

**Evidencia de desempeño: GA7-220501096-AA5-EV01 diseño y desarrollo de servicios web-
proyecto**

Hector Carlos Guevara Romero

Belkis Yurany Medina Sarmiento

Instructor

Henry Alfonso Garzón Sánchez

Servicio Nacional de Aprendizaje - SENA

Centro de Materiales y Ensayos - CME

Análisis y Desarrollo de Software – ADSO

2024

Contenido

Introducción a la evidencia	3
Título del proyecto	4
Objetivo General del Proyecto	4
Objetivos Específicos.....	4
Servicio web para un registro y un inicio de sesión.....	5
Enlace a repositorio GIT	16
Conclusiones de la Evidencia	16
Referencias Bibliográficas	17

Introducción a la evidencia

En el desarrollo de aplicaciones modernas, la implementación de servicios web es fundamental para garantizar una comunicación eficiente entre clientes y servidores. Esta evidencia tiene como objetivo diseñar y codificar un servicio web de autenticación, que permita el registro y el inicio de sesión de usuarios en el proyecto desarrollado.

El servicio web se encargará de recibir credenciales de acceso (usuario y contraseña) y verificar su validez en la base de datos. En caso de autenticación exitosa, se mostrará un mensaje de confirmación; de lo contrario, se devolverá un error de autenticación.

Título del proyecto

Software para la contratación de personal idóneo por eventos.

Objetivo General del Proyecto

Implementar en el departamento técnico un sistema para la contratación de profesionales idóneos que integre todos los procesos para facilitar la rápida y eficaz colocación del personal en cualquier evento que requiera el cliente y lograr tener una base de datos de todo el personal para atender todas las zonas a la ciudad de Bogotá.

Objetivos Específicos

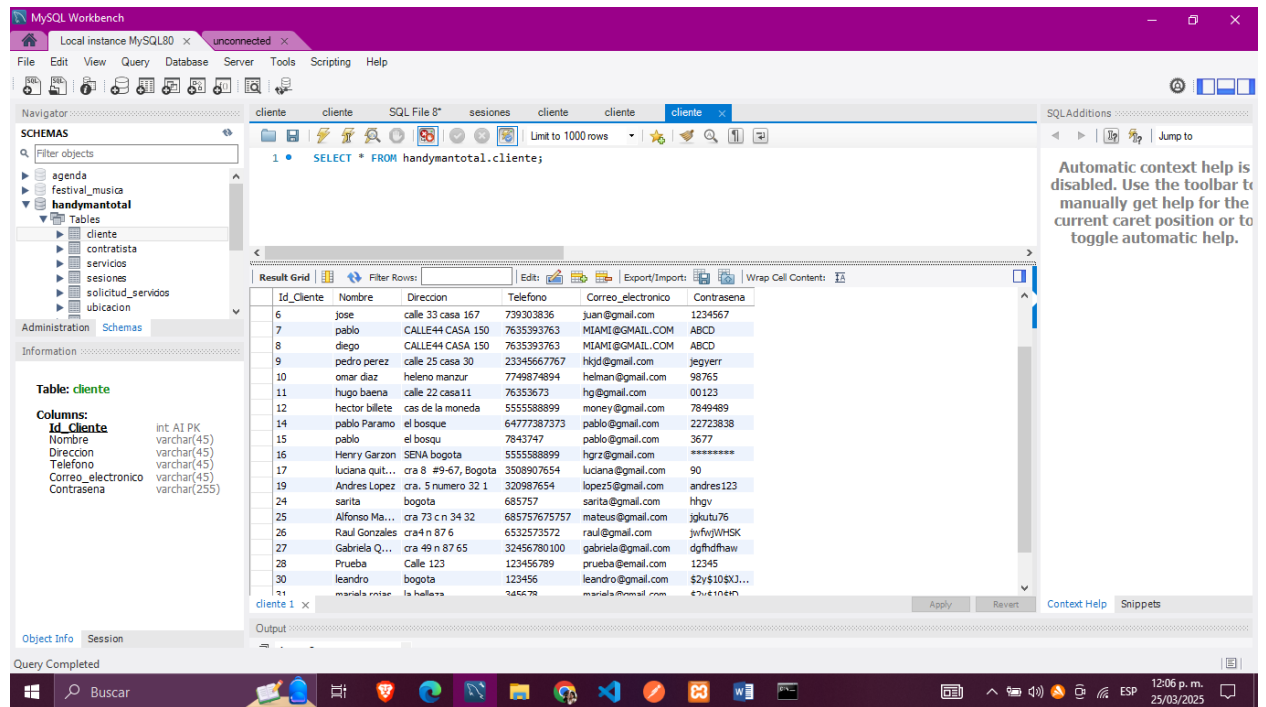
- Levantar información de personal de profesionales idóneos o contratistas creando en la página Web de la empresa las utilidades necesarias para llenar formularios de inscripción del personal. Crear publicidad en redes sociales con el propósito de recaudar información de personas que deseen participar como Contratistas que sean egresados y certificados del SENA preferiblemente.
- Mediante el análisis de la información plantear soluciones que ayuden a conseguir la mayor cantidad de profesionales disponibles.
- Mediante el análisis de la información diseñar un software para este proceso que optimice la contratación del personal.
- Teniendo el diseño de la solución desarrollar el software de contratación para personal idóneo por eventos con el presupuesto para tal fin.
- Probar y validar el sistema de manera que se pueda usar 24/7 sin fallas.

- Implementar el Software de contratación en el departamento de personal y departamento técnico y abrir la plaza para contratar un operario.

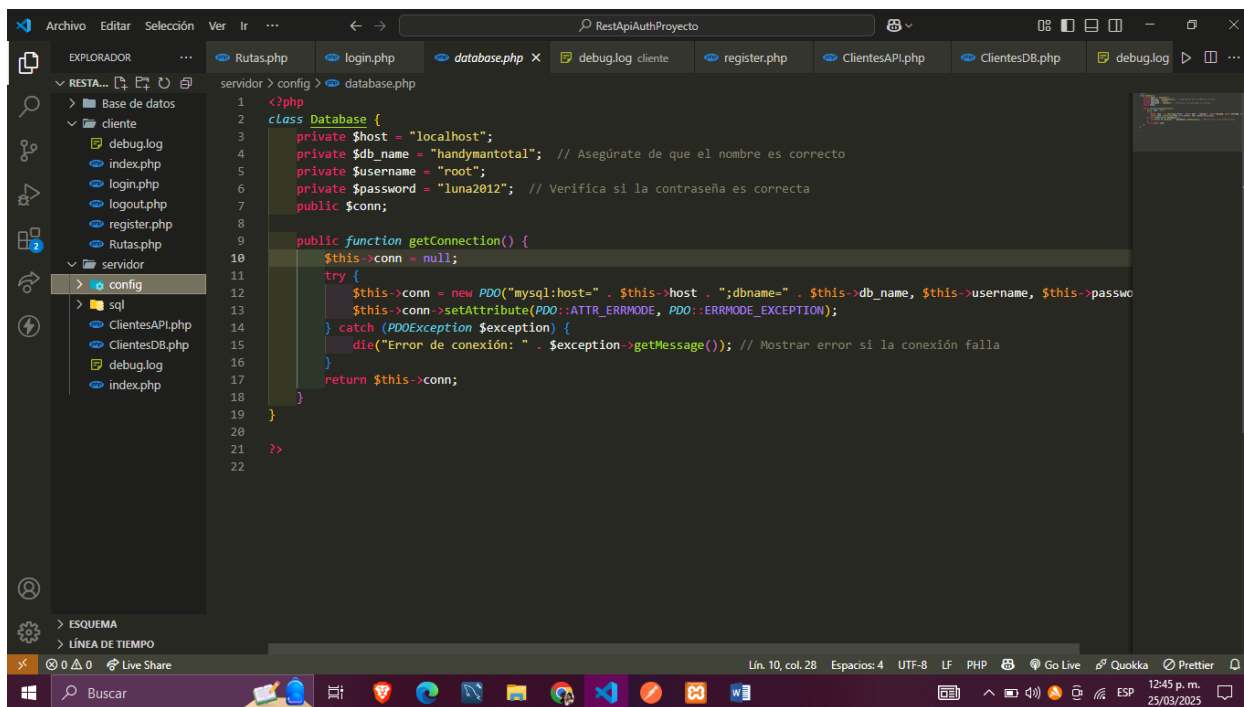
Servicio web para un registro y un inicio de sesión

Para realizar el servicio web para aun registro e inicio de sesión, debemos conectarnos a la base de datos con nuestras credenciales, encender el servidor de xampp apache, y como editor de código usaremos visual studio code.

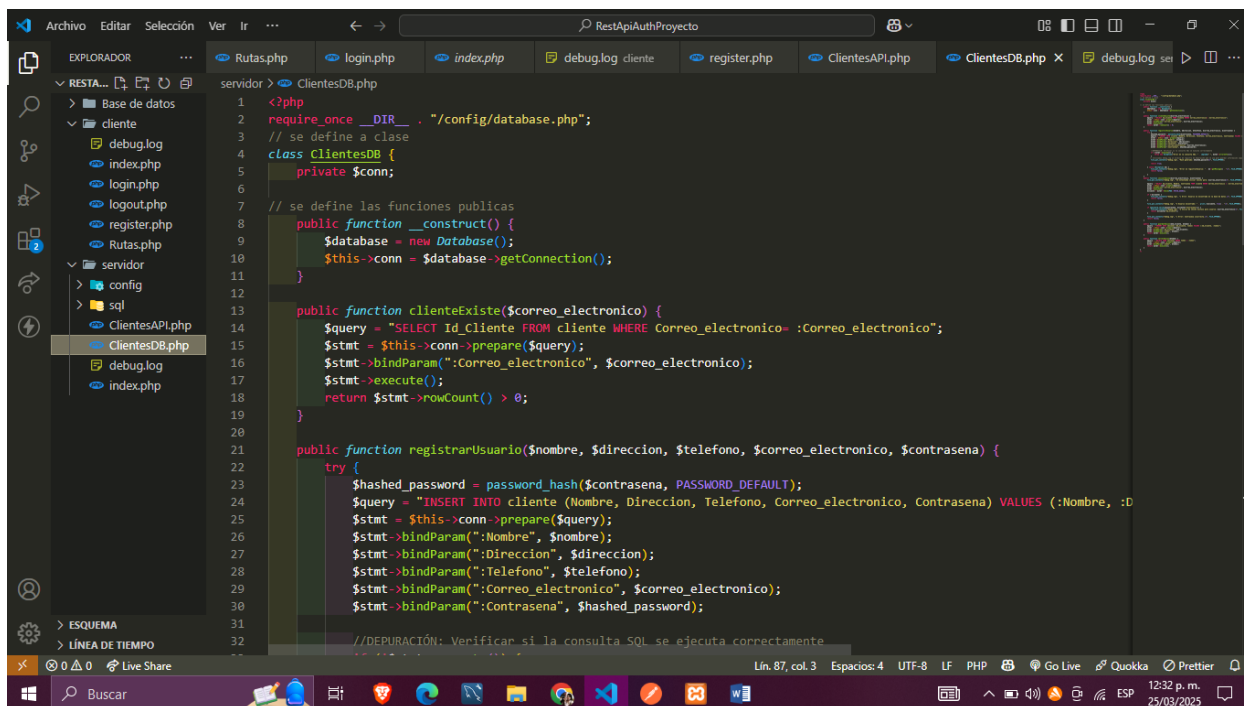
1. Utilizamos la base de datos del proyecto



2. Dentro de visual crearemos la carpeta de nuestro proyecto, y dentro de ella la carpeta config, creamos el archivo database.php, en el cual estarán las credenciales de nuestra base de datos y la configuración para acceder a ella.



3. Creamos la carpeta servidor, y dentro de ella el archivo Clientes.php



```

7 // se define las funciones publicas
21 public function registrarUsuario($nombre, $direccion, $telefono, $correo_electronico, $contrasena) {
22     try {
33         if (!$stmt->execute()) {
34             throw new Exception("Error en la consulta SQL: " . implode(" ", $stmt->errorInfo()));
35         }
36         // utilizamos debug.log un archivo de registro de depuración que se usa para guardar información importante sobre
37         file_put_contents("debug.log", "Hash generado: $hashed_password\n", FILE_APPEND);
38     }
39     return true;
40 }
41 } catch (Exception $e) {
42     file_put_contents("debug.log", "Error en registrarUsuario: " . $e->getMessage() . "\n", FILE_APPEND);
43     return false;
44 }
45 }
46 }
47 public function iniciarSesion($correo_electronico, $contrasena) {
48     file_put_contents("debug.log", "Intentando iniciar sesión para: $correo_electronico\n", FILE_APPEND);
49 }
50 $query = "SELECT Id_Cliente, Nombre, Contraseña FROM cliente WHERE Correo_electronico = :Correo_electronico";
51 $stmt = $this->conn->prepare($query);
52 $stmt->bindParam(":Correo_electronico", $correo_electronico);
53 $stmt->execute();
54 $cliente = $stmt->fetch(PDO::FETCH_ASSOC);
55 }
56 if (!$cliente) {
57     file_put_contents("debug.log", "X Error: Usuario no encontrado en la base de datos.\n", FILE_APPEND);
58     return false;
59 }
60 }
61 file_put_contents("debug.log", "U Usuario encontrado: " . print_r($cliente, true) . "\n", FILE_APPEND);

```

```

47 // se define las funciones publicas
48 public function iniciarSesion($correo_electronico, $contrasena) {
49     file_put_contents("debug.log", "U Usuario encontrado: " . print_r($cliente, true) . "\n", FILE_APPEND);
50 }
51 if (password_verify($contrasena, $cliente["Contraseña"])) {
52     file_put_contents("debug.log", "U Inicio de sesión exitoso para usuario: $correo_electronico\n", FILE_APPEND);
53     return $cliente["Id_Cliente"];
54 }
55 }
56 file_put_contents("debug.log", "X Error: Contraseña incorrecta.\n", FILE_APPEND);
57 return false;
58 }
59 }
60 public function guardarSesion($Id_Cliente, $token) {
61     $query = "INSERT INTO sesiones (Id_Cliente, token) VALUES (:Id_Cliente, :token)";
62     $stmt = $this->conn->prepare($query);
63     $stmt->bindParam(":Id_Cliente", $Id_Cliente);
64     $stmt->bindParam(":token", $token);
65     return $stmt->execute();
66 }
67 }
68 public function cerrarSesion($token) {
69     $query = "DELETE FROM sesiones WHERE token = :token";
70     $stmt = $this->conn->prepare($query);
71     $stmt->bindParam(":token", $token);
72     return $stmt->execute();
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }

```

4. Se crea el archivo CientesAPI.php, en este archivo se definen las funciones y métodos

```

1 <?php
2 require_once "ClientesDB.php";
3
4 //Se define la clase
5 class ClientesAPI {
6     private $db;
7     //se define la funciones publicas
8
9     public function __construct() {
10         $this->db = new ClientesDB();
11     }
12     public function API() {
13         header("Content-Type: application/json");
14         //utilizamos debug.logs un archivo de registro de depuración
15         // que se usa para guardar información importante sobre el funcionamiento del código .
16
17         file_put_contents("debug.log", " ♦ Método recibido: " . $_SERVER["REQUEST_METHOD"] . "\n", FILE_APPEND);
18         file_put_contents("debug.log", " ♦ Parámetros GET: " . print_r($_GET, true) . "\n", FILE_APPEND);
19
20         $method = $_SERVER["REQUEST_METHOD"];
21         file_put_contents("debug.log", " ♦ Contenido de \$_SERVER[\\"REQUEST_URI\\": " . $_SERVER["REQUEST_URI"] . "\n", FILE_APPEND);
22         file_put_contents("debug.log", " ♦ Contenido de \$_GET: " . print_r($_GET, true) . "\n", FILE_APPEND);
23
24         file_put_contents("debug.log", " ♦ Método recibido en API: " . $_SERVER["REQUEST_METHOD"] . "\n", FILE_APPEND);
25         file_put_contents("debug.log", " ♦ Parámetros GET en API: " . print_r($_GET, true) . "\n", FILE_APPEND);
26
27         file_put_contents("debug.log", " ♦ Método recibido en API: " . $_SERVER["REQUEST_METHOD"] . "\n", FILE_APPEND);
28         file_put_contents("debug.log", " ♦ Parámetros GET en API: " . print_r($_GET, true) . "\n", FILE_APPEND);
29         file_put_contents("debug.log", " ♦ Parámetros POST en API: " . file_get_contents("php://input") . "\n", FILE_APPEND);
30
31         file_put_contents("debug.log", " ♦ Método recibido en API: " . $_SERVER["REQUEST_METHOD"] . "\n", FILE_APPEND);
32         file_put_contents("debug.log", " ♦ Contenido de \$_SERVER[\\"REQUEST_URI\\": " . $_SERVER["REQUEST_URI"] . "\n", FILE_APPEND);

```

```

36 //se definen los metodos
37 switch ($method) {
38     case "POST":
39         if (isset($_GET["action"])) {
40             if ($_GET["action"] == "register") {
41                 $this->registrarUsuario();
42                 return;
43             } elseif ($_GET["action"] == "login") {
44                 $this->iniciarSesion();
45                 return;
46             }
47         }
48         break;
49     case "DELETE":
50         if (isset($_GET["action"]) && $_GET["action"] == "logout") {
51             $this->cerrarSesion();
52             return;
53         }
54         break;
55     default:
56         $this->response(405, "error", "Método no permitido");
57         return;
58 }
59
60 file_put_contents("debug.log", " ♦ JSON generado antes de enviar: " . $json . "\n", FILE_APPEND);
61 file_put_contents("debug.log", " ♦ JSON generado antes de enviar: " . json_encode(["status" => "success", "message" => ""]) . "\n", FILE_APPEND);
62
63 echo json_encode(["status" => "error", "message" => "Acción no válida"]);
64 file_put_contents("debug.log", " ♦ JSON enviado por la API: " . json_encode(["status" => "success", "message" => ""]) . "\n", FILE_APPEND);
65
66 exit;
67 return;
68 }
69 //se definen las funciones privadas

```



```

66 exit;
70 private function registrarUsuario() {
71     try {
72         // Leer y decodificar datos del cuerpo de la petición
73         $datos = json_decode(file_get_contents("php://input"), true);
74         // DEPURACIÓN: Verificar si los datos llegan correctamente
75         file_put_contents("debug.log", print_r($datos, true));
76         if (!$datos) {
77             throw new Exception("Error al decodificar JSON o JSON vacío");
78         }
79         // Validar que los datos necesarios están presentes
80         if (!isset($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos["Correo_electronico"], $datos["Contraseña"])) {
81             throw new Exception("Datos incompletos: Se requieren nombre, direccion, telefono, correo_electronico y contraseña");
82         }
83         // Intentar registrar al usuario en la base de datos
84         $registroExitoso = $this->db->registrarUsuario($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos["Correo_electronico"], $datos["Contraseña"]);
85         if (!$registroExitoso) {
86             return $this->response(200, "error", "El usuario ya se encuentra registrado");
87         }
88         // Respuesta exitosa
89         return $this->response(201, "success", "Usuario registrado");
90     } catch (Exception $e) {
91         var_dump($e);
92     }
93 }
94 // Manejo de errores y respuesta adecuada
95 return $this->response(500, "error", $e->getMessage());

```

```

102 } catch (Exception $e) {
103     return $this->response(500, "error", $e->getMessage());
104 }
105 private function iniciarSesion() {
106     header("Content-Type: application/json");
107     ob_clean(); // Limpia cualquier salida anterior
108     file_put_contents("debug.log", "Entrando a iniciarSesion() en ClientesAPI.php\n", FILE_APPEND);
109     $datos = json_decode(file_get_contents("php://input"), true);
110     file_put_contents("debug.log", "Datos recibidos: " . print_r($datos, true) . "\n", FILE_APPEND);
111     if (!$datos || !isset($datos["Correo_electronico"], $datos["Contraseña"])) {
112         file_put_contents("debug.log", "Error: Datos incompletos en iniciarSesion().\n", FILE_APPEND);
113         $json = json_encode(["status" => "error", "message" => "Datos incompletos"]);
114         file_put_contents("debug.log", "JSON enviado por la API (error): " . $json . "\n", FILE_APPEND);
115         echo $json;
116         exit;
117     }
118     file_put_contents("debug.log", "Buscando usuario en la base de datos...\n", FILE_APPEND);
119     $sid_cliente = $this->db->iniciarSesion($datos["Correo_electronico"], $datos["Contraseña"]);
120     if ($sid_cliente) {
121         file_put_contents("debug.log", "Inicio de sesión exitoso para: " . $datos["Correo_electronico"] . "\n", FILE_APPEND);
122         $token = bin2hex(random_bytes(32));
123         $this->db->guardarSesion($sid_cliente, $token);
124     }
125     $respuesta = ["status" => "success", "message" => "Inicio de sesión exitoso", "token" => $token];

```

```

66  exit;
104 private function iniciarSesion() {
125     if ($id_cliente) {
130         $respuesta = ["status" => "success", "message" => ["token" => $token]];
131         $json = json_encode($respuesta);
132
133         file_put_contents("debug.log", "♦ Respuesta enviada por la API: " . $json . "\n", FILE_APPEND);
134         echo $json;
135         exit;
136     } else {
137         file_put_contents("debug.log", "✗ Error: Credenciales incorrectas para: " . $datos["Correo_electronico"] . "\n", FILE_APPEND);
138         $json = json_encode(["status" => "error", "message" => "Credenciales incorrectas"]);
139         file_put_contents("debug.log", "♦ JSON enviado por la API (error): " . $json . "\n", FILE_APPEND);
140         echo $json;
141         exit;
142     }
143 }
144 private function cerrarSesion() {
145     $token = $_GET["token"] ?? "";
146     if ($token && $this->db->cerrarSesion($token)) {
147         $this->response(200, "success", "Sesión cerrada");
148     } else {
149         $this->response(400, "error", "Error al cerrar sesión");
150     }
151 }
152 private function response($code, $status, $message) {
153     http_response_code($code);
154     echo json_encode(["status" => $status, "message" => $message], JSON_PRETTY_PRINT);
155 }
156
157 $api = new ClientesAPI();
158 $api->API();

```

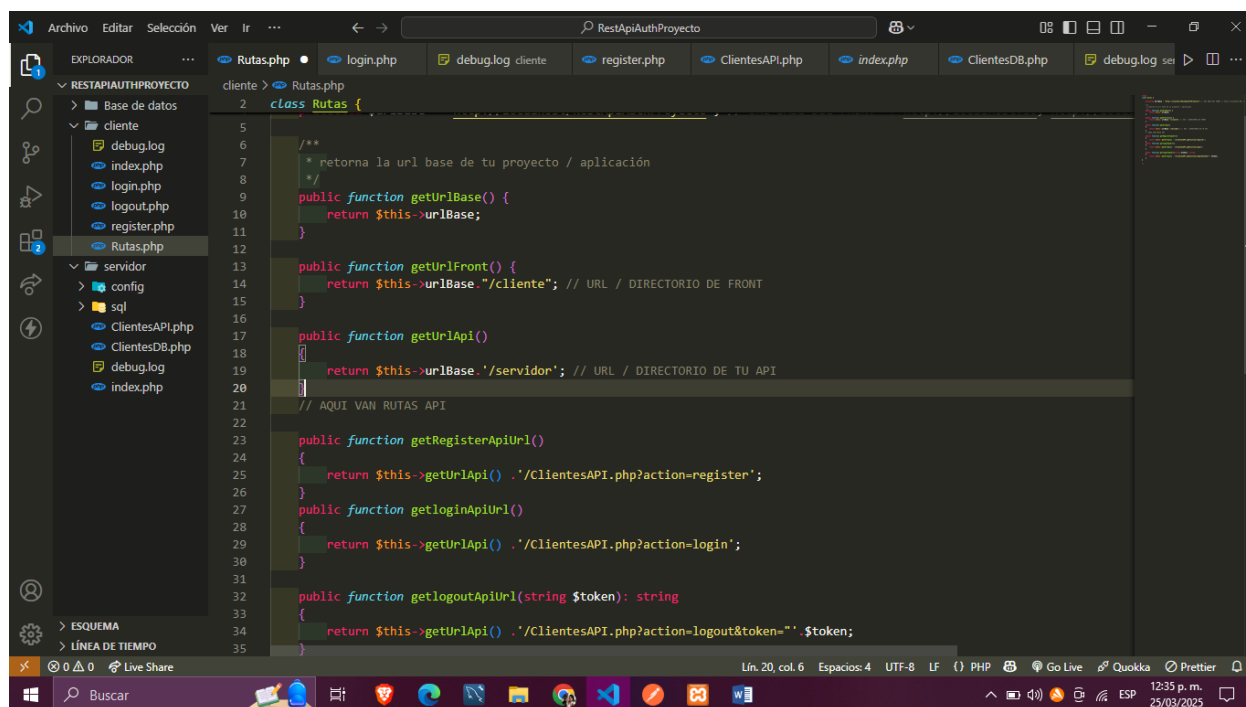
5. Se crea el archivo index.php

```

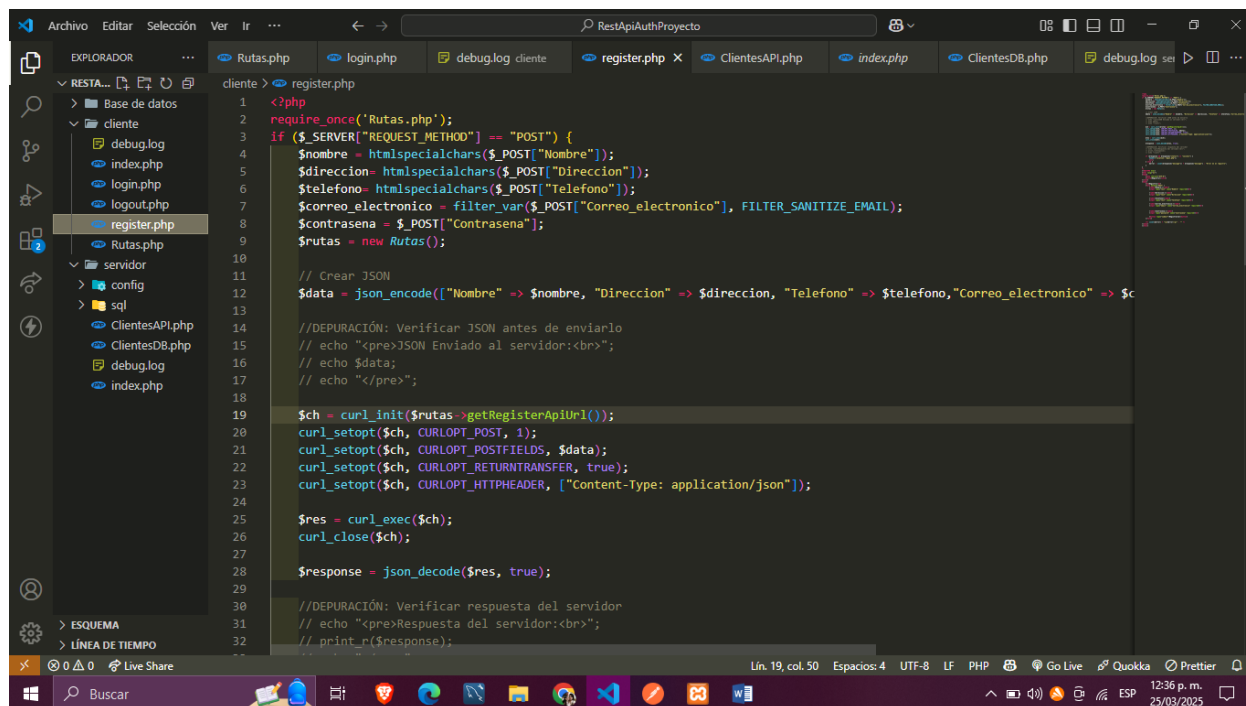
1  <?php
2  echo json_encode(["message" => "API de Clientes en funcionamiento"]);
3  ?>

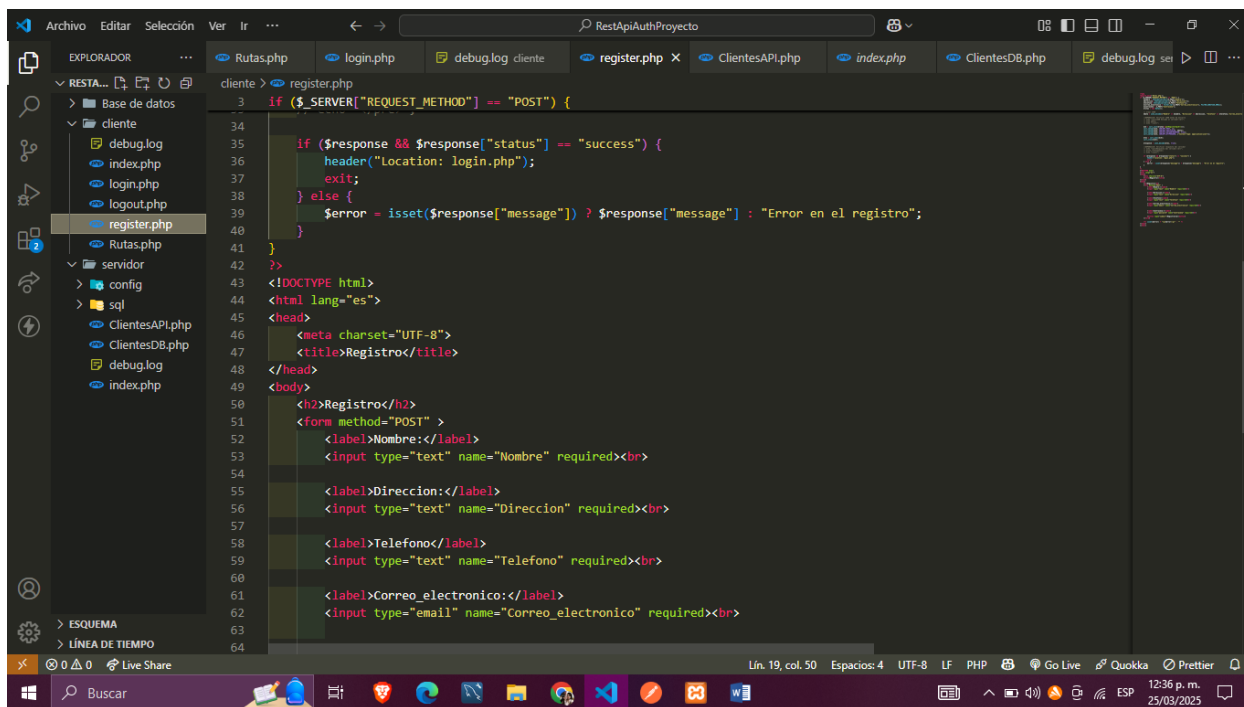
```

6. En la carpeta clientes creamos el archivo Rutas.php, donde se definirá la url del front y las rutas api



7. Creamos el archivo register.php, donde creamos el registro de un cliente



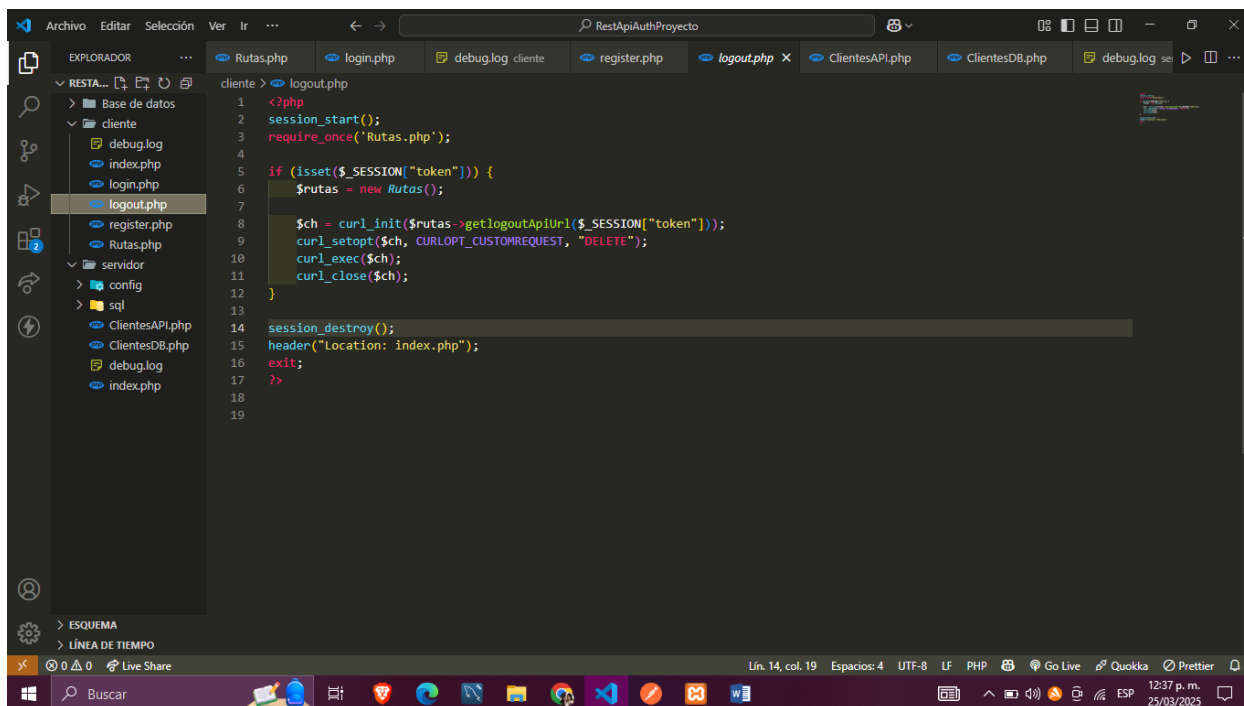


```

3  if ($SERVER["REQUEST_METHOD"] == "POST") {
34
35     if ($response && $response["status"] == "success") {
36         header("Location: login.php");
37         exit;
38     } else {
39         $error = isset($response["message"]) ? $response["message"] : "Error en el registro";
40     }
41 }
42 }>
43 <!DOCTYPE html>
44 <html lang="es">
45 <head>
46     <meta charset="UTF-8">
47     <title>Registro</title>
48 </head>
49 <body>
50     <h2>Registro</h2>
51     <form method="POST" >
52         <label>Nombre:</label>
53         <input type="text" name="Nombre" required><br>
54
55         <label>Direccion:</label>
56         <input type="text" name="Direccion" required><br>
57
58         <label>Telefono:</label>
59         <input type="text" name="Telefono" required><br>
60
61         <label>Correo_electronico:</label>
62         <input type="email" name="Correo_electronico" required><br>
63
64

```

8. Creamos el archivo Logout.php



```

1  <?php
2  session_start();
3  require_once('Rutas.php');
4
5  if (isset($_SESSION["token"])) {
6      $rutas = new Rutas();
7
8      $ch = curl_init($rutas->getlogoutApiUrl($_SESSION["token"]));
9      curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
10     curl_exec($ch);
11     curl_close($ch);
12 }
13
14 session_destroy();
15 header("Location: index.php");
16 exit;
17 ?>
18
19

```

9. Creamos el archivo login.php

```

1 <?php
2 session_start();
3 require_once('Rutas.php');
4
5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6     $rutas = new Rutas();
7
8     $correo_electronico = trim(filter_var($_POST["Correo_electronico"], FILTER_SANITIZE_EMAIL));
9     $contrasena = trim($_POST["Contrasena"]);
10    $data = json_encode(["Correo_electronico" => $correo_electronico, "Contrasena" => $contrasena]);
11
12    $ch = curl_init($rutas->getLoginApiUrl());
13    curl_setopt($ch, CURLOPT_POST, 1);
14    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
15    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
16    curl_setopt($ch, CURLOPT_HTTPHEADER, ["Content-Type: application/json"]);
17
18    $res = curl_exec($ch);
19    file_put_contents("debug.log", "♦ Respuesta JSON cruda de la API: " . $res . "\n", FILE_APPEND);
20
21    curl_close($ch);
22
23    // Decodificar la respuesta JSON
24    $response = json_decode($res, true);
25    file_put_contents("debug.log", "♦ Respuesta decodificada en PHP: " . print_r($response, true) . "\n", FILE_APPEND);
26
27    // Verificar la estructura de la respuesta
28    if ($response && isset($response["status"]) && $response["status"] == "success") {
29        if (isset($response["message"]["token"])) {
30            file_put_contents("debug.log", "✅ Sesión iniciada correctamente en login.php\n", FILE_APPEND);
31            $_SESSION["cliente"] = $correo_electronico;
32            $_SESSION["token"] = $response["message"]["token"];
33        }
34    }
35 }

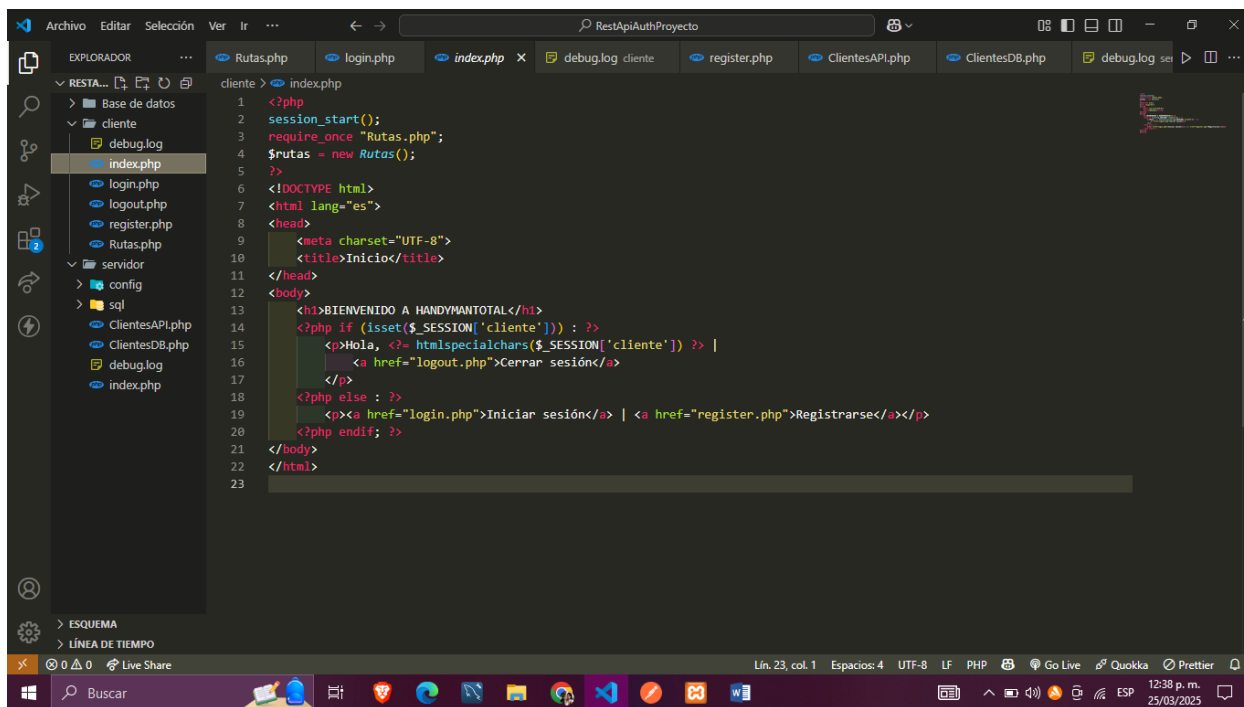
```

```

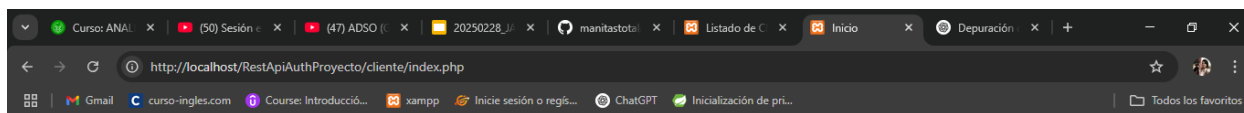
28 if ($_SERVER["REQUEST_METHOD"] == "POST") {
29     if ($response && isset($response["status"]) && $response["status"] == "success") {
30         if (isset($response["message"]["token"])) {
31             header("Location: index.php");
32             exit;
33         } else {
34             file_put_contents("debug.log", "⚠️ La respuesta no contiene un token\n", FILE_APPEND);
35             $error = "Error en la autenticación";
36         }
37     } else {
38         file_put_contents("debug.log", "❌ Error en login.php: " . print_r($response, true) . "\n", FILE_APPEND);
39         $error = "Credenciales incorrectas";
40     }
41 }
42
43 <?php
44
45 <DOCTYPE html>
46 <html lang="es">
47 <head>
48     <meta charset="UTF-8">
49     <title>Login</title>
50 </head>
51 <body>
52     <h2>Iniciar Sesión</h2>
53     <form method="POST">
54         <label>Correo_electronico:</label>
55         <input type="text" name="Correo_electronico" required><br>
56         <label>Contrasena:</label>
57         <input type="password" name="Contrasena" required><br>
58         <button type="submit">Ingresar</button>
59     </form>
60     <?php if ($error) {
61         echo "<p>$error</p>";
62     }
63 }

```

10. Creamos el archivo index.php



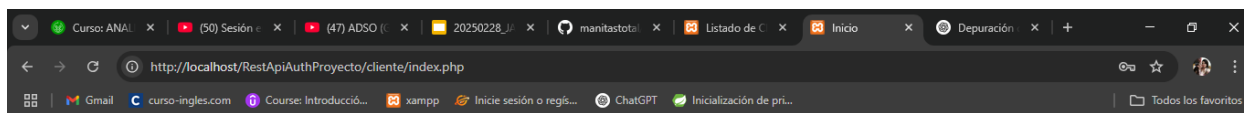
11. Registro e inicio de sesión en funcionamiento



BIENVENIDO A HANDYMANTOTAL

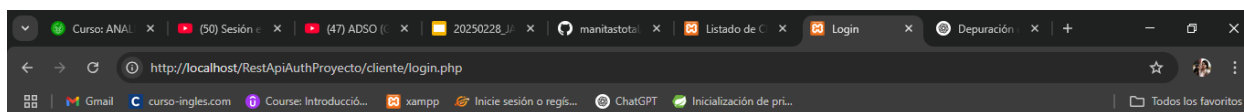
[Iniciar sesión](#) | [Registrarse](#)





BIENVENIDO A HANDYMANTOTAL

Hola, belkyz.ymys@gmail.com | [Cerrar sesión](#)

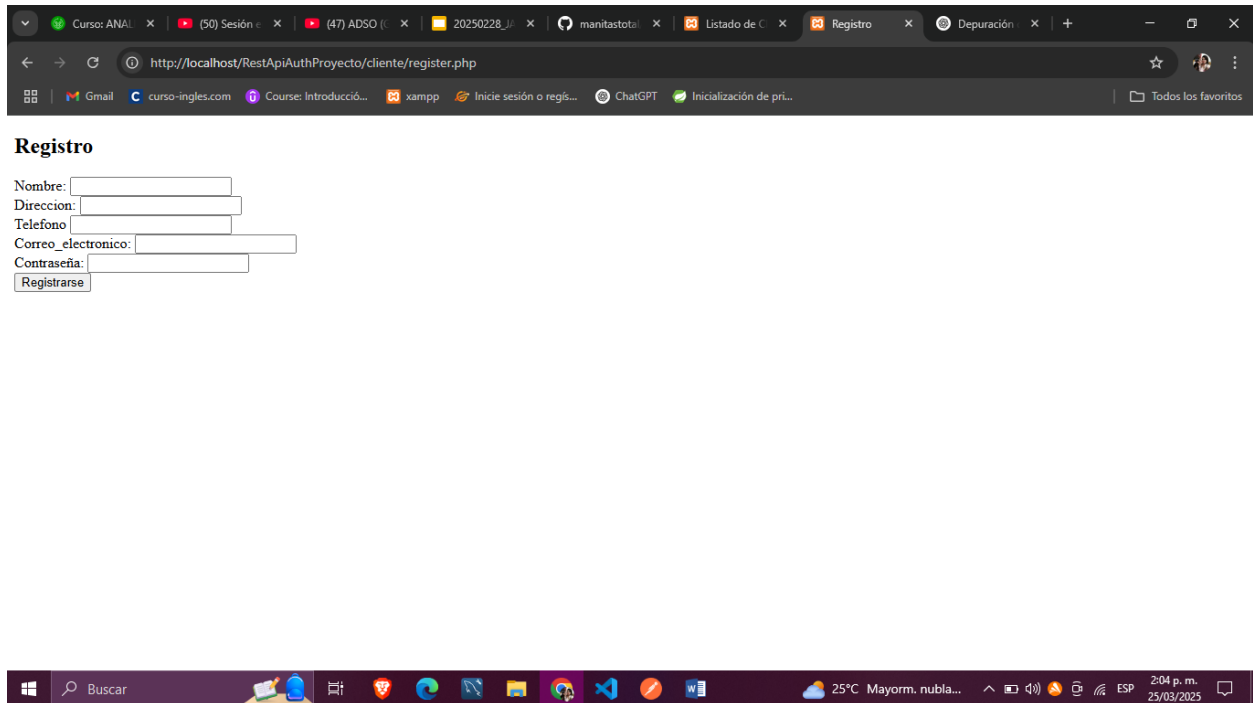


Iniciar Sesión

Correo_electronico:

Contraseña:





The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Registro' and displays a registration form. The form is titled 'Registro' and contains the following fields: 'Nombre:', 'Direccion:', 'Telefono:', 'Correo_electronico:', and 'Contraseña:'. Each field is followed by a text input box. Below the 'Contraseña:' field is a button labeled 'Registrarse'. The browser's address bar shows the URL 'http://localhost/RestApiAuthProyecto/cliente/register.php'. The Windows taskbar at the bottom shows the system clock as 2:04 p.m. on 25/03/2025, with a temperature of 25°C and weather 'Mayorm. nubla...'. The taskbar also includes icons for various applications like Chrome, VS Code, and Word.

Enlace a repositorio GIT

<https://github.com/handymantotal/PROYECTO-HANDYMANTOTAL.git>

Conclusiones de la Evidencia

La implementación de un servicio web para registro e inicio de sesión es un paso fundamental en el desarrollo de aplicaciones seguras y eficientes. Durante esta evidencia, se diseñó y programó un servicio que permite autenticar usuarios de manera confiable, garantizando el acceso controlado a la aplicación.

El uso de técnicas de manejo de errores, validación de datos y respuestas estructuradas en formato JSON ha permitido mejorar la fiabilidad del sistema. Además, se aplicaron buenas

prácticas de codificación y comentarios en el código, lo que facilita su mantenimiento y escalabilidad.

Finalmente, este servicio web proporciona una base sólida para futuras mejoras, como la implementación de tokens de sesión, encriptación avanzada de contraseñas y la integración con otros módulos del sistema.

Referencias Bibliográficas

<https://chatgpt.com/>

<https://www.youtube.com/watch?v=CRW8wf76kgQ>

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

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF34/index.html#/>

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF35/index.html#/curso/tema1>