *stakeholders*.

Em um sistema de uma pizzeria, por exemplo, poderíamos ter como requisitos funcionais os seguintes requisitos:

- Cadastrar Cliente;
- Cadastrar Pedido;
- Aplicar Cupom de Desconto;
- Cadastrar Pizza.

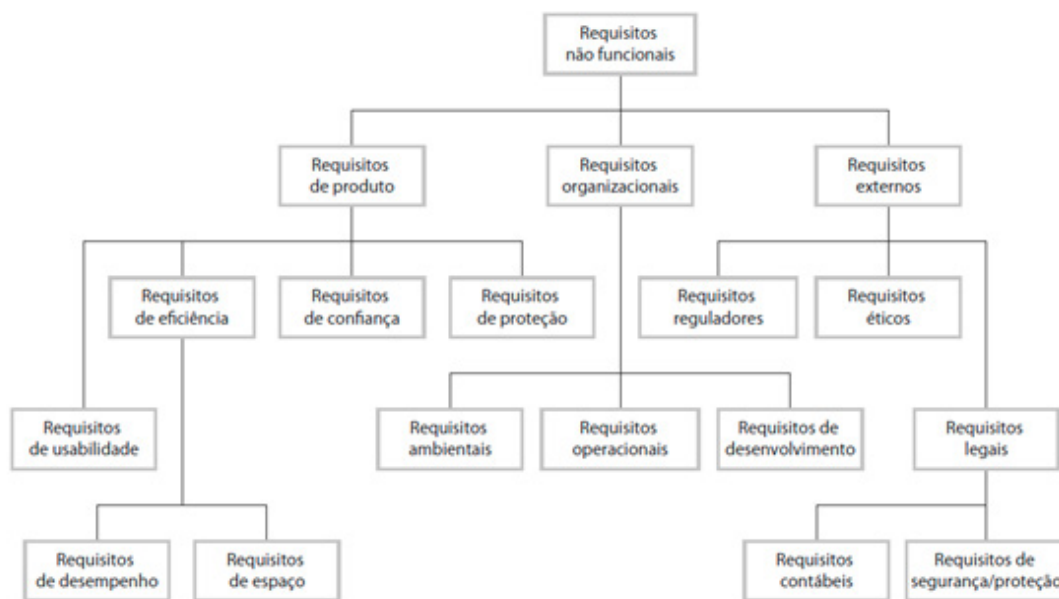
Já os Requisitos Não-Funcionais dizem respeito às condições que um sistema deve atender, ou seja, rapidez, segurança, usabilidade, confidencialidade, confiabilidade, regulação, dentre outros que não especifiquem uma funcionalidade. Eles definem as restrições de funcionamento do sistema.

Considerando o hipotético sistema de uma pizzaria, alguns requisitos funcionais do sistema poderiam ser:

- O sistema deverá exibir o cardápio em, no máximo, 2 segundos;
- O aplicativo da pizzaria deverá ser responsivo;
- Os dados do cartão de crédito não devem estar acessíveis.

De acordo com Sommerville (2011, p. 61), os requisitos não funcionais podem ser provenientes das características requeridas para o software (requisitos de produto), da organização que desenvolve o software (requisitos organizacionais) ou de fontes externas:

Figura 1: Diagrama de Requisitos Não-Funcionais



Fonte: Sommerville (2011, p. 61).

Na Figura 1 vemos as possíveis ramificações que os Requisitos Funcionais podem ter, o que facilita o entendimento de que muitas características podem ter influência no desenvolvimento de um sistema, para que as especificações sejam atendidas.

Tendo em vista essas ramificações, podemos afirmar que os requisitos são caracterizados da seguinte forma:

- **Requisitos de Produto:** tem a ver com o funcionamento do software, diz respeito ao desempenho, rapidez, uso de memória, confiabilidade, usabilidade e segurança, por exemplo;
- **Requisitos Organizacionais:** vão listar características inerentes às políticas e procedimentos da organização e do desenvolvedor, incluindo questões como sistema operacional, linguagem de programação e normas gerais da organização;
- **Requisitos Externos:** apresentam fatores externos que o sistema deve cumprir, como normas reguladoras, requisitos legais e requisitos éticos.

Quanto aos requisitos não funcionais, é importante que eles sejam especificados quantitativamente, para que ele seja testável e que não gere interpretações equivocadas ou dúbias.

Por exemplo, podemos especificar um requisito funcional da seguinte forma:

O sistema deverá realizar o backup em, no máximo, cinco minutos.

Para ajudar no processo de especificação de requisitos não funcionais, Sommerville (2011, p. 63) sugere as seguintes métricas dispostas na Tabela 1:

Tabela 1: Métricas para Requisitos Não-Funcionais

Requisito	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes/Gigabytes
Usabilidade	Tempo de Treinamento
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo

Fonte: Adaptada de Sommerville (2011, p. 63).

Os Requisitos Não-Funcionais podem ainda se basear na sigla UCDS (Confiabilidade, Desempenho e Suportabilidade), que determinam as características desejáveis do software.

Em termos práticos, cada característica dessas tem o objetivo de:

- Usabilidade: tratar da facilidade com que o software pode ser utilizado pelos usuários;
- Confiabilidade: tratar da habilidade do sistema de se comportar de forma consistente e aceitável;
- Desempenho: tratar da velocidade ou eficiência de execução do sistema;
- Suportabilidade: tratar da habilidade do software em ser facilmente modificável para acomodar melhorias e reparos ao longo de sua operação.

2. Outros Tipos de Requisitos

Em um contexto mais amplo, podemos dizer que todos os requisitos são requisitos de projeto, e ainda que eles possam ter subdivisões que podem ser assim representadas:

a) Requisitos de Projeto:

- Requisitos de Marketing;
- Requisitos de Hardware;
- Requisitos de Produção;
- Requisitos de Processos Organizacionais;
- Requisitos de Software;
- Requisitos Funcionais;
- Requisitos Não-Funcionais.

Exemplos:

- Requisitos de Marketing – aponta um direcionamento do produto, por exemplo: o sistema deverá ser direcionado para estudantes de doutorado;
- Requisitos de Hardware – uma condição de hardware para uso do sistema, por exemplo: o sistema precisa de um processador, no mínimo, i7 ou equivalente;
- Requisitos de Produção – direciona o processo de produção, por exemplo: o sistema deverá ser desenvolvido utilizando o método ágil Scrum;
- Requisitos de Processos Organizacionais – aborda processos e regras de negócio da organização, por exemplo: o sistema deverá permitir o cadastro de atividades de alunos monitores.

3. Documento de Requisitos

Após a identificação de requisitos, o próximo passo é documentá-los. A documentação de requisitos é especialmente importante porque ela define o escopo do sistema, isto é, tudo aquilo que o sistema deverá apresentar ao seu término.

A literatura da **Engenharia de Software** propõe a construção de documentos guias para que o analista possa documentar e especificar os requisitos descobertos. No entanto, não há a necessidade de seguir o padrão apresentado por essa literatura, o mais importante é a padronização deste tipo de documentação em uma organização ou projeto e, além disso, também é importante que o documento seja útil, e ajude aos **stakeholders** a conhecer as funcionalidades que serão implantadas no sistema.

Outro ponto importante é que seja considerado o tempo de produção deste documento, principalmente em casos em que os requisitos mudam rapidamente. Não se deve perder tempo demais na construção deste documento, para que isso não impacte no tempo total de elaboração do projeto.

Como boa prática, um documento de requisitos deve conter:

- Índice – para ajudar a organizar o trabalho e para que o leitor encontre as informações desejadas;
- Histórico de Versões – para controlar as mudanças no documento;
- Introdução – para descrever o sistema de forma geral;
- Glossário – para descrever os termos técnicos utilizados no projeto;
- Diagrama de Caso de Uso – trata-se de um diagrama da UML (Linguagem de Modelagem Unificada) que deve exibir os requisitos funcionais do sistema;
- Requisitos Funcionais e Não-Funcionais – deve-se listar e descrever os requisitos funcionais e não funcionais encontrados;
- Especificação ou Descrição dos Requisitos Funcionais e Não-Funcionais – para detalhar como os requisitos funcionais e não funcionais devem funcionar dentro do sistema.

Novamente, é importante dizer que isso é apenas uma sugestão de como elaborar um documento de requisitos, mas este artefato pode trazer muitas outras informações que sejam úteis para equipe. A elaboração de um **template** para ser usado pela equipe é outra boa prática. Vale a pena lembrar que, embora esse documento seja muito útil, precisamos tomar cuidado com o tempo que levamos para produzi-lo.

Considerações Finais

Nesta aula abordamos os tipos de requisitos, que podem ser geralmente classificados em Funcionais e Não-Funcionais, mas vimos também que existem outras formas de classificar os requisitos, segmentando ainda mais o que é desejado pelos *stakeholders*.

Quanto aos requisitos Não-Funcionais, há uma infinidade de classificações, trazendo as características como eficiência, usabilidade e portabilidade, além dos requisitos que tratam de questões legais e regulatórias.

Vimos ainda que a elaboração de um **template** para documentar os requisitos é uma boa prática, já que este documento apoia a organização, a auditabilidade e o rastreo de mudanças ao longo do ciclo de vida do produto.

Materiais Complementares



Site

Link: *Como escrever requisitos de software de forma simples e garantir o mínimo de erros no sistema/app?* Disponível em: <https://medium.com/lfdev-blog/como-escrever-requisitos-de-software-de-forma-simples-e-garantir-o-m%C3%ADnimo-de-erros-no-sistema-app-74df2ee241cc>. Acesso em: 23 nov. 2022.



Vídeo

Entenda a diferença entre Requisitos Funcionais e Não-Funcionais. Disponível em: <https://www.youtube.com/watch?v=YLd6AWKVyas>. Acesso em: 23 nov. 2022.



Vídeo

Qual é a melhor forma de documentar software? Disponível em: https://www.youtube.com/watch?v=S3NLU898_Cc. Acesso em: 23 nov. 2022.

Referências

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison. Wesley, 2019.