

## ThinkPrint Galeria Facial – Documento Técnico

### Visão Geral:

O projeto ThinkPrint Galeria Facial é uma plataforma de gerenciamento de galerias de fotos e vídeos com reconhecimento facial voltada para eventos. A solução é composta por um backend escrito em Python/Flask, três aplicações web (dashboard administrativo, microsite do participante e slideshow), além de um script de upload em Python para o servidor.

### Arquitetura:

- Backend: Flask expõe uma API REST com autenticação de administradores, gerenciamento de eventos, registro de participantes e listagem de galerias.
- Frontend Dashboard: Interface administrativa que permite login, criação de eventos e visualização da lista de eventos.
- Microsite: Site responsivo usado por participantes para cadastro (telefone + selfie) e consulta à galeria individual.
- Slideshow: Aplicação que apresenta todas as mídias de um evento em sequência automática para projeção em tela.
- Uploader: Script que monitora uma pasta no Windows e envia automaticamente novas fotos/vídeos para o servidor.

### Fluxos Principais:

1. O administrador cria um evento via dashboard. É gerado um link/QR Code para o cadastro dos participantes.
2. Participantes acessam o microsite, informam telefone e enviam uma selfie. O backend cria um registro e associa as mídias existentes do evento ao participante (reconhecimento facial via AWS Rekognition – nesta referência a função é simulada).
3. Durante o evento, o uploader envia novos arquivos para o backend. O sistema tenta identificar rostos e envia notificações por SMS/WhatsApp (via Twilio).
4. Os participantes recebem um link para sua galeria personalizada, onde podem ver e baixar suas fotos e vídeos.
5. Um slideshow público pode ser exibido consumindo todas as mídias do evento.

### Configuração:

- Defina credenciais AWS (Rekognition/S3) e Twilio em `thinkprint/backend/config.py` ou via variáveis de ambiente.
- Instale dependências do backend com pip e execute `python app.py``.
- Para cada aplicativo React, rode ``npm install`` e ``npm run dev``.
- O script uploader é executado via linha de comando: `python uploader.py --folder <pasta> --event <id> --ap`