# Requisitos de Software

## Requisitos funcionais

Um requisito funcional define uma função de um software ou parte dele. Ele é o conjunto de entradas, seu comportamento e sua saída, ou seja, envolve cálculos, lógicas de trabalho, manipulação e processamento de dados, entre outros. Dentro dos requisitos funcionais também se encontram a arquitetura do aplicativo, diferentemente da arquitetura técnica, que pertence aos requisitos não funcionais.

Muitos autores ainda dividem os requisitos funcionais em três: evidente, escondida e friso. Requisitos evidentes é quando o usuário final do sistema está ciente do que está sendo executado. Escondida é quando uma função está sendo feita, mas é invisível ao usuário. Além das duas, ainda há a do tipo friso, quando a execução da funcionalidade não afeta outras funções do software.

Registrar a venda de produtos em lojas virtuais, calcular o total de uma venda, capturar a informação de um formulário, fazer o login com uma identificação e senha e exibir a descrição e preço de um item são exemplos de requisitos funcionais evidentes. Já dar baixa em um item, manter um log de vendas feitas e prover a integração com outros sistemas são requisitos escondidos.

## Requisitos não funcionais

Requisitos não funcionais são relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, disponibilidade, segurança e tecnologias envolvidas. Muitas vezes, os requisitos não funcionais acabam gerando restrições aos funcionais.

Com o crescente número de usuários vindos de dispositivos móveis, requisitos não funcionais como portabilidade e mobilidade tornam-se fundamentais em praticamente todos os sistemas web. Além deles, a preocupação com interoperabilidade, localização geográfica e alta disponibilidade também fazem parte do mapeamento e desenvolvimento de requisitos não funcionais.

Entender profundamente os tipos de requisitos no momento de definir seu software pode decidir o futuro dele. Por outro lado, também é preciso ter em mente que o software pode ser beneficiado se houver um pouco de flexibilidade no seu desenvolvimento. Assim, unindo os conceitos, é possível ter um sistema de qualidade sem um alto custo de aquisição. Esse tipo de mapeamento também facilita e otimiza a manutenção e correção de erros, além de prover uma visão clara do serviço que está sendo implementado. Não apenas do lado do cliente, mas também da empresa que está vendendo o software.

## Requisito funcional

Em [engenharia de software](https://pt.m.wikipedia.org/wiki/Engenharia_de_software), um **requisito funcional** define uma função de um sistema de [software](https://pt.m.wikipedia.org/wiki/Software) ou seu componente. Uma função é descrita como um conjunto de entradas, seu comportamento e as saídas. Os [requisitos](https://pt.m.wikipedia.org/wiki/An%C3%A1lise_de_requerimento_de_software) funcionais podem ser cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar. Requisitos comportamentais, que descrevem todos os casos em que o sistema utiliza os requisitos funcionais, são extraídos dos [casos de uso](https://pt.m.wikipedia.org/wiki/Casos_de_Uso). Também, os requisitos funcionais são suportados por [requisitos não-funcionais](https://pt.m.wikipedia.org/wiki/Requisito_n%C3%A3o-funcional) (também conhecidos como requisitos de qualidade), que impõem restrições sobre o projeto ou execução (tais como requisitos de desempenho, segurança ou confiabilidade). O plano para a implementação dos requisitos funcionais é detalhado no projeto do sistema. Já o plano para a implementação de requisitos não funcionais é detalhado na arquitetura do sistema.

Tal como definido na engenharia de requisitos, os requisitos funcionais especificam resultados particulares de um sistema. Isto deve ser contrastado com [requisitos não-funcionais](https://pt.m.wikipedia.org/wiki/Requisito_n%C3%A3o-funcional), os quais especificam características gerais, tais como custo e confiabilidade. Os requisitos funcionais fazem parte da arquitetura do aplicativo de um sistema, enquanto os requisitos não funcionais denotam a arquitetura técnica de um sistema.

Em alguns casos, um analista de requisitos gera [casos de uso](https://pt.m.wikipedia.org/wiki/Casos_de_Uso) após a coleta e validação de um conjunto de requisitos funcionais. A hierarquia de requisitos funcionais é: usuário / pedido das partes interessadas - característica> -> caso de uso -> regra de negócio. Cada caso de uso ilustra cenários de comportamento através de um ou mais requisitos funcionais. Muitas vezes, porém, um analista começará por evocar um conjunto de casos de uso, a partir do qual o analista pode derivar os requisitos funcionais, que devem ser implementados para permitir que um usuário possa realizar cada caso de uso.

## Requisitos não funcionais

Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Não é preciso o cliente dizer sobre eles, pois eles são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos ou não.

**'Requisitos Não funcionais'**

* Demonstram qualidade acerca dos serviços ou funções disponibilizadas pelo sistema. Ex.: tempo, o processo de desenvolvimento, padrões, etc.
* Surgem conforme a necessidade dos usuários, em razão de orçamento e outros fatores.
* Podem estar relacionados à confiabilidade, tempo de resposta e espaço nas mídias de armazenamento disponíveis.
* Caso ocorra falha do não atendimento a um requisito não funcional, poderá tornar todo o sistema ineficaz. Ex.: requisito confiabilidade em um sistema de controle de voos.

**Classificação dos Requisitos Não Funcionais**

* Requisitos de produtos: Requisitos que especificam o comportamento do produto. Ex. portabilidade; tempo na execução; confiabilidade, mobilidade, etc.
* Requisitos da organização: Requisitos decorrentes de políticas e procedimentos corporativos. Ex. padrões, infraestrutura, etc.
* Requisitos externos: Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento. Ex. requisitos de interoperabilidade, legislação, localização geográfica etc.
* Requisitos de facilidade de uso. Ex.: usuários deverão operar o sistema após um determinado tempo de treinamento.
* Requisitos de eficiência. Ex.: o sistema deverá processar n requisições por um determinado tempo.
* Requisitos de confiabilidade. Ex.: o sistema deverá ter alta disponibilidade, p. exemplo, 99% do tempo.
* Requisitos de portabilidade. Ex.: o sistema deverá rodar em qualquer plataforma.
* Requisitos de entrega. Ex.: um relatório de acompanhamento deverá ser fornecido toda segunda-feira.
* Requisitos de implementação: Ex.: o sistema deverá ser desenvolvido na linguagem Java.
* Requisitos de padrões: Ex. uso de programação orientada a objeto sob a plataforma A.
* Requisitos de interoperabilidade: Ex. o sistema deverá se comunicar com o SQL Server.
* Requisitos éticos. Ex.: o sistema não apresentará aos usuários quaisquer dados de cunho privativo.
* Requisitos legais. Ex.: o sistema deverá atender às normas legais, tais como padrões, leis, etc.
* Requisitos de Integração. Ex.: o sistema integra com outra aplicação.