

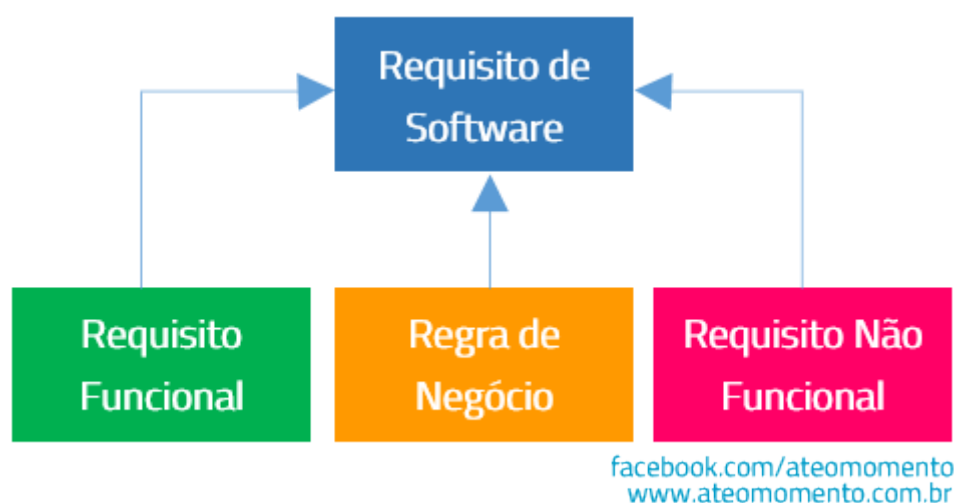
# O que é Requisito Funcional

By **Plínio Ventura**

O que é um Requisito Funcional? Vamos primeiro ao que é Requisito. Requisito é uma **exigência**, solicitação, desejo, **necessidade**.

Quando falamos de um Requisito Funcional estamos nos referindo à requisição de uma função que um software deverá atender/realizar. Ou seja, exigência, solicitação, desejo, necessidade, que um software deverá materializar.

Um Requisito Funcional é um [Requisito de Software](#).

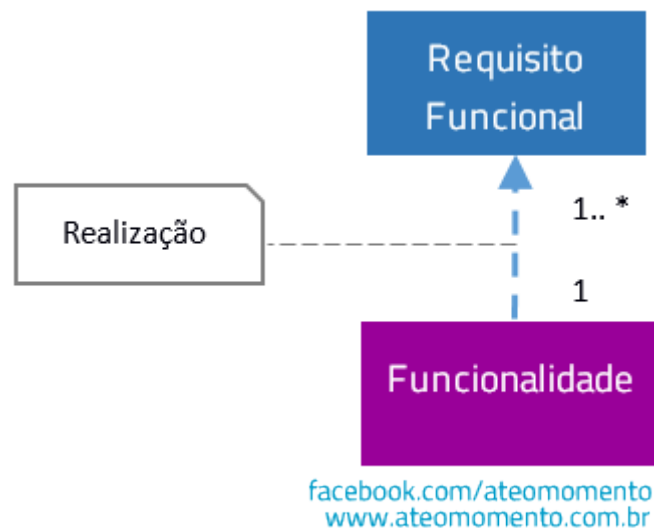


É comum os profissionais de engenharia de software associarem a ideia de um requisito funcional a uma tela, uma rotina, que no fim serão as funcionalidades de fato de um sistema.

Mas é necessário entender que **uma funcionalidade não necessariamente realizará apenas um Requisito Funcional**.

Uma funcionalidade pode realizar **vários** Requisitos Funcionais – significa que em uma funcionalidade **um ou mais** Requisitos Funcionais podem ser **atendidos**, não necessariamente apenas um.

Se pensarmos em **multiplicidade**, uma funcionalidade pode realizar um ou muitos Requisitos Funcionais (1.. \*).



Para entender melhor isso vamos a um exemplo mais básico. Imaginemos um sistema que possui uma tela para “Manutenção de Clientes”, que mantém os dados cadastrais de um cliente no sistema.

Estamos falando de uma **única funcionalidade**. Nesta tela é possível incluir/alterar/consultar/excluir clientes dos tipos pessoa física e pessoa jurídica.

Mas quantos requisitos são realizados (atendidos) por esta funcionalidade? **Oito requisitos**.

Vejamos a lista a seguir:

Requisitos Funcionais (Identificador e Nome)
RF001 – Incluir cliente pessoa física
RF002 – Alterar cliente pessoa física
RF003 – Consultar cliente pessoa física
RF004 – Excluir cliente pessoa física

Requisitos Funcionais (Identificador e Nome)
RF005 – Incluir cliente pessoa jurídica
RF006 – Alterar cliente pessoa jurídica
RF007 – Consultar cliente pessoa jurídica
RF008 – Excluir cliente pessoa jurídica

## O que é um Requisito Funcional, agora sabemos. E o que não é?

### *O que **não** é um Requisito Funcional?*

*É comum quando se fala de Requisito Funcional associar a isto funcionalidade, caso de uso, regra de negócio ou até mesmo requisito não funcional. São coisas muito diferentes.*

*Uma funcionalidade pode realizar um ou mais Requisitos Funcionais. Requisito funcional não é uma funcionalidade, é uma necessidade funcional (uma função) que o software deve atender. Uma funcionalidade será executada por um ator (um ator sistêmico [pelo próprio sistema] ou um ator humano [usuário final]). É onde Requisitos Funcionais serão viabilizados.*

*Um **Caso de Uso** é uma especificação do **comportamento** de uma funcionalidade. Nele se tem detalhes sobre como a funcionalidade “funcionará”, com restrições, premissas e diretrizes pertinentes à funcionalidade.*

*Regra de negócio refere-se a premissas ou restrições de negócio que o sistema deverá atender, regras que poderão ou não estar associadas a um requisito funcional, mas que sempre serão realizadas por uma ou mais funcionalidades do sistema. Na visão da **modelagem conceitual**, Regras de negócio são o “como”, requisitos funcionais são o “o que”.*

*Requisitos Não Funcionais são premissas ou restrições que o sistema deverá atender, mas que não são realizados através de funcionalidades. Podem ou não estar associados a Requisitos Funcionais, mas não tem, necessariamente, relação com o negócio, na visão do usuário.*

# Importância dos Requisitos Funcionais

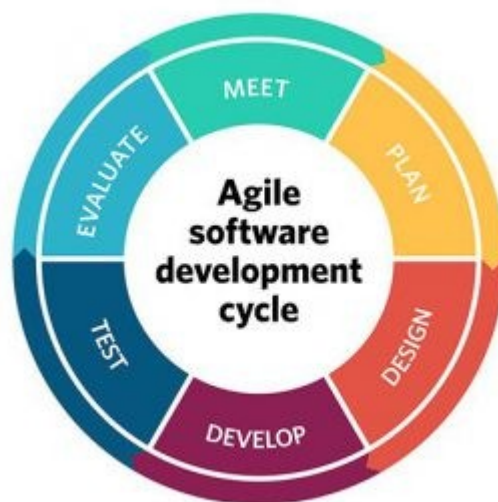
Funcionalidades somente existem para realizar Requisitos Funcionais. Logo, sem requisitos funcionais não há funcionalidades e sem funcionalidades não há sistema.

Este raciocínio por si só demonstra a importância absoluta e inquestionável dos requisitos funcionais no escopo de um sistema.

E por ser algo importante como é, todo cuidado é pouco para que estes requisitos possuam a maior qualidade possível, pois apenas a existência deles no escopo não garante um bom sistema, eles precisam ter qualidade em termos de [sintaxe e semântica](#).

Precisam ser bem feitos. Mas o que devemos entender como qualidade de um requisito?

## Requisitos e Agilidade



**Agilidade não é produzir software com pressa**, é produzir software com velocidade, entregando valor no menor espaço de tempo possível, e melhorando isso continuamente.

Para ser ágil, **é fundamental ser eficiente**.

Não é plausível falar em agilidade sem eficiência, com desperdício.

Em projetos de software, um dos maiores desafios é a boa **comunicação**.

Deixar claro o que se quer, o que será entregue, como será produzido etc. Isso não é simples quando produzimos software, devido à **complexidade inerente a esta atividade**.

Quando se entende um **requisito** do jeito errado, sempre há o custo de fazer errado, desfazer, e fazer certo. Obviamente, este tipo de desperdício custa 3 vezes mais que se tivéssemos **feito certo da primeira vez**.

E neste contexto, fica claro que o uso racional da Modelagem de Requisitos com o objetivo de **transmitir ideias** entre membros de um mesmo time, é uma ferramenta que favorece muito uma **cultura ágil**.

## **Atributos de um bom Requisito Funcional**

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos.

Na literatura, principalmente estrangeira, existem várias recomendações de atributos que um requisito deve atender para ter

qualidade. Mas vou me ater apenas aos que realmente considero relevantes na prática, que fazem diferença no dia a dia.

A seguir a lista dos atributos que considero relevantes.

Atributo	Referente a
Unidade	O RF deve propor uma única coisa apenas. Não deve atender a mais de uma exigência. O RF “Incluir cliente” não é unitário, pois se refere a incluir clientes de tipos diferentes (pessoa física e jurídica), assumindo assim várias responsabilidades, quando deveria assumir apenas uma.
Compleitude	O RF deve ser autocontido, deve ter “início/meio/fim”, ser completo. O RF “Pagar fatura” não é completo, só conta “parte da estória”. Para ser completo deveria ser algo como “Pagar fatura com cartão de crédito para cliente pessoa física”.
Consistência	O RF não deve contradizer outro RF do mesmo escopo do projeto. É como termos dois RFs se propondo a fazer uma mesma coisa, mas cada RF se propondo a fazer esta coisa de uma forma diferente.
Atomicidade	Um RF para ser atômico precisa também ter unidade, pois atomicidade remete a assumir apenas uma responsabilidade. Mas também deve ser algo indivisível, não podendo ser decomposto. Muitos RFs possuem conjunção, dependem de outros para se realizarem. Onde temos dois RFs “Realizar compra de produto” e “Realizar pagamento com cartão de crédito” na realidade, se pensarmos em atomicidade, temos um único RF que é “Realizar compra de produto com pagamento em cartão de crédito”.
Não-Ambiguidade	Um RF não pode ser ambíguo, não pode propor algo que não fica claro o que é. O RF “Emitir relatório” não quer dizer nada. Relatório de que, para que? “Emitir relatório de saldo” já é melhor, mas ainda é ruim. Saldo de que? Seria não ambíguo se não deixasse dúvidas, algo como “Emitir relatório de saldo da conta corrente do cliente pessoa física”.
Verificável	Não adianta ter um RF se ele não é palpável, possível de associar com um artefato de construção, de testes. Um RF tem que ser testável, tem que ser possível atestar que o RF foi atendido, foi construído, foi homologado. Para isso tem que ser também rastreável.
Rastreável	Deve ser possível achar o RF no sistema pronto, funcional e executável. Como saber se um RF foi atendido? Para isso é necessário ter rastreabilidade, e isso só é possível ligando as pontas (associar o RF à interface gráfica, que será associada a um caso de uso, que será associado a funcionalidades, que serão implementadas etc.).
Prioridade	Um RF Essencial é algo muito diferente de um RF Desejável, possuem valores para o negócio completamente diferentes. O RF deve possuir sua prioridade, isso interfere diretamente no projeto do software.

Um detalhe fundamental é o uso do tempo verbal no nome do RF. Um RF, em tempo de especificação, **refere-se a algo que será feito, uma ação a ser realizada pelo sistema**. Por isso o nome precisa estar no tempo verbal infinitivo. Um RF que fala sobre “expurgo de registros de clientes inativos” não pode ter este nome, deve se chamar “Expurgar registros de clientes inativos”.

É uma necessidade, mas que precisa ser verbalizada como uma ação a ser realizada.