L L E V O N

DOCUMENTAÇÃO TÉCNICA

LLEVON — Guia de Deploy e Administração

Sistema de Gerenciamento do Site

PHP 7.4+ | SQLite | 2FA TOTP | API REST | MCP JSON-RPC 2.0 | cPanel

Novembro de 2026 · Baseado em README-ADMIN.md

LLEVON — Sistema de Gerenciamento do Site

Painel administrativo próprio (PHP + SQLite, sem Composer/npm/CDN) para o site da LLEVON. Reúne em um único lugar: métricas de acesso, landing pages com teste A/B, blog estático, usuários com 2FA, configurações do site e uma interface MCP para controle por IA.

1. Índice

2. **Visão geral** 3
3. **Requisitos** 3
4. **Estrutura de arquivos** 4
5. **Instalação** 5
6. **Deploy no cPanel** 6
7. **Usando o painel** 7
   - 7.1 Login e 2FA 7
   - 7.2 Dashboard 7
   - 7.3 Landing Pages 7
   - 7.4 Blog 7
   - 7.5 Usuários (somente `super_admin`) 8
   - 7.6 Configurações 8
8. **Pixel tracker** 8
9. **API REST interna** 10
10. **Interface MCP (controle por IA)** 10
11. **Cron jobs** 12
12. **Backup e restauração** 12
13. **Segurança** 13
14. **Troubleshooting** 13

2. Visão geral

Módulo	O que faz
Dashboard	KPIs do período, gráficos (visitas/dia, tipo de página, top páginas, dispositivos, fontes) e sub-dashboard de LPs com drill-down por variante.
Landing Pages	CRUD de LPs, múltiplas variantes (A/B) com rotação sequencial, aleatória, ponderada ou manual, upload de imagens, preview e publicação.
Blog	CRUD de posts com editor WYSIWYG leve, categorias, tags, agendamento e geração de HTML estático.
Usuários	Contas com papéis (`super_admin` , `editor` , `viewer`), 2FA TOTP obrigatório e API Keys para MCP.
Configurações	Configurações globais do site, instalação do tracker nas páginas estáticas, sitemap, backups, logs MCP e atividade recente.
MCP	Endpoint JSON-RPC 2.0 (`/api/mcp.php`) com 30 ferramentas para que uma IA opere o site via Bearer Token.

Identidade visual: tema escuro (`#0b1426`) com acentos laranja (`#ff6b2b`), sidebar colapsável, toasts de feedback e confirmação antes de qualquer exclusão.

3. Requisitos

Obrigatórios

- PHP **7.4 ou superior** (testado também em PHP 8.x)
- Extensões: `pdo_sqlite` , `mbstring` , `openssl`
- Pastas graváveis: `/data` e `/data/uploads`
- Apache com `mod_rewrite` (roteamento das LPs e do blog)

Recomendados

- Extensão `gd` — redimensionamento automático de imagens no upload (sem ela, a imagem é salva no tamanho original)
- Raiz do site gravável — necessária para gerar os HTML estáticos do blog, os stubs das LPs e o `sitemap.xml`
- HTTPS ativo — obrigatório na prática para o painel e para o MCP

Sem Composer, sem npm, sem build. Chart.js e a biblioteca de QR Code já estão versionados em `admin/assets/js/` .

4. Estrutura de arquivos

```
public_html/
├── admin/                      # Painel administrativo
│   ├── index.php               # Login + 2FA
│   ├── dashboard.php           # Métricas
│   ├── lps.php / lp-editor.php # Landing pages
│   ├── blog.php / blog-editor.php
│   ├── usuarios.php            # Usuários e API Keys (super_admin)
│   ├── configuracoes.php       # Config, backups, logs MCP
│   ├── .htaccess               # Força HTTPS + headers de segurança
│   ├── assets/
│   │   ├── admin.css           # Tema escuro completo
│   │   ├── admin.js            # Helpers globais (objeto LLV)
│   │   └── js/                 # chart.umd.min.js, qrcode.min.js e JS por página
│   └── includes/               # (protegido por .htaccess)
│       ├── auth.php            # Login, 2FA TOTP, permissões, API Keys
│       ├── db.php              # PDO SQLite, migrações, cache
│       ├── functions.php       # Helpers, validação, uploads, config
│       ├── layout.php          # Header/sidebar/footer do painel
│       ├── session.php
│       ├── services/           # lps, blog, metrics, site, backup
│       └── mcp/tools.php       # Registro das ferramentas MCP
├── api/                        # API REST interna (JSON)
│   ├── _bootstrap.php, index.php
│   ├── auth.php, lps.php, blog.php, metrics.php, site.php
│   └── mcp.php                 # Endpoint JSON-RPC 2.0
├── tracker/t.php               # Pixel tracker (GIF 1x1)
├── lp/                         # Roteamento público das LPs
│   ├── index.php
│   └── .htaccess
├── blog/                       # HTML estático gerado + .htaccess
├── data/                       # (bloqueado por .htaccess)
│   ├── config.php              # Constantes versionadas (sem segredos)
│   ├── config.local.php        # Segredos gerados na instalação — NÃO versionado
│   ├── site.db                 # Banco SQLite
│   ├── backups/                # Backups diários
│   └── uploads/{lps,blog,site} # Imagens enviadas pelo painel
├── install.php                 # Instalador (remover após o uso)
├── backup.php                  # Rotina diária de manutenção (cron)
└── README-ADMIN.md
```

5. Instalação

1. **Envie os arquivos** para a raiz do site (`public_html`).
2. **Garanta as permissões** de escrita:

```
bash chmod 755 data data/uploads data/uploads/lps data/uploads/blog data/uploads/site data/backups
```

1. **Acesse** `https://seusite.com.br/install.php` no navegador. O instalador:
2. verifica os requisitos e mostra o que está faltando;
3. gera `data/config.local.php` com `ENCRYPTION_KEY` e `IP_HASH_SALT` aleatórios;
4. cria o banco `data/site.db` com todas as tabelas e índices;
5. cria o usuário **Super Admin** com a senha informada;
6. gera o segredo TOTP e exibe o **QR Code** (leia com Google Authenticator, Authy, 1Password etc.);
7. instala o snippet do tracker nas páginas HTML existentes da raiz;
8. gera o `sitemap.xml`;
9. importa o blog legado de `blog-data/blog.db`, se o arquivo existir (opcional).
10. **Guarde o segredo TOTP** mostrado na tela — ele não é exibido novamente.
11. **Remova o instalador** assim que terminar:

```
bash rm install.php
```

O instalador cria `data/install.lock`; enquanto esse arquivo existir, uma nova execução é recusada. Ainda assim, **apague o arquivo** — é a única garantia real.

1. **Entre no painel**: `https://seusite.com.br/admin/` → e-mail, senha e código de 6 dígitos.

> **Importante:** `data/config.local.php` guarda a chave que criptografa os segredos 2FA. Nunca sobrescreva nem apague esse arquivo ao reenviar o código — todos os segredos TOTP ficariam ilegíveis e seria necessário reconfigurar o 2FA de todos os usuários.

6. Deploy no cPanel

1. **Upload** — envie os arquivos via Gerenciador de Arquivos (ZIP + extrair) ou FTP para `public_html`. Não envie `data/config.local.php`, `data/site.db` nem `data/backups/` de outro ambiente.
2. **Versão do PHP** — em *Selecionar versão do PHP*, escolha 7.4+ (de preferência 8.1/8.2) e confirme que `pdo_sqlite`, `mbstring`, `openssl` e `gd` estão marcadas.
3. **Permissões** — pastas `755`, arquivos `644`. `data/` e `data/uploads/` precisam ser graváveis pelo usuário do PHP.
4. **HTTPS** — ative o SSL (AutoSSL/Let's Encrypt) antes de instalar. O `admin/.htaccess` já força o redirecionamento para HTTPS.
5. **Rode o `install.php`** e, em seguida, apague-o.
6. **Cron job** — *Cron Jobs* → adicionar tarefa diária (veja a seção 11).
7. **Checagem rápida** após o deploy:
8. `https://seusite.com.br/admin/` abre a tela de login;
9. `https://seusite.com.br/data/site.db` retorna **403**;
10. `https://seusite.com.br/tracker/t.php?p=/` devolve uma imagem GIF;
11. `https://seusite.com.br/lp/alguma-lp/` abre a LP publicada.

7. Usando o painel

7.1 Login e 2FA

Login em `/admin/` com e-mail + senha e, na sequência, o código TOTP de 6 dígitos. Após 5 falhas o IP fica bloqueado por 15 minutos. A sessão expira com 30 minutos de inatividade — o painel avisa antes de expirar e renova a sessão enquanto a aba estiver em uso.

7.2 Dashboard

- **KPIs**: visitas totais, visitantes únicos, média diária, LPs ativas e posts publicados, cada um com a variação percentual em relação ao período anterior de mesmo tamanho.
- **Filtros de período**: hoje, últimos 7 dias, últimos 30 dias, mês atual, mês anterior ou intervalo personalizado. Todo o carregamento é feito por AJAX.
- **Gráficos** (Chart.js, renderizados no cliente): visitas por dia (linha), distribuição por tipo de página (pizza), top 10 páginas (barras), dispositivos (pizza) e fontes de tráfego (barras).
- **Sub-dashboard de LPs**: tabela com visitas por LP e, ao expandir, o desempenho de cada variante e a série diária.
- As agregações ficam em cache por 5 minutos; o botão **Atualizar** força o recálculo.

7.3 Landing Pages

- **Listagem** com filtros por status (rascunho, publicada, arquivada) e busca por slug/título; ações de publicar, despublicar, arquivar e excluir.
- **Editor** com os dados da LP (slug, título, modo de exibição, meta tags, Open Graph) e abas de variantes.
- **Modos de exibição**: `sequential` (round-robin), `random` (sorteio), `weighted` (por peso) e `manual` (sempre a variante principal).
- **Variantes**: nome, HTML/CSS/JS em editor de código com realce leve, peso, ordem, ativo/inativo, duplicar, definir como principal, importar HTML existente e preview em nova aba.
- **Upload de imagens** direto no editor; os arquivos vão para `data/uploads/lps/`.
- Uma LP só aparece publicamente quando está **publicada** e possui ao menos **uma variante ativa**.

7.4 Blog

- **Listagem** com filtros por status (rascunho, agendado, publicado, arquivado), categoria e busca.
- **Categorias e tags** gerenciadas na própria tela de listagem.

- **Editor**: título, slug gerado automaticamente, categoria, tags, resumo, imagem de destaque, meta tags/OG e um editor WYSIWYG em JS puro (negrito, itálico, títulos, listas, links, imagens, citação, código) com alternância para edição em HTML.
- **Ações**: salvar rascunho, publicar, agendar para uma data futura, despublicar e preview.
- Ao publicar, o HTML estático é gerado em `/blog/<slug>/index.html`, o índice `/blog/index.html` é atualizado e o post entra no `sitemap.xml`. Posts agendados são publicados pelo cron diário.
- **Regerar estáticos** reconstrói todos os arquivos do blog a partir do banco.

7.5 Usuários (somente `super_admin`)

- Criar usuário informando nome, e-mail e papel — o sistema devolve uma **senha temporária** exibida uma única vez; o usuário precisa trocá-la e configurar o 2FA no primeiro acesso.
- Editar papel, ativar/desativar, redefinir senha, resetar o 2FA e excluir.
- **Papéis**: `super_admin` (tudo), `editor` (LPs, blog, uploads, métricas) e `viewer` (somente leitura).
- **API Keys**: criar (nome + permissão `all`, `read_only` ou `metrics_only`), revogar e excluir. A chave é exibida **uma única vez** na criação; depois só restam o prefixo e o hash.

7.6 Configurações

- **Gerais**: nome do site, URL, descrição, logos, imagem OG padrão, WhatsApp, GTM, dados do blog, HTML customizado no `<head>` / `<body>`, tracker ligado/desligado, retenção dos logs MCP.
- **Páginas estáticas**: lista as páginas HTML da raiz e permite instalar o snippet do tracker em lote (cada página alterada é copiada antes para `data/backups/pages/`).
- **Sitemap**: regeração manual.
- **Backups**: executar backup agora, baixar e excluir arquivos.
- **Logs MCP**: últimas chamadas com método, status e IP, além da limpeza por retenção.
- **Sistema**: versões de PHP/SQLite, extensões, permissões de pastas e tamanho do banco.
- **Atividade recente**: trilha de auditoria das ações do painel.

8. Pixel tracker

`tracker/t.php` recebe uma requisição GET e devolve um GIF 1x1 transparente, registrando em `page_views`:

Campo	Origem
`page_path`	parâmetro `p` (normalizado)

`page_type`	deduzido do caminho: `lp`, `blog` ou `page`
`referer`	parâmetro `r` ou cabeçalho `Referer`
`ip_hash`	HMAC-SHA256 do IP com `IP_HASH_SALT` (LGPD — o IP nunca é gravado)
`device_type`	`desktop`, `mobile` ou `tablet`, do User-Agent
`user_agent`	User-Agent bruto

Bots conhecidos são descartados e qualquer erro é silenciado — o GIF é sempre entregue, nunca quebrando a página. As visitas às LPs são registradas pelo próprio `/lp/index.php` (com o `variant_id`), sem depender do pixel.

Snippet instalado antes de `</body>`:

```
<!-- LLEVON tracker -->
<script>
(function(){
    var img = new Image();
    img.src = '/tracker/t.php?p=' + encodeURIComponent(location.pathname)
        + '&r=' + encodeURIComponent(document.referrer)
        + '&t=' + Date.now();
})();
</script>
```

Em *Configurações → Páginas* o snippet pode ser instalado automaticamente em todas as páginas estáticas; o tracker pode ser pausado a qualquer momento pela configuração `tracker_enabled`.

9. API REST interna

Base: `/api/`. Respostas em JSON no formato `{"ok":true,"data":{...}}` ou `{"ok":false,"error":{"message":"..."}}`.

Autenticação: **sessão do painel** (cookie) + **token CSRF** no header `X-CSRF-Token` para qualquer método que altere dados. O token é obtido em `GET /api/auth.php?action=csrf`.

Endpoint	Ações principais
`auth.php`	`csrf`, `login`, `totp`, `setup`, `me`, `password`, `ping`, `logout`
`lps.php`	`list`, `get`, `create`, `update`, `delete`, `publish`, `unpublish`, `variant_create`, `variant_update`, `variant_toggle`, `variant_duplicate`, `variant_primary`, `variants_reorder`, `variant_import`, `upload_image`
`blog.php`	`list`, `get`, `create`, `update`, `delete`, `publish`, `schedule`, `unpublish`, `categories`, `category_save`, `category_delete`, `tags`, `upload_image`, `preview`, `regenerate`
`metrics.php`	`dashboard`, `summary`, `timeseries`, `by_type`, `top_pages`, `devices`, `sources`, `countries`, `lp_performance`, `lp_daily`
`site.php`	`pages`, `page_update`, `config`, `config_update`, `system`, `activity`, `users`, `user_create`, `user_update`, `user_reset_password`, `user_delete`, `api_keys`, `api_key_create`, `api_key_toggle`, `api_key_delete`, `mcp_logs`, `mcp_logs_purge`, `backups`, `backup_run`, `backup_download`, `backup_delete`, `tracker_install`, `sitemap`, `cache_clear`, `upload`

Exemplo:

```
curl -s "https://seusite.com.br/api/metrics.php?action=summary&preset=30d" \
  -H "Cookie: <sessão do painel>"
```

10. Interface MCP (controle por IA)

- **URL**: `https://seusite.com.br/api/mcp.php`
- **Método**: `POST` com `Content-Type: application/json`
- **Protocolo**: JSON-RPC 2.0 (padrão MCP)
- **Autenticação**: `Authorization: Bearer <api_key>` — chave criada em *Usuários → API Keys*
- **Rate limit**: 60 requisições por minuto por chave
- **HTTPS obrigatório** (localhost liberado para testes)

Métodos: `initialize`, `tools/list` e `tools/call`.

```
curl -X POST https://seusite.com.br/api/mcp.php \
  -H "Authorization: Bearer SUA_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

```
curl -X POST https://seusite.com.br/api/mcp.php \
  -H "Authorization: Bearer SUA_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","id":2,"method":"tools/call","params":
{"name":"create_landing_page","arguments":{"slug":"promo-verao","title":"Promoção de
Verão","display_mode":"random"}}}'
```

Ferramentas disponíveis

Landing Pages — `list_landing_pages`, `get_landing_page`, `create_landing_page`, `update_landing_page`, `delete_landing_page`, `publish_landing_page`, `create_lp_variant`, `update_lp_variant`, `toggle_lp_variant`, `delete_lp_variant`

Blog — `list_blog_posts`, `get_blog_post`, `create_blog_post`, `update_blog_post`, `delete_blog_post`, `publish_blog_post`, `list_blog_categories`, `manage_blog_category`

Métricas — `get_metrics_summary`, `get_page_views`, `get_lp_performance`, `get_traffic_sources`, `get_device_breakdown`, `get_top_pages`

Site — `get_site_pages`, `get_page_content`, `update_page_content`, `upload_image`, `regenerate_sitemap`, `get_site_config`, `update_site_config`

As permissões da chave valem para o MCP: `read_only` bloqueia as ferramentas de escrita e `metrics_only` libera apenas as de métricas. Todas as chamadas ficam registradas em `mcp_logs` (método, parâmetros sanitizados, status, IP) e podem ser consultadas em *Configurações → Logs MCP*.

11. Cron jobs

`backup.php` executa, em uma só passagem:

1. backup do `data/site.db` em `data/backups/site_AAAA-MM-DD.db` (mantém os 30 mais recentes);
2. publicação dos posts agendados cuja data já chegou;
3. regeração do `sitemap.xml`;
4. limpeza dos logs MCP fora da retenção configurada.

cPanel → Cron Jobs (uma vez por dia, de madrugada):

```
0 3 * * * /usr/local/bin/php /home/USUARIO/public_html/backup.php >> /home/USUARIO/cron-llevon.log 2>&1
```

Ajuste o caminho do binário PHP conforme a versão selecionada no cPanel (ex.: `/opt/cpanel/ea-php82/root/usr/bin/php`).

Se o cron por CLI não estiver disponível, a rotina pode ser chamada por URL — nesse caso é obrigatório um token:

```
https://seusite.com.br/backup.php?token=SEU_TOKEN
```

Defina o token como a constante `CRON_TOKEN` em `data/config.local.php` ou como a configuração `cron_token` (gravável pela ferramenta MCP `update_site_config`). Sem token configurado, a chamada por navegador responde **403**.

> Posts **agendados** só são publicados quando o cron roda. Sem cron diário, publique manualmente.

12. Backup e restauração

Backup

- Automático: diário, pelo cron (`data/backups/site_AAAA-MM-DD.db`, 30 cópias).
- Manual: *Configurações → Backups → Executar backup agora*, com download pelo painel.
- O backup usa `VACUUM INTO` quando disponível (cópia consistente mesmo com o site em uso).
- As páginas estáticas alteradas pelo painel são copiadas para `data/backups/pages/` antes da gravação.

Restauração

1. Coloque o painel fora do ar (ou avise a equipe).
2. Substitua `data/site.db` pelo arquivo de backup desejado e remova `site.db-wal` / `site.db-shm`, se existirem.
3. Ajuste as permissões (`644` para o arquivo, pasta gravável).
4. Em *Blog → Regerar estáticos* e *Configurações → Sitemap*, reconstrua os arquivos públicos.

Nunca restaure um `data/config.local.php` diferente do que estava em uso: a `ENCRYPTION_KEY` precisa ser a mesma que criptografou os segredos TOTP do banco.

13. Segurança

- **Senhas**: `password_hash()` com Bcrypt; política mínima de 10 caracteres com letras e números.
- **2FA TOTP** obrigatório, com os segredos criptografados (AES-256-CBC + HMAC) usando a `ENCRYPTION_KEY`.
- **Brute force**: 5 tentativas por IP e bloqueio de 15 minutos (tabela `login_attempts`).
- **Sessão**: cookie `HttpOnly` / `SameSite=Lax` (`Secure` sob HTTPS), expiração por inatividade de 30 minutos e regeneração do ID a cada elevação de privilégio.
- **CSRF**: token obrigatório em todas as requisições de escrita do painel.
- **LGPD**: o IP do visitante nunca é armazenado — apenas o HMAC-SHA256 com `IP_HASH_SALT`.
- **.htaccess**: `/data/` bloqueado (exceto imagens de upload), `/admin/includes/` bloqueado, `/admin/` com HTTPS forçado e headers `X-Content-Type-Options`, `X-Frame-Options`, `X-XSS-Protection` e `Referrer-Policy`.
- **Uploads**: apenas imagens `jpg`, `png`, `gif` e `webp` (tipo real verificado com `getimagesize`, não pela extensão), limite de 8 MB, nome higienizado e redimensionamento para 1920px (com GD).
- **API Keys**: armazenadas como SHA256, exibidas uma única vez, revogáveis, com rate limit e log.
- **Auditoria**: `activity_log` registra autor, ação, entidade e data de cada operação do painel.

Checklist pós-instalação: `install.php` removido, HTTPS ativo, `data/site.db` inacessível pela web, cron configurado e segredo TOTP do Super Admin guardado em local seguro.

14. Troubleshooting

Sintoma	Causa provável / solução

"ENCRYPTION_KEY não configurada"	`data/config.local.php` não existe ou não é legível. Rode o `install.php` (em instalação nova) ou restaure o arquivo original do servidor.
Erro 500 no painel	Veja o `error_log` do cPanel. Normalmente é versão de PHP < 7.4, extensão `pdo_sqlite` desativada ou `data/` sem permissão de escrita.
"Não foi possível gravar no banco"	A pasta `data/` precisa ser gravável pelo usuário do PHP (o SQLite cria `site.db-wal` / `site.db-shm` ao lado do banco).
Código 2FA sempre inválido	Relógio do servidor fora de hora. Confirme o fuso (`America/Sao_Paulo`) e sincronize o horário; o sistema aceita ±1 janela de 30s.
Perdi o acesso ao 2FA	Outro `super_admin` pode resetar o 2FA em *Usuários*. Sem nenhum acesso, apague `data/install.lock`, reenvie o `install.php` e recrie o Super Admin — **nunca apague** `config.local.php`.
Bloqueado por tentativas de login	Aguarde 15 minutos ou limpe a tabela `login_attempts` no banco.
LP retorna 404	A LP precisa estar **publicada** e ter ao menos uma variante **ativa**. Verifique também se o `mod_rewrite` está ativo e se `lp/.htaccess` foi enviado.
URLs do blog dão 404	Os arquivos estáticos ainda não foram gerados: use *Blog → Regerar estáticos*. Confirme que a raiz do site é gravável.
Dashboard zerado	(1) O tracker não está instalado nas páginas — use *Configurações → Páginas*; (2) `tracker_enabled` está em `0` ; (3) cache de 5 minutos — clique em **Atualizar**. Acessos de bots e via `curl` são ignorados de propósito.
Gráficos não aparecem	`admin/assets/js/chart.umd.min.js` não foi enviado ou está bloqueado. Verifique o console do navegador.
MCP responde 401/403	Chave inválida, revogada, sem permissão para a ferramenta chamada, ou requisição em HTTP (o endpoint exige HTTPS).
MCP responde 429	Rate limit de 60 req/min por chave atingido.
Posts agendados não publicam	O cron diário não está rodando — veja a seção 11.
Upload de imagem falha	Arquivo acima de 8 MB, extensão não permitida, `data/uploads/` sem permissão ou `upload_max_filesize` / `post_max_size` baixos no PHP.
Sessão caindo o tempo todo	Expiração de 30 minutos de inatividade (`SESSION_TIMEOUT` em `data/config.php`) ou sessão do PHP sem pasta gravável no servidor.

Convenções do código

- PHP 7.4 compatível, `declare(strict_types=1)` nos arquivos novos, PDO com prepared statements.
- JS e CSS em vanilla, sem build; os helpers do painel ficam no objeto global `LLV` (`admin/assets/admin.js`).

- Todas as mensagens, labels e comentários em português do Brasil.