Universidade de Brasília
INSTITUTO DE EXATAS - IE
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO - CIC
ENGENHARIA DE SOFTWARE 1/2025

ESPECIFICAÇÃO DE REQUISITOS NÃO FUNCIONAIS

Discentes: BRUNO EDUARDO DOS SANTOS - 211066249
GEILSON DOS SANTOS SA - 231006239
MARCOS ALEXANDRE DA SILVA NERES - 211055334
PEDRO FARIAS DE OLIVEIRA - 211055577
THELMA EVANGELISTA DOS SANTOS - 231003513
WESLEY HENRIQUE FERREIRA - 231021496

Sistema de Gestão de Feiras	
Requisitos não funcionais	Data: 23/05/2025

1. Introduction

Este documento especifica os requisitos não funcionais e de suporte do sistema de software que viabiliza a criação, leitura, atualização e exclusão de registros relacionados a feiras, expositores, produtos e ingressos. Além disso, contempla os serviços disponíveis tanto para usuários autenticados quanto para visitantes.

2. System-Wide Functional Requirements

- O sistema deve validar permissões de cada usuário para ações como criação, edição e exclusão de registros, conforme seu perfil (administrador, expositor, visitante).
- O sistema deve restringir a exclusão de registros vinculados a outras entidades, garantindo a integridade dos dados.
- O sistema deve permitir a emissão e leitura de ingressos válidos por data e evento, impedindo reutilizações fora do prazo.
- O sistema deve validar se o usuário que realiza determinada operação é o criador do recurso, quando aplicável.
- O sistema deve registrar ações administrativas relevantes (criação, edição, exclusão de feiras, usuários e expositores), incluindo data, hora e autor da ação.
- O sistema deve permitir a geração de relatórios em PDF ou CSV contendo informações consolidadas de feiras, expositores e visitas.
- O sistema deve notificar usuários por e-mail em eventos importantes, como aprovações de inscrição e alterações de eventos.
- O sistema deve aplicar validações sistêmicas para impedir alterações que comprometam a consistência entre módulos e dados relacionados.

3. System Qualities

3.1 Usability

- A interface do sistema deverá oferecer experiência adaptada para os diferentes perfis de usuário (administrador, expositor e visitante), priorizando a simplicidade para visitantes e

ferramentas completas para administradores e expositores.
- Padrões consistentes de nomenclatura, layout e interações em todas as páginas.

3.2 Reliability
- Sistema deve permitir recuperação de falhas com perda mínima de dados (máximo de 5 minutos).
- Mecanismos de backup automáticos diários.
- O sistema deve implementar autenticação segura com hash de senhas (ex: bcrypt).
- As sessões devem expirar automaticamente após 30 minutos de inatividade.
- Tentativas múltiplas de login inválido devem ativar um mecanismo de bloqueio temporário.

3.3 Performance
- Tempo máximo de resposta: 2 segundos para listagens e consultas.
- Suporte mínimo para 200 usuários simultâneos.
- Tempo de inicialização do sistema: até 10 segundos.
- O sistema deve tratar corretamente acessos simultâneos a um mesmo recurso (como compra de ingressos), evitando condições de corrida ou conflitos de transações.

3.4 Supportability
- Arquitetura modular com suporte a atualizações independentes por módulo (feira, expositor, produto, ingresso).
- Código documentado para facilitar manutenção.
- Compatibilidade com navegadores modernos (Chrome, Firefox, Edge).
- Capacidade de escalonamento horizontal em ambiente de nuvem.

3.5 Security Requirements
- O sistema deve implementar controle de acesso baseado em perfis de usuário (administrador, expositor e visitante), garantindo que cada perfil tenha permissões específicas para visualização, edição e exclusão de conteúdos.

3.6 Scalability Requirements
- O sistema deve ser capaz de operar simultaneamente com múltiplos eventos ativos e suportar aumento gradual no número de usuários, feiras e expositores sem perda significativa de desempenho.
- O sistema deverá permitir expansão da infraestrutura para atender à demanda crescente sem necessidade de reestruturação completa da arquitetura.

3.7. Internationalization Requirements
- O sistema deverá oferecer suporte a idioma português brasileiro como padrão e permitir futura adição de novos idiomas.
- O sistema deve exibir datas, moedas e formatos numéricos de acordo com as convenções brasileiras (pt-BR).

4. System Interfaces
- API RESTful para integração com sistemas terceiros.
- Respostas em formato JSON com status HTTP apropriados.

4.1 User Interfaces
- Interface web responsiva com acesso via dispositivos móveis e desktops.
- Tela de login, cadastro, gerenciamento de feiras, expositores, produtos e ingressos.
- Campos de busca, filtros e paginação em listagens.

4.1.1 Look & Feel

- Estilo moderno e limpo, com paleta de cores baseada em tons neutros e verdes.
- Tipografia legível e ícones informativos.
- Compatibilidade mínima com leitores de tela (acessibilidade básica para usuários com deficiência visual).
- Garantia de responsividade em dispositivos com telas menores que 5.

4.1.2 Layout and Navigation Requirements

- Menu lateral fixo com navegação hierárquica.
- Áreas de conteúdo centralizadas com agrupamento lógico por contexto (feira, expositor, produto).

4.1.3 Consistency

- Elementos de interface (botões, campos, tabelas) padronizados por meio de biblioteca de componentes.
- Terminologia uniforme (e.g., “Feira”, “Expositor”, “Produto”).

4.1.4 User Personalization & Customization Requirements

- Exibição de registros criados pelo usuário autenticado.
- Possibilidade de personalização futura de notificações e filtros salvos.

4.2 Interfaces to External Systems or Devices

- Sem integração prevista com dispositivos físicos neste momento.

4.2.1 Software Interfaces

- Integração na primeira versão prevista com sistemas de pagamento para venda de ingressos (via APIs de terceiros).
- O sistema deve permitir a integração com APIs de redes sociais (como Facebook, Instagram e X/Twitter) para facilitar o compartilhamento de eventos e páginas de expositores diretamente pelos usuários.

4.2.2 Hardware Interfaces

- Sistema totalmente baseado na web, sem dependência de hardware específico.

4.2.3 Communications Interfaces

- Requisições HTTP/HTTPS via rede pública.
- Proteção de dados em trânsito com TLS 1.2 ou superior.

5. Business Rules

5.1 Validation and Permission Rules

R001 - CreatorOwnership

- Descrição: Apenas o criador do registro pode editá-lo ou excluí-lo
- Padrão:
  if [currentUser.id != record.createdBy] then [blockAction(), returnError(403, "Acesso negado: você não é o proprietário deste registro")]

R002 - AdminPrivileges

- Descrição: Administradores podem editar qualquer registro mas só excluem registros sem vínculos
- Padrão:

if [user.role == "admin" && record.dependencies.count == 0] then [allowAction()]

R003 - BlockDependentDeletion

- Descrição: Bloqueia exclusão de registros com dependências ativas.
- Padrão:
  if [record.hasActiveDependencies()] then [showWarning("Exclusão bloqueada: existem registros vinculados")]

R004 - MandatoryFieldsValidation

- Descrição: Campos obrigatórios devem ser preenchidos antes do salvamento.
- Padrão:
  if [form.validateRequiredFields() == false] then [highlightEmptyFields(), preventSubmission()]

5.2 Time-Based Rules

R005 - TicketExpiration

- Descrição: Ingressos são válidos apenas na data do evento correspondente
- Padrão:
  if [ticket.eventDate != currentDate] then [invalidateTicket(), notifyUser("Ingresso expirado")]

R006 - EditDeadline

- Descrição: Registros só podem ser editados até 24h antes do evento
- Padrão:
  if [event.startTime - currentTime < 24.hours] then [disableEditing(), showMessage("Período de edição encerrado")]

5.3 Financial Rules

R007 - RefundPolicy

- Descrição: Reembolsos só são permitidos até 48h antes do evento.
- Padrão:
  if [event.startTime - currentTime < 48.hours] then [denyRefund(), showPolicy("Reembolso não permitido neste prazo")]

R008 - DiscountApplication

- Descrição: Descontos só aplicam para compras acima de 3 ingressos.
- Padrão:
  if [cart.tickets.count >= 3] then [applyDiscount(10%)]

6. System Constraints

- Desenvolvimento em linguagem de programação moderna (sugestão: JavaScript/TypeScript com Node.js no backend e React no frontend).
- Uso de banco de dados relacional (PostgreSQL).
- Hospedagem em nuvem (AWS, Azure ou similar), sem dependência de hardware proprietário.
- Ferramentas de versionamento como **Git** (via GitHub) e integração contínua serão obrigatórias para rastreamento das alterações.
- O desenvolvimento deverá seguir padrões estabelecidos de codificação (como Airbnb Style Guide para JavaScript), com revisão de código via pull requests.
- As dependências externas devem seguir as regras de licenciamento livre (ex: MIT,

Apache 2.0), conforme alinhado com os objetivos de software aberto descritos no Documento de Visão.

7. System Compliance

7.1 Licensing Requirements

- O sistema deve implementar controle de licenciamento para usuários premium, se aplicável em versões futuras.

7.2 Legal, Copyright, and Other Notices

- Todo conteúdo cadastrado no sistema deve estar sujeito à política de privacidade e termos de uso da plataforma.
- A aplicação deve exibir e exigir aceitação explícita dos termos de uso e política de privacidade no momento do cadastro do usuário.

7.3 Applicable Standards

- A aplicação deve adotar princípios de segurança amplamente reconhecidos, como o OWASP Top 10.
- A interface deve seguir diretrizes básicas de acessibilidade e responsividade, conforme WCAG 2.1 AA, para garantir uma experiência positiva a todos os perfis de usuário.

8. System Documentation

- Manual do usuário acessível via interface do sistema.
- Ajuda contextual (tooltips e tutoriais iniciais).
- Documentação da API para desenvolvedores externos.