

Regras de Verificação e Análise de Requisitos

Ficha Técnica

Equipe Responsável pela Elaboração

Daniel Silva Ferraz Neto
João Gabriel Salomão Baldim
José Victor Miranda

Público Alvo

Este manual destina-se à equipe de desenvolvimento e ao professor.

Regras de Verificação e Análise de Requisitos

Este documento estabelece os padrões e critérios que devem ser obedecidos durante a especificação, análise e verificação dos requisitos do sistema Calendars, baseando-se nas melhores práticas de Engenharia de Software (referência: Rogério Magela).

1. Padronização de Nomenclatura e Identificação

Para garantir a organização e rastreabilidade, todos os requisitos devem seguir estritamente a seguinte codificação:

- **Requisitos Funcionais:** Devem ser identificados pela sigla RF seguida de um número sequencial de três dígitos.
 - **Exemplo:** [RF001], [RF002].
- **Requisitos Não Funcionais:** Devem ser identificados pela sigla NF (ou RNF) seguida de um número sequencial de três dígitos.
 - **Exemplo:** [NF001], [NF003].

2. Classificação de Prioridades

A priorização dos requisitos deve utilizar as seguintes denominações para classificar a importância de cada funcionalidade:

- **Essencial:** Requisito imprescindível. Sem ele, o sistema não entra em funcionamento.
- **Importante:** O sistema funciona sem ele, mas de forma não satisfatória. Deve ser implementado se houver tempo hábil.
- **Desejável:** Funcionalidade que não compromete o funcionamento básico do sistema. Pode ser deixada para versões futuras (backlog).

3. Regras de Especificação e Verificação (Baseado em Rogério Magela)

Todo requisito descrito no documento deve passar pelo crivo das seguintes regras de análise para ser considerado "Aprovado" ou "Pronto para Desenvolvimento":

Regra 1: Completude e Padronização de Estrutura

Todo Requisito Funcional deve ser descrito utilizando o template padrão de Caso de Uso adotado no projeto. A verificação deve garantir que todos os seguintes campos estejam preenchidos:

1. **Identificador e Nome:** Único e iniciado por verbo no infinitivo (ex: "Autenticar Usuário").
2. **Prioridade e Atores:** Quem executa a ação.
3. **Resumo:** Breve descrição do objetivo.
4. **Pré-condição e Pós-condição:** O estado do sistema antes e depois da execução.
5. **Fluxo Principal:** O "caminho feliz" passo-a-passo.
6. **Fluxo Alternativo:** Tratamento de erros ou caminhos de exceção (ex: "Senha inválida").

Regra 2: Clareza e Mensurabilidade (Evitar Ambiguidade)

Os requisitos, especialmente os Não Funcionais, não podem ser subjetivos. Termos como "rápido", "seguro" ou "amigável" são proibidos se não vierem acompanhados de métricas.

- **Critério de Verificação:** O requisito possui uma métrica clara de aceitação?
- **Exemplo Correto no Projeto:** "[NF002] ...executar, no mínimo, 30 requisições de leitura mantendo um tempo de resposta até 500 milissegundos".
- **Exemplo Correto no Projeto:** "[NF003] ...funcional em computadores/laptops com, no mínimo, 4 gigabytes de memória RAM".

Regra 3: Atomicidade e Unicidade

Cada requisito deve tratar de uma única funcionalidade ou restrição. Não se deve agrupar múltiplos comportamentos complexos em um único identificador se eles puderem ser testados e implementados separadamente.

- **Critério de Verificação:** O requisito descreve apenas uma função ou característica? Se o requisito contiver a conjunção "e" ligando dois processos distintos (ex: "Cadastrar usuário E emitir relatório"), ele deve ser quebrado em dois (RFXXX e RFXYY).

4. Rastreabilidade

Para a gestão do backlog no GitHub (Tasks #14 e #15), deve ser garantido que:

1. **Todo item no quadro Kanban (Issue) tenha o identificador do requisito correspondente no título (ex: "RF001 - Implementar Tela de Login").**

- 2. Interfaces desenhadas no protótipo devem ser citadas no campo "Interfaces" do documento de requisitos (ex: I_Login, I_Dashboard).**