

# Sistema de Monitoramento Hidrológico CEMADEN

Sistema automatizado de coleta, processamento e monitoramento de dados hidrológicos das estações CEMADEN com sistema de alertas inteligente.

## Índice

- [Características](#)
- [Requisitos](#)
- [Instalação](#)
- [Configuração](#)
- [Uso](#)
- [Estrutura do Banco de Dados](#)
- [Monitoramento](#)
- [Manutenção](#)

## Características

### Funcionalidades Principais

- **✓ Coleta automatizada** de múltiplas APIs CEMADEN
- **✓ Processamento em lote** para alta performance
- **✓ Sistema de retry** automático para requisições
- **✓ Detecção de duplicidade** otimizada (memória + banco)
- **✓ Alertas inteligentes** em 3 níveis (atenção, alerta, transbordamento)
- **✓ Cotas personalizadas** por estação
- **✓ Logs estruturados** com múltiplos níveis
- **✓ Histórico de execuções** com estatísticas
- **✓ Validação robusta** de dados
- **✓ Índices otimizados** para consultas rápidas

### Melhorias Implementadas

#### Arquitetura

- Código orientado a objetos com separação de responsabilidades
- Padrão Repository para acesso a dados

- Services para lógica de negócio
- Configurações centralizadas

## Performance

- Inserções em lote (batch insert)
- Prepared statements reutilizáveis
- Cache de cotas em memória
- Índices compostos no banco

## Confiabilidade

- Sistema de retry com backoff
- Tratamento robusto de erros
- Validação de dados em múltiplos níveis
- Logs detalhados para debugging

## Monitoramento

- Estatísticas de execução
- Histórico de alertas
- Views para consultas rápidas
- Stored procedures para manutenção

## Requisitos

### Sistema

- PHP 7.4 ou superior
- MySQL 5.7 ou superior / MariaDB 10.3+
- Extensões PHP:
  - PDO
  - pdo\_mysql
  - curl
  - json
  - mbstring

## Opcional (mas recomendado)

- Composer (para Monolog e outras dependências)
- Cron ou similar para execução automática
- Servidor SMTP para envio de alertas

## Instalação

### 1. Clone ou baixe os arquivos

```
bash  
  
mkdir /var/www/cemaden  
cd /var/www/cemaden
```

### 2. Instale dependências (se usar Composer)

```
bash  
  
composer require monolog/monolog
```

Ou use a versão standalone sem dependências externas.

### 3. Configure permissões

```
bash  
  
chmod 755 hidrologicocemadem.php  
mkdir -p logs  
chmod 777 logs
```

### 4. Crie o banco de dados

```
bash  
  
mysql -u root -p
```

```
sql  
  
CREATE DATABASE cemaden_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE USER 'cemaden_user'@'localhost' IDENTIFIED BY 'senha_segura';  
GRANT ALL PRIVILEGES ON cemaden_db.* TO 'cemaden_user'@'localhost';  
FLUSH PRIVILEGES;
```

### 5. Execute os scripts SQL

```
bash
```

```
mysql -u cemaden_user -p cemaden_db < migrations.sql
```

## ⚙️ Configuração

### 1. Configure variáveis de ambiente

Copie o arquivo `.env.example` para `.env` e ajuste:

```
bash
```

```
cp .env.example .env  
nano .env
```

### 2. Configure o banco de dados

Edito `config/configbd.php`:

```
php
```

```
class Database {  
    private static $host = 'localhost';  
    private static $db = 'cemaden_db';  
    private static $user = 'cemaden_user';  
    private static $pass = 'sua_senha';  
    // ...  
}
```

### 3. Configure estações

Edito as cotas específicas de cada estação no banco:

```
sql
```

```
INSERT INTO estacoes_config (codigo_estacao, cota_atencao, cota_alerta, cota_transbordamento)  
VALUES ('3121', 40.0, 60.0, 80.0);
```

### 4. Configure alertas

Edito a classe `Config` em `hidrologicocemadem.php`:

```
php
```

```
private static $alertRecipients = [  
    'default' => ['monitor@example.com'],  
    'critical' => ['emergencia@example.com', 'diretor@example.com']  
];
```

## 🔧 Uso

## Execução Manual

```
bash  
php hidrologicocemadem.php
```

## Execução Automática (Cron)

Edite o crontab:

```
bash  
crontab -e
```

Adicione (executa a cada hora):

```
bash  
0 * * * * /usr/bin/php /var/www/cemaden/hidrologicocemadem.php >> /var/www/cemaden/logs/cron.log 2>&1
```

Exemplos de frequência:

```
bash  
# A cada 30 minutos  
*/30 * * * * /usr/bin/php /caminho/script.php  
  
# A cada 15 minutos  
*/15 * * * * /usr/bin/php /caminho/script.php  
  
# A cada 6 horas  
0 */6 * * * /usr/bin/php /caminho/script.php
```

## Verificar Logs

```
bash  
# Log de aplicação  
tail -f logs/app.log  
  
# Log de erros  
tail -f logs/errors.log  
  
# Log do cron  
tail -f logs/cron.log
```

## Estrutura do Banco de Dados

### Tabelas Principais

#### **leituras\_cemaden**

Armazena todas as leituras das estações.

#### **estacoes\_config**

Configurações específicas de cada estação (cotas, offset).

#### **execucoes\_log**

Histórico de execuções do script com estatísticas.

#### **alertas\_enviados**

Registro de todos os alertas enviados.

### Views

#### **v\_ultimas\_leituras**

Última leitura de cada estação com status atual.

sql

```
SELECT * FROM v_ultimas_leituras WHERE status_atual = 'ALERTA';
```

#### **v\_estatisticas\_diarias**

Estatísticas agregadas por dia/estação.

sql

```
SELECT * FROM v_estatisticas_diarias WHERE data_leitura = CURDATE();
```

## Monitoramento

### Consultas Úteis

#### Execuções recentes:

sql

```
SELECT
    data_execucao,
    status,
    registros_inseridos,
    tempo_execucao,
    JSON_LENGTH(erros) as qtd_erros
FROM execucoes_log
ORDER BY data_execucao DESC
LIMIT 10;
```

## Estações em alerta:

```
sql

SELECT
    codigo_estacao,
    estacao_nome,
    cidade_nome,
    valor,
    status_atual,
    data_hora_completa
FROM v_ultimas_leituras
WHERE status_atual IN ('ALERTA', 'TRANSBORDAMENTO')
ORDER BY valor DESC;
```

## Performance do sistema:

```
sql

SELECT
    DATE(data_execucao) as data,
    COUNT(*) as execucoes,
    AVG(tempo_execucao) as tempo_medio,
    SUM(registros_inseridos) as total_inseridos,
    SUM(CASE WHEN status = 'falha' THEN 1 ELSE 0 END) as falhas
FROM execucoes_log
WHERE data_execucao >= DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY DATE(data_execucao)
ORDER BY data DESC;
```

## Histórico de alertas:

```
sql
```

```

SELECT
    DATE(enviado_em) as data,
    tipo_alerta,
    COUNT(*) as quantidade
FROM alertas_enviados
WHERE enviado_em >= DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY DATE(enviado_em), tipo_alerta
ORDER BY data DESC;

```

## Dashboard Simples

Crie um arquivo `dashboard.php`:

```

php

<?php
require_once 'config/configbd.php';
$pdo = Database::getConnection();

// Última execução
$stmt = $pdo->query("SELECT * FROM execucoes_log ORDER BY data_execucao DESC LIMIT 1");
$ultimaExecucao = $stmt->fetch(PDO::FETCH_ASSOC);

// Estações em alerta
$stmt = $pdo->query("SELECT COUNT(*) FROM v_ultimas_leituras WHERE status_atual != 'NORMAL'");
$estacoesAlerta = $stmt->fetchColumn();

echo "Status do Sistema CEMADEN\n";
echo "=====\\n\\n";
echo "Última Execução: {$ultimaExecucao['data_execucao']}\\n";
echo "Status: {$ultimaExecucao['status']}\\n";
echo "Registros Inseridos: {$ultimaExecucao['registros_inseridos']}\\n";
echo "Tempo: {$ultimaExecucao['tempo_execucao']} s\\n\\n";
echo "Estações em Alerta: $estacoesAlerta\\n";

```

## 🔧 Manutenção

### Limpeza de Dados Antigos

Execute mensalmente para manter apenas últimos 90 dias:

```

sql

CALL sp_limpar_dados_antigos(90);

```

Ou via cron (todo dia 1º do mês):

```
bash
```

```
0 2 1 * * mysql -u user -p database -e "CALL sp_limpar_dados_antigos(90)"
```

## Otimização de Tabelas

```
sql
```

```
OPTIMIZE TABLE leituras_cemaden;  
OPTIMIZE TABLE execucoes_log;  
OPTIMIZE TABLE alertas_enviados;
```

## Backup Automático

```
bash
```

```
#!/bin/bash  
# backup_cemaden.sh  
  
DATA=$(date +%Y%m%d)  
mysqldump -u cemaden_user -p'senha' cemaden_db | gzip > backup_cemaden_$DATA.sql.gz  
  
# Manter apenas últimos 30 dias  
find /caminho/backups -name "backup_cemaden_*.sql.gz" -mtime +30 -delete
```

Adicione ao cron:

```
bash
```

```
0 3 * * * /caminho/backup_cemaden.sh
```

## Troubleshooting

### Script não executa

1. Verifique permissões:

```
bash
```

```
ls -la hidrologicocemadem.php
```

2. Teste execução manual:

```
bash
```

```
php -f hidrologicocemadem.php
```

### 3. Verifique logs:

```
bash
```

```
tail -f logs/errors.log
```

## Sem inserções no banco

### 1. Verifique conectividade com APIs:

```
bash
```

```
curl -v "https://mapservices.cemaden.gov.br/MapaInterativoWS/resources/horario/3121/8"
```

### 2. Verifique tabela execucoes\_log:

```
sql
```

```
SELECT * FROM execucoes_log ORDER BY data_execucao DESC LIMIT 5;
```

## Alertas não são enviados

### 1. Verifique configuração SMTP

### 2. Teste função de email manualmente

### 3. Verifique logs de alertas:

```
sql
```

```
SELECT * FROM alertas_enviados ORDER BY enviado_em DESC LIMIT 10;
```

## Licença

Sistema desenvolvido para monitoramento interno. Ajuste conforme necessário.

## Contribuição

Para melhorias ou correções, documente as mudanças e atualize este README.

## Suporte

Para dúvidas ou problemas:

- Verifique os logs em [/logs](#)
- Consulte a tabela [execucoes\\_log](#)
- Revise as configurações em [.env](#)

