

# Descrição da Infraestrutura de Produção

## 1. Introdução

Este documento detalha a arquitetura de infraestrutura proposta para a implantação e operação do "**Sistema de Divulgação e Gestão de Feiras**" em um ambiente de produção. O objetivo é descrever os componentes de hardware, software e serviços externos necessários para garantir que o sistema opere de forma eficiente, segura e confiável, atendendo aos requisitos de disponibilidade e desempenho para os usuários finais.

## 2. Visão Geral da Arquitetura de Implantação

A implantação do sistema seguirá um modelo de arquitetura em camadas, um padrão consolidado para aplicações web modernas, que garante a separação de responsabilidades e a escalabilidade. O fluxo de uma requisição do usuário passará pelas seguintes camadas:

[Cliente/Navegador] -> [Internet] -> [Servidor Web (Nginx)] -> [Servidor de Aplicação (Gunicorn)] <-> [Aplicação (Django)] -> [Banco de Dados (PostgreSQL)]

- **Cliente:** O usuário final interagindo com o sistema através de um navegador web.
- **Servidor Web:** A primeira camada de contato no servidor, responsável por lidar com o tráfego de entrada e servir arquivos estáticos.
- **Servidor de Aplicação:** Camada responsável por executar a lógica de negócios da aplicação Django.
- **Banco de Dados:** Camada de persistência onde todos os dados do sistema são armazenados.

## 3. Componentes de Hardware (Recursos em Nuvem)

A aplicação será hospedada em um ambiente de nuvem, utilizando um Servidor Virtual Privado (VPS) ou uma instância de nuvem equivalente. Esta abordagem oferece flexibilidade, escalabilidade e um custo-benefício superior ao hardware físico dedicado.

- **Tipo de Servidor:** Instância de Nuvem / VPS (Virtual Private Server).
- **Especificações Iniciais Recomendadas:**
  - **CPU:** 1 vCPU
  - **Memória RAM:** 2 GB

- **Armazenamento:** 50 GB de armazenamento em SSD (Solid State Drive) para maior velocidade de leitura/escrita.
- **Escalabilidade:** A configuração pode ser facilmente escalada verticalmente (aumentando os recursos da instância) ou horizontalmente (adicionando mais instâncias) conforme a demanda de tráfego aumentar.

#### 4. Componentes de Software

O ambiente do servidor será configurado com um conjunto de softwares padrão da indústria para hospedar aplicações Django de forma otimizada e segura.

- **4.1. Sistema Operacional:**
  - **Software:** Linux, distribuição **Ubuntu 22.04 LTS (Long-Term Support)**.
  - **Justificativa:** É um sistema operacional de código aberto, robusto, seguro e com vasto suporte da comunidade, sendo o padrão de mercado para servidores web. A versão LTS garante atualizações de segurança por um longo período.
- **4.2. Banco de Dados:**
  - **Software:** **PostgreSQL** (versão 14 ou superior).
  - **Justificativa:** Enquanto o SQLite é utilizado para o desenvolvimento devido à sua simplicidade, um ambiente de produção exige um sistema de gerenciamento de banco de dados mais robusto. O PostgreSQL é altamente recomendado pela comunidade Django por sua confiabilidade, performance com grande volume de dados e suporte a funcionalidades avançadas.
- **4.3. Servidor Web / Proxy Reverso:**
  - **Software:** **Nginx**.
  - **Justificativa:** Nginx atuará como a porta de entrada para todas as requisições. Suas principais responsabilidades serão:
    - **Servir Arquivos Estáticos:** Entregar arquivos como CSS, JavaScript e imagens diretamente ao cliente, de forma muito mais rápida e eficiente do que a aplicação Django.
    - **Proxy Reverso:** Receber as requisições dinâmicas e encaminhá-las para o servidor de aplicação (Gunicorn), atuando como um intermediário que protege a aplicação.
    - **Terminação SSL/TLS:** Gerenciar a criptografia HTTPS, descriptografando as requisições antes de passá-las para a aplicação.
- **4.4. Servidor de Aplicação (WSGI):**
  - **Software:** **Gunicorn (Green Unicorn)**.
  - **Justificativa:** O servidor de desenvolvimento do Django (runserver) não é adequado para produção. Gunicorn é um servidor WSGI (Web Server Gateway Interface) padrão para Python, responsável por

executar o código da aplicação Django, gerenciar múltiplos processos de trabalho para lidar com requisições concorrentes e se comunicar de forma eficiente com o Nginx.

- **4.5. Aplicação Principal:**

- **Software:** O código do projeto "**Sistema de Divulgação e Gestão de Feiras**" (Python 3.10+ e Django 4.x+).

## *5. Serviços Externos e Dependências*

- **5.1. Provedor de Hospedagem em Nuvem:**

- **Serviço:** Plataformas como **DigitalOcean**, **AWS (Amazon Web Services)**, ou **Google Cloud Platform**.
  - **Descrição:** Fornecerão a infraestrutura de hardware virtualizada (VPS) onde todos os softwares serão instalados.

- **5.2. Certificado SSL/TLS:**

- **Serviço:** **Let's Encrypt**.
- **Descrição:** Uma autoridade de certificação gratuita e automatizada que fornecerá o certificado SSL/TLS necessário para habilitar o tráfego seguro via **HTTPS**, um requisito essencial de segurança para qualquer aplicação web.

- **5.3. Serviço de Envio de E-mails Transacionais:**

- **Serviço:** **SendGrid**, **Amazon SES** ou **Mailgun**.
- **Descrição:** Para funcionalidades como "esqueci minha senha" e outras notificações, não é confiável enviar e-mails diretamente do servidor, pois eles podem ser bloqueados ou marcados como spam. Um serviço transacional garante alta taxa de entrega e gerenciamento profissional dos envios.

## *6. Segurança e Manutenção*

- **Firewall:** Configuração do UFW (Uncomplicated Firewall) no servidor para permitir tráfego apenas nas portas necessárias (80 para HTTP e 443 para HTTPS), bloqueando todo o resto.
- **Atualizações:** Implementação de uma rotina para manter o sistema operacional e todos os pacotes de software atualizados com os últimos patches de segurança.
- **Backups:** Configuração de backups automáticos e periódicos do banco de dados PostgreSQL e dos arquivos da aplicação, armazenados em um local seguro e externo.