Guia Técnico Completo - Estatística de Visitação com Docker e Banco de Dados

# Arquitetura (visão rápida)



# 2) Estrutura de diretórios

Finalidade: Organização clara do projeto para manutenção.

 web → contém o site e scripts de rastreio.

 db → scripts de inicialização do banco.

 ops → rotinas de backup e automação.

project-root/  
├─ docker-compose.yml  
├─ .env  
├─ web/  
│ ├─ Dockerfile  
│ ├─ vhost/000-default.conf  
│ ├─ analytics/  
│ │ ├─ config/myphp\_config.php  
│ │ └─ track.php  
│ └─ www/  
│ └─ index.html  
├─ db/  
│ └─ init/001\_schema.sql  
└─ ops/  
 ├─ backup.sh  
 └─ crontab  
3) Arquivo .env

Finalidade:Centralizar senhas, portas e parâmetros (boa prática de DevOps).

# WEB  
WEB\_PORT=8080  
PHPMYADMIN\_PORT=8081  
DB\_ROOT\_PASSWORD=super\_root\_secret  
DB\_NAME=analytics\_db  
DB\_USER=analytics\_user  
DB\_PASSWORD=analytics\_pass  
HASH\_SALT=troque-este-salt-robusto

# 4) docker-compose.yml

**Finalidade:** Orquestrar os containers (subir tudo com um único comando).

Com esse arquivo, evitamos instalar manualmente PHP, Apache e MariaDB em servidores diferentes. Basta docker compose up.

version: "3.9"  
  
services:  
 web:  
 build:  
 context: ./web  
 dockerfile: Dockerfile  
 container\_name: analytics\_web  
 ports:  
 - "${WEB\_PORT}:80"  
 volumes:  
 - ./web/www:/var/www/html:ro  
 - ./web/analytics:/var/www/html/analytics:ro  
 - ./web/vhost/000-default.conf:/etc/apache2/sites-available/000-default.conf:ro  
 environment:  
 HASH\_SALT: ${HASH\_SALT}  
 PHP\_MEMORY\_LIMIT: 128M  
 depends\_on:  
 - db  
 restart: unless-stopped  
  
 db:  
 image: mariadb:10.11  
 container\_name: analytics\_db  
 environment:  
 MARIADB\_ROOT\_PASSWORD: ${DB\_ROOT\_PASSWORD}  
 MARIADB\_DATABASE: ${DB\_NAME}  
 MARIADB\_USER: ${DB\_USER}  
 MARIADB\_PASSWORD: ${DB\_PASSWORD}  
 volumes:  
 - db\_data:/var/lib/mysql  
 - ./db/init:/docker-entrypoint-initdb.d:ro  
 restart: unless-stopped  
  
 phpmyadmin:  
 image: phpmyadmin/phpmyadmin:latest  
 container\_name: analytics\_pma  
 environment:  
 PMA\_HOST: db  
 PMA\_USER: ${DB\_USER}  
 PMA\_PASSWORD: ${DB\_PASSWORD}  
 ports:  
 - "${PHPMYADMIN\_PORT}:80"  
 depends\_on:  
 - db  
 restart: unless-stopped  
volumes:  
 db\_data:

# 5) Dockerfile do web

**Finalidade:** Configurar Apache + PHP de forma endurecida (desabilitar funções perigosas, ativar módulos certos).

Evita brechas de segurança comuns em hospedagens compartilhadas.

FROM php:8.2-apache  
RUN docker-php-ext-install pdo pdo\_mysql  
RUN { echo "expose\_php=0"; echo "disable\_functions=exec,passthru,shell\_exec,system,proc\_open,popen,curl\_multi\_exec,parse\_ini\_file,show\_source"; echo "memory\_limit=${PHP\_MEMORY\_LIMIT:-128M}"; } > /usr/local/etc/php/conf.d/custom.ini  
RUN a2enmod rewrite headers  
RUN chown -R www-data:www-data /var/www/html  
USER www-data

# 6) VirtualHost do Apache

**Finalidade:** Controlar headers de segurança e impedir listagem de diretórios.

<VirtualHost \*:80>  
 ServerAdmin admin@localhost  
 DocumentRoot /var/www/html  
 <Directory "/var/www/html">  
 AllowOverride All  
 Require all granted  
 Options -Indexes  
 </Directory>  
 Header always set X-Content-Type-Options "nosniff"  
 Header always set X-Frame-Options "SAMEORIGIN"  
 Header always set Referrer-Policy "strict-origin-when-cross-origin"  
 Header always set Content-Security-Policy "default-src 'self' 'unsafe-inline' 'unsafe-eval' data: https:"  
 ErrorLog ${APACHE\_LOG\_DIR}/error.log  
 CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
</VirtualHost>

# 7) Esquema SQL inicial

**Finalidade: Estruturar onde as visitas ficam registradas.**

CREATE TABLE IF NOT EXISTS t\_visitas (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 pagina VARCHAR(255) NOT NULL,  
 ip\_hash CHAR(64) NOT NULL,  
 user\_agent VARCHAR(512) NOT NULL,  
 referrer VARCHAR(512) NULL,  
 device VARCHAR(32) NULL,  
 navegador VARCHAR(128) NULL,  
 pais VARCHAR(64) NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 KEY idx\_pagina (pagina),  
 KEY idx\_created\_at (created\_at),  
 KEY idx\_device (device)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

# 8) Configuração myphp\_config.php

**Finalidade: Abstrair a conexão PDO com segurança.**

**Centralizar a conexão permite mudar credenciais sem mexer no track.php.**

<?php  
$host = "db";  
$dbname = getenv("DB\_NAME") ?: "analytics\_db";  
$user = getenv("DB\_USER") ?: "analytics\_user";  
$pass = getenv("DB\_PASSWORD") ?: "analytics\_pass";  
$charset = "utf8mb4";  
$dsn = "mysql:host=$host;dbname=$dbname;charset=$charset";  
$options = [  
 PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,  
 PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,  
 PDO::ATTR\_EMULATE\_PREPARES => false,  
];  
try {  
 $pdo = new PDO($dsn, $user, $pass, $options);  
} catch (PDOException $e) {  
 http\_response\_code(500);  
 die("DB connection failed");  
}  
?>

# 9) Coletor track.php

**Finalidade: Capturar e registrar a visita (IP anon, UA, página).**

**Ele é o coração do sistema — cada visita dispara esse endpoint.**

<?php  
header('Content-Type: image/gif');  
header('Cache-Control: no-store, no-cache, must-revalidate, max-age=0');  
header('Pragma: no-cache');  
function tiny\_gif(){ echo base64\_decode('R0lGODlhAQABAPAAAP///wAAACH5BAAAAAAALAAAAAABAAEAAAICRAEAOw=='); }  
$ua = $\_SERVER['HTTP\_USER\_AGENT'] ?? '';  
$page = $\_GET['page'] ?? ($\_SERVER['REQUEST\_URI'] ?? '/');  
$referrer = $\_SERVER['HTTP\_REFERER'] ?? null;  
$ip = $\_SERVER['REMOTE\_ADDR'] ?? '0.0.0.0';  
$device = 'desktop';  
if (preg\_match('/mobile|android|iphone|ipad/i', $ua)) $device = 'mobile';  
$naveg = 'other';  
foreach (['Chrome','Firefox','Safari','Edge'] as $b) {  
 if (stripos($ua, $b) !== false) { $naveg = $b; break; }  
}  
$salt = getenv('HASH\_SALT') ?: 'change-me';  
$ip\_hash = hash('sha256', $ip.$salt);  
require \_\_DIR\_\_.'/config/myphp\_config.php';  
try {  
 $stmt = $pdo->prepare(  
 "INSERT INTO t\_visitas (pagina, ip\_hash, user\_agent, referrer, device, navegador)  
 VALUES (?, ?, ?, ?, ?, ?)"  
 );  
 $stmt->execute([substr($page,0,255), $ip\_hash, $ua, $referrer, $device, $naveg]);  
} catch (Throwable $e) {}  
tiny\_gif();  
?>

# 10) Snippet no HTML

Finalidade: Inserir no site o código que chama o track.php.

<img src="/analytics/track.php?page={ENCODED\_PATH}" alt="" width="1" height="1" style="display:none">  
<script>  
 (function(){  
 var p = encodeURIComponent(location.pathname + location.search);  
 var el = document.querySelector('img[src^="/analytics/track.php"]');  
 if (el) el.src = "/analytics/track.php?page=" + p;  
 })();  
</script>

<script>  
 (function(){  
 var p = encodeURIComponent(location.pathname + location.search);  
 fetch("/analytics/track.php?page=" + p, {cache:"no-store",mode:"no-cors"});  
 })();  
</script>