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	
Descrição da Arquitetura do Software	Data: 11/06/2025

1. Architectural Goals and Philosophy

A arquitetura será orientada a serviços (Service-Oriented Architecture - SOA) e construída com base no padrão MVC (Model-View-Controller), visando modularidade, manutenibilidade, escalabilidade e segurança. Como o sistema lida com dados sensíveis de usuários autenticados e regras de negócio restritivas (e.g., controle de autoria e vínculos entre registros), a arquitetura deverá também oferecer mecanismos de autenticação, autorização e integridade referencial.

Objetivos principais:

- Separação clara de responsabilidades (MVC);
- Suporte a múltiplos usuários e sessões autenticadas;
- Camadas bem definidas para facilitar manutenção futura;
- Integração com banco de dados relacional;
- Potencial para migração futura para plataformas móveis ou web progressiva.

2. Assumptions and Dependencies

- O sistema será acessado via web, portanto depende de um servidor web e de um banco de dados relacional utilizando o PostgreSQL.
- Utilizaremos ferramentas/frameworks como Node.js com Express.
- Autenticação será baseada em tokens (JWT).
- A equipe tem conhecimento intermediário de programação e modelagem de dados, mas experiência limitada com microsserviços e arquiteturas distribuídas.
- Espera-se que o sistema opere em um único servidor durante a fase inicial.

3. Architecturally Significant Requirements

- Requisitos com impacto direto na arquitetura incluem:
- CRUD completo de Feiras, Expositores, Produtos e Ingressos.
- Controle de acesso por autenticação e associação de registros ao usuário que os criou.
- Regra de exclusão condicional com integridade referencial (não excluir feiras que tenham

expositores, por exemplo).
- Listagem pública (sem autenticação) e modificações privadas (com autenticação).

4. Decisions, Constraints, and Justifications

- Padrão MVC adotado: Permite separação clara de lógica de negócio, visualização e controle, facilitando testes e manutenção.
- Framework com ORM embutido (Node.js com Express): agiliza desenvolvimento e garante integridade dos dados.
- Validações no backend: mesmo que existam validações no frontend, a integridade será reforçada no servidor.
- Autenticação JWT: oferece escalabilidade e segurança para operações autenticadas.
- Não serão utilizados microsserviços: sistema será monolítico nesta fase, por simplicidade e foco no aprendizado.

5. Architectural Mechanisms

- Mecanismo de Autenticação e Autorização: Implementado com JWT ou sessões; protege operações de modificação.
- Controle de Autoria de Registros: Cada entidade tem campo usuario_criador_id; validações bloqueiam alterações de registros por usuários não autorizados.
- Integridade Referencial e Regras de Exclusão: Implementadas com constraints no banco e lógica de negócio no backend.
- Persistência com ORM: Simplifica acesso a dados e sincronização com modelo conceitual.

6. Key Abstractions

- **Feira:** nome, descrição, data início/fim, local, cidade, estado.
- **Expositor:** nome, descrição, contato.
- **Produto:** nome, descrição, preço.
- **Ingresso:** feira, número, data.
- **Usuário:** autenticação, associação com entidades criadas.
- **Sessão:** controle de acesso do usuário autenticado.

7. Layers or Architectural Framework

- **Apresentação:** interface web utilizando HTML/CSS/JS e framework frontend React.
- **Controle:** endpoints REST que recebem e processam requisições.
- **Negócio:** lógica para CRUD, regras de negócio, validações de acesso e integridade.
- **Persistência:** acesso ao banco de dados via ORM.

8. Architectural Views

- **Lógica:** diagramas de classes e pacotes; relacionamentos entre Feira, Expositor, Produto, Ingresso e Usuário.
- **Operacional:** sistema monolítico rodando em servidor local com serviço de banco de dados separado.
- **Casos de Uso:** acesso público a listagens; CRUD restrito ao usuário autenticado; controle de integridade e exclusão.