Aquí tienes un resumen con toda la información para implantar el contador en local (por ejemplo, en XAMPP):

**1. Estructura de Archivos**

* **Raíz del sitio (dentro de htdocs en XAMPP):**
  + index.html (o la página principal)
  + contadores.json
    - **Ubicación:** Raíz
    - **Función:** Guarda los datos de cada post (visitas, likes y fecha de creación).
    - **Creación:** Se crea automáticamente si no existe (si PHP tiene permisos de escritura).
    - **Nota:** Si hay problemas, puedes crear el archivo manualmente con contenido:
    - {}
* **Carpeta /php/:**
* contador.php
  + **Función:** Lee y actualiza el JSON, incrementa visitas y likes. Se ejecuta al recibir solicitudes desde el JavaScript.
* **Carpeta /js/:**
* contador-visitas.js
  + **Función:** Detecta el nombre del post a través de la URL, solicita datos a contador.php y actualiza el HTML. También maneja el evento de "like" (clic en el icono de corazón).
* **Carpeta /blog/:**
* Archivos HTML de tus posts (por ejemplo, menopausia-y-ayurveda o menopausia-y-ayurveda.html).

### 2. Código en el HTML (para Visualizar los Contadores)

Dentro de cada post, añade el siguiente código para mostrar la fecha, visitas y likes alineados a la derecha:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Ejemplo de Post</title>

<script src="/js/contador-visitas.js" defer></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

<style>

.info-container {

display: flex;

justify-content: flex-end; /\* Alinea a la derecha \*/

align-items: center;

gap: 20px; /\* Espacio entre elementos \*/

font-size: 16px;

color: #333;

margin-bottom: 15px;

}

.info-container span {

display: flex;

align-items: center;

gap: 5px;

}

.info-container i {

color: #666;

cursor: pointer;

}

#like-heart {

color: #e74c3c;

font-size: 18px;

cursor: pointer;

}

#like-heart:hover {

color: #c0392b;

}

</style>

</head>

<body>

<h1>Ejemplo de Post</h1>

<!-- Contenedor para mostrar fecha, visitas y el icono de like (corazón) -->

<div class="info-container">

<span><i class="fa fa-calendar"></i> <span id="fecha\_creacion"></span></span>

<span><i class="fa fa-eye"></i> <span id="contador"></span> visitas</span>

<span><i class="fa fa-heart" id="like-heart"></i> <span id="like-count">0</span></span>

</div>

</body>

</html>

**3. Archivo PHP: /php/contador.php**

Este archivo se ejecuta cada vez que el JavaScript hace una solicitud. Hace lo siguiente:

* **Lee o crea** el archivo contadores.json (ubicado en la raíz).
* **Detecta el post** mediante el parámetro post en la URL.
* **Si el post no existe** en el JSON, lo inicializa con:
  + visitas: 0
  + likes: 0
  + fecha\_creacion: Fecha actual en formato DD/MM/YYYY.
* **Si se envía** el parámetro like=1, incrementa los likes y devuelve ese dato.
* **Si no se envía like**, incrementa las visitas y devuelve todos los datos del post en formato JSON.

**Código:**

<?php

$archivo = '../contadores.json'; // Ubicación del JSON en la raíz

// Si el archivo no existe, lo crea con un JSON vacío

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

file\_put\_contents($archivo, '{}');

}

if (!isset($\_GET['post'])) {

die(json\_encode(["error" => "No se ha especificado un post."]));

}

$slug = preg\_replace('/[^a-zA-Z0-9\_-]/', '', $\_GET['post']);

$datos = json\_decode(file\_get\_contents($archivo), true);

if (!isset($datos[$slug])) {

$datos[$slug] = [

"visitas" => 0,

"likes" => 0,

"fecha\_creacion" => date("d/m/Y")

];

}

if (isset($\_GET['like']) && $\_GET['like'] == "1") {

$datos[$slug]["likes"]++;

file\_put\_contents($archivo, json\_encode($datos, JSON\_PRETTY\_PRINT));

echo json\_encode(["likes" => $datos[$slug]["likes"]]);

exit;

}

$datos[$slug]["visitas"]++;

file\_put\_contents($archivo, json\_encode($datos, JSON\_PRETTY\_PRINT));

echo json\_encode($datos[$slug]);

?>

**4. Archivo JavaScript: /js/contador-visitas.js**

Este script:

* **Detecta el nombre del post** usando la URL (elimina la extensión .html si está presente).
* **Solicita datos** a contador.php y actualiza el HTML.
* **Maneja el evento de like** (al hacer clic en el icono de corazón).

**Código:**

document.addEventListener("DOMContentLoaded", function () {

let postSlug = window.location.pathname.split('/').pop();

postSlug = postSlug.replace(/\.html$/, '');

fetch('/php/contador.php?post=' + postSlug)

.then(response => response.json())

.then(data => {

document.getElementById('contador').innerText = data.visitas;

document.getElementById('fecha\_creacion').innerText = data.fecha\_creacion;

document.getElementById('like-count').innerText = data.likes;

})

.catch(error => {

console.error("Error al cargar el contador: ", error);

document.getElementById('contador').innerText = "Error";

document.getElementById('fecha\_creacion').innerText = "Desconocida";

document.getElementById('like-count').innerText = "Error";

});

document.getElementById('like-heart').addEventListener("click", function () {

fetch('/php/contador.php?post=' + postSlug + '&like=1')

.then(response => response.json())

.then(data => {

document.getElementById('like-count').innerText = data.likes;

})

.catch(error => console.error("Error al actualizar los likes: ", error));

});

});

**5. Ubicación y Creación del Archivo JSON**

* **Ubicación:** contadores.json debe estar en la raíz del sitio (al mismo nivel que index.html).
* **Creación:**
  + El código PHP intenta crearlo automáticamente si no existe.
  + Si PHP no tiene permisos, crea el archivo manualmente con el contenido: {}

**6. Ejecución en Local con XAMPP**

1. Copia toda la estructura de carpetas en la carpeta htdocs de XAMPP.
2. Asegúrate de que Apache esté iniciado.
3. Accede a http://localhost/tu-carpeta/ para ver el sitio.
4. Abre un post para ver cómo se actualiza el archivo contadores.json y se muestran visitas, fecha y likes en el HTML.
5. Al hacer clic en el icono de corazón, se sumarán los likes en tiempo real.

Con esta información podrás implantar y probar el contador en local usando XAMPP. Si tienes alguna duda o necesitas más ajustes, ¡avísame!