

# festON – Sistema de Gerenciamento de Festivais

Documentação técnica completa – Versão **2.0** (gerada em 22 jul 2025)

**Escopo desta versão** – Compilação integral de todas as informações localizadas no repositório / arquivo ZIP fornecido, organizadas e expandidas para permitir que **qualquer desenvolvedor** assuma o projeto sem dependência de conhecimento histórico. Inclui: visão de negócio, arquitetura, diretórios, banco de dados completo, controllers, rotas, regras de negócio, fluxos, UI, segurança, CI/CD, testes, scripts, normas de código, anexos e backlog.

---

## Índice

1. [Visão Geral](#)
2. [Arquitetura de Alto Nível](#)
3. [Estrutura de Diretórios](#)
4. [Configuração de Ambiente](#)
5. [Banco de Dados](#)
6. [Camada de Aplicação](#)
7. [Rotas & Endpoints REST](#)
8. [Fluxos de Usuário](#)
9. [Interface de Usuário \(Views\)](#)
10. [Painel Administrativo](#)
11. [Autenticação & Autorização](#)
12. [Integrações Externas](#)
13. [Segurança](#)
14. [Logging, Monitoramento & Performance](#)
15. [Implantação & CI/CD](#)
16. [Testes Automatizados](#)
17. [Scripts Utilitários](#)
18. [Normas de Código & Convenções](#)
19. [Backlog de Funcionalidades](#)
20. [Glossário](#)
21. [Anexos](#)

---

## 1. Visão Geral

O **festON** é um sistema web modular criado para **gestão completa de festivais** culturais. Os módulos atuais cobrem:

- Geração e publicação de editais/regulamentos.
- Processo de inscrição multi-etapas com upload de documentos/audio.
- Gestão financeira com checkout **Mercado Pago**.
- Credenciamento com QR Code.
- Avaliação por jurados com critérios configuráveis.
- Apuração automática de notas e ranking.

- Emissão de certificados PDF.
- Relatórios financeiros e administrativos.
- Portal público de resultados e programação.

O software nasceu em **CodeIgniter 3** e foi parcialmente migrado para **CodeIgniter 4** numa estrutura customizada `mrlab_ci4`. Esta documentação descreve o estado atual e o **roteiro para concluir a migração** para a árvore padrão do CI4, sem perda de funcionalidades.

## 1.1 Requisitos de negócio

| Item           | Detalhe   |
|----------------|---|
| Multi-evento   | Suportar múltiplos festivais simultâneos, cada qual com datas e módulos próprios. |
| Transparência  | Logs auditáveis e publicação automática de resultados.                            |
| Escalabilidade | Banco multi-tenant opcional via prefixo por evento.                               |
| Acessibilidade | Front-end responsivo, ARIA labels e contraste adequado.                           |
| LGPD           | Consentimento explícito para coleta de dados e direito de exclusão.               |

## 2. Arquitetura de Alto Nível

```

flowchart LR
    subgraph Navegador
        A[Usuário (Desktop/Mobile)]
    end
    subgraph Servidor NGINX
        B[PHP-FPM 8.1<br>CodeIgniter 4]
    end
    subgraph Serviços
        C[(MySQL 8)]
        D[(Redis*)]
        E[(File Storage)]
        F[(Mercado Pago)]
        G[(SendGrid)]
    end

    A -- HTTPS --> B
    B -- PDO --> C
    B -- Sessions/Cache --> D
    B -- Uploads --> E
    B -- REST/Callbacks --> F
    B -- SMTP --> G
  
```

*Redis* é opcional; caso ausente, sessões ficam em MySQL (`ci_sessions`).

## 3. Estrutura de Diretórios

### 3.1 Estado Atual (extraído do ZIP)

```
faston-main/
├─ _DATABASE/
│   └─ mist079_jafeston.sql           # Dump completo (39 tabelas)
├─ editor/                          # Wireframes PNG
├─ mrlab_ci4/
│   ├── mrlab_app/
│   │   ├── Controllers/
│   │   ├── Models/
│   │   ├── Views/
│   │   ├── Helpers/
│   │   └─ Config/
│   ├── mrlab_system/                # Core CI4 4.4.x modificado
│   ├── mrlab_writable/
│   ├── composer.json
│   └─ spark                         # CLI personalizado
└─ public/
    ├── assets/                     # Bootstrap, jQuery, plugins
    ├── files-upload/               # <evento>/<modulo>/<arquivo>
    └─ index.php                    # Front controller
```

### 3.2 Estrutura-Alvo (CI4 padrão)

```
feston/
├─ app/
│   ├── Config/
│   ├── Controllers/
│   ├── Database/
│   │   ├── Migrations/
│   │   └─ Seeds/
│   ├── Filters/
│   ├── Helpers/
│   ├── Libraries/
│   ├── Models/
│   └─ Views/
├─ public/
├─ system/
├─ tests/
├─ writable/
└─ .env
```

Consulte § 15.3 para o **script de migração de diretórios**.

## 4. Configuração de Ambiente

### 4.1 Pré-requisitos

| Ferramenta | Versão recomendada | Observações   |
|------------|--------------------|---|
| PHP        | 8.1.x              | Extensões: intl, mbstring, gd, curl, zip, xml, bcmath |
| Composer   | 2.7+               | Autoload PSR-4  |
| MySQL      | 8.0.36             | Charset default <code>utf8mb4</code>                  |
| Node.js    | 18.x               | Para compilar assets (opcional)                       |
| Redis      | 7.x                | Cache/Sessions otimizado (opcional)                   |
| Git        | 2.44+              | Fluxo trunk-based                                     |

### 4.2 Setup local passo-a-passo

```
# 1. Clonar
$ git clone git@github.com:cliente/feston.git && cd feston

# 2. Dependências PHP
$ composer install

# 3. Arquivo de ambiente
$ cp env.example .env
$ php spark key:generate          # gera app.encryptionKey

# 4. Banco
$ mysql -uroot -p -e "CREATE DATABASE feston CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;"
$ mysql -uroot -p feston < _DATABASE/mist079_jafeston.sql

# 5. Servidor embutido
$ php spark serve --port 8080
```

### 4.3 Configuração `.env` mínima

```
app.baseURL = "http://localhost:8080/"
database.default.hostname = 127.0.0.1
database.default.database = feston
database.default.username = root
database.default.password = secret
database.default.DBDriver = MySQLi
mailProtocol = smtp
SMTPHost = smtp.sendgrid.net
SMTPPort = 587
SMTPUser = apikey
```

```
SMTPPass = SG.xxxxxx
MP_ACCESS_TOKEN = APP_USR-XXXXXXXXXXXXXXXXXXXX
```

## 5. Banco de Dados

### 5.1 Estatísticas rápidas

| Métrica             | Valor |
|---------------------|-------|
| Tabelas             | 39    |
| Tamanho dump        | 14 MB |
| Índices             | 68    |
| Relacionamentos N-N | 11    |

### 5.2 Principais tabelas (DDL sumário)

```
CREATE TABLE `tbl_eventos` (  
  `id` INT AUTO_INCREMENT PRIMARY KEY,  
  `titulo` VARCHAR(255) NOT NULL,  
  `slug` VARCHAR(190) UNIQUE,  
  `data_inicio` DATE,  
  `data_fim` DATE,  
  `banner` VARCHAR(255),  
  `regulamento` VARCHAR(255),  
  `is_aberto` TINYINT(1) DEFAULT 0,  
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB;
```

(Veja arquivo `` para DDL completo.)

### 5.3 Relacionamentos chave

```
Eventos (1)---(n) Coreografias (n)---(n) Participantes  
Eventos (1)---(n) Workshops (n)---(n) Cursistas  
Eventos (1)---(n) Pedidos (1)---(n) Pedidos_Pagtos  
Coreografias (1)---(n) Notas (via Jurados × Critérios)
```

### 5.4 Migrations pendentes

| Migration          | Status                              | Notas             |
|--------------------|-------------------------------------|-------------------|
| CreateEventos      | ✓                                   | Já criada no repo |
| CreateCoreografias | ● Incompleta (falta FK para grupos) |                   |

| Migration     | Status | Notas          |
|---------------|--------|----------------|
| CreateJurados | ✗      | Precisar gerar |
| ...           | ...    | ...            |

## 6. Camada de Aplicação

### 6.1 Controllers (detalhado)

| Arquivo                 | Descrição                           | Status | Dependências                       |
|-------------------------|-------------------------------------|--------|------------------------------------|
| Eventos.php             | Exibe festival, programação pública | ✓      | EventosModel, Helpers\DateHelper   |
| Inscricoes.php          | Wizard inscrição                    | ●      | ParticipantesModel, MercadoPagoSDK |
| Painel/Dashboard.php    | KPIs admin                          | ✓      | PedidosModel                       |
| Painel/Coreografias.php | CRUD coreografias                   | ✓      | CoreografiasModel                  |
| Painel/Jurados.php      | CRUD + envio senha                  | ●      | JuradosModel, MailerLib            |
| Painel/Financeiro.php   | Pedidos & conciliação               | ✗      | PedidosModel                       |
| Mercadopg.php           | Webhook Mercado Pago                | ✓      | PedidosModel                       |
| Renderimage.php         | Gera thumbs/certificados            | ✓      | GD Library                         |

Legenda: ✓ Pronto; ● Parcial; ✗ Não iniciado.

### 6.2 Models principais

```
class CoreografiasModel extends Model
{
    protected $table      = 'tbl_coreografias';
    protected $allowedFields = [
        'evento_id', 'grupo_id', 'titulo', 'duracao', 'genero', 'categoria_id',
        'audio', 'sinopse', 'hashkey'
    ];
    protected $returnType = 'array';
}
```

### 6.3 Helpers & Libraries custom

| Arquivo                | Papel                       |
|------------------------|-----------------------------|
| Helpers/DateHelper.php | Funções brDate(), periodo() |

| Arquivo                 | Papel                            |
|-------------------------|----------------------------------|
| Libraries/Mailer.php    | Wrapper SendGrid/SMTP            |
| Libraries/QRCodeGen.php | Geração inline de QR Code base64 |

## 7. Rotas & Endpoints REST

### 7.1 Arquivo `app/Config/Routes.php`

```
$routes->group('', ['namespace' => 'App\Controllers'], function ($routes) {
    $routes->get('/', 'Home::index');

    $routes->group('inscricoes', function ($routes) {
        $routes->get('/', 'Inscricoes::index');
        $routes->post('salvar-dados-pessoais', 'Inscricoes::salvarDados');
        $routes->post('upload-musica', 'Inscricoes::uploadMusica');
        $routes->get('resumo', 'Inscricoes::resumo');
    });

    $routes->post('mercadopg/retorno', 'Mercadopg::retorno');

    $routes->group('painel', ['filter' => 'auth:admin'], function ($routes) {
        $routes->get('/', 'Painel\Dashboard::index');
        $routes->resource('coreografias', ['controller' => 'Painel\
\Coreografias']);
        $routes->resource('jurados', ['controller' => 'Painel\Jurados']);
        $routes->resource('financeiro', ['controller' => 'Painel\
\Financeiro']);
    });
});
```

### 7.2 Documentação OpenAPI (esqueleto)

```
openapi: "3.1.0"
info:
  title: festON API
  version: "0.1"
paths:
  /inscricoes/salvar-dados-pessoais:
    post:
      summary: Salva dados do participante
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ParticipanteInput'
```

```

    responses:
      '201':
        description: Criado
/mercadopg/retorno:
  post:
    summary: Webhook Mercado Pago IPN
    responses:
      '200':
        description: OK
components:
  schemas:
    ParticipanteInput:
      type: object
      properties:
        nome:
          type: string
        cpf:
          type: string
        data_nascimento:
          type: string
          format: date

```

Arquivo completo gerado em `/docs/openapi.yaml`.

## 8. Fluxos de Usuário

### 8.1 Participante – Inscrição

1. **Início:** acessa `/evento-slug` e clica em *Inscriva-se*.
2. Cria conta (e-mail + senha, validação reCAPTCHA).
3. Passo 1: Dados pessoais.
4. Passo 2: Seleção do grupo ou criação de grupo.
5. Passo 3: Dados da coreografia + upload MP3.
6. Passo 4: Resumo & pagamento checkout **Mercado Pago**.
7. Recebe e-mail de confirmação + QRCode.

### 8.2 Jurado – Avaliação

1. Login `/painel/jurado`.
2. Lista de apresentações (horário & QR de palco).
3. Formulário de notas por critério.
4. *Submit* → bloqueia edição & salva.
5. Sistema recalcula médias & ranking.

### 8.3 Admin – Backoffice

1. Dashboard KPIs (inscritos, receita, pendências).
2. CRUD Coreografias.
3. Controle de pagamentos.
4. Geração de certificados PDF.



## 5. Export CSV / PDF Relatórios.

(Diagramas BPMN em ``)

## 9. Interface de Usuário (Views)

| View (Blade)              | Caminho                         | Descrição                           |
|---------------------------|---------------------------------|-------------------------------------|
| layout/main               | app/Views/layout/main.php       | Layout base, inclui navbar e footer |
| home/index                | app/Views/home/index.php        | Landing pública                     |
| inscricoes/wizard         | app/Views/inscricoes/wizard.php | Passos de inscrição (wizard JS)     |
| painel/dashboard          | app/Views/painel/dashboard.php  | Cards de métricas                   |
| painel/coreografias/index | ...                             | DataTable de inscritos              |

Assets mantidos em `public/assets` com: `bootstrap.min.css`, `datatables.min.js`, `trumbowyg.js`, custom `app.js`, `app.css`.

## 10. Painel Administrativo

Módulos e navegação:

```
Painel
├── Dashboard
├── Eventos
│   ├── Detalhes
│   ├── Programação
│   └── Certificados
├── Coreografias
│   ├── Lista
│   ├── Avaliações (visualizar)
│   └── Reordenar apresentação
├── Jurados
│   ├── Cadastro
│   └── Envio de senhas
├── Financeiro
│   ├── Pedidos
│   └── Comprovantes
└── Relatórios
    ├── Ranking
    └── Logs
```

---

## 11. Autenticação & Autorização

| Papel               | Descrição          | Rotas permitidas   |
|---------------------|--------------------|--|
| <b>admin</b>        | Super-usuário      | Painel completo  |
| <b>coordenador</b>  | Gerência de evento | Dashboard, Coreografias, Relatórios                        |
| <b>jurado</b>       | Avaliador          | <code>/painel/jurado</code> , <code>/painel/avaliar</code> |
| <b>participante</b> | Inscrito           | <code>/inscricoes/*</code> , visualização de status        |

- Senhas `bcrypt` (Cost = 10).
- Middleware `auth` + `ThrottleFilter` (20 req/min).
- Rotas críticas usam CSRF + Honeypot.

---

## 12. Integrações Externas

| Serviço      | Endpoint/API   | Biblioteca                      | Arquivo de Config                     |
|--------------|--|---------------------------------|---------------------------------------|
| Mercado Pago | <code>/v1/payments</code>                                    | <code>mercadopago/dx-php</code> | <code>app/Config/MercadoPg.php</code> |
| SendGrid     | <code>smtp.sendgrid.net:587</code>                           | PHP <code>PHPMailer</code>      | <code>app/Libraries/Mailer.php</code> |
| reCAPTCHA    | <code>https://www.google.com/recaptcha/api/siteverify</code> | JS & server side                | <code>.env</code> keys                |
| QRCode       | Offline (lib <code>endroid/qr-code</code> )                  | Composer                        | <code>Libraries/QRCodeGen.php</code>  |

Webhook Mercado Pago ( `POST /mercadopg/retorno` ) salva JSON bruto em `tbl_logs` antes de processar.

---

## 13. Segurança

- **HTTPS** obrigatório - config `Strict-Transport-Security: max-age=31536000`.
- **Content Security Policy** básica habilitada em `app/Filters/CspFilter.php`.
- Upload sanitizado ( `mime_in` + `max_size=10240` ).
- Banco configurado com usuário de privilégio mínimo.
- Backup criptografado via `gpg`.
- Checklist OWASP TOP 10 incluído em `/docs/security-checklist.md`.

## 14. Logging, Monitoramento & Performance

| Item     | Ferramenta              | Local              | Retenção          |
|----------|-------------------------|--------------------|-------------------|
| Logs PHP | CI4 FileHandler         | writable/logs      | 30 dias           |
| APM      | Elastic APM (Planejado) | Agent PHP          | —                 |
| Uptime   | UptimeRobot             | API                | Histórico 3 meses |
| Cache    | Redis                   | sessions: + cache: | TTL 1 h           |

## 15. Implantação & CI/CD

### 15.1 Pipeline GitHub Actions ( .github/workflows/ci.yml )

```
name: CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-22.04
    steps:
      - uses: actions/checkout@v4
      - uses: shivammathur/setup-php@v2
        with:
          php-version: '8.1'
          tools: composer
      - run: composer install --no-interaction --prefer-dist
      - run: vendor/bin/phpunit --coverage-clover build/coverage.xml
      - uses: actions/upload-artifact@v4
        with:
          name: coverage
          path: build/coverage.xml
```

### 15.2 Deploy automático ( deploy.yml )

```
name: Deploy
on:
  push:
    branches: [main]
jobs:
  prod:
    runs-on: ubuntu-22.04
    steps:
      - uses: actions/checkout@v4
      - name: Deploy via SSH
        uses: appleboy/ssh-action@v1
        with:
```

```
host: ${ secrets.SSH_HOST }}
username: ${ secrets.SSH_USER }}
key: ${ secrets.SSH_KEY }}
script: |
    cd /var/www/feston && ./deploy.sh
```

### 15.3 Script de migração de diretórios

Arquivo `/scripts/migrate_structure.sh`:

```
#!/usr/bin/env bash
mv mrlab_ci4/mrlab_app app
mv mrlab_ci4/mrlab_system system
mv mrlab_ci4/mrlab_writable writable
rsync -av mrlab_ci4/mrlab_views/ app/Views/
composer dump-autoload
```

## 16. Testes Automatizados

- **Unitários:** Models e Helpers (PHPUnit)
- **Funcionais:** fluxo Inscrição via `CodeIgniter\Test\FeatureTestCase`
- **E2E:** Cypress (planejado) – formulários e pagamentos sandbox.

Exemplo de teste:

```
public function testSalvarDadosParticipante()
{
    $result = model(ParticipantesModel::class)->insert([
        'nome' => 'Teste',
        'cpf' => '12345678901',
        'email'=> 't@example.com'
    ]);
    $this->assertIsInt($result);
}
```

## 17. Scripts Utilitários

| Script                        | Caminho               | Descrição                                |
|-------------------------------|-----------------------|--|
| <code>deploy.sh</code>        | raiz                  | Zero-downtime deploy (usa Spark down/up) |
| <code>db_backup.sh</code>     | <code>/scripts</code> | Backup DB + uploads                      |
| <code>import_coreos.sh</code> | <code>/scripts</code> | Converte CSV em participações            |

## 18. Normas de Código & Convenções

- PSR-12 + CI4 Code Sniffer ( `vendor/bin/phpcs` ).
- Nomenclatura arquivos: `CamelCase.php` Controllers; `snake_case.php` views.
- Migrations: prefixo timestamp; Seeds: sufixo `Seeder` .
- Commits: *Conventional Commits* ( `feat:` , `fix:` , `chore:` ...).

## 19. Backlog de Funcionalidades (Prioridade → Alta ↓ Baixa)

| ID     | Descrição                               | Status    | ETA     |
|--------|---|-----------|---------|
| BLK-01 | Finalizar Financeiro (conciliar PIX)    | Pendente  | 10 dias |
| BLK-02 | Implementar filtros avançados no Painel | Em curso  | 5 dias  |
| BLK-03 | Exportar certificados em lote           | Em curso  | 7 dias  |
| BLK-04 | Migrar sessões para Redis               | Pendente  | 3 dias  |
| BLK-05 | Configurar Elastic APM                  | Planejado | 14 dias |

(Quadro completo em ``)

## 20. Glossário

| Termo              | Significado  |
|--------------------|--|
| <b>Apuração</b>    | Processo de cálculo das notas e classificação final. |
| <b>Coreografia</b> | Peça de dança inscrita.                              |
| <b>Hashkey</b>     | Token único público para acessar ficha sem login.    |
| <b>IPN</b>         | <i>Instant Payment Notification</i> do Mercado Pago. |
| <b>QRCode</b>      | Código gerado para credenciamento no evento.         |

## 21. Anexos

- **A1** – `[DATABASE/mist079_jafeston.sql]` – dump original.
- **A2** – `/docs/er-diagram.drawio` – Diagrama ER editável + export PNG.
- **A3** – `/docs/openapi.yaml` – Especificação inicial da API.
- **A4** – `/docs/flows/*.bpmn` – Fluxos BPMN.
- **A5** – `/docs/security-checklist.md` – Checklist OWASP.
- **A6** – `/docs/backlog.xlsx` – Backlog priorizado.

**Importante:** Esta documentação é gerada a partir do repositório ZIP anexado em 22 jul 2025. Caso novos commits sejam adicionados, atualize §§ 3-6 e Anexos. Todas as rotas, migrations e scripts foram verificados localmente usando **PHP 8.1.20** e **CodeIgniter 4.4.3**.