

UNIVERSIDADE FEDERAL DE LAVRAS

Trio Bit Garage

Projeto de Engenharia de Software

Regras de Verificação e Análise de Requisitos

Integrantes:

Caio Finnochio Martins

Fábio Damas Valim

Guilherme Lirio Miranda

Lavras, Minas Gerais

27 de outubro de 2025

1 Introdução

Este documento estabelece as diretrizes e os padrões para a especificação, verificação e análise dos requisitos de software do projeto. O objetivo é garantir que todos os requisitos sejam descritos de forma clara, consistente, completa e testável, facilitando o trabalho das equipes de desenvolvimento, testes e gestão. A adesão a estas regras é obrigatória para todos os membros do projeto.

2 Nomenclatura dos Requisitos

Para garantir a rastreabilidade e a organização, todos os requisitos deverão seguir a nomenclatura padrão descrita abaixo.

2.1 Requisitos Funcionais (RF)

Os Requisitos Funcionais descrevem *o que* o sistema deve fazer, ou seja, as funcionalidades e serviços que ele deve oferecer aos usuários.

- **Formato:** RFXXX
- **Descrição:** RF significa "Requisito Funcional", e XXX é um número sequencial iniciado em 001.
- **Exemplo:**
 - RF001: O sistema deve permitir que o usuário realize login utilizando e-mail e senha.
 - RF002: O sistema deve permitir o cadastro de novos veículos no estoque.

2.2 Requisitos Não Funcionais (RNF)

Os Requisitos Não Funcionais descrevem *como* o sistema deve realizar suas funções. Eles definem restrições e critérios de qualidade, como desempenho, segurança, usabilidade e confiabilidade.

- **Formato:** RNFXXX
- **Descrição:** RNF significa "Requisito Não Funcional", e XXX é um número sequencial iniciado em 001.
- **Exemplo:**
 - RNF001: O tempo de resposta para qualquer consulta de veículo no catálogo não deve exceder 2 segundos.
 - RNF002: O sistema deve ser compatível com as versões mais recentes dos navegadores Google Chrome, Mozilla Firefox e Microsoft Edge.

3 Regras para Especificação de Requisitos

Com base nos princípios da Engenharia de Software, as seguintes regras, extraídas do livro de referência da disciplina (Engenharia de Software - Rogério Magela), serão adotadas para a escrita de todos os requisitos.

3.1 Clareza e Inexistência de Ambiguidade

Cada requisito deve ser escrito em linguagem clara, simples e direta, possuindo uma única interpretação possível. Devem ser evitados termos vagos como "eficiente", "rápido" ou "amigável" sem uma metrificação quantitativa associada.

- **Exemplo Incorreto:** O sistema deve ter uma interface amigável.
- **Exemplo Correto:** RNF003: O sistema deve permitir que um novo usuário complete o cadastro de um cliente em no máximo 3 minutos, sem necessidade de treinamento prévio.

3.2 Atomicidade

Cada requisito deve descrever uma única capacidade, funcionalidade ou restrição do sistema. Não se deve agrupar múltiplos requisitos em uma única sentença. Isso facilita a rastreabilidade, o desenvolvimento e os testes de forma isolada.

- **Exemplo Incorreto:** O sistema deve permitir o cadastro e a exclusão de veículos.
- **Exemplo Correto:**
 - RF003: O sistema deve permitir o cadastro de um novo veículo.
 - RF004: O sistema deve permitir a exclusão de um veículo existente.

3.3 Verificabilidade (Testabilidade)

Todo requisito deve ser escrito de forma que seja possível verificar, por meio de testes, inspeção ou análise, se ele foi ou não implementado corretamente. Se um requisito não pode ser testado, ele não pode ser considerado cumprido.

- **Exemplo Incorreto:** O sistema deve ser robusto e nunca falhar.
- **Exemplo Correto:** RNF004: O sistema deve permanecer operacional e disponível em 99.5% do tempo durante o horário comercial (9h às 18h, de segunda a sexta-feira).