Arquitetura de Software

Propósito

Neste documento estão as principais decisões arquiteturais e justificativas adotadas para no desenvolvimento do sistema hipotético de gerenciamento de feiras, utilizando o framework **Flask** (Python). O objetivo é garantir clareza técnica, consistência e facilidade de manuntencão e desenvolvimento.

Metas e Filosofia

A arquitetura foi pensada com os seguintes objetivos:

- **Simplicidade e clareza**: Usar um framework level, fácil de entender.
- **Separação de responsabilidades**: Manter o código organizado em camadas.
- **Fácil prototipagem**: Permitir alterações e testes rápidos durante o ciclo iterativo de desenvolvimento.
- **Reutilização de componentes**: Utilizar templetes, rotas e funções reutilizáveis.
- **Documentação e testes integrados**: Facilitar testes manuais.

Premissas e Dependências

- O sistema será executado localmente ou em um ambiente controlado (localhost)
- O banco de dados será SQLite.
- As operações CRUD são o coração do sistema.
- A equipe tem conhecimento básico em python, html
- O projeto não exige alta escalabilidade.

Requisitos Significativos

Os principais requisitos que direcionam a arquitetura são:

- Cadastro e autenticação de usuários.
- CRUD Completo para feiras, expositores e produtos.
- Associação lógica entre entidades
- Controle de acesso(Somente criador pode modificar)
- Geração e listagem de ingressos.
- Interface amigável e navegação simples.

Decisões, Restrições e Justificativas

Decisão/Restrição	Justificativa
Uso do Flask	Framework minimalista e simples
Banco de dados SQLite	Simples, sem necessidade de servidor de banco externo
Validação mínima de entrada	Simples, pois o foco está no funcionamento básico e entrega do sistema.

Abstrações Principais

Abstração	Descrição
Feira	Evento com nome, local, data, cidade e estado.
Expositor	Entidade associada a uma feira, com nome, descrição e contato.
Produto	Item de um expositor, com nome, descrição e preço.
Ingresso	Acesso à feira com número e data.
Usuário	Responsável por criar/editar/deletar entidades.

Visões Arquiteturais

Visão Lógica

- Entidade inter-relacionadas
    - Feira -> Expositores -> Produtos
- Usuário -> [Feiras, Expositores, Produtos]
- As operações CRUD seguem uma estrutura REST-like:
    - **GET** para visualização
    - **POST** para criação
    - **PUT/POST** para atualização
    - **DELETE** para remoção

Visão de casos de uso

Casos de usos significativos:

- Cadastrar/login
- Criar uma feira
- Adicionar expositores e produtos
- Emitir ingresso
- Ver informações públicas de feiras, expositores e produtos