

**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

2025

## 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**

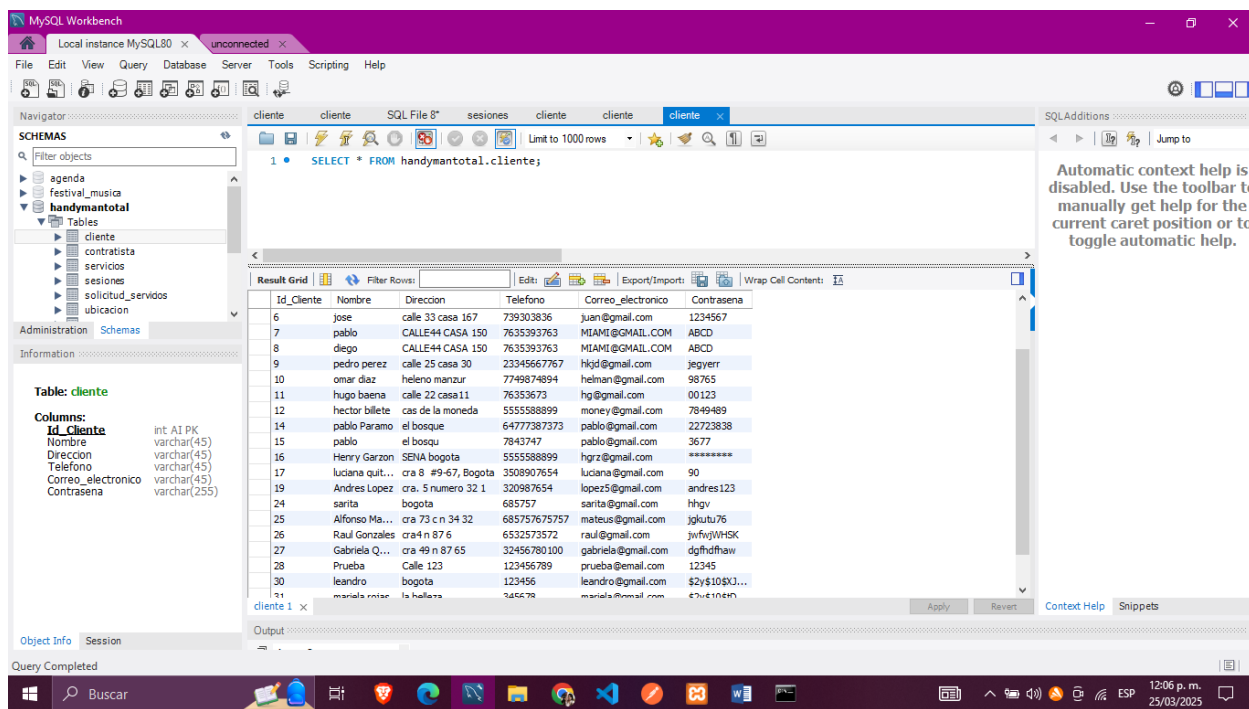
- Levantar información de personal de profesionales idóneos o contratistas creando en la página Web de la empresa las utilidades necesarias para llenar formularios de inscripción del personal. Crear publicidad en redes sociales con el propósito de recaudar información de personas que deseen participar 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

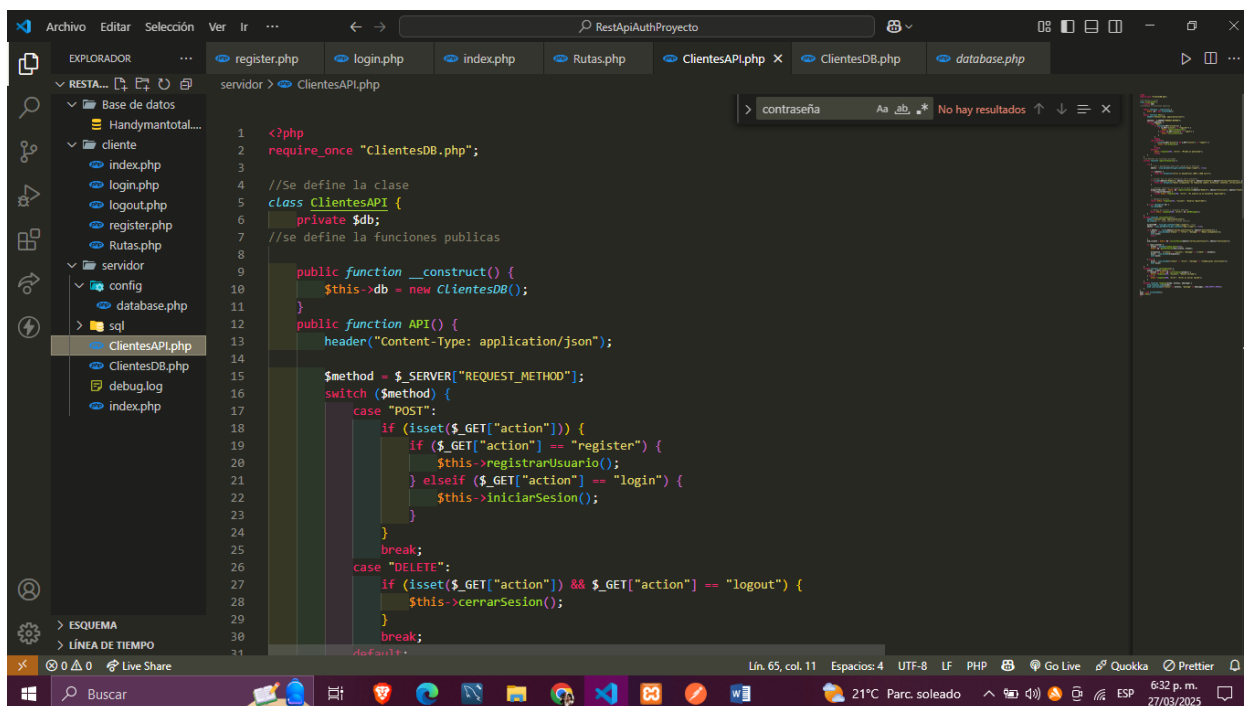
Para realizar el servicio web para un registro e inicio de sesión, debemos conectarnos a la base de datos con nuestras credenciales, encender el servidor de xampp apache, y como editor de código usaremos visual studio code.

#### 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 `ClientesDB.php`



```

7 //se define la funciones publicas
12 public function API() {
16     switch ($method) {
26         case "DELETE":
30             break;
31         default:
32             $this->response(405, "error", "Método no permitido");
33             break;
34     }
35 }
36 //se definen las funciones privadas
37 private function registrarUsuario() {
38
39     try {
40         // Leer y decodificar datos del cuerpo de la petición
41         $datos = json_decode(file_get_contents("php://input"), true);
42
43         if (!$datos) {
44             throw new Exception("Error al decodificar JSON o JSON vacío");
45         }
46
47         // Validar que los datos necesarios están presentes
48         if (!isset($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos["Correo_electronico"], $datos["Cont
49             throw new Exception("Datos incompletos: Se requieren nombre, direccion, telefono, correo_electronico y cont
50         }
51
52         // Intentar registrar al usuario en la base de datos
53         $registroExitoso = $this->db->registrarUsuario($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos
54         if (!$registroExitoso) {
55             return $this->response(200, "error", "El usuario ya se encuentra registrado");
56         }
57     }

```

```

37 //se define la funciones publicas
38 private function registrarUsuario() {
39     try {
40
41     } catch (Exception $e) {
42         var_dump($e);
43     }
44     // Manejo de errores y respuesta adecuada
45     return $this->response(500, "error", $e->getMessage());
46 }
47
48 private function iniciarSesion() {
49     header("Content-Type: application/json");
50     ob_clean(); // Limpia cualquier salida anterior
51
52     $inputJSON = file_get_contents("php://input"); //esto
53     $datos = json_decode(file_get_contents("php://input"), true);
54
55     if (!$datos || !isset($datos["Correo_electronico"], $datos["Contrasena"])) {
56         $json = json_encode(["status" => "error", "message" => "Datos incompletos"]);
57         echo $json;
58         exit;
59     }
60
61     $id_cliente = $this->db->iniciarSesion($datos["Correo_electronico"], $datos["Contrasena"]);
62
63     if ($id_cliente) {
64         $token = bin2hex(random_bytes(32));
65         $this->db->guardarSesion($id_cliente, $token);
66
67         $respuesta = ["status" => "success", "message" => ["token" => $token]];
68         $json = json_encode($respuesta);
69         echo $json;
70     }

```





```

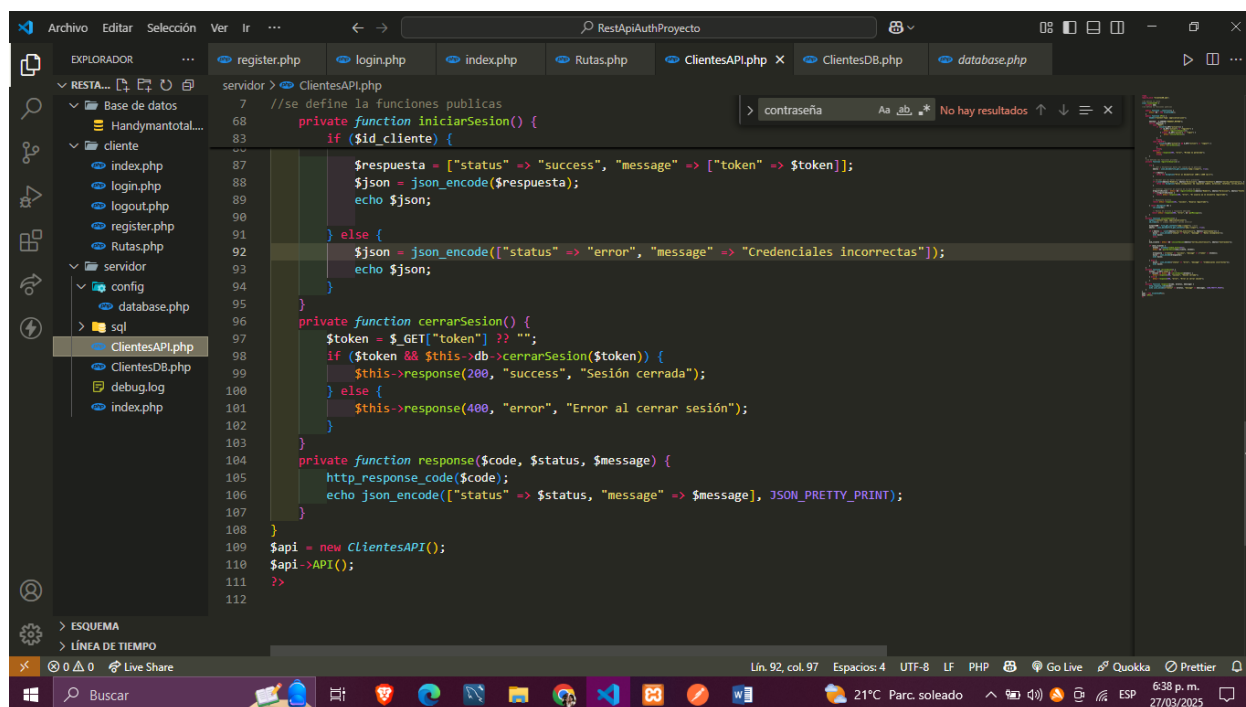
7 //se define la funciones publicas
8 public function API() {
9     switch ($method) {
10         default:
11             $this->response(405, "error", "Metodo no permitido");
12             break;
13     }
14 }
15
16 //se definen las funciones privadas
17 private function registrarUsuario() {
18     try {
19         // Leer y decodificar datos del cuerpo de la petición
20         $datos = json_decode(file_get_contents("php://input"), true);
21
22         if (!$datos) {
23             throw new Exception("Error al decodificar JSON o JSON vacio");
24         }
25
26         // Validar que los datos necesarios están presentes
27         if (!isset($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos["Correo_electronico"], $datos["Cont"]
28             throw new Exception("Datos incompletos: Se requieren nombre, direccion, telefono, correo_electronico y cont
29         }
30
31         // Intentar registrar al usuario en la base de datos
32         $registroExitoso = $this->db->registrarUsuario($datos["Nombre"], $datos["Direccion"], $datos["Telefono"], $datos
33         if (!$registroExitoso) {
34             return $this->response(200, "error", "El usuario ya se encuentra registrado");
35         }
36
37         // Respuesta exitosa
38         return $this->response(201, "success", "Usuario registrado");
39     }
40 }

```

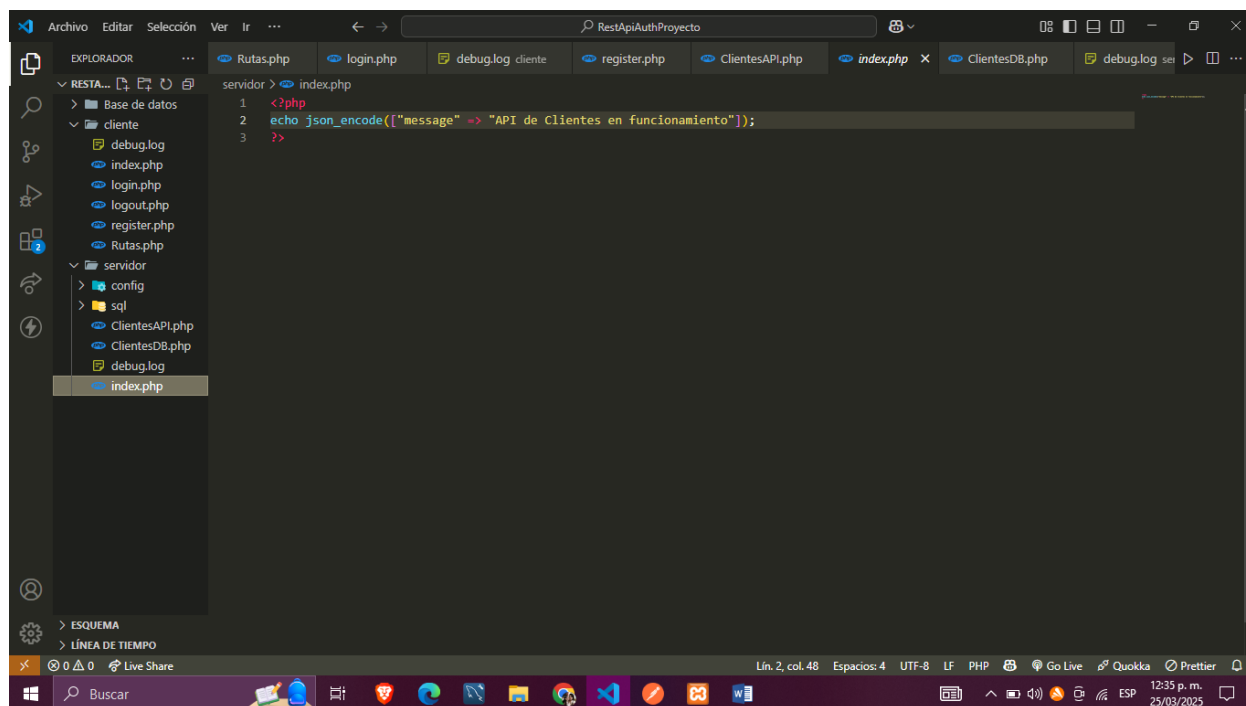
```

37 //se define la funciones privadas
38 private function registrarUsuario() {
39     try {
40         // Manejo de errores y respuesta adecuada
41         return $this->response(500, "error", $e->getMessage());
42     } catch (Exception $e) {
43         var_dump($e);
44     }
45 }
46
47 private function iniciarSesion() {
48     header("Content-Type: application/json");
49     ob_clean(); // Limpia cualquier salida anterior
50
51     $inputJSON = file_get_contents("php://input"); //esto
52     $datos = json_decode(file_get_contents("php://input"), true);
53
54     if (!$datos || !isset($datos["Correo_electronico"], $datos["Contraseña"])) {
55         $json = json_encode(["status" => "error", "message" => "Datos incompletos"]);
56         echo $json;
57         exit;
58     }
59
60     $id_cliente = $this->db->iniciarSesion($datos["Correo_electronico"], $datos["Contraseña"]);
61
62     if ($id_cliente) {
63         $token = bin2hex(random_bytes(32));
64         $this->db->guardarSesion($id_cliente, $token);
65
66         $respuesta = ["status" => "success", "message" => ["token" => $token]];
67         $json = json_encode($respuesta);
68         echo $json;
69     }
70 }

```



## 5. Se crea el archivo index.php



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

```

1  class Rutas {
2
3      /**
4       * retorna la url base de tu proyecto / aplicación
5       */
6      public function getUrlBase() {
7          return $this->urlBase;
8      }
9
10     public function getUrlFront() {
11         return $this->urlBase."/cliente"; // URL / DIRECTORIO DE FRONT
12     }
13
14     public function getUrlApi()
15     {
16         return $this->urlBase."/servidor"; // URL / DIRECTORIO DE TU API
17     }
18
19     // AQUÍ VAN RUTAS API
20
21     public function getRegisterApiUrl()
22     {
23         return $this->getUrlApi() ."/ClientesAPI.php?action=register";
24     }
25     public function getloginApiUrl()
26     {
27         return $this->getUrlApi() ."/ClientesAPI.php?action=login";
28     }
29
30     public function getlogoutApiUrl(string $token): string
31     {
32         return $this->getUrlApi() ."/ClientesAPI.php?action=logout&token='".$token;
33     }
34 }

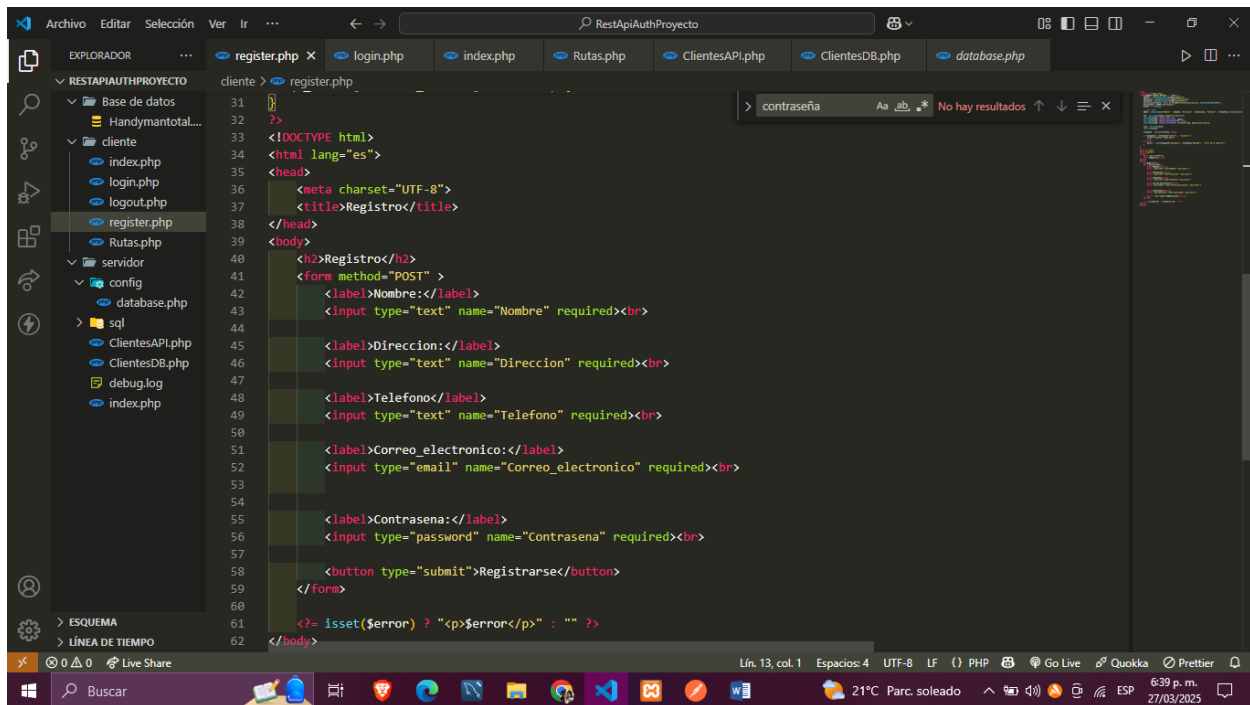
```

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

```

1  <?php
2  require_once('Rutas.php');
3  if ($SERVER["REQUEST_METHOD"] == "POST") {
4      $nombre = htmlspecialchars($_POST["Nombre"]);
5      $direccion = htmlspecialchars($_POST["Direccion"]);
6      $telefono = htmlspecialchars($_POST["Telefono"]);
7      $correo_electronico = filter_var($_POST["Correo_electronico"], FILTER_SANITIZE_EMAIL);
8      $contrasena = $_POST["Contrasena"];
9      $rutas = new Rutas();
10
11      // Crear JSON
12      $data = json_encode(["Nombre" => $nombre, "Direccion" => $direccion, "Telefono" => $telefono, "Correo_electronico" => $correo_electronico, "Contrasena" => $contrasena]);
13
14      $ch = curl_init($rutas->getRegisterApiUrl());
15      curl_setopt($ch, CURLOPT_POST, 1);
16      curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
17      curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
18      curl_setopt($ch, CURLOPT_HTTPHEADER, ["Content-Type: application/json"]);
19
20      $res = curl_exec($ch);
21      curl_close($ch);
22
23      $response = json_decode($res, true);
24
25      if ($response && $response["status"] == "success") {
26          header("Location: login.php");
27          exit;
28      } else {
29          $error = isset($response["message"]) ? $response["message"] : "Error en el registro";
30      }
31  }

```

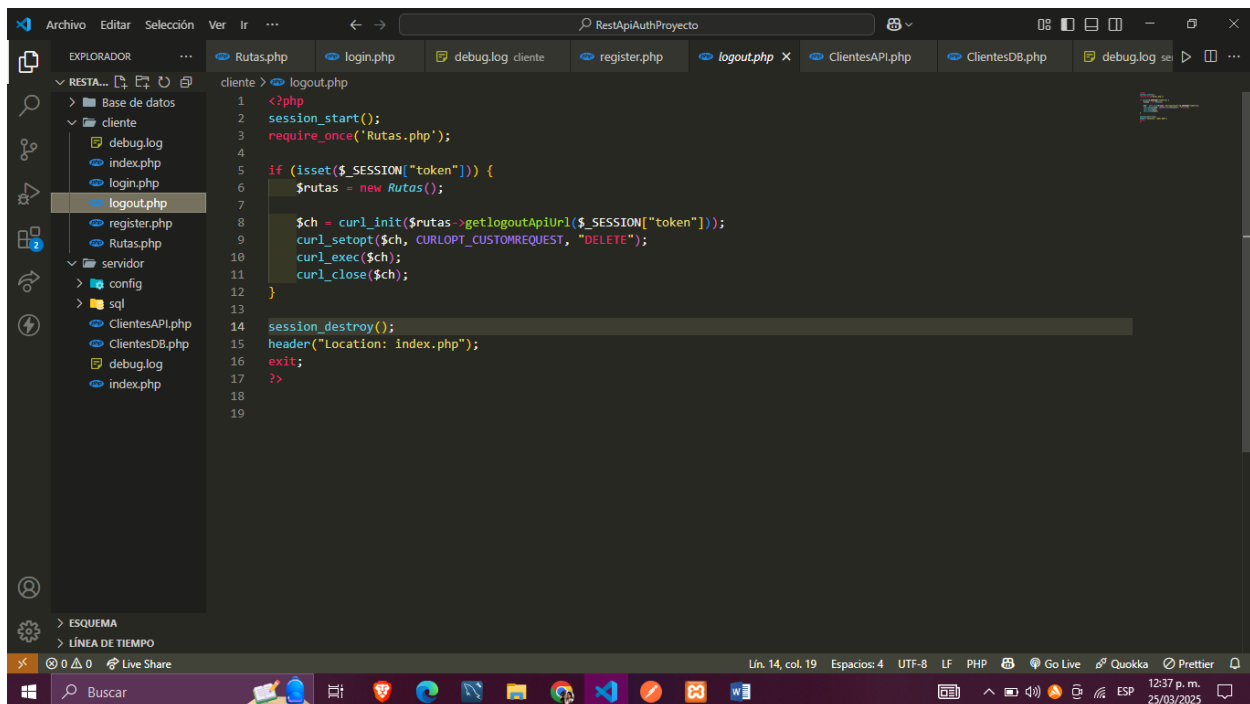


```

31 <?php
32 >
33 <!DOCTYPE html>
34 <html lang="es">
35 <head>
36 <meta charset="UTF-8">
37 <title>Registro</title>
38 </head>
39 <body>
40 <h2>Registro</h2>
41 <form method="POST">
42 <label>Nombre:</label>
43 <input type="text" name="Nombre" required><br>
44
45 <label>Direccion:</label>
46 <input type="text" name="Direccion" required><br>
47
48 <label>Telefono:</label>
49 <input type="text" name="Telefono" required><br>
50
51 <label>Correo_electronico:</label>
52 <input type="email" name="Correo_electronico" required><br>
53
54 <label>Contraseña:</label>
55 <input type="password" name="Contraseña" required><br>
56
57 <button type="submit">Registrarse</button>
58 </form>
59
60 <?= isset($error) ? "<p>$error</p>" : "" >
61
62 </body>

```

## 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    $sch = curl_init($rutas->getLoginApiUrl());
13    curl_setopt($sch, CURLOPT_POST, 1);
14    curl_setopt($sch, CURLOPT_POSTFIELDS, $data);
15    curl_setopt($sch, CURLOPT_RETURNTRANSFER, true);
16    curl_setopt($sch, CURLOPT_HTTPHEADER, ["Content-Type: application/json"]);
17
18    $res = curl_exec($sch);
19    curl_close($sch);
20
21    // Decodificar la respuesta JSON
22    $response = json_decode($res, true);
23
24    // Verificar la estructura de la respuesta
25    if ($response && isset($response["status"]) && $response["status"] == "success") {
26        if (isset($response["message"]["token"])) {
27            $_SESSION["cliente"] = $correo_electronico;
28            $_SESSION["token"] = $response["message"]["token"];
29            header("Location: index.php");
30            // exit;
31        }
32    }
33    else {
34        $error = "Error en la autenticación";
35    }
36    else {
37        $error = "Credenciales incorrectas";
38    }
39
40    <?php
41    <DOCTYPE html>
42    <html lang="es">
43    <head>
44        <meta charset="UTF-8">
45        <title>Login</title>
46    </head>
47    <body>
48        <h2>Iniciar Sesión</h2>
49        <form method="POST">
50            <label>Correo_electronico:</label>
51            <input type="text" name="Correo_electronico" required><br>
52            <label>Contrasena:</label>
53            <input type="password" name="Contrasena" required><br>
54            <button type="submit">Ingresar</button>
55        </form>
56        <?php
57        <?= isset($error) ? "<p>$error</p>" : "" ?>
58    </body>
59    </html>

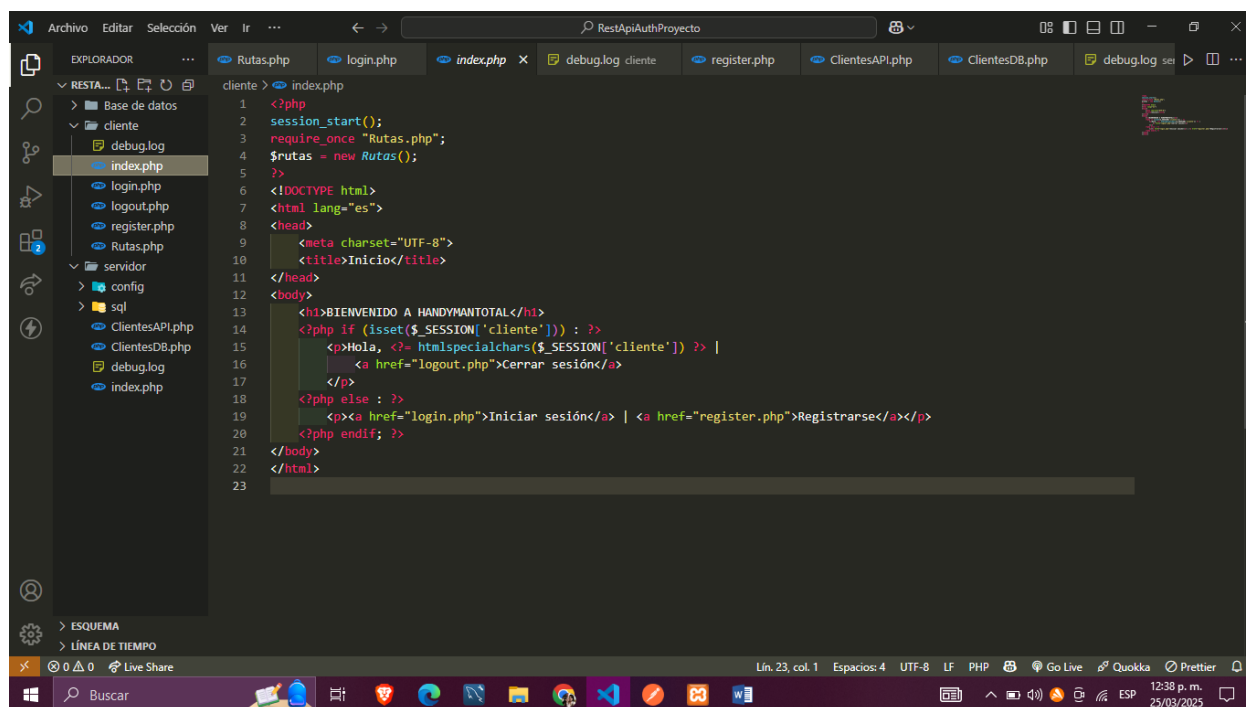
```

```

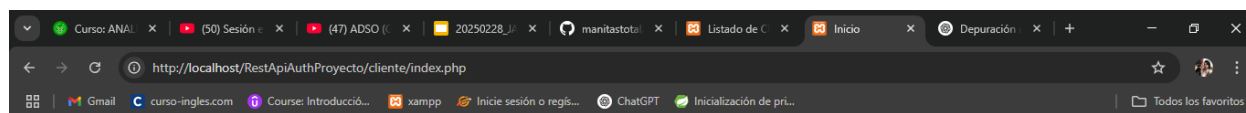
1 <?php
2 session_start();
3 require_once('Rutas.php');
4
5 if ($_SERVER["REQUEST_METHOD"] == "GET") {
6     if ($response && isset($response["status"]) && $response["status"] == "success") {
7         if (isset($response["message"]["token"])) {
8             $_SESSION["cliente"] = $correo_electronico;
9             $_SESSION["token"] = $response["message"]["token"];
10            header("Location: index.php");
11            // exit;
12        }
13    }
14    else {
15        $error = "Error en la autenticación";
16    }
17    else {
18        $error = "Credenciales incorrectas";
19    }
20
21    <?php
22    <DOCTYPE html>
23    <html lang="es">
24    <head>
25        <meta charset="UTF-8">
26        <title>Login</title>
27    </head>
28    <body>
29        <h2>Iniciar Sesión</h2>
30        <form method="POST">
31            <label>Correo_electronico:</label>
32            <input type="text" name="Correo_electronico" required><br>
33            <label>Contrasena:</label>
34            <input type="password" name="Contrasena" required><br>
35            <button type="submit">Ingresar</button>
36        </form>
37        <?php
38        <?= isset($error) ? "<p>$error</p>" : "" ?>
39    </body>
40    </html>

```

## 10. Creamos el archivo index.php



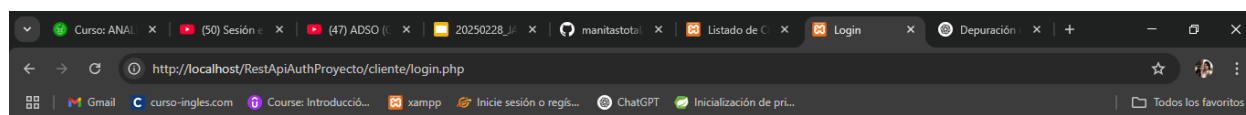
## 11. Registro e inicio de sesión en funcionamiento



### BIENVENIDO A HANDYMANTOTAL

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

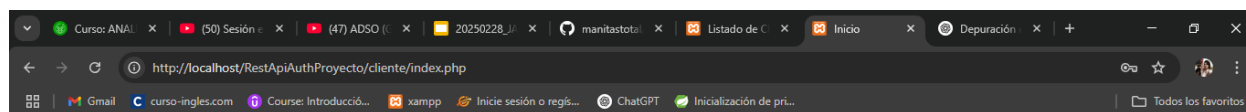




## Iniciar Sesión

Correo\_electronico:

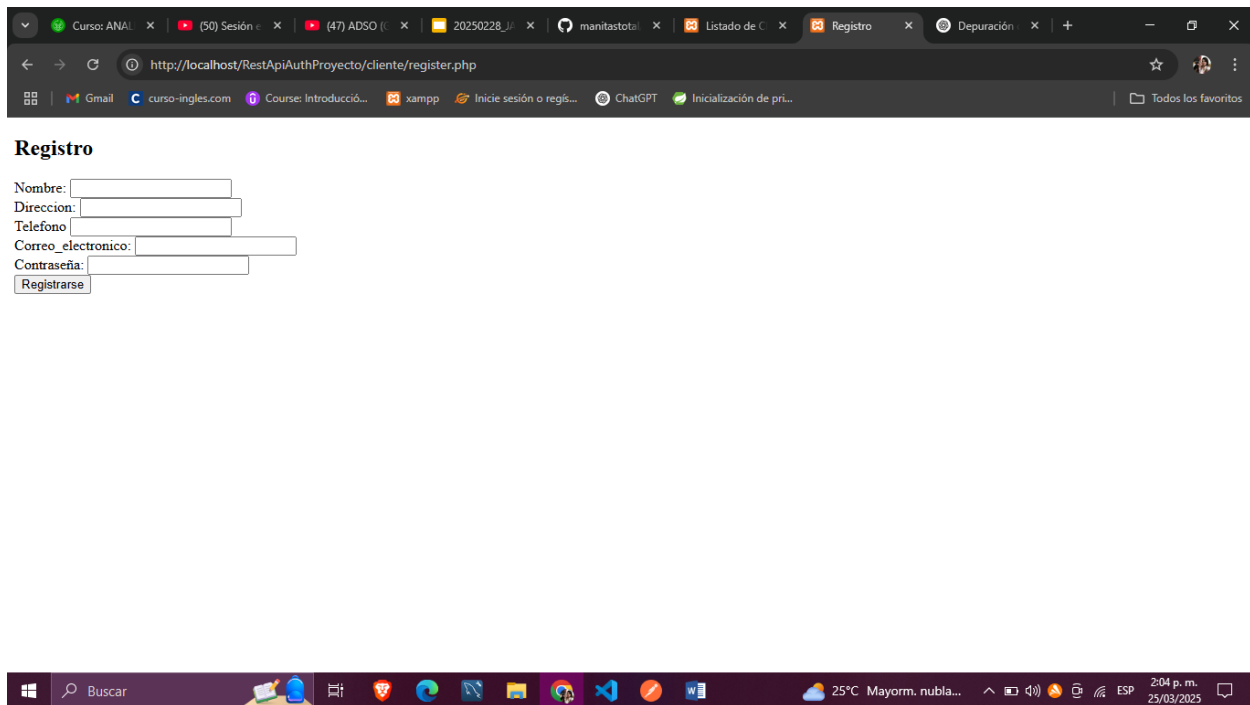
Contraseña:



## BIENVENIDO A HANDYMANTOTAL

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





### 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>