

**DISEÑO Y DESARROLLO DE SERVICIOS WEB-PROYECTO
GA7-220501096-AA5-EV01**

NOMBRE DEL APRENDIZ

Carlos Arturo Santos Castellar

Instructor

SERGIO ESTEBAN BERNAL CASTILLO

FICHA 3070302

Servicio Nacional de Aprendizaje - SENA

Centro de Materiales y Ensayos - CME

Análisis y Desarrollo de Software – ADSO

2025

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	3
2. OBJETIVO.....	4
3. OBJETIVOS ESPECÍFICOS.....	4
4. DESARROLLO DE EVIDENCIA.....	4
5. DISEÑO Y DESARROLLO DE SERVICIOS WEB-PROYECTO.....	4
6. CONCLUSIONES.....	7
7. REFERENCIAS.....	7

INTRODUCCIÓN

En esta evidencia desarrollé un servicio web que permite el registro y la autenticación de usuarios, integrando un frontend construido con HTML, CSS y JavaScript con un backend implementado en PHP y un archivo JSON como almacenamiento temporal. Entender el flujo de comunicación (cliente, servidor) mediante peticiones HTTP y validar la correcta gestión de datos en formato JSON.

OBJETIVO

Implementar un servicio web que gestione el registro y la autenticación de usuarios utilizando PHP.

OBJETIVOS ESPECÍFICOS

- Crear servicios web con PHP utilizando el método POST.
- Guardar información en un archivo JSON simulando una base de datos.
- Probar las rutas con Postman para asegurar el funcionamiento.

DESARROLLO DE EVIDENCIA:

DISEÑO Y DESARROLLO DE SERVICIOS WEB-PROYECTO: En esta evidencia desarrollo un servicio web básico que permite registrar usuarios y realizar autenticación mediante archivos PHP, utilizando JSON como almacenamiento. Además, realicé pruebas desde Postman para verificar el correcto funcionamiento de los servicios.

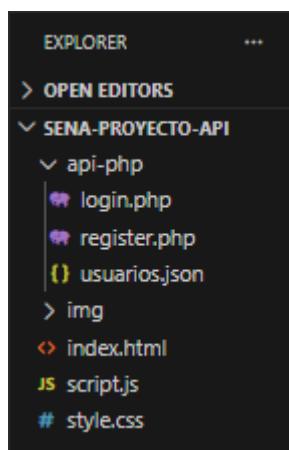
Herramientas utilizadas

Para desarrollar esta actividad utilicé:

- XAMPP como servidor (Apache).
- Postman para probar las peticiones.
- Editor de código (VSCode).
- JavaScript para realizar pruebas desde el frontend.

1. Creación de la estructura del proyecto

Captura 1: Estructura de carpetas del proyecto

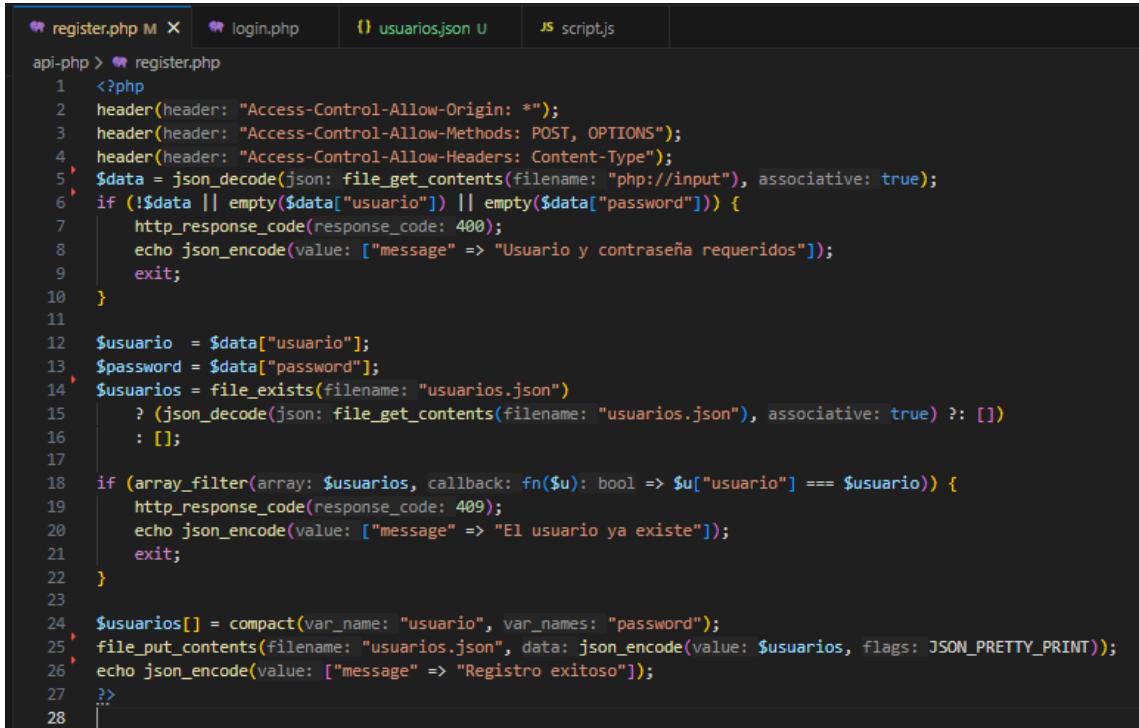


Explicación:

Aquí se muestra la organización inicial del proyecto donde se separan los archivos del cliente (interfaz) y el servidor (API). Esto permite trabajar de forma ordenada.

2. Desarrollo del proyecto

Captura 2: Archivo register.php

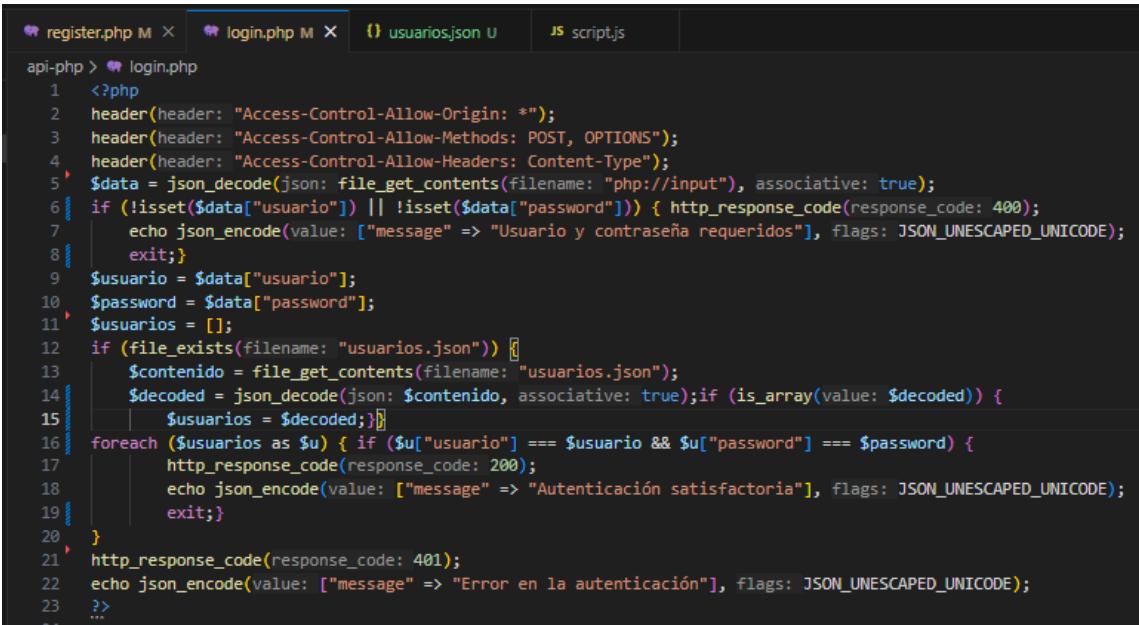


```
register.php M X login.php usuarios.json U script.js
api-php > register.php
1 <?php
2 header(header: "Access-Control-Allow-Origin: *");
3 header(header: "Access-Control-Allow-Methods: POST, OPTIONS");
4 header(header: "Access-Control-Allow-Headers: Content-Type");
5 $data = json_decode(json: file_get_contents(filename: "php://input"), associative: true);
6 if (!empty($data["usuario"]) || !empty($data["password"])) {
7     http_response_code(response_code: 400);
8     echo json_encode(value: ["message" => "Usuario y contraseña requeridos"]);
9     exit;
10 }
11
12 $usuario = $data["usuario"];
13 $password = $data["password"];
14 $usuarios = file_exists(filename: "usuarios.json")
15 ? (json_decode(json: file_get_contents(filename: "usuarios.json"), associative: true) ?: [])
16 : [];
17
18 if (array_filter(array: $usuarios, callback: fn($u): bool => $u["usuario"] === $usuario)) {
19     http_response_code(response_code: 409);
20     echo json_encode(value: ["message" => "El usuario ya existe"]);
21     exit;
22 }
23
24 $usuarios[] = compact(var_name: "usuario", var_names: "password");
25 file_put_contents(filename: "usuarios.json", data: json_encode(value: $usuarios, flags: JSON_PRETTY_PRINT));
26 echo json_encode(value: ["message" => "Registro exitoso"]);
27 ?>
28 ...
```

Explicación:

Primero valido que los datos lleguen correctamente, luego verifico si el usuario ya existe.

Captura 3: Archivo login.php



```
register.php M X login.php M X usuarios.json U script.js
api-php > login.php
1 <?php
2 header(header: "Access-Control-Allow-Origin: *");
3 header(header: "Access-Control-Allow-Methods: POST, OPTIONS");
4 header(header: "Access-Control-Allow-Headers: Content-Type");
5 $data = json_decode(json: file_get_contents(filename: "php://input"), associative: true);
6 if (!empty($data["usuario"]) || !empty($data["password"])) { http_response_code(response_code: 400);
7     echo json_encode(value: ["message" => "Usuario y contraseña requeridos"], flags: JSON_UNESCAPED_UNICODE);
8     exit;
9 }
10 $usuario = $data["usuario"];
11 $password = $data["password"];
12 $usuarios = [];
13 if (file_exists(filename: "usuarios.json")) {
14     $contenido = file_get_contents(filename: "usuarios.json");
15     $decoded = json_decode(json: $contenido, associative: true);if (is_array(value: $decoded)) {
16         $usuarios = $decoded;}}
17 foreach ($usuarios as $u) { if ($u["usuario"] === $usuario && $u["password"] === $password) {
18     http_response_code(response_code: 200);
19     echo json_encode(value: ["message" => "Autenticación satisfactoria"], flags: JSON_UNESCAPED_UNICODE);
20     exit;
21 }
22 http_response_code(response_code: 401);
23 echo json_encode(value: ["message" => "Error en la autenticación"], flags: JSON_UNESCAPED_UNICODE);
24 ...
```

Explicación:

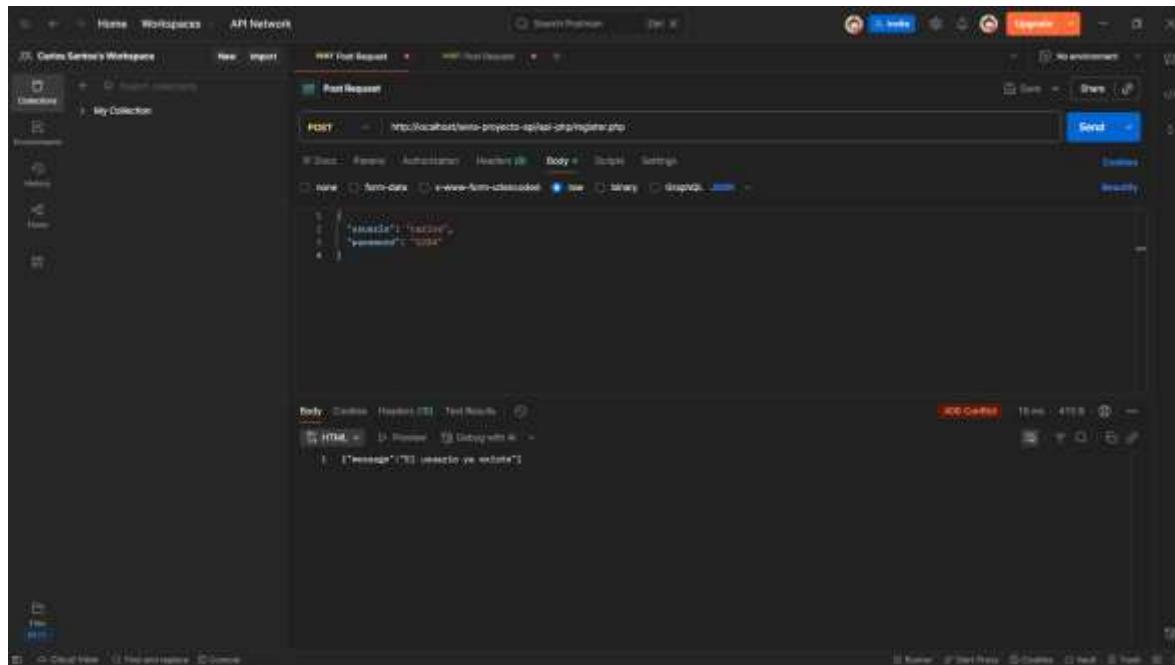
Este archivo se encarga de verificar que el usuario exista y que la contraseña coincida.

3. Pruebas realizadas en Postman

Para registrar un usuario utilicé el método POST, con la siguiente URL:
<http://localhost/sena-proyecto-api/api-php/register.php>

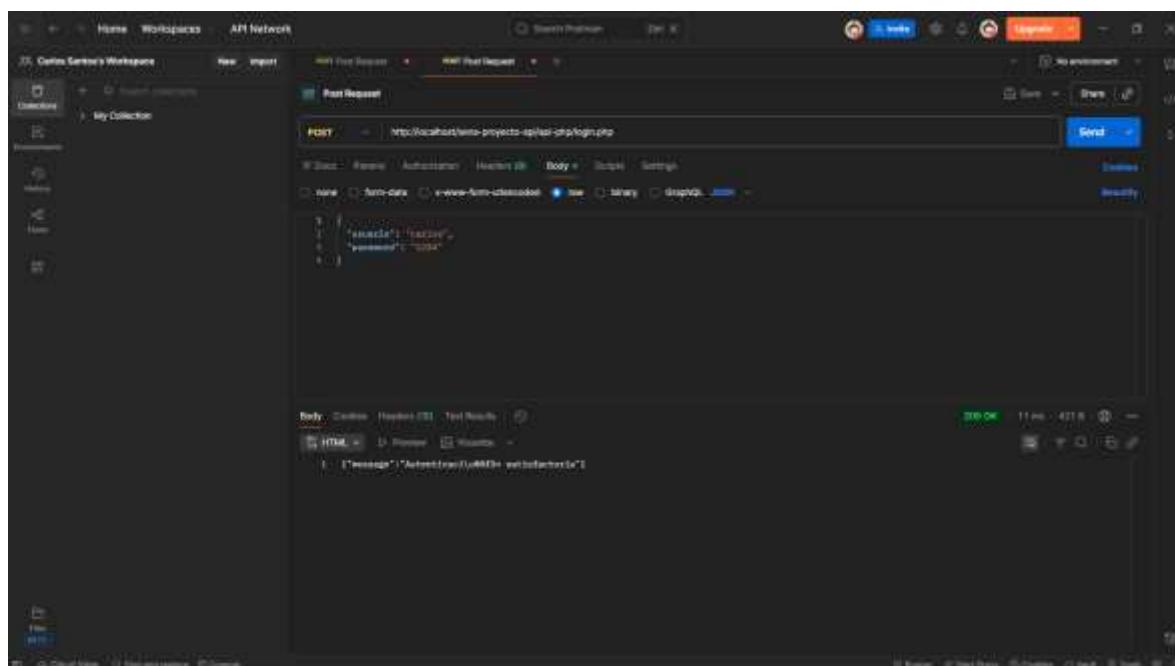
Luego probé el inicio de sesión con la ruta: <http://localhost/sena-proyecto-api/api-php/login.php>

Captura 4: Prueba del registro



The screenshot shows the Postman application interface. A 'Post Request' is being made to the URL <http://localhost/sena-proyecto-api/api-php/register.php>. The request method is set to POST. The body is defined as form-data with two fields: 'username' containing 'carlos' and 'password' containing '1234'. The response status is 200 OK, and the message is "User registered successfully".

Captura 5: Prueba del login



The screenshot shows the Postman application interface. A 'Post Request' is being made to the URL <http://localhost/sena-proyecto-api/api-php/login.php>. The request method is set to POST. The body is defined as form-data with two fields: 'username' containing 'carlos' and 'password' containing '1234'. The response status is 200 OK, and the message is "Authentication successful".

CONCLUSIONES

Logré implementar correctamente los servicios de registro y autenticación usando PHP. Aprendí a manejar archivos JSON como almacenamiento simple para usuarios. Las pruebas realizadas en Postman confirmaron que ambas rutas funcionan según lo esperado.

REFERENCIAS

SENA. (s.f.). *Tema 1* [Página web].

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF33/index.html#/curso/tema1>

SENA. (s.f.). *Tema 2* [Página web].

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF33/index.html#/curso/tema2>

SENA. (s.f.). *Tema 3* [Página web].

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF33/index.html#/curso/tema3>