

I.

---

# Projeto: Event Manager

## Descrição Geral

Desenvolva um sistema web para gerenciamento e inscrição em eventos. O sistema terá dois tipos de usuários: administradores e participantes. Os administradores poderão criar, editar e excluir eventos, enquanto os participantes poderão visualizar os eventos disponíveis, se inscrever neles e gerenciar suas inscrições.

---

## Requisitos Funcionais

### 1. Autenticação e Autorização

- Utilize o sistema de autenticação do Laravel para registro, login, logout e recuperação de senha.
- Diferencie dois papéis de usuário:
  - **Administrador:** Pode gerenciar eventos e visualizar inscrições de participantes.
  - **Participante:** Pode visualizar eventos e realizar inscrições.
- Garanta que cada usuário tenha acesso apenas às funcionalidades permitidas pelo seu papel.

### 2. Gerenciamento de Eventos (Admin)

- **CRUD de Eventos:** Permita que administradores criem, visualizem, editem e excluam eventos.
- Cada evento deve possuir, no mínimo, os seguintes campos:
  - **Título**
  - **Descrição**
  - **Data e hora de início e término**
  - **Localização**
  - **Capacidade máxima de participantes**
  - **Status** (ex.: aberto para inscrições, encerrado, cancelado)

### 3. Inscrições em Eventos (Participante)

- Participantes devem poder visualizar a lista de eventos abertos para inscrições.
- Possibilitar a inscrição em eventos, respeitando a capacidade máxima definida.
- Permitir que o participante visualize e cancele suas inscrições.

- (Opcional) Enviar notificações ou e-mails de confirmação ao se inscrever ou cancelar inscrição.

#### 4. Interface de Usuário

- Utilize Blade templates para a construção das views.
- Crie uma interface responsiva e amigável, usando Tailwind ou outro framework CSS de sua preferência.
- Tenha uma página inicial que exiba os eventos (para participantes) e um painel de controle para administradores.

#### 5. API RESTful (Opcional para desafios avançados)

- Desenvolva endpoints para operações de CRUD de eventos e gerenciamento de inscrições.
- Proteja os endpoints com autenticação via tokens, utilizando Laravel Sanctum ou Passport.
- Permita que a API retorne respostas em JSON seguindo boas práticas.

#### 6. Testes

- Implemente testes unitários e de feature utilizando PHPUnit para as principais funcionalidades, como:
  - Criação e gerenciamento de eventos.
  - Inscrição e cancelamento de eventos.
  - Validações (ex.: verificação da capacidade máxima).

#### 7. Banco de Dados

- Utilize migrations para a criação e modificação das tabelas.
- Crie seeders e factories para popular o banco com dados fictícios, facilitando testes e demonstrações.
- Considere relações entre tabelas, por exemplo:
  - Um evento pode ter muitas inscrições.
  - Cada inscrição está associada a um participante e a um evento.

---

## Requisitos Técnicos

- **Linguagem/Framework:** PHP (7.4+ ou 8.x) com Laravel (versão 8, 9 ou 10, conforme definição do teste).
  - **Versionamento:** Uso do Git para versionamento. O projeto final deve ser publicado em um repositório público no GitHub.
  - **Código:** Manter a organização e seguir os padrões PSR.
  - **Documentação:** Incluir um README.md detalhado com instruções de instalação, configuração, execução dos testes e uso da aplicação.
-

## Critérios de Avaliação

- **Qualidade do Código:** Organização, clareza e aderência às boas práticas do Laravel.
  - **Implementação das Funcionalidades:** Correta execução do CRUD de eventos, sistema de inscrições e diferenciação de acessos (admin x participante).
  - **Testes:** Cobertura e qualidade dos testes unitários e de feature.
  - **Documentação:** Clareza das instruções e organização do README.md.
  - **Versionamento com Git:** Commits claros e uma estrutura de branch bem definida (se aplicável).
- 

## Entrega

O candidato deve finalizar o projeto e publicá-lo em um repositório público no GitHub, compartilhando o link para avaliação.