<?php

// Ruta de la carpeta donde se encuentran las entradas (archivos HTML de cada entrada)

$blogDir = \_\_DIR\_\_ . "/../";  // Asume que las entradas están en public\_html/blog/

// Ruta al archivo entries.json

$entriesJsonPath = \_\_DIR\_\_ . '/entries.json';

$entriesPerPage = 6;

// Verificar si el archivo entries.json existe, si no, crearlo vacío

if (!file\_exists($entriesJsonPath)) {

    file\_put\_contents($entriesJsonPath, "[]");

}

// Buscar todos los archivos HTML en la carpeta que no sean los de paginación ni el blog principal

$files = glob($blogDir . "\*.html");

$entries = [];

foreach ($files as $file) {

    $filename = basename($file);

    // Ignorar archivos de paginación y blog.html

    if (stripos($filename, 'blog.html') !== false || stripos($filename, 'page') === 0) {

        continue;

    }

    $url = "/blog/" . $filename;  // URL relativa de la entrada

    $creationTime = filectime($file); // Fecha de creación del archivo

    // Leer el contenido del archivo HTML de la entrada

    $htmlContent = file\_get\_contents($file);

    // ✅ Extraer el <h1> dentro de <div class="cabecera-blog-adaptable">

    preg\_match('/<div class="cabecera-blog-adaptable">.\*?<h1[^>]\*>(.\*?)<\/h1>/si', $htmlContent, $titleMatch);

    $title = isset($titleMatch[1]) ? '<h2 class="mini-entrada-titulo">' . trim($titleMatch[1]) . '</h2>' : '<h2 class="mini-entrada-titulo">Sin título</h2>';

    // ✅ Extraer la primera imagen dentro del primer <div class="contenido-blog">

    preg\_match('/<div class="contenido-blog-container">.\*?<div class="contenido-blog">.\*?<img[^>]+src=["\'](.\*?)["\']/si', $htmlContent, $imageMatch);

    $image = isset($imageMatch[1]) ? trim($imageMatch[1]) : "/images/default.jpg";

    // ✅ Extraer el primer párrafo dentro del primer <div class="contenido-blog">

    preg\_match('/<div class="contenido-blog-container">.\*?<div class="contenido-blog">.\*?<p[^>]\*>(.\*?)<\/p>/si', $htmlContent, $excerptMatch);

    $excerpt = isset($excerptMatch[1]) ? strip\_tags($excerptMatch[1], '<i><em>') : "Sin descripción";

    // Agregar la entrada al array, incluyendo la fecha de creación para ordenar

    $entries[] = [

        "image"      => $image,

        "title"      => $title,  // Se almacena como <h2>

        "link"       => $url,

        "excerpt"    => $excerpt,

        "created\_at" => $creationTime

    ];

}

// Ordenar las entradas de más reciente a más antigua

usort($entries, function ($a, $b) {

    return $b['created\_at'] - $a['created\_at'];

});

// Eliminar el campo created\_at antes de guardar en JSON

foreach ($entries as &$entry) {

    unset($entry['created\_at']);

}

// Evitar escribir en disco si los datos no han cambiado

$prevEntries = file\_exists($entriesJsonPath) ? json\_decode(file\_get\_contents($entriesJsonPath), true) : [];

if ($prevEntries !== $entries) {

    file\_put\_contents($entriesJsonPath, json\_encode($entries, JSON\_PRETTY\_PRINT | JSON\_UNESCAPED\_UNICODE));

}

// Calcular el total de páginas necesarias

$totalPages = ceil(count($entries) / $entriesPerPage);

// 📌 CREAR `blog.html` AUTOMÁTICAMENTE

$blogMainPage = \_\_DIR\_\_ . "/../blog.html";

$latestEntries = array\_slice($entries, 0, $entriesPerPage);

$blogContent = '<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Blog de Ayurveda Tradicional</title>

  <link rel="stylesheet" href="/css/styles.css">

  <link rel="stylesheet" href="/css/blog.css">

</head>

<body>

<main>

  <div class="cabecera-paginas">

    <img id="cabecera-img" src="/images/blog-de-ayurveda-tradicional.webp" alt="Blog de Ayurveda Tradicional">

    <h1>Blog de Ayurveda Tradicional</h1>

  </div>

  <div class="distribucion-blog">

    <div class="contenido-blog-container">

      <div class="content-container">';

foreach ($latestEntries as $entry) {

    $blogContent .= '<div class="content-box">

      <img src="' . $entry['image'] . '" alt="' . htmlspecialchars(strip\_tags($entry['title'])) . '">

      ' . $entry['title'] . '

      <p>' . $entry['excerpt'] . '</p>

      <a href="' . $entry['link'] . '" class="entry-button">Leer más</a>

    </div>';

}

// Agregar el contenedor de paginación

$blogContent .= '</div>

      <div id="pagination-container"></div>

    </div>

  </main>

  <script src="/blog/blog-js/selector.js"></script>

</body>

</html>';

file\_put\_contents($blogMainPage, $blogContent);

// 📌 CREAR LAS PÁGINAS `page1.html`, `page2.html`, etc.

for ($i = 1; $i <= $totalPages; $i++) {

    $pageName = \_\_DIR\_\_ . "/../page{$i}.html";

    $pageEntries = array\_slice($entries, ($i - 1) \* $entriesPerPage, $entriesPerPage);

    $content = '<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Blog de Ayurveda Tradicional - Página ' . $i . '</title>

  <link rel="stylesheet" href="/css/styles.css">

  <link rel="stylesheet" href="/css/blog.css">

</head>

<body>

<main>

  <div class="cabecera-paginas">

    <img id="cabecera-img" src="/images/blog-de-ayurveda-tradicional.webp" alt="Blog de Ayurveda Tradicional">

    <h1>Blog de Ayurveda Tradicional - Página ' . $i . '</h1>

  </div>

  <div class="distribucion-blog">

    <div class="contenido-blog-container">

      <div class="content-container">';

    foreach ($pageEntries as $entry) {

        $content .= '<div class="content-box">

          <img src="' . $entry['image'] . '" alt="' . htmlspecialchars(strip\_tags($entry['title'])) . '">

          ' . $entry['title'] . '

          <p>' . $entry['excerpt'] . '</p>

          <a href="' . $entry['link'] . '" class="entry-button">Leer más</a>

        </div>';

    }

    // Agregar el contenedor de paginación en cada página

    $content .= '</div>

      <div id="pagination-container"></div>

    </div>

  </main>

  <script src="/blog/blog-js/selector.js"></script>

</body>

</html>';

    file\_put\_contents($pageName, $content);

}

echo json\_encode(["status" => "success", "message" => "Entradas detectadas y páginas actualizadas correctamente."]);

?>