

# Artefato 3: Especificação de Requisitos Não Funcionais

## Sistema de Gestão de Artefatos Scrum (SGAS)

Arthur Luiz Lima de Araújo  
Arthur da Silva Pereira Bispo  
João Carlos Gonçalves de Oliveira Filho  
Ana Luísa de Souza Paraguassu  
Tauã Valentim de A. M. Frade

06 Novembro de 2025

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Requisitos Funcionais de Escopo Global</b>	<b>3</b>
<b>3</b>	<b>Qualidades do Sistema</b>	<b>4</b>
3.1	Usabilidade (Usability) . . . . .	4
3.2	Confiabilidade (Reliability) . . . . .	5
3.3	Desempenho (Performance) . . . . .	5
3.4	Suportabilidade (Supportability) . . . . .	5
<b>4</b>	<b>Interfaces do Sistema</b>	<b>5</b>
4.1	Interfaces de Usuário (GUI) . . . . .	6
4.2	Interfaces com Sistemas Externos . . . . .	6
4.3	Interfaces de Software . . . . .	6
4.4	Interfaces de Hardware . . . . .	6
4.5	Interfaces de Comunicação . . . . .	6
<b>5</b>	<b>Regras de Negócio</b>	<b>7</b>
<b>6</b>	<b>Restrições do Sistema</b>	<b>8</b>
6.1	Restrições Técnicas . . . . .	8
6.2	Restrições de Design e Arquitetura . . . . .	8
6.3	Restrições de Escopo e Operação . . . . .	8
<b>7</b>	<b>Conformidade e Licenciamento</b>	<b>9</b>
7.1	Licenciamento Open-Source . . . . .	9
7.2	Conformidade Legal . . . . .	9
7.3	Padrões Aplicáveis . . . . .	9
<b>8</b>	<b>Documentação do Sistema</b>	<b>9</b>

# 1. Introdução

Este documento apresenta a especificação de requisitos não funcionais do Sistema de Gestão de Artefatos Scrum (SGAS). O objetivo é descrever as propriedades globais de qualidade, restrições técnicas e normas de conformidade que o sistema deve atender, complementando os requisitos funcionais definidos nas Histórias de Usuário.

Esses requisitos não definem novas funcionalidades, mas estabelecem critérios sobre como o sistema deve operar — abrangendo aspectos como segurança, desempenho, usabilidade, confiabilidade, manutenção e conformidade legal. Este documento servirá de referência para as decisões de arquitetura, implementação e implantação do SGAS, garantindo a aderência aos padrões de qualidade e aos objetivos estabelecidos no Documento de Visão e Escopo.

## 2. Requisitos Funcionais de Escopo Global

Esta seção detalha requisitos funcionais que se aplicam ao sistema de forma global, complementando as Histórias de Usuário (Artefato 4). Eles definem regras de negócio, segurança e usabilidade consistentes em múltiplas funcionalidades.

Tabela 1: Requisitos Funcionais de Escopo Global

ID	Descrição do Requisito	Prioridade
RFS-01	O sistema deve garantir que um usuário só possa visualizar e interagir com os artefatos (backlogs, sprints) de projetos aos quais ele foi formalmente adicionado.	Essencial
RFS-02	Ao realizar o logout, o cliente (frontend) deve remover o token de autenticação do armazenamento local, invalidando a sessão.	Essencial
RFS-03	O sistema deve exibir uma caixa de diálogo de confirmação (ex: "Você tem certeza?") antes de executar qualquer ação de exclusão permanente.	Essencial
RFS-04	O sistema não deve permitir a criação de duas histórias de usuário com exatamente o mesmo título dentro do mesmo projeto.	Desejável
RFS-05	Após o login, o usuário deve ser direcionado para uma página principal (dashboard) que lista todos os projetos em que ele está ativo.	Essencial

Tabela 1: Requisitos Funcionais de Escopo Global

ID	Descrição do Requisito	Prioridade
RFS-06	O sistema deve ter um menu de navegação persistente (na parte superior ou lateral da tela) que permita acesso rápido ao dashboard de projetos e à página de perfil do usuário.	Essencial
RFS-07	O sistema deve prover uma funcionalidade de busca simples no backlog do produto para filtrar histórias de usuário pelo título.	Desejável
RFS-08	O sistema deve impor rigorosamente as permissões baseadas em papéis (PO, SM, Dev). Um usuário com um papel (ex: 'Developer') não deve poder executar ações de outro papel (ex: 'Criar Sprint' ou 'Excluir História do Product Backlog').	Essencial
RFS-09	O sistema não deve permitir a exclusão de um 'Projeto' se ele ainda contiver 'Histórias de Usuário' associadas (seja no Product Backlog ou em Sprints ativos).	Essencial
RFS-10	Os artefatos 'Histórias de Usuário' e 'Backlog do Produto' devem ser globais ao projeto, permitindo que sejam acessados por outros módulos (como a criação do Backlog de Sprint).	Essencial

### 3. Qualidades do Sistema

Esta seção detalha os atributos de qualidade (requisitos não funcionais) que o sistema SGAS deve atender, baseada no modelo FURPS+ [1].

#### 3.1. Usabilidade (Usability)

Requisitos relacionados à facilidade de uso, aprendizado e consistência da interface.

- O sistema deve possuir uma interface coerente e previsível, mantendo padrões de design consistentes em todas as telas.
- As ações primárias (como 'Salvar' e 'Criar') e secundárias/destrutivas (como 'Cancelar' e 'Excluir') devem utilizar estilos visuais distintos e padronizados.
- O sistema deve oferecer mensagens claras de feedback visual ou textual a cada ação executada (confirmação, erro, alerta).

### **3.2. Confiabilidade (Reliability)**

Requisitos relacionados à disponibilidade, integridade dos dados e robustez do sistema.

- O sistema deve garantir a integridade referencial dos dados, impedindo a exclusão de entidades com dependências ativas.
- O sistema deve garantir a autorização de todas as operações de CRUD com base nas permissões de cada papel (PO, SM, Dev).
- As senhas devem ser armazenadas utilizando algoritmo de hash com salt (ex: bcrypt), impedindo o armazenamento em texto plano.
- Em caso de falha de conexão ou erro de servidor, o sistema deve exibir uma mensagem de erro comprehensível.

### **3.3. Desempenho (Performance)**

Requisitos relacionados ao tempo de resposta, capacidade e eficiência do sistema.

- O tempo de carregamento inicial do sistema, após autenticação, não deve exceder 3 segundos em conexão banda larga padrão.
- Operações de CRUD de artefatos (histórias, sprints, backlog) devem apresentar retorno visual ou textual (feedback) em até 1,5 segundo.

### **3.4. Suportabilidade (Supportability)**

Requisitos que facilitam a manutenção (manutenibilidade), portabilidade e compatibilidade do sistema.

- O sistema deve ser implementado utilizando tecnologias open-source.
- A arquitetura adotada deve ser N-Tier (Frontend, API e Banco de Dados), possibilitando substituição modular de componentes.
- A compatibilidade deve ser garantida com as duas últimas versões estáveis dos navegadores Google Chrome e Mozilla Firefox.
- O código-fonte deve seguir boas práticas de documentação interna e versionamento no GitHub.

## **4. Interfaces do Sistema**

Esta seção descreve os requisitos de interface do SGAS, detalhando como ele se conecta com usuários e outros componentes de software.

#### **4.1. Interfaces de Usuário (GUI)**

A interface com o usuário é um requisito chave, podendo ser implementada em uma das duas modalidades a seguir:

- O SGAS será disponibilizado como uma aplicação web (GUI) projetada para ser executada localmente (ex: *localhost*) e acessada via navegador, com foco em clareza e simplicidade.

#### **4.2. Interfaces com Sistemas Externos**

Conforme definido no Documento de Visão e Escopo ("Fora do Escopo"), o sistema não terá interfaces com sistemas externos de terceiros (ex: GitHub, Jira).

#### **4.3. Interfaces de Software**

A arquitetura N-Tier define as seguintes interfaces de software internas:

- Frontend (Apresentação): Aplicação web (SPA).
- Backend (Aplicação): API RESTful responsável por autenticação, autorização e lógica de negócios.
- Banco de Dados (Persistência): Banco de dados relacional para armazenamento transacional dos dados.
- A camada de persistência é acessada exclusivamente pelo backend, utilizando o ORM nativo do Django.

#### **4.4. Interfaces de Hardware**

O sistema não possui requisitos de interface com hardware especializado. Ele é projetado para ser executado em hardware de usuário padrão (desktops, notebooks).

#### **4.5. Interfaces de Comunicação**

- O frontend e o backend se comunicam via protocolo HTTP, seguindo o padrão REST (utilizando JSON), consistente com uma aplicação de execução local.
- O backend (API) comunica-se com o banco de dados (SQLite) por meio do ORM do Django, com acesso direto ao sistema de arquivos do servidor.

## 5. Regras de Negócio

Esta seção define as regras de negócio que restringem o comportamento do sistema, derivadas diretamente do framework Scrum [3] e das responsabilidades dos papéis (PO, SM, Dev). O usuário do sistema é neutro e as permissões são definidas por projeto.

Tabela 2: Regras de Negócio por Papel no Projeto

ID	Categoria/Papel	Descrição da Regra
RN-01	Sistema (Acesso)	Usuários não autenticados não podem visualizar, criar ou editar qualquer artefato do sistema.
RN-02	Sistema (Acesso)	Ao ser removido de um projeto, o usuário perde automaticamente todas as permissões associadas a esse contexto.
RN-03	Sistema (Projeto)	Qualquer usuário autenticado pode criar um novo projeto e será automaticamente definido como Product Owner (PO) desse projeto.
RN-04	Sistema (Visibilidade)	Todos os membros de um projeto podem visualizar seu backlog, sprints e histórias, respeitando os níveis de permissão para edição.
RN-05	Product Owner	O PO de um projeto pode atribuir outros usuários a esse projeto, definindo seus papéis (SM, Dev).
RN-06	Scrum Master	Pode Criar e configurar um novo Sprint.
RN-07	Scrum Master	A mudança de status de uma Sprint (de planejada para ativa ou concluída) só pode ser executada pelo SM.
RN-08	Scrum Master	Pode criar o Backlog de Sprint selecionando 'Itens do Backlog do Produto'.
RN-09	Developer	Pode consultar o Plano de Sprint e o Sprint Backlog.
RN-10	Developer	Pode atualizar o status do trabalho (tarefas) no Plano de Sprint.
RN-11	Sistema (Integridade)	Nenhum artefato pode ser excluído se estiver associado a outros dependentes (ex: não excluir projeto com sprints abertas).
RN-12	Sistema (Integridade)	Cada artefato criado (projeto, backlog, etc.) deve estar vinculado a um único projeto e possuir autor responsável.

Tabela 2: Regras de Negócio por Papel no Projeto

ID	Categoria/Papel	Descrição da Regra
RN-13	Sistema (Integridade)	Uma sprint só pode ser encerrada se todas as suas histórias estiverem concluídas ou justificadas como não entregues.

## 6. Restrições do Sistema

Esta seção detalha as restrições (constraints) do projeto, que são decisões ou limitações impostas que afetam o design e a implementação do SGAS.

### 6.1. Restrições Técnicas

- O sistema deve ser desenvolvido exclusivamente com tecnologias open-source.
- O backend deve ser implementado em Python, utilizando o framework Django Rest Framework (DRF).
- O frontend deve ser desenvolvido em React, utilizando Vite como ferramenta de construção (build tool).
- O banco de dados utilizado deve ser SQLite, o banco de dados padrão do Django, garantindo simplicidade na implantação local.

### 6.2. Restrições de Design e Arquitetura

- A arquitetura do sistema deve seguir o modelo N-Tier (Apresentação, Aplicação e Persistência).
- A comunicação entre as camadas deve ser realizada por chamadas REST. A autenticação será baseada em token (ex: JWT), que deve ser enviado no cabeçalho de cada requisição para validação.
- O design da interface deve priorizar o uso em navegadores desktop, não havendo obrigatoriedade de compatibilidade mobile.
- O sistema deve suportar o idioma português do Brasil.

### 6.3. Restrições de Escopo e Operação

- O SGAS destina-se exclusivamente ao apoio de equipes que utilizam o framework Scrum.

- O sistema não contempla gestão financeira, controle de tempo ou métricas de produtividade avançadas.
- O acesso ao sistema é feito via navegador moderno, primariamente em ambiente local (*localhost*).
- O sistema não dependerá de integração obrigatória com plataformas externas (ex: GitHub, Jira).
- O sistema não proverá funcionalidade para um usuário deletar a própria conta.

## 7. Conformidade e Licenciamento

Esta seção detalha os requisitos de conformidade legal (normas e leis) e os aspectos de licenciamento do software.

### 7.1. Licenciamento Open-Source

- O projeto SGAS será licenciado sob uma licença permissiva open-source (ex: MIT).
- O sistema depende de bibliotecas de terceiros (ex: Django, React, Vite). A equipe deve garantir que o uso dessas bibliotecas esteja em conformidade com suas respectivas licenças (ex: MIT, BSD).

### 7.2. Conformidade Legal

- O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD) do Brasil (Lei nº 13.709/2018) [2].

### 7.3. Padrões Aplicáveis

- Padrões Web: A aplicação deve seguir os padrões web modernos.
- Padrões de API: A comunicação deve seguir o padrão de arquitetura RESTful sobre HTTP.

## 8. Documentação do Sistema

Esta seção descreve os requisitos de documentação que acompanharão o SGAS.

- Documentação de Implantação: Deve ser fornecido um guia ('README.md') detalhando os passos para configurar e executar o ambiente de desenvolvimento localmente, como a instalação de dependências e a configuração de variáveis de ambiente.
- Documentação Interna (Código): O código-fonte deve conter comentários claros para facilitar a manutenção.

## Referências

- [1] **FURPS+:** Modelo de classificação de requisitos de software, popularizado pela Hewlett-Packard. Os atributos de qualidade são baseados em padrões internacionais como a ISO/IEC 25010:2011.
- [2] **LGPD:** Lei Geral de Proteção de Dados Pessoais, Lei nº 13.709, de 14 de agosto de 2018.
- [3] **Scrum:** SCHWABER, Ken; SUTHERLAND, Jeff. O Guia do Scrum: O Guia Definitivo para o Scrum: As Regras do Jogo. Scrum.org, 2020.