

## O que é Requisito Funcional

By **Plínio Ventura**

On 11 maio, 2016

Last updated 7 fev, 2021

 53

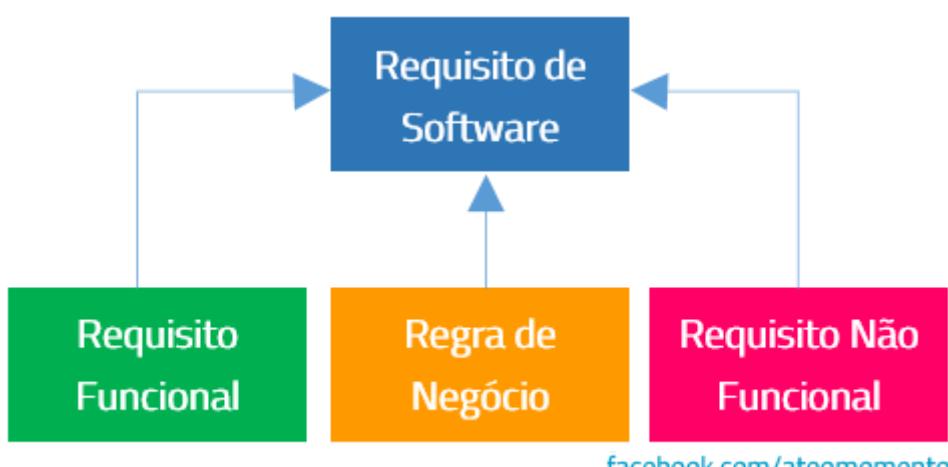
### Compartilhe!



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.



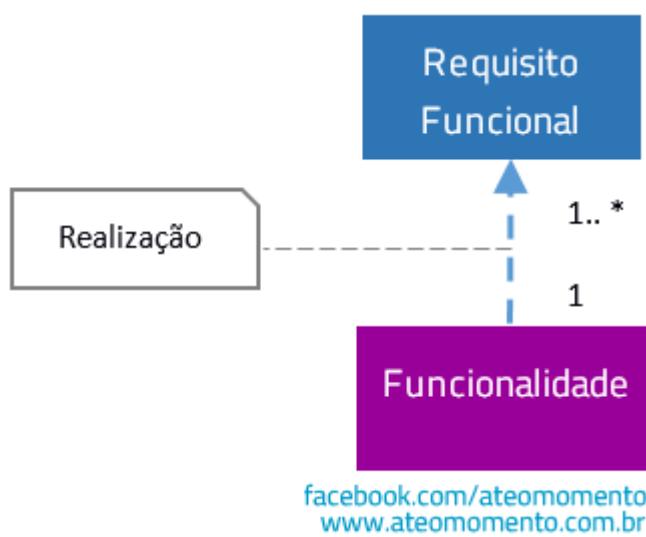
É comum os profissionais de engenharia de software associarem a ideia de um requisito funcional a uma tela,

[facebook.com/ateomomento](http://facebook.com/ateomomento)  
[www.ateomomento.com.br](http://www.ateomomento.com.br)

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

Requisitos Funcionais (Identificador e Nome)
RF004 – Excluir cliente pessoa física
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

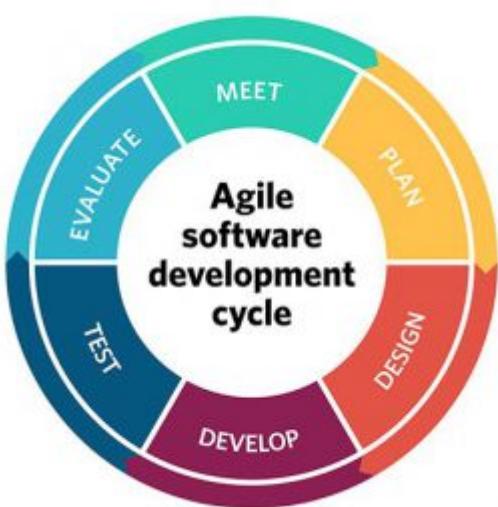
Funcionalidades somente existem para realizar Requisitos Funcionais. Logo, sem requisitos funcionais não há funcionalidades e sem funcionalidades não há sistema.

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
<b>Unidade</b>	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.
<b>Completude</b>	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”.
<b>Consistência</b>	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.

Atributo	Referente a
<b>Atomicidade</b>	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”.
<b>Não-Ambiguidade</b>	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”.
<b>Verificável</b>	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.
<b>Rastreável</b>	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.).

Atributo	Referente a
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.

## Estrutura de um Requisito Funcional

Não há um padrão estabelecido sobre a estrutura de um RF.

Mas a maioria das empresas utiliza um formato semelhante, contendo campos específicos. O modelo a seguir contempla os campos mais relevantes, com posterior descrição de cada um.

O modelo citado é aplicável em especificações produzidas em Editores de Texto como o Microsoft Word, por exemplo.

É recomendado que se utilize, sempre que possível, alguma ferramenta Case que dê suporte a Engenharia de Requisitos, para melhorar a produtividade na modelagem e a gestão dos requisitos.

Identificador	
Nome	

<b>Módulo</b>			
<b>Data de criação</b>		<b>Autor</b>	
<b>Data da última alteração</b>		<b>Autor</b>	
<b>Versão</b>	<b>Prioridade</b>		
<b>Descrição</b>			

## Explicando cada campo

Campo	Descrição
<b>Identificador</b>	Sufixo seguido de um identificador único. O sufixo geralmente utilizado é RF (Requisito Funcional) e o identificador único geralmente é composto de quatro dígitos.
<b>Nome</b>	Nome curto do RF, mas que possibilite entender bem o que RF faz apenas pelo nome.
<b>Módulo</b>	Módulo ao qual o RF pertence. Se for um sistema pequeno que não possua nenhum módulo, somente o próprio sistema, deve ser preenchido com N/A (não se aplica).
<b>Data de criação</b>	Data da criação do RF, ou a data em que ele foi especificado.
<b>Autor</b>	Profissional que especificou o RF pela primeira vez, quem o criou.
<b>Data da última alteração</b>	Data em que houve a última alteração no RF.

Campo	Descrição
<b>Autor</b>	Profissional que alterou a especificação do RF pela última vez.
<b>Versão</b>	Número da versão do RF. Geralmente utiliza-se algo simples, como 1, 2. A versão inicial sempre é a 1, e a cada alteração incrementa-se a versão (na criação versão 1, na primeira alteração versão 2 e por aí vai).
<b>Prioridade</b>	Se o RF é Essencial, Importante ou Desejável.
<b>Descrição</b>	Descrição detalhada (a mais detalhada possível) do RF.

”

*/\* O uso de identificador é fundamental para uma melhor organização do projeto. Localizar requisitos pelo nome não funciona, devido ao volume dos requisitos e ambiguidades nos nomes. E também, é importante ter algum recurso de numeração automática, pois controlar os números dos identificadores manualmente é inviável. \*/*

Anúncio fechado pela  criteo.

Denunciar este anúncio

Ad choices 

# Exemplo de um Requisito Funcional especificado

<b>Identificador</b>	RF0003		
<b>Nome</b>	Emitir carta de cobrança para clientes inadimplentes conforme critérios pré-estabelecidos		
<b>Módulo</b>	Cobrança		
<b>Data de criação</b>	31/01/2016	<b>Autor</b>	Hípias de Elis
<b>Data da última alteração</b>	N/A	<b>Autor</b>	N/A
<b>Versão</b>	1	<b>Prioridade</b>	Essencial

<b>Descrição</b>	<p>Para todo cliente que esteja inadimplente deverá ser possível a emissão de carta de cobrança através do sistema. O resultado da emissão será a carta impressa, que será posteriormente despachada por correios ou outro agente de entrega.</p> <p>Os critérios para definir se um cliente é ou não inadimplente devem estar cadastrados no sistema utilizando regras de negócio específicas. Nem todo cliente com pagamento em atraso será considerado inadimplente, pois deverá ser levado em consideração o score de crédito do cliente, sua renda mensal, patrimônio etc. Obs.: verificar os Requisitos Funcionais e regras de negócio específicas para manutenção dos critérios citados.</p> <p>Não haverá diferença na emissão da carta de cobrança para clientes pessoa física (particular) ou jurídica (empresarial), logo, o processo será o mesmo para ambos.</p>
------------------	---

Obrigado pela leitura, espero que possa ter sido útil.

Se você gostou do conteúdo deste post, deixe seus comentários, para ajudar outras pessoas a encontrarem este material! 😊

Grande abraço!

## Related Posts via Categories

- O que é UML (Unified Modeling Language)
- Entendendo o Diagrama de Instalação da UML
- Entendendo o Diagrama de Sequência da UML
- Entendendo o Diagrama de Classes da UML
- Caso de Uso ou Estória de Usuário?
- Analista de Infraestrutura e Requisito Não Funcional
- FAQ04 – Perguntas e Respostas sobre Engenharia de Software
- FAQ03 – Perguntas e Respostas sobre Engenharia de Software