

# Especificação de requisitos não funcionais

Responsável: João Henrique Jácomo Lemes

Última vez atualizado: 31/10/2025

## Requisitos funcionais do sistema não especificados nas histórias de usuário:

- **Auditoria:**

- O Mestre Scrum e o Product Owner devem ser capazes de verificar se os usuários integrantes de um projeto estão acessando o software por meio de quando foi a última vez que um dado usuário interagiu com o projeto.
- O sistema deve registrar o momento e quais alterações foram realizadas pelos usuários.
- O sistema deve registrar todas as ações realizadas por cada usuário, incluindo data, hora e tipo de operação.
- O sistema deve permitir que o Scrum Master e o Product Owner consultem o histórico de interações dos membros do projeto.  
O sistema deve manter histórico de versões dos artefatos, registrando autor, data e conteúdo da modificação.
- O sistema deve permitir restaurar versões anteriores de artefatos a partir do histórico.
- O sistema deve gerar trilhas de auditoria (audit trails) consultáveis por usuários com permissão administrativa.

- **Autenticação:**

- Um usuário deve ter uma conta cadastrada para acessar o sistema.
- O sistema deve exigir autenticação de usuário antes do acesso a qualquer funcionalidade.
- O sistema deve permitir criação de conta contendo nome, ID e e-mail.
- O sistema deve permitir redefinição e recuperação de senha via e-mail.
- O sistema deve encerrar automaticamente a sessão do usuário após período de inatividade.
- O sistema deve oferecer autenticação multifator (MFA) para contas com privilégios administrativos.
- O sistema deve permitir que o administrador bloquee, ative e atribua papéis a usuários.
- O sistema deve permitir associação de usuários a grupos e projetos.

- **Licenciamento:**

- Softwares utilizados que requerem licenças devem ter essas licenças adquiridas antes do lançamento do produto.
- Licenças dos softwares utilizados devem ser instaladas nas máquinas com os respectivos softwares instalados.
- Licenças adquiridas devem ser registradas e monitoradas para quaisquer mudanças de acordo.
- O sistema deve registrar licenças de softwares utilizados e seus respectivos períodos de validade.
- O sistema deve monitorar o status das licenças, emitindo alertas antes da expiração.

- O sistema deve impedir a execução de componentes cuja licença esteja vencida ou não instalada.
- O sistema deve manter registro das licenças adquiridas e dos termos de uso associados.
- O sistema deve garantir o cumprimento dos acordos de software de código aberto integrados ao produto.
- **Impressão:**
  - O sistema deve permitir imprimir relatórios gerados diretamente pela interface do usuário.
  - O sistema deve permitir exportar relatórios para formato PDF, preservando a formatação de impressão.
  - O sistema deve permitir ao usuário selecionar quais informações incluir na impressão (por exemplo, histórico, backlog ou plano de sprint).
  - O sistema deve ajustar automaticamente a formatação de impressão para diferentes tamanhos de papel (A4, Carta, etc.).
- **Relatórios:**
  - O sistema deve permitir gerar relatórios de progresso dos projetos, incluindo número de tarefas concluídas e pendentes.
  - O sistema deve permitir gerar relatórios de alterações realizadas por período, usuário ou artefato.
  - O sistema deve permitir exportar relatórios em formato PDF e CSV.
  - O sistema deve apresentar indicadores de produtividade (histórias concluídas por sprint, média de tarefas por desenvolvedor, etc.).
  - O sistema deve permitir que o Product Owner defina filtros e intervalos de tempo para geração de relatórios.
- **Agendamento:**
  - O sistema deve permitir agendar reuniões, revisões e retrospectivas de sprint.
  - O sistema deve permitir definir datas de início e término de sprints.
  - O sistema deve notificar os participantes de eventos com antecedência de 5 dias e diariamente a partir de 3 dias antes.
  - O sistema deve enviar notificações automáticas sobre início e término de sprints.
  - O sistema deve alertar o Product Owner sobre tarefas atrasadas ou bloqueadas.
  - O sistema deve permitir integração com calendários externos (por exemplo, Google Calendar).
- **Segurança:**
  - O sistema deve impedir acesso a informações ou funções restritas sem as devidas permissões.
  - O sistema deve criptografar senhas e dados sensíveis armazenados.
  - O sistema deve registrar todas as tentativas de login, bem-sucedidas ou não.
  - O sistema deve realizar backups automáticos em intervalos configuráveis.
  - O sistema deve restaurar o backup mais recente em caso de falha.
  - O sistema deve validar os dados inseridos pelos usuários antes de armazená-los.
  - O sistema deve proteger comunicações entre cliente e servidor usando HTTPS/TLS.

# Qualidades do Sistema

## Usabilidade

- **Facilidade de aprendizagem:**

- Um usuário que não é familiar com softwares e começou a usar o sistema deve ser capaz de entender suas principais funcionalidades em menos de 45 minutos.
- Um usuário que é familiar com softwares e começou a usar o sistema deve ser capaz de entender suas principais funcionalidades em menos de 20 minutos.
- O sistema deve prover um manual simples e claro para usuários com dúvidas.

- **Eficiência de tarefa:**

- Um usuário deve ser capaz de cadastrar sua conta em menos de 5 minutos.
- Um usuário deve ser capaz de entrar em sua conta já cadastrada em menos de 1 minuto.
- Um usuário deve ser capaz de criar projetos em menos de 10 minutos.
- Um usuário deve ser capaz de adicionar participantes a um projeto em cerca de 5 cliques.
- Um usuário deve ser capaz de acessar diferentes projetos em cerca de 3 cliques.
- Um usuário deve ser capaz de realizar mudanças em projetos em cerca de 6 cliques.

- **Facilidade de lembrar:**

- O usuário deve ser capaz de lembrar como realizar o login após mais de 2 meses sem usar o sistema.
- O usuário deve ser capaz de lembrar como acessar seus projetos após mais de 2 meses sem usar o sistema.
- O usuário deve ser capaz de lembrar como criar projetos após mais de 2 meses sem usar o sistema.
- O usuário deve ser capaz de lembrar como realizar mudanças em projetos após mais de 2 meses sem usar o sistema.

- **Compreensibilidade:**

- Um usuário deve ser capaz de compreender o que os botões do sistema fazem apenas com base em seus ícones e nomes.
- Um usuário deve ser capaz de compreender que mudanças está realizando em um projeto apenas com base no que está escrito e destacado.

- **Satisfação subjetiva:**

- 50% dos usuários devem estar muito satisfeitos com o sistema.
- 25% dos usuários devem estar satisfeitos com o sistema.

## Confiabilidade

- **Precisão:**
  - O sistema deve registrar todas as mudanças realizadas em um projeto.
  - O sistema deve apresentar todas as mudanças realizadas em um projeto.
  - O sistema deve registrar todos os agendamentos realizados.
- **Disponibilidade:**
  - O sistema deve estar disponível 90% do tempo.
  - O sistema só deve estar indisponível quando ocorrer falhas no servidor ou durante manutenções, que devem corresponder a 10% do tempo.
  - O sistema deve ter um tempo médio entre falhas de 30 dias.
- **Recuperabilidade:**
  - O sistema deve ter um tempo médio para reparar de 1 hora.
  - O sistema deve recuperar o backup mais recente na ocorrência de falhas.
- **Frequência e severidade das falhas:**
  - Uma falha menor é uma inconveniência para o usuário.
  - Uma falha significante é um problema nas funcionalidades do produto.
  - Uma falha crítica é uma falha que causa perda de dados ou inutilização de uma funcionalidade do produto.
  - Um tempo de resposta acima de 10 segundos do normal é uma falha menor.
  - Uma mudança em um projeto não ser registrada é uma falha significante.
  - Um projeto perder o registro de mudanças é uma falha crítica.
  - Um usuário perder sua conta é uma falha crítica.
  - O produto deve apresentar no máximo 3 falhas por funcionalidade.

## Performance

- **Tempos de resposta:**
  - O tempo de resposta de qualquer interface com o usuário deve ser de no máximo 4 segundos.
  - O produto deve mostrar mudanças realizadas em um projeto no log de mudanças dentro de 8 segundos.
- **Vazão:**
  - O produto deve suportar 10 alterações por segundo dentro de um projeto.
- **Capacidade:**
  - O produto deve suportar 300 usuários durante qualquer momento do dia.
- **Inicialização:**
  - O produto deve demorar no máximo 30 segundos para inicializar.
- **Encerramento:**
  - O produto deve demorar no máximo 30 segundos para desligar.

## Suportabilidade

- **Adaptabilidade:**

- O sistema deve ser fácil de instalar em qualquer máquina, independente do sistema operacional.
- O sistema deve ser fácil de acessar com uma conta já criada em qualquer máquina.
- **Compatibilidade:**
  - O sistema deve ser compatível entre suas versões, sem perda de contas cadastradas ou projetos já criados e suas mudanças.
- **Configurabilidade:**
  - O sistema deve permitir configurar os canais de notificação (e-mail, SMS, push) e a frequência de envio de notificações.
  - Os usuários devem poder ativar ou desativar tipos específicos de alertas sem alterar o código.
- **Instalação:**
  - O sistema deve fornecer um script de instalação automatizado que configure banco de dados, variáveis de ambiente e permissões.
  - A instalação não deve exigir acesso root permanente após a configuração inicial.
- **Nível de suporte:**
  - Deve ser disponibilizado suporte técnico ao usuário final durante o horário comercial, via e-mail e chat online.
  - O suporte deve responder às solicitações de usuários em até 24 horas úteis.
  - O suporte deve incluir atendimento de primeiro e segundo nível:
    - Nível 1: dúvidas de uso, redefinição de senha, problemas simples.
    - Nível 2: falhas técnicas, bugs, integração com outros sistemas.
  - O sistema deve oferecer um módulo de ajuda embutido, acessível a partir de qualquer tela.
  - O sistema deve conter tutoriais interativos ou documentação integrada para orientar novos usuários.
  - O sistema deve permitir que usuários reportem erros diretamente pela interface, enviando logs e descrições automáticas do problema.
  - O sistema deve exibir mensagens de erro descritivas, que orientem o usuário quanto à causa e solução possível.
- **Manutenabilidade:**
  - O sistema deve ser capaz de ser mantido por usuários finais ou desenvolvedores que não são parte do time de desenvolvimento.
  - Os lançamentos de manutenção ocorrerão uma vez por ano.
- **Escalabilidade:**
  - O sistema deve suportar até 10.000 usuários cadastrados.
  - O sistema deve permitir até 300 usuários simultaneamente conectados sem degradação perceptível de desempenho.
  - O sistema deve ser projetado para suportar crescimento de 50% no número de usuários por ano, mantendo o tempo médio de resposta abaixo de 4 segundos por requisição.
- **Testabilidade:**
  - O sistema deve permitir a execução automatizada de testes unitários, de integração e de regressão.
  - O sistema deve expor APIs e interfaces estáveis para permitir o teste automatizado de funcionalidades principais.

- O sistema deve ser compatível com ferramentas de automação de teste, como JUnit, Pytest, Selenium, ou Postman.
- O sistema deve atingir no mínimo 80% de cobertura de código em testes unitários.
- O sistema deve passar em todos os testes de integração críticos antes de cada release.
- O tempo total de execução do conjunto de testes automatizados não deve ultrapassar 30 minutos.

## Interfaces do Sistema

### Interfaces de usuário

- **Look & Feel:**
  - A interface do produto deve ter a opção entre um modo claro e um modo escuro.
  - A interface do produto deve ter uma interface simples e objetiva.
  - A interface deve ter um estilo simples, com as partes de interação como botões e caixas de texto claramente identificadas e destacadas.
  - A interface deve permitir interação apenas onde há indicação de tal, como os botões e caixas de texto.
- **Layout e requisitos de navegação:**
  - O produto deve ter uma interface inicial de login que possui os campos “nome de usuário/e-mail” e “senha”.
  - O produto deve ter uma tela na qual o usuário pode selecionar entre os projetos em que participa.
  - O produto deve ter uma tela que apresenta o trabalho, as histórias de usuário, backlog de produto, backlog de sprint e plano de sprint.
  - O produto deve ter botões para a criação, leitura, atualização e exclusão das histórias de usuário, backlog de produto, backlog de sprint e plano de sprint.
  - O produto deve ter uma tela de plano de sprint que apresenta o objetivo, o trabalho, a equipe, as tarefas e a saída.
- **Consistência:**
  - Botões para a criação de artefatos terão o símbolo “+”.
  - Botões para a leitura de artefatos terão o símbolo “...”.
  - Botões para a edição de artefatos terão o símbolo de um lápis.
  - Botões para a exclusão de artefatos terão o símbolo “X”.
  - Uma tela de confirmação aparecerá para a edição e a exclusão.
  - Os módulos das interfaces terão o formato de um quadrilátero.
- **Personalização de usuário e requisitos de customização:**
  - Apenas o usuário com o papel “Product Owner” pode ter acesso para editar o backlog de produto.
  - Apenas o usuário com o papel “Scrum Master” pode ter acesso para editar o backlog de sprint e o plano de sprint.

- O usuário deve ser capaz de selecionar quais eventos agendados devem ser notificados para ele.

## Interfaces para Sistemas Externos ou Dispositivos

- **Interfaces de software:**

- O sistema deve integrar-se a um serviço de autenticação seguro (ex.: módulo interno de login ou provedor externo via OAuth 2.0) para validar credenciais de usuário.
- O protocolo de comunicação entre o módulo de autenticação e o sistema deve ser HTTPS, garantindo a confidencialidade das informações transmitidas.
- O sistema deve se comunicar com um sistema gerenciador de banco de dados relacional (por exemplo, PostgreSQL ou MySQL) por meio de uma camada de persistência que use SQL padrão.
- A interface deve permitir operações de criação, leitura, atualização e exclusão (CRUD) dos artefatos: história de usuário, backlog de produto, backlog de sprint e plano de sprint.
- Caso a interface gráfica seja web-based, o sistema deve interagir com navegadores modernos compatíveis com HTML5, CSS3 e JavaScript.
- A comunicação entre o frontend e o backend deve ocorrer por meio de APIs RESTful com respostas em formato JSON.
- O sistema deve disponibilizar endpoints para consulta e atualização de dados de projetos, permitindo integração com ferramentas externas de gestão (por exemplo, Jira, Trello ou sistemas internos).

- **Interfaces de hardware:**

- O sistema deve ser executável em estações de trabalho que atendam aos seguintes requisitos mínimos:
  - Processador com suporte a instruções de 64 bits;
  - 8 GB de memória RAM;
  - Resolução mínima de tela de 1366×768 (para interfaces gráficas);
  - Acesso à rede local ou Internet.
- O sistema deve suportar dispositivos de entrada padrão, como teclado e mouse.
- O software deve poder ser implantado em servidores físicos ou virtuais que utilizem arquitetura x86\_64, com sistema operacional Linux ou Windows Server.
- Deve haver compatibilidade com ambiente de hospedagem em nuvem (ex.: AWS, Azure, GCP).
- O sistema deve operar corretamente em redes TCP/IP com suporte a IPv4 e IPv6.

- O tempo máximo de resposta a requisições de autenticação não deve exceder 2 segundos em condições normais de rede (<100 ms de latência).
- **Interfaces de comunicação:**
  - O sistema deve operar em rede local (LAN) com largura de banda mínima de 100 Mbps.
  - O sistema deve suportar comunicação via internet utilizando protocolo TCP/IP sobre HTTPS.
  - O sistema deve manter conexão persistente com o servidor principal, reconectando-se automaticamente em caso de falha.
  - O sistema deve ser compatível com redes IPv4 e IPv6.
  - Em caso de falha de conexão, os dados devem ser armazenados localmente e retransmitidos automaticamente quando a conexão for restabelecida.
  - O sistema deve suportar até 500 conexões simultâneas de dispositivos remotos.
  - O tempo máximo para restabelecimento de conexão após falha deve ser inferior a 10 segundos.

## Regras de Negócio

### Classe de Regra: Contas de Usuário e Autenticação

Regras relacionadas à criação de contas, autenticação e controle de acesso de usuários no sistema.

#### ID: UA-01

**Nome da Regra:** Criação de Conta de Usuário

**Descrição:**

Se um usuário fornecer um endereço de e-mail válido e uma senha que atenda aos critérios de segurança (mínimo de 8 caracteres, contendo pelo menos uma letra maiúscula, um número e um símbolo especial), então o sistema deve permitir a criação de uma nova conta de usuário.

#### ID: UA-02

**Nome da Regra:** Exclusividade de E-mail

**Descrição:**

Se um usuário tentar registrar-se com um e-mail que já esteja associado a uma conta existente, então o sistema deve rejeitar o registro e notificar o usuário de que o e-mail já está em uso.

#### ID: UA-03

**Nome da Regra:** Autenticação de Usuário

**Descrição:**

Se um usuário fornecer credenciais válidas (e-mail e senha) que correspondam a uma conta

registrada, então o sistema deve autenticar o usuário e conceder acesso aos serviços de acordo com seu(s) papel(is).

#### **ID: UA-04**

**Nome da Regra:** Controle de Acesso Baseado em Papéis

**Descrição:**

Se um usuário autenticado tentar acessar um serviço ou artefato, então o sistema deve verificar o papel do usuário no projeto atual e permitir apenas ações autorizadas para aquele papel.

#### **Classe de Regra: Gestão de Projetos e Papéis**

Regras que definem como usuários se associam a projetos e como papéis do Scrum são atribuídos.

#### **ID: PR-01**

**Nome da Regra:** Associação a Projetos

**Descrição:**

Se um usuário for convidado para participar de um projeto, então ele deve aceitar o convite antes de ter acesso aos artefatos e serviços desse projeto.

#### **ID: PR-02**

**Nome da Regra:** Atribuição de Papéis

**Descrição:**

Se um usuário for designado a um papel em um projeto, então o sistema deve garantir que cada projeto possui exatamente um Product Owner e um Scrum Master, mas pode conter múltiplos desenvolvedores.

#### **ID: PR-03**

**Nome da Regra:** Conflito de Papéis

**Descrição:**

Se um usuário já ocupar o papel de Product Owner ou Scrum Master em um projeto, então o sistema deve impedir que ele assuma ambos os papéis simultaneamente no mesmo projeto.

#### **Classe de Regra: Gestão de Backlogs de Produto e de Sprint**

Regras que tratam da criação, modificação e consistência dos artefatos Scrum.

#### **ID: BL-01**

**Nome da Regra:** Criação de História de Usuário

**Descrição:**

Se um usuário com o papel de Product Owner criar uma nova história de usuário, então a história deve conter, no mínimo, título, descrição, critérios de aceitação e nível de prioridade antes de ser salva.

## **ID: BL-02**

**Nome da Regra:** Consistência do Backlog de Sprint

**Descrição:**

Se uma história de usuário for incluída em um Backlog de Sprint, então ela deve pertencer ao mesmo projeto e possuir o status “Pronta” no Backlog do Produto.

## **ID: BL-03**

**Nome da Regra:** Permissão de Edição de Artefato

**Descrição:**

Se um usuário tentar excluir ou modificar um item de backlog, então o sistema deve verificar se o usuário possui permissão de edição para aquele item, conforme o papel que desempenha no projeto.

## **ID: BL-04**

**Nome da Regra:** Completude do Plano de Sprint

**Descrição:**

Se um Plano de Sprint for marcado como “Ativo”, então ele deve conter um Objetivo de Sprint definido, membros da equipe atribuídos, tarefas associadas e o incremento esperado.

# **Classe de Regra: Execução e Acompanhamento de Sprints**

Regras relacionadas à execução, monitoramento e encerramento de sprints.

## **ID: SP-01**

**Nome da Regra:** Início de Sprint

**Descrição:**

Se o Backlog de Sprint tiver sido aprovado pelo Product Owner e o Plano de Sprint estiver completo, então o Scrum Master pode iniciar o Sprint.

## **ID: SP-02**

**Nome da Regra:** Atribuição de Tarefas

**Descrição:**

Se uma tarefa for criada dentro de um Sprint, então ela deve ser atribuída a pelo menos um Desenvolvedor membro da equipe do Sprint.

## **ID: SP-03**

**Nome da Regra:** Encerramento de Sprint

**Descrição:**

Se todas as tarefas de um Sprint estiverem marcadas como “Concluídas” ou o tempo do Sprint tiver expirado, então o Sprint deve ser encerrado automaticamente e o sistema deve gerar um resumo de Revisão do Sprint.

## **Classe de Regra: Segurança e Auditoria**

Regras que garantem rastreabilidade, segurança e controle de acesso.

### **ID: SA-01**

**Nome da Regra:** Registro de Auditoria

**Descrição:**

Se qualquer modificação (criação, atualização ou exclusão) ocorrer em um artefato do projeto, então o sistema deve registrar o usuário, o horário e a ação executada para fins de auditoria.

### **ID: SA-02**

**Nome da Regra:** Expiração de Sessão

**Descrição:**

Se um usuário permanecer inativo por mais de 30 minutos, então o sistema deve encerrar automaticamente a sessão e exigir nova autenticação.

### **ID: SA-03**

**Nome da Regra:** Expiração de Senha

**Descrição:**

Se a senha de um usuário não for alterada em um período de 180 dias, então o sistema deve solicitar a atualização da senha no próximo login.

## **Classe de Regra: Comportamento Geral do Sistema**

Regras que definem o comportamento operacional do sistema em termos de disponibilidade e tratamento de erros.

### **ID: SY-01**

**Nome da Regra:** Disponibilidade do Sistema

**Descrição:**

Se o sistema estiver em operação normal, então ele deve garantir 99% de disponibilidade para acesso aos serviços de gestão de projetos durante o horário comercial.

### **ID: SY-02**

**Nome da Regra:** Tratamento de Erros

**Descrição:**

Se ocorrer um erro inesperado no sistema, então ele deve registrar o evento e apresentar ao usuário uma mensagem compreensível, sem revelar detalhes técnicos.

## Restrições do Sistema

### 1. Plataforma e Ambiente de Execução

- O sistema deve ser desenvolvido para execução em ambiente **multiplataforma**, com suporte mínimo a **Windows, Linux e macOS**.
- O sistema deve ser implementado para execução em **servidores compatíveis com arquitetura x86-64**.
- O sistema deve ser compatível com **bancos de dados relacionais** baseados em **SQL** (por exemplo, **PostgreSQL ou MySQL**).

### 2. Linguagens e Tecnologias

- O sistema deve ser desenvolvido utilizando **linguagens de programação open source** (como **Python, JavaScript/TypeScript ou Java**).
- O uso de bibliotecas de terceiros deve ser documentado e respeitar as condições de suas licenças.

### 3. Segurança e Privacidade

- Todos os dados trocados entre cliente e servidor devem utilizar **criptografia TLS 1.3** via **HTTPS**.
- O sistema deve estar em conformidade com a **Lei Geral de Proteção de Dados (LGPD)**, garantindo o consentimento e a proteção das informações pessoais.
- Backups devem ser armazenados de forma criptografada e acessíveis apenas por administradores autorizados.

### 4. Ferramentas de Desenvolvimento e Controle de Versão

- O desenvolvimento deve ser gerenciado por meio de **sistema de controle de versão Git** (por exemplo, GitHub ou GitLab).

### 5. Recursos e Limitações de Hardware

- O sistema deve funcionar em máquinas cliente com, no mínimo:
  - Processador dual-core de 2 GHz ou superior;
  - 4 GB de memória RAM;
  - 500 MB de espaço livre em disco;
  - Conexão de rede estável (mínimo de 5 Mbps).

### 6. Restrições de Interface

- O sistema deve seguir o **padrão de design responsivo**, ajustando-se automaticamente a resoluções entre **1280x720 e 1920x1080**.
- A interface deve utilizar o **português do Brasil** como idioma padrão.

## Conformidade do Sistema

- **Requisitos de licenciamento:**
  - LC-01 — Verificação de Licenças de Software de Terceiros  
O sistema deve verificar, antes da implantação, se todas as bibliotecas, frameworks e ferramentas de terceiros utilizadas possuem licenças válidas e compatíveis com o modelo de distribuição do produto (por exemplo, MIT, Apache 2.0, GPLv3).
  - LC-02 — Registro de Licenças Adquiridas  
O sistema deve manter um registro atualizado de todas as licenças de softwares de terceiros utilizados, incluindo nome do produto, tipo de licença, data de aquisição e validade.
  - LC-03 — Restrição de Uso sem Licença Válida  
Se o sistema detectar que uma licença obrigatória está ausente, expirada ou inválida, então o sistema não deve permitir a inicialização ou deve operar em modo de funcionalidade reduzida até a regularização.
  - LC-04 — Monitoramento e Auditoria de Licenças  
O sistema deve gerar relatórios mensais automáticos sobre o estado das licenças, destacando aquelas próximas do vencimento (menos de 30 dias).
  - LC-05 — Conformidade com Termos de Licenciamento  
O sistema e seus desenvolvedores devem garantir conformidade com os termos de uso, redistribuição e modificação definidos pelas licenças de software de terceiros empregados.
  - LC-06 — Licenciamento de Usuários Finais  
Cada instância do sistema implantada deve estar associada a uma licença de uso registrada e vinculada a um identificador único da organização usuária (por exemplo, ID de cliente ou domínio de empresa).
  - LC-07 — Restrição de Número de Usuários por Licença  
Se uma licença limitar o número máximo de usuários simultâneos, então o sistema deve monitorar a quantidade de conexões ativas e impedir novos logins quando o limite for atingido.
  - LC-08 — Renovação de Licenças  
O sistema deve notificar o administrador responsável com 15 dias de antecedência sobre a necessidade de renovação de licenças de uso do produto ou de softwares de terceiros.
  - LC-09 — Auditoria de Conformidade Legal  
O sistema deve permitir exportar relatórios de conformidade de licenciamento para fins de auditoria interna ou externa, incluindo informações sobre versões de software, licenças e períodos de validade.
  - LC-10 — Licença para Atualizações  
A instalação de atualizações ou novas versões do sistema deve ser restrita a usuários ou organizações que possuam uma licença de uso válida e ativa no momento da atualização
- **Legalidade, direitos autorais, e outros avisos:**

- **LG-01 — Direitos Autorais (Copyright Notice)**

Todo o código-fonte, documentação, interfaces e demais artefatos produzidos no desenvolvimento deste sistema são protegidos por direitos autorais e pertencem à organização proprietária do produto.

Nenhuma parte do software pode ser copiada, modificada, distribuída ou reproduzida sem autorização expressa por escrito do detentor dos direitos autorais.

- **LG-02 — Licenciamento de Uso (End-User License Agreement - EULA)**

O uso do sistema é regido por um contrato de licença de uso (EULA) aceito pelo usuário durante a instalação ou primeiro acesso.

A violação dos termos de uso descritos no EULA resultará na suspensão ou revogação imediata da licença.

- **LG-03 — Isenção de Garantia (Warranty Disclaimer)**

O sistema é fornecido “no estado em que se encontra” (*as is*), sem qualquer garantia explícita ou implícita de desempenho, disponibilidade, adequação a um propósito específico ou ausência de falhas.

Os desenvolvedores e distribuidores não serão responsáveis por danos diretos, indiretos ou consequenciais resultantes do uso ou incapacidade de uso do sistema.

- **LG-04 — Marcas Registradas (Trademark Notice)**

Todos os nomes de produtos, logotipos e marcas citados no sistema são de propriedade de seus respectivos titulares.

O uso de marcas registradas de terceiros é apenas para fins de identificação e não implica endosso ou afiliação.

- **LG-05 — Notas de Patente (Patent Notice)**

Caso o sistema implemente métodos, algoritmos ou tecnologias patenteadas, os respectivos números de patente e titulares devem ser informados na documentação técnica.

O uso comercial de componentes patenteados requer a devida licença dos titulares dos direitos.

- **LG-06 — Política de Privacidade e Proteção de Dados (Data Protection and Privacy Policy)**

O sistema deve estar em conformidade com as legislações aplicáveis de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD — Lei nº 13.709/2018) e o Regulamento Geral de Proteção de Dados da União Europeia (GDPR).

O sistema deve garantir que dados pessoais dos usuários sejam armazenados e processados apenas para fins legítimos e com o consentimento explícito do titular.

- **LG-07 — Responsabilidade de Uso (Liability Limitation)**

O usuário é responsável por assegurar o uso adequado do sistema em conformidade com leis locais, políticas organizacionais e restrições contratuais.

A organização proprietária do software não se responsabiliza por uso indevido, adulteração ou violação de direitos de propriedade intelectual decorrentes de modificações não autorizadas.

- **LG-08 — Logotipos e Identidade Visual (Logo and Wordmark Compliance)**

O uso do logotipo, nome comercial e identidade visual do sistema deve seguir o manual de marca oficial.

É proibido alterar cores, proporções, tipografia ou elementos visuais do logotipo em qualquer material digital ou impresso sem autorização do detentor dos direitos.

- **LG-9 — Jurisdição e Foro (Governing Law)**

Qualquer disputa relacionada ao uso, distribuição ou modificação do sistema será regida pelas leis do país de origem do desenvolvimento e submetida ao foro da comarca da sede da organização detentora dos direitos.

- **Padrões Aplicáveis:**

- **ST-01 — Padrões de Qualidade de Software (Software Quality Standards)**

O sistema deve seguir as diretrizes e boas práticas definidas pela norma ISO/IEC 25010:2011 — Systems and Software Quality Models, que estabelece as características de qualidade de software, incluindo:

- Funcionalidade (adequação e correção funcional),
- Usabilidade,
- Confiabilidade,
- Eficiência de desempenho,
- Manutenibilidade,
- Portabilidade e
- Segurança.

- **ST-02 — Padrões de Engenharia de Software (Software Engineering Process Standards)**

O desenvolvimento, documentação e manutenção do sistema devem estar em conformidade com os processos descritos em ISO/IEC/IEEE 12207:2017 — Systems and Software Engineering — Software Life Cycle Processes, abrangendo atividades de:

- Planejamento e gestão do ciclo de vida,
- Implementação, verificação e validação,
- Manutenção e suporte.

- **ST-03 — Padrões de Segurança da Informação (Information Security Standards)**

O sistema deve estar em conformidade com a ISO/IEC 27001:2022 — Information Security Management Systems (ISMS), assegurando:

- Controle de acesso baseado em papéis,
- Registro de auditoria,
- Proteção de dados sensíveis, e
- Mecanismos de recuperação de falhas.

- **ST-04 — Padrões de Proteção de Dados e Privacidade (Data Protection Standards)**

O sistema deve obedecer às normas e legislações aplicáveis à privacidade e proteção de dados, incluindo:

- Lei Geral de Proteção de Dados Pessoais (LGPD — Lei nº 13.709/2018, Brasil),
- Regulamento Geral de Proteção de Dados (GDPR — EU 2016/679),

- Princípios de minimização e consentimento explícito no tratamento de informações pessoais.

## Documentação do Sistema

- **SD-01 — Notificações “Help About” (Ajuda → Sobre)**

A seção “Ajuda → Sobre” deve conter informações sobre:

- Nome e versão do produto (de acordo com o controle de versionamento SemVer);
- Nome do fornecedor/desenvolvedor do sistema;
- Informações de licenciamento;
- Créditos e avisos legais (copyright, trademarks e patentes, conforme seção *Legal, Copyright, and Other Notices*).

- **SD-02 — Documentação Técnica (Technical Documentation)**

A equipe de desenvolvimento deve produzir e manter atualizada a documentação técnica do sistema, contendo:

- Arquitetura do sistema e principais componentes;
- Modelos de dados e diagramas de entidade-relacionamento;
- Especificação de interfaces e APIs (RESTful endpoints, formatos JSON, autenticação, etc.);
- Guia de instalação e configuração do ambiente;
- Instruções para manutenção, atualização e backup do sistema.

- **SD-03 — Requisitos de Qualidade da Documentação**

A documentação deve seguir os seguintes requisitos de qualidade:

- Clareza, concisão e consistência terminológica;
- Linguagem acessível, com exemplos e capturas de tela;
- Conformidade com a norma ISO/IEC/IEEE 26514:2022 — Systems and Software Engineering — Design and Preparation of User Documentation;
- Atualização obrigatória a cada nova versão liberada do sistema.