

UNIVERSIDADE DE PERNAMBUCO

ESCOLA POLITÉCNICA DE PERNAMBUCO

## BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Análise e Projeto de Software: Sistema Integrado de Estágio, Monitoria e TCC’s

**Aluno: Paulo Geraldo**

**Prof. Dr. Joabe Bezerra de Jesus Júnior**

**Disciplina: Análise e Projeto de Software**

**Recife**

****Julho/2025****

Sumário

[1. Objetivo 3](#__RefHeading___Toc7806_1374593248)

[2. Escopo 3](#__RefHeading___Toc7808_1374593248)

[3. Topologia Física 3](#__RefHeading___Toc7810_1374593248)

[4. Descrição dos Nós de Implantação 3](#__RefHeading___Toc7812_1374593248)

[4.1. Cliente Web (Navegador) 3](#__RefHeading___Toc7814_1374593248)

[4.2. Servidor de Aplicação 4](#__RefHeading___Toc7816_1374593248)

[4.3. Banco de Dados 4](#__RefHeading___Toc12381_1374593248)

[4.4. Armazenamento de Documentos 4](#__RefHeading___Toc12383_1374593248)

[4.5. Google OAuth 4](#__RefHeading___Toc12385_1374593248)

[4.6. Serviço de Assinatura Digital 5](#__RefHeading___Toc12387_1374593248)

[4.7. Serviço de Notificações 5](#__RefHeading___Toc12389_1374593248)

[4.8. Monitoramento e Logs (Opcional) 5](#__RefHeading___Toc12391_1374593248)

[4.9. Balanceador de Carga (Opcional) 5](#__RefHeading___Toc12393_1374593248)

[5. Relações entre os Nós 5](#__RefHeading___Toc12395_1374593248)

[6. Considerações de Segurança 5](#__RefHeading___Toc12397_1374593248)

[7. Disponibilidade e Escalabilidade 6](#__RefHeading___Toc12399_1374593248)

[8. Diagramas 6](#__RefHeading___Toc12401_1374593248)

[9. Conclusão 8](#__RefHeading___Toc6922_1771842369)

**Índice de figuras**

[Figura 1: Módulo de estágio 6](#Figura!0|sequence)

[Figura 2: Módulo de monitoria 7](#Figura!1|sequence)

[Figura 3: Módulo de TCC 7](#Figura!2|sequence)

[Figura 4: Tela de Login 8](#Figura!3|sequence)

[Figura 5: Tela de Início 8](#Figura!4|sequence)

[Figura 6: Tela do Módulo de Estágio 9](#Figura!5|sequence)

[Figura 7: Tela do Módulo de Monitoria 9](#Figura!6|sequence)

[Figura 8: Tela do Módulo de TCC 10](#Figura!7|sequence)

[Figura 9: Tela do Módulo de Administração 10](#Figura!8|sequence)

**Documento de Análise e Projeto de Software**

**Sistema Integrado de Gestão Acadêmica: Estágio, Monitoria e TCC**

|  |  |
| --- | --- |
| **Instituição:** | UNIVERSIDADE DE PERNAMBUCO - ESCOLA POLITÉCNICA DE PERNAMBUCO |
| **Curso:** | BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO |
| **Projeto:** | Análise e Projeto de Software: Sistema Integrado de estágio, Monitoria e TCC’s – Projeto Arquitetural |
| **Aluno:** | Paulo Geraldo |
| **Disciplina:** | Análise e Projeto de Software |
| **Professor:** | Prof. Dr. Joabe Bezerra de Jesus Júnior |
| **Data:** | Julho/2025 |
| **Versão:** | 1.0 |

### 1. Objetivo

Este documento descreve a visão de implantação do sistema proposto, com foco na topologia física, nos nós de hardware e software, nos relacionamentos entre eles, nas tecnologias empregadas e nas diretrizes arquiteturais. O objetivo é garantir que os requisitos não funcionais (segurança, disponibilidade, desempenho, escalabilidade, entre outros) estejam cobertos pela arquitetura implantada, conforme recomendado pelo RUP.

### 2. Escopo

O sistema tem como escopo principal fornecer uma plataforma integrada que permita a discentes, docentes e coordenadores da universidade gerenciar processos relacionados a Estágio, TCC e Monitoria. A implantação considera o uso de autenticação por Google OAuth, assinatura digital de documentos e comunicação com serviços externos para notificações.

### 3. Topologia Física

A solução é implantada em uma arquitetura distribuída cliente-servidor, composta por:

* Cliente Web (Frontend)
* Servidor de Aplicação (Backend)
* Banco de Dados Relacional
* Serviços Externos (Google OAuth, Serviço de Assinatura Digital, Serviço de Notificações)
* Armazenamento de Documentos (Storage)
* Monitoramento (Opcional)
* Balanceador de Carga (Opcional)

### 4. Descrição dos Nós de Implantação

#### 4.1. Cliente Web (Navegador)

* Papel: Interface de usuário acessada por Alunos, Professores e Coordenadores.
* Tecnologia: HTML5, CSS3, JavaScript (framework: React ou VueJS).
* Acesso: Via HTTPS.
* Requisitos atendidos: Usabilidade, portabilidade.

#### 4.2. Servidor de Aplicação

* Papel: Processamento da lógica de negócio, controle de sessão, autenticação, serviços REST.
* Componentes internos:  
  • API REST  
  • Controle de Sessão  
  • Controle de Acesso (RBAC)  
  • ControleEstagio  
  • ControleTCC  
  • ControleMonitoria  
  • ControleReuniao  
  • ControleDocumentos
* Tecnologia: Node.js, Java (Spring Boot) ou Python (Django/Flask).
* Requisitos atendidos: Escalabilidade, modularidade, rastreabilidade com casos de uso.

#### 4.3. Banco de Dados

* Papel: Persistência das entidades de negócio e dados transacionais.
* Tipo: Relacional (PostgreSQL).
* Componentes armazenados:  
  • Usuários, Papéis, Documentos, Relatórios, Editais, Solicitações, Atividades, etc.
* Requisitos atendidos: Durabilidade, integridade dos dados.

#### 4.4. Armazenamento de Documentos

* Papel: Armazenar arquivos submetidos no sistema (PDFs, relatórios, documentos assinados).
* Tecnologia: Amazon S3, Firebase Storage, File System remoto.
* Requisitos atendidos: Acesso seguro, backup, escalabilidade de armazenamento.

#### 4.5. Google OAuth

* Papel: Serviço de autenticação e autorização baseado em conta institucional (@upe.br).
* Protocolo: OAuth 2.0
* Requisitos atendidos: Segurança, integração com identidade da universidade.

#### 4.6. Serviço de Assinatura Digital

* Papel: Assinar digitalmente documentos e validar assinaturas.
* Integração: API REST de serviços como GOV.BR Assinatura, D4Sign, DocuSign.
* Requisitos atendidos: Autenticidade, não repúdio, conformidade legal.

#### 4.7. Serviço de Notificações

* Papel: Enviar e-mails transacionais e lembretes de ações pendentes.
* Protocolo: SMTP, Webhook.
* Requisitos atendidos: Comunicabilidade, interatividade, engajamento.

#### 4.8. Monitoramento e Logs (Opcional)

* Papel: Observabilidade do sistema, análise de erros e desempenho.
* Ferramentas: Prometheus, Grafana, ELK Stack, Sentry.
* Requisitos atendidos: Diagnóstico, confiabilidade, manutenção.

#### 4.9. Balanceador de Carga (Opcional)

* Papel: Distribuir requisições entre múltiplas instâncias da aplicação.
* Tecnologia: NGINX, HAProxy, AWS Load Balancer.
* Requisitos atendidos: Escalabilidade horizontal, disponibilidade.

### 5. Relações entre os Nós

* Cliente Web → Load Balancer → Servidor de Aplicação (via HTTPS)
* Servidor de Aplicação → Banco de Dados (via JDBC / ORM)
* Servidor de Aplicação → Armazenamento de Documentos (via REST API / SDK)
* Servidor de Aplicação → Google OAuth (via OAuth 2.0)
* Servidor de Aplicação → Serviço de Assinatura Digital (via API)
* Servidor de Aplicação → Serviço de Notificação (via SMTP / Webhook)
* Servidor de Aplicação → Monitoramento (logs e métricas)

### 6. Considerações de Segurança

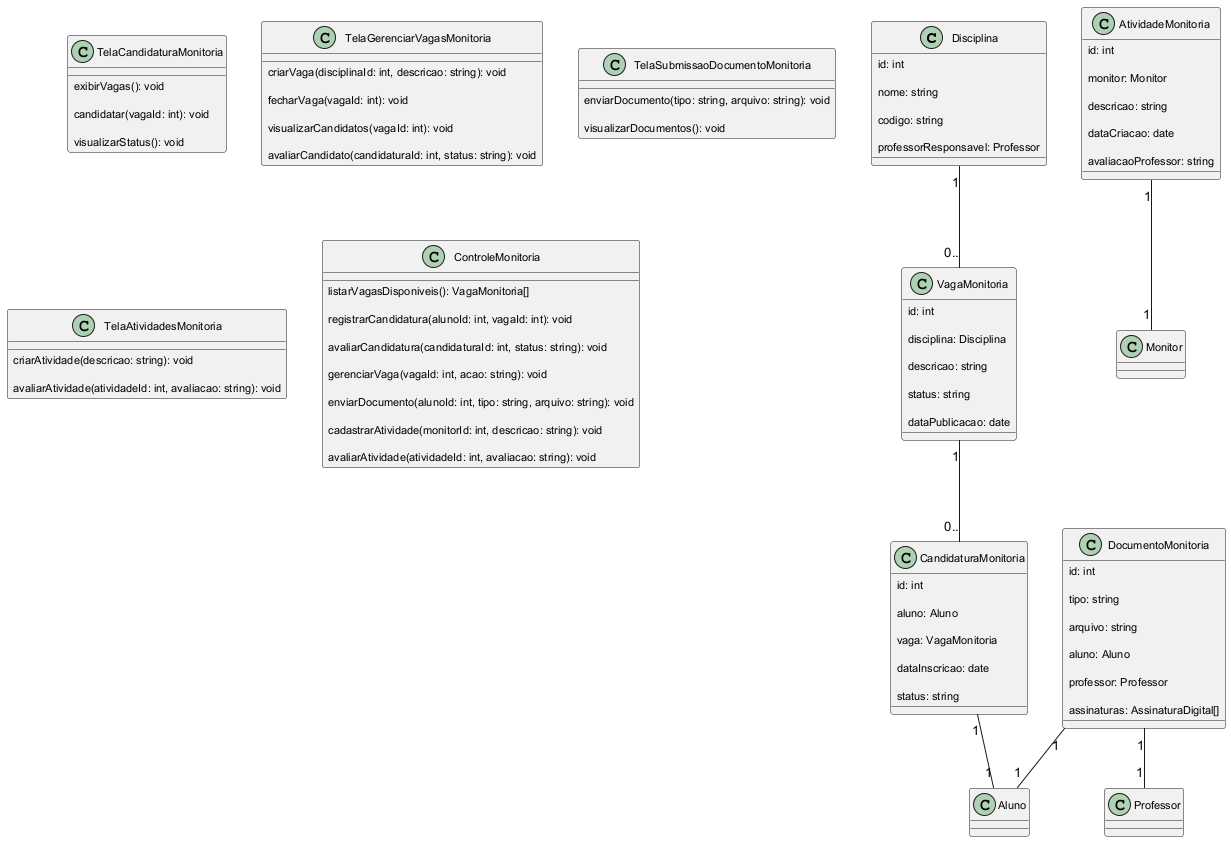
* Todo tráfego entre Cliente e Servidor deve ser criptografado com HTTPS.
* A autenticação é centralizada via OAuth com domínio institucional.
* As assinaturas digitais garantem integridade dos documentos.
* A separação de papéis (RBAC) assegura que apenas usuários autorizados executem determinadas ações.

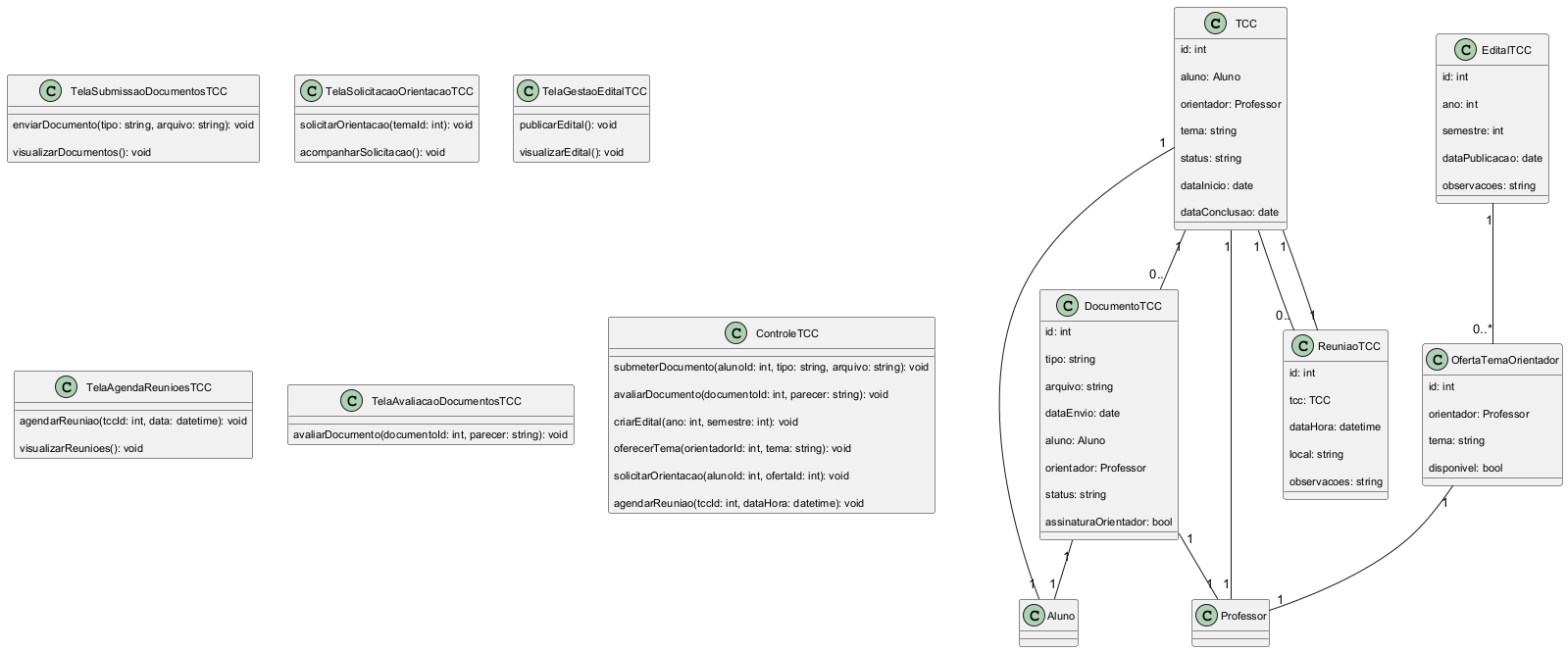
### 7. Disponibilidade e Escalabilidade

* O uso de load balancer permite alta disponibilidade e tolerância a falhas.
* A aplicação pode ser escalada horizontalmente em múltiplas instâncias.
* Armazenamento e banco de dados com provisionamento elástico.

### 8. Diagramas

Figura 1: Módulo de estágio

Figura 2: Módulo de monitoria

Figura 3: Módulo de TCC

### 9. Imagens Protótipo

### 

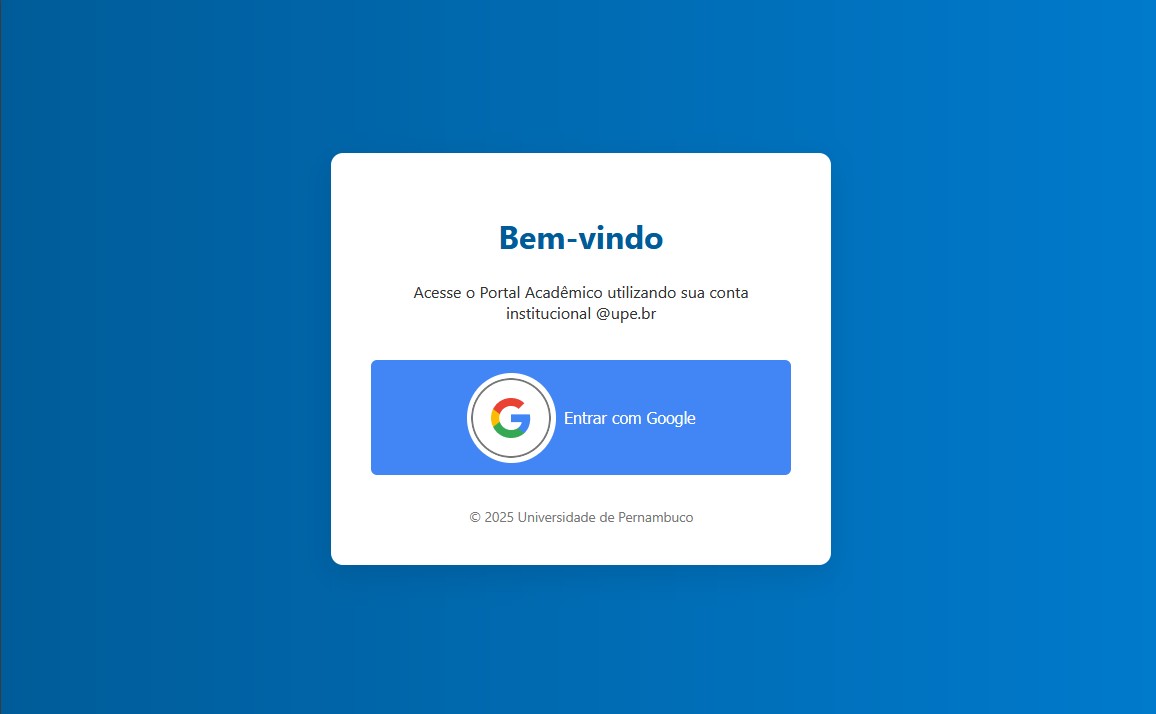
Figura 4: Tela de Login

Figura 5: Tela de Início

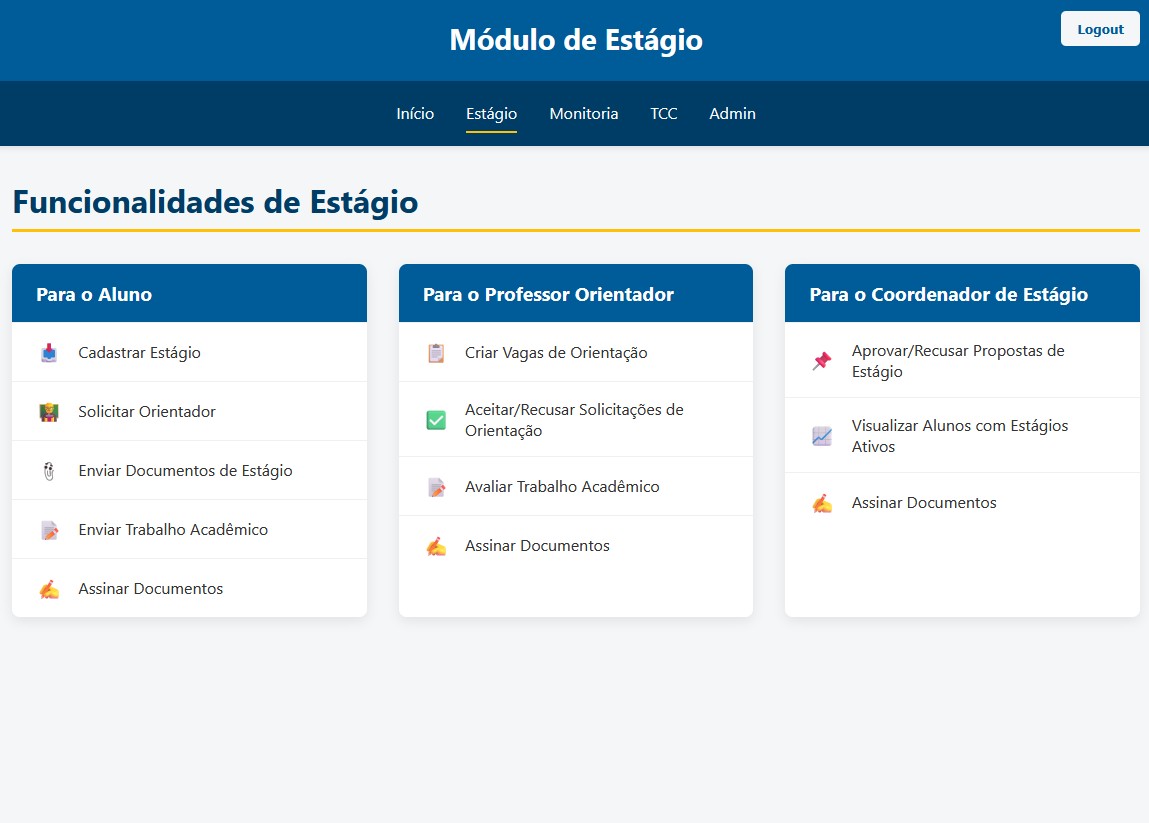
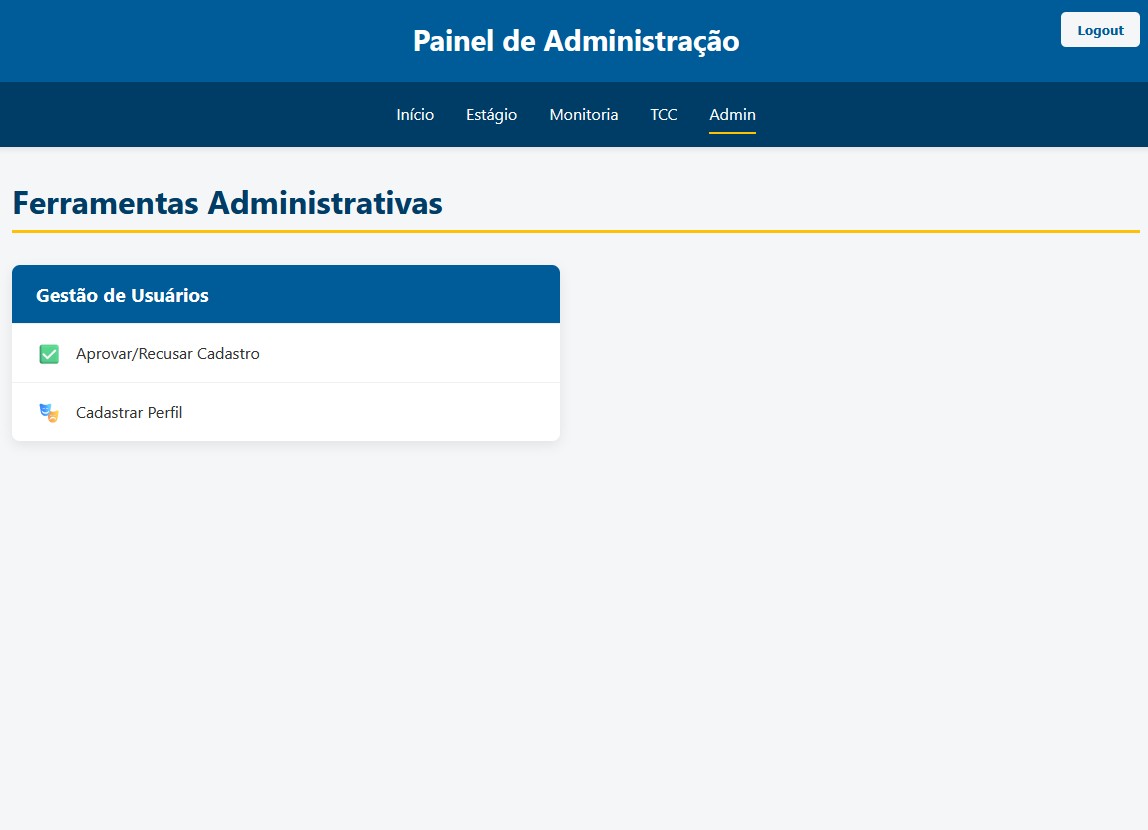
Figura 6: Tela do Módulo de Estágio

Figura 7: Tela do Módulo de Monitoria

Figura 8: Tela do Módulo de TCC

Figura 9: Tela do Módulo de Administração

### 10. Conclusão

Esta Visão de Implantação fornece uma base sólida para a construção e entrega do sistema, suportando todos os casos de uso fundamentais, integrando com segurança os serviços externos e respeitando os requisitos de desempenho e segurança. É um artefato essencial para as próximas fases de construção e testes dentro do processo RUP.