

Artefato 9: Infraestrutura de Implantação

Sistema de Gestão de Artefatos Scrum (SGAS)

Arthur Luiz Lima de Araújo
Arthur da Silva Pereira Bispo
João Carlos Goncalves de Oliveira Filho
Ana Luisa de Souza Paraguassu
Taua Valentim de A. M. Frade

15 de novembro de 2025

Conteúdo

1	Introdução	3
1.1	Propósito	3
1.2	Escopo	3
1.3	Visão Geral do Sistema	3
2	Requisitos de Hardware	4
2.1	Ambiente de Desenvolvimento	4
2.2	Ambiente de Produção	4
3	Requisitos de Software	5
3.1	Sistema Operacional	5
3.2	Ambiente de Execução	5
3.2.1	Backend	5
3.2.2	Frontend	5
3.3	Frameworks e Bibliotecas Principais	5
3.3.1	Backend	5
3.3.2	Frontend	6
3.4	Banco de Dados	6
3.5	Controle de Versão	6
4	Requisitos de Serviços	7
4.1	Serviços de Rede	7
4.1.1	Conectividade e Protocolos	7
4.1.2	Portas de Comunicação	7
4.2	Infraestrutura de Hospedagem	7
4.2.1	Serviços de Servidor	7
4.3	Serviços de Desenvolvimento	7
4.3.1	Controle de Versão	7
4.4	Serviços de Segurança	8
4.4.1	Autenticação e Autorização	8
4.5	Serviços de Backup	8
5	Arquitetura de Implantação	9
5.1	Ambiente de Desenvolvimento	9
5.2	Ambiente de Produção	9
5.3	Fluxo de Dados	9
6	Conclusão	10

1 Introdução

1.1 Propósito

Este documento descreve a infraestrutura necessária para a implantação do Sistema de Gestão de Artefatos Scrum, fornecendo informações sobre os requisitos de hardware, software e serviços utilizados no desenvolvimento e implantação do sistema.

1.2 Escopo

Este documento abrange:

- Requisitos de hardware para implantação
- Software utilizado e suas dependências
- Serviços necessários para operação do sistema
- Arquitetura de implantação

1.3 Visão Geral do Sistema

O SGAS é um sistema web para gerenciamento de projetos Scrum, composto por:

- **Backend:** API RESTful em Django REST Framework
- **Frontend:** Aplicação React com Vite e TypeScript
- **Banco de Dados:** SQLite

2 Requisitos de Hardware

2.1 Ambiente de Desenvolvimento

Para o ambiente de desenvolvimento local, cada desenvolvedor necessita de uma estação de trabalho com as seguintes características mínimas:

Componente	Especificação Mínima
Processador	Dual-core 2.0 GHz ou superior
Memória RAM	4 GB (8 GB recomendado)
Armazenamento	10 GB de espaço livre
Conectividade	Conexão de rede com acesso à internet

Tabela 1: Requisitos de hardware para desenvolvimento

2.2 Ambiente de Produção

O sistema foi projetado para ser executado em um servidor com as seguintes especificações básicas:

Componente	Especificação
Processador	Mínimo 2 núcleos virtuais (vCPU)
Memória RAM	Mínimo 4 GB
Armazenamento	Mínimo 20 GB de espaço em disco
Rede	Conexão estável com largura de banda adequada

Tabela 2: Requisitos de hardware para produção

Observações:

- Os requisitos podem variar de acordo com o número de usuários simultâneos e volume de dados
- Recomenda-se utilizar armazenamento SSD para melhor desempenho
- Para ambientes de produção, é recomendado provisionar recursos adicionais para garantir estabilidade

3 Requisitos de Software

Esta seção descreve o software utilizado no desenvolvimento e implantação do sistema.

3.1 Sistema Operacional

O sistema foi desenvolvido e testado para funcionar em ambientes baseados em sistemas operacionais modernos:

- **Servidores de Produção:** Linux (Ubuntu Server 20.04 LTS ou superior)
- **Desenvolvimento:** Linux, macOS ou Windows

3.2 Ambiente de Execução

3.2.1 Backend

Software	Versão	Descrição
Python	3.10 ou superior	Linguagem de programação
pip	21.0 ou superior	Gerenciador de pacotes Python

Tabela 3: Ambiente de execução do backend

3.2.2 Frontend

Software	Versão	Descrição
Node.js	18.0 ou superior	Plataforma JavaScript
npm	9.0 ou superior	Gerenciador de pacotes Node

Tabela 4: Ambiente de execução do frontend

3.3 Frameworks e Bibliotecas Principais

3.3.1 Backend

Framework/Biblioteca	Função
Django 4.2+	Framework web principal
Django REST Framework	Construção da API REST
djangorestframework-simplejwt	Autenticação baseada em tokens JWT
django-cors-headers	Configuração de políticas CORS
drf-nested-routers	Roteamento aninhado para APIs

Tabela 5: Frameworks e bibliotecas do backend

Framework/Biblioteca	Função
React 18.x	Biblioteca para construção da interface
Vite 5.x	Ferramenta de build e servidor de desenvolvimento
TypeScript 5.x	Superset JavaScript com tipagem estática
TanStack Query 5.x	Gerenciamento de estado e cache de dados
Radix UI	Componentes de interface acessíveis
Tailwind CSS 3.x	Framework CSS utilitário
React Router	Gerenciamento de rotas

Tabela 6: Frameworks e bibliotecas do frontend

3.3.2 Frontend

3.4 Banco de Dados

O sistema utiliza SQLite como sistema de gerenciamento de banco de dados:

Software	Descrição
SQLite 3.35+	Sistema de banco de dados relacional leve e embutido

Tabela 7: Sistema de banco de dados

Características do SQLite:

- Não requer servidor de banco de dados separado
- Armazenamento em arquivo único
- Adequado para aplicações de pequeno a médio porte

3.5 Controle de Versão

Software	Descrição
Git	Sistema de controle de versão distribuído

Tabela 8: Controle de versão

4 Requisitos de Serviços

Esta seção descreve os serviços necessários para a operação do sistema.

4.1 Serviços de Rede

4.1.1 Conectividade e Protocolos

O sistema requer conectividade de rede e utiliza os seguintes protocolos:

Protocolo/Serviço	Descrição
HTTP/HTTPS	Protocolo de comunicação para acesso ao sistema
TCP/IP	Protocolo base de comunicação em rede

Tabela 9: Protocolos de rede utilizados

4.1.2 Portas de Comunicação

Serviço	Porta	Uso
Backend (Desenvolvimento)	8000	Servidor de desenvolvimento Django
Frontend (Desenvolvimento)	5173	Servidor de desenvolvimento Vite
HTTP (Produção)	80	Acesso web padrão
HTTPS (Produção)	443	Acesso web seguro

Tabela 10: Portas de comunicação

4.2 Infraestrutura de Hospedagem

4.2.1 Serviços de Servidor

Para implantação do sistema, é necessário um ambiente de hospedagem que forneça:

- **Servidor Web:** Capacidade de executar aplicações Python e servir arquivos estáticos
- **Acesso ao Sistema Operacional:** Permissões para instalação de dependências
- **Armazenamento Persistente:** Para banco de dados e arquivos do sistema
- **Acesso à Internet:** Para comunicação com clientes e atualizações

4.3 Serviços de Desenvolvimento

4.3.1 Controle de Versão

- **Repositório Git:** Sistema de controle de versão para gerenciar o código-fonte
- **Plataforma de Hospedagem:** GitHub, GitLab ou similar para repositório remoto

4.4 Serviços de Segurança

4.4.1 Autenticação e Autorização

O sistema implementa os seguintes serviços de segurança:

- **Autenticação JWT (JSON Web Token):** Tokens de autenticação para acesso seguro à API
- **CORS (Cross-Origin Resource Sharing):** Configuração de políticas de compartilhamento de recursos entre origens, para controlar quais sites podem fazer requisições a API
- **HTTPS:** Criptografia de comunicação (recomendado para produção)

4.5 Serviços de Backup

Para garantir a integridade dos dados, recomenda-se:

- **Backup de Banco de Dados:** Cópias periódicas do arquivo SQLite
- **Backup de Código:** Versionamento através do Git
- **Armazenamento Externo:** Serviço de armazenamento para manter backups

5 Arquitetura de Implantação

5.1 Ambiente de Desenvolvimento

No ambiente de desenvolvimento, o sistema é executado localmente com:

- **Frontend:** Servidor Vite (porta 5173) servindo a aplicação React
- **Backend:** Servidor Django (porta 8000) fornecendo a API REST
- **Banco de Dados:** Arquivo SQLite local (db.sqlite3)

O frontend consome a API do backend via requisições HTTP, seguindo a arquitetura cliente-servidor.

5.2 Ambiente de Produção

Em produção, a arquitetura mantém a separação entre frontend e backend:

- **Frontend:** Arquivos estáticos (HTML/CSS/JavaScript) gerados pelo build do Vite
- **Backend:** Aplicação Django servindo exclusivamente a API REST
- **Banco de Dados:** Arquivo SQLite persistente no servidor

5.3 Fluxo de Dados

O fluxo de comunicação no sistema segue o padrão:

1. Usuário acessa a aplicação através do navegador
2. Frontend é carregado (arquivos estáticos)
3. Usuário realiza ações na interface
4. Frontend envia requisições HTTP para a API backend
5. Backend processa requisições, consulta/atualiza o banco de dados
6. Backend retorna resposta em formato JSON
7. Frontend atualiza a interface com os dados recebidos

6 Conclusão

Este documento apresentou a infraestrutura necessária para implantação do Sistema de Gestão de Artefatos Scrum, descrevendo os requisitos de hardware, software e serviços essenciais para operação do sistema.

A infraestrutura proposta utiliza tecnologias modernas e consolidadas (Python/Django, React/Vite, SQLite) que permitem desenvolvimento ágil e implantação eficiente. Os requisitos de hardware são modestos, tornando o sistema acessível para diferentes contextos de implantação.

O sucesso da implantação depende da correta instalação das dependências, configuração adequada do ambiente e adoção de práticas de segurança e manutenção. A arquitetura cliente-servidor com API REST facilita a manutenção e evolução independente do frontend e backend.