

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

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

1. Introdução

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

Em conformidade com a proposta apresentada no Artefato 2, o SGAS mantém foco em simplicidade, leveza e aderência estrita ao framework Scrum, evitando funcionalidades excessivas e privilegiando uma experiência essencialista. Assim, este documento reforça diretrizes que preservem tais princípios durante a arquitetura, implementação e implantação do sistema.

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 artefatos de projetos aos quais foi formalmente adicionado.	Essencial
RFS-02	Ao realizar o logout, o frontend deve remover o token armazenado, invalidando a sessão.	Essencial
RFS-03	Antes de executar qualquer ação de exclusão permanente, o sistema deve solicitar confirmação explícita ao usuário.	Essencial
RFS-04	O sistema não deve permitir a criação de duas histórias de usuário com exatamente o mesmo título dentro de um mesmo projeto.	Desejável
RFS-05	Após o login, o usuário deve ser direcionado ao dashboard contendo a lista de projetos nos quais participa.	Essencial
RFS-06	O sistema deve fornecer um menu de navegação persistente, consistente e sempre acessível, permitindo acesso ao dashboard.	Essencial
RFS-07	O sistema deve permitir busca simples no Product Backlog pelo título das histórias.	Desejável

Tabela 1: Requisitos Funcionais de Escopo Global

ID	Descrição do Requisito	Prioridade
RFS-08	O sistema deve aplicar rigorosamente as permissões definidas pelos papéis (PO, SM, Dev). A autorização deve ser verificada no backend, garantindo que um usuário só execute ações compatíveis com seu papel em cada projeto.	Essencial
RFS-09	O sistema não deve permitir a exclusão de um projeto caso existam histórias ou sprints associados a ele.	Desejável
RFS-10	Os artefatos “Histórias de Usuário” e “Product Backlog” devem ser globais ao projeto, permitindo acesso e utilização por outros módulos (ex.: criação do Sprint Backlog).	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.

- A interface deve manter consistência visual e funcional em todas as telas.
- As ações primárias (Salvar, Criar) devem utilizar botão azul e sólido, conforme definido no Artefato 4.
- Ações de edição devem utilizar botão branco.
- Ações destrutivas devem utilizar botão vermelho.
- O sistema deve prover feedback visual claro em menos de 1.5s após cada operação.

3.2. Confiabilidade (Reliability)

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

- O sistema deve garantir integridade referencial, impedindo exclusão de entidades com dependências.
- Todas as operações de CRUD devem validar permissões no backend.
- Senhas devem ser armazenadas utilizando hash com salt.
- Mensagens de erro devem ser claras em caso de falha de conexão ou servidor.

3.3. Desempenho (Performance)

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

- O carregamento inicial após login deve ocorrer em até 3 segundos.
- Operações CRUD devem retornar feedback em no máximo 1.5 segundos.

3.4. Suportabilidade (Supportability)

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

- O sistema deve utilizar tecnologias open-source.
- A arquitetura deve seguir o modelo N-Tier.
- Compatibilidade garantida com as duas últimas versões estáveis de Chrome e Firefox.
- 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)

O sistema será acessado via navegador web, com interface clara e simplificada, projetada para ser executada localmente em desktops e notebooks.

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 artefatos.
RN-02	Sistema (Acesso)	Ao ser removido de um projeto, o usuário perde automaticamente todas as permissões associadas ao projeto.
RN-03	Sistema (Projeto)	Ao criar um projeto, o usuário é automaticamente definido como Scrum Master (SM).
RN-04	Sistema (Visibilidade)	Todos os membros podem visualizar backlog, sprints e histórias, respeitando permissões de edição.
RN-05	Product Owner	O PO pode criar, editar, priorizar e excluir histórias do Product Backlog.
RN-06	Scrum Master	O SM pode criar e configurar Sprints.

Tabela 2: Regras de Negócio por Papel no Projeto

ID	Categoria/Papel	Descrição da Regra
RN-07	Scrum Master	Apenas o SM pode alterar o status de uma Sprint (planejada, ativa, concluída).
RN-08	Developer	O desenvolvedor cria o Sprint Backlog selecionando itens do Product Backlog.
RN-09	Developer	O desenvolvedor pode consultar o Plano de Sprint e o Sprint Backlog.
RN-10	Developer	O desenvolvedor pode atualizar o status do trabalho no Plano de Sprint.
RN-11	Sistema (Integridade)	Nenhum artefato pode ser excluído se estiver associado a dependentes ativos.
RN-12	Sistema (Integridade)	Cada artefato criado deve estar vinculado a um único projeto e possuir autor responsável.
RN-13	Sistema (Integridade)	Uma sprint só pode ser encerrada se todas as histórias estiverem concluídas ou justificadas.

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.
- O frontend deve ser desenvolvido em React, utilizando Vite como ferramenta de construção.
- 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.
- A comunicação entre as camadas deve ser realizada por chamadas REST. A autenticação será baseada em token, 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.
- 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.
- 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.