

Universidad Don Bosco



Asignatura:

Desarrollo de Aplicaciones Web con Software Interpretado en el Servidor

Investigación Aplicada 01:

Estrategias para Escalar Aplicaciones Web

Docente: Ing. Jorge Rivera

Integrantes del grupo:

Fernando Josue Anzora Aquino - AA222744
David Alejandro Alvarez Moreira - AM240104
Asly Eduvina Hernandez Melara - HM240071
Dylan Alfonso Quintanilla Rivera - QR240095
Ashley Gabriela Valdez González - VG240970

Documentación del Proceso:

- En el panel de control de XAMPP habilitaremos Apache junto a MySQL para que creamos la base de datos llamada “investigacionaplicada” que va a contener una tabla “usuarios” con 5 columnas (id, nombre, email, password, teléfono), para guardar las solicitudes que realicemos.

Servidor: 127.0.0.1 » Base de datos: investigacionaplicada » Tabla: usuarios

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0002 segundos.)

```
SELECT * FROM `usuarios`
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 ▼

Opciones extra

id	nombre	email	password	telefono
1	Achi	achi@practica.com	Achi	7191 6676
2	David	David@practica.com	David	7866 5007
0	Dylan	dylan@practica.com	Dylan	6858 7234

☐ Mostrar todo | Número de filas: 25 ▼

- En nuestro proyecto, dentro de la carpeta “config”, crearemos el archivo “database.php” que contendrá la conexión a la base de datos haciendo uso de la clase mysqli, además de detener la ejecución en caso de fallos.

File Edit Selection View Go Run Terminal Help ← → investigacionAplicada

EXPLORER

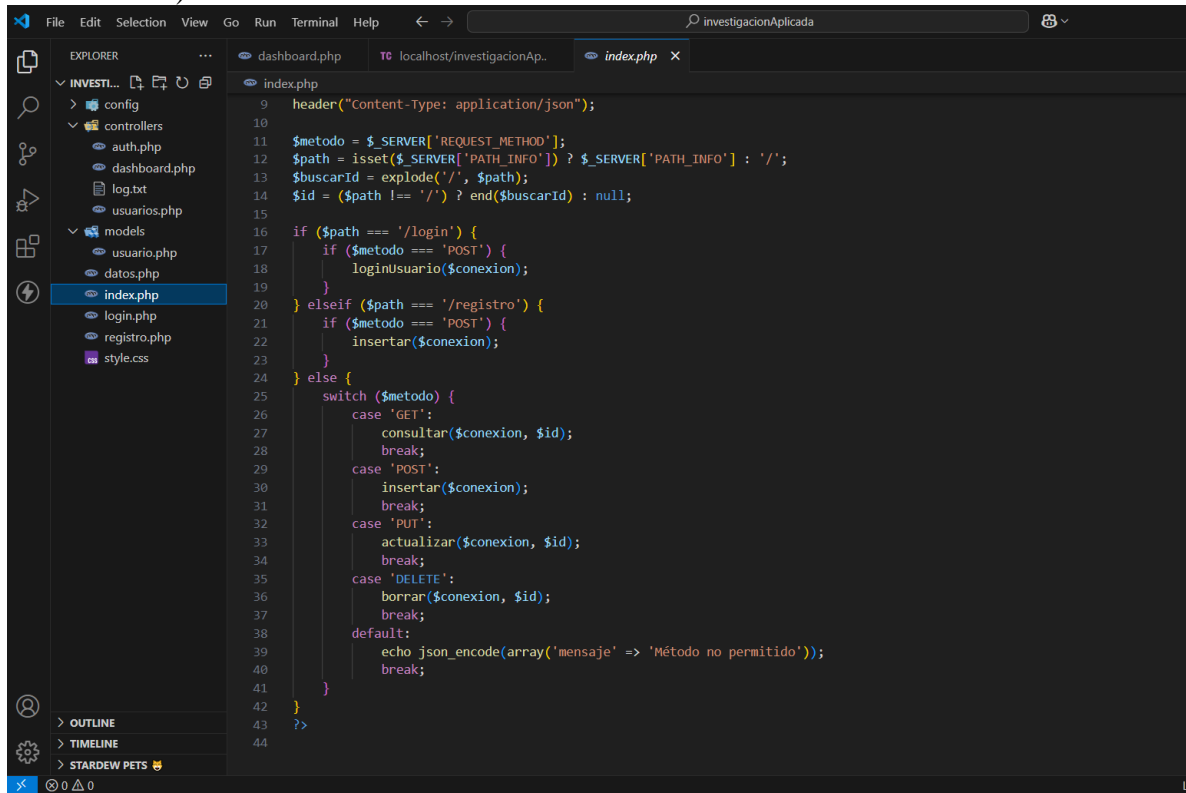
- INVESTIGACIONAPLICADA
 - config
 - database.php
 - controllers
 - auth.php
 - dashboard.php
 - usuarios.php
 - models
 - usuario.php
 - datos.php
 - index.php
 - login.php
 - registro.php
 - style.css

config > database.php

```
1 <?php
2 $host = "localhost";
3 $user = "root";
4 $password = "";
5 $database = "investigacionaplicada";
6
7 $conn = new mysqli($host, $user, $password, $database);
8
9 if ($conn->connect_error) {
10     die("Error de conexión: " . $conn->connect_error);
11 }
12 ?>
13
```

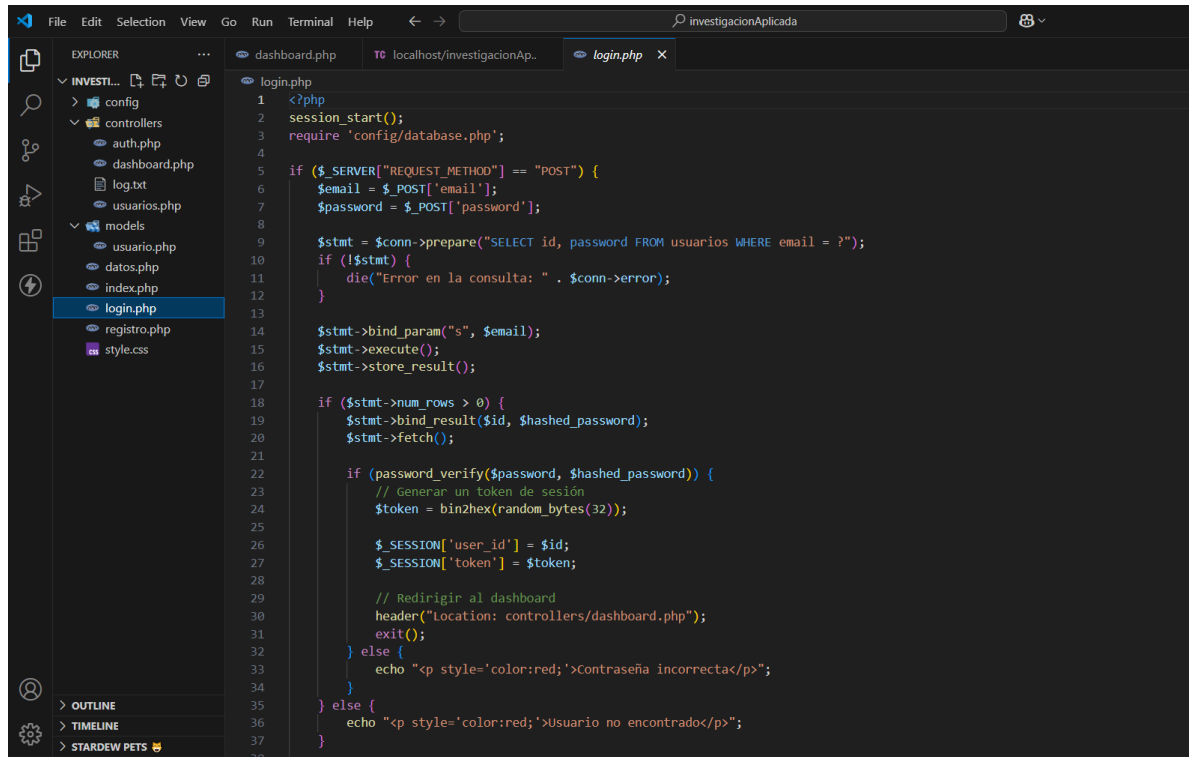
- Fuera de la carpeta “config” procederemos a crear los archivos que interactúan con la base de datos, especialmente con la de usuarios pues es donde gestionamos la autenticación, registro y CRUD de usuarios.

Index.php: es el punto de entrada a la aplicación que también lleva a login.php. Definimos las rutas de la API gestionando las solicitudes HTTP (GET, POST, PUT, DELETE).



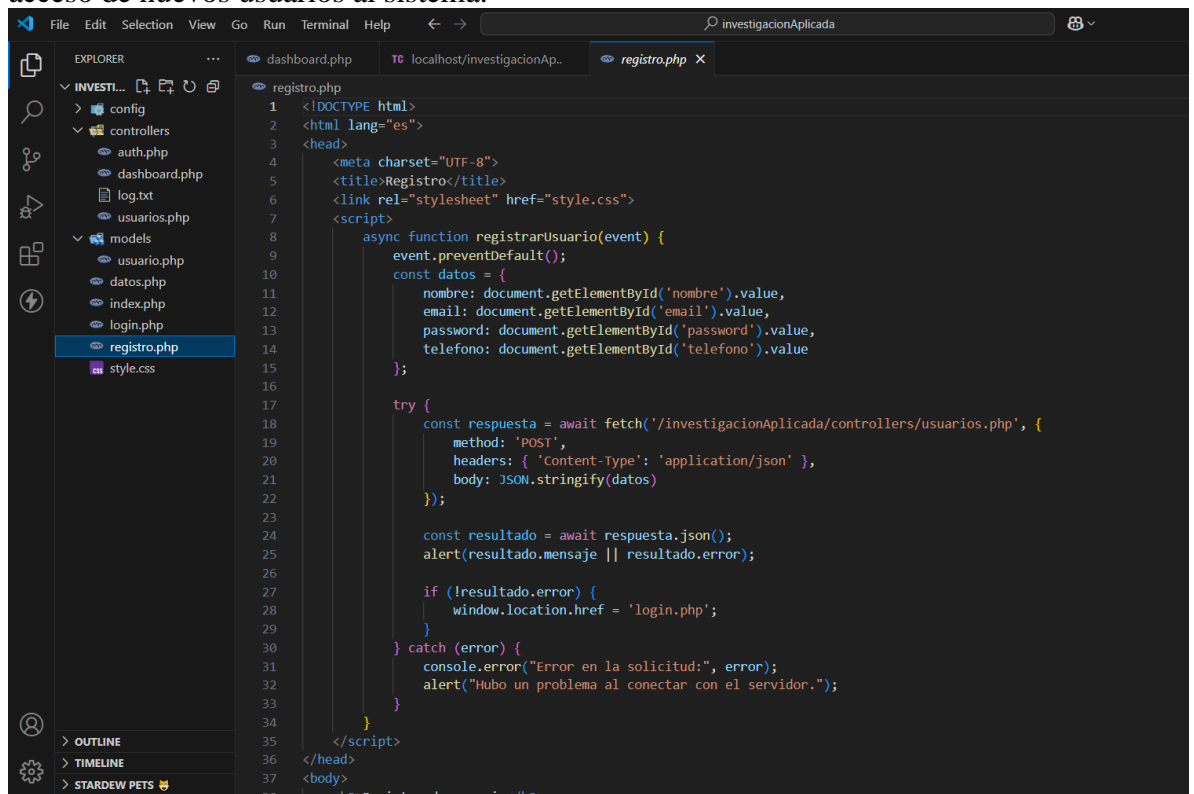
```
9 header("Content-Type: application/json");
10
11 $metodo = $_SERVER['REQUEST_METHOD'];
12 $path = isset($_SERVER['PATH_INFO']) ? $_SERVER['PATH_INFO'] : '/';
13 $buscarId = explode('/', $path);
14 $id = ($path !== '/') ? end($buscarId) : null;
15
16 if ($path === '/login') {
17     if ($metodo === 'POST') {
18         loginUsuario($conexion);
19     }
20 } elseif ($path === '/registro') {
21     if ($metodo === 'POST') {
22         insertar($conexion);
23     }
24 } else {
25     switch ($metodo) {
26         case 'GET':
27             consultar($conexion, $id);
28             break;
29         case 'POST':
30             insertar($conexion);
31             break;
32         case 'PUT':
33             actualizar($conexion, $id);
34             break;
35         case 'DELETE':
36             borrar($conexion, $id);
37             break;
38         default:
39             echo json_encode(array('mensaje' => 'Método no permitido'));
40             break;
41     }
42 }
43 ?>
44
```

Login.php: gestionará la autenticación de usuarios mediante sesiones, verifica credenciales, inicia sesión y redirige a un dashboard si el login es exitoso.



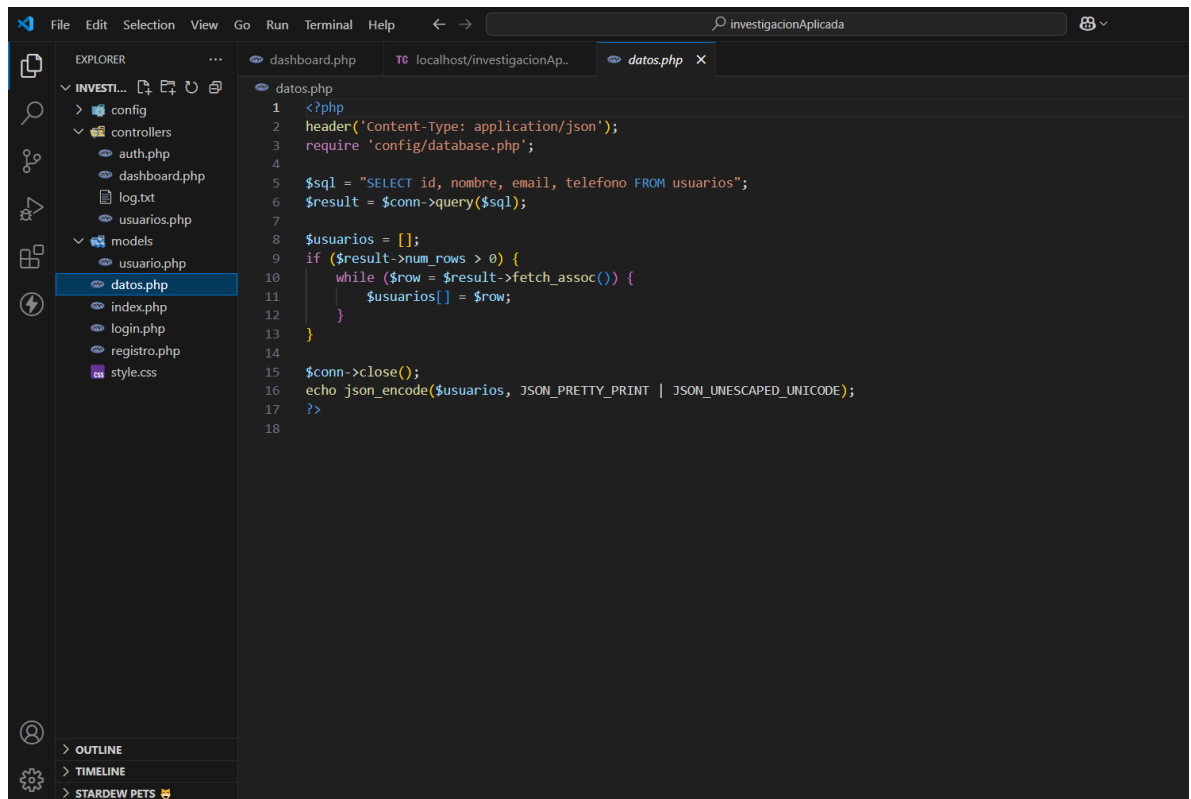
```
1 <?php
2 session_start();
3 require 'config/database.php';
4
5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6     $email = $_POST['email'];
7     $password = $_POST['password'];
8
9     $stmt = $conn->prepare("SELECT id, password FROM usuarios WHERE email = ?");
10    if (!$stmt) {
11        die("Error en la consulta: " . $conn->error);
12    }
13
14    $stmt->bind_param("s", $email);
15    $stmt->execute();
16    $stmt->store_result();
17
18    if ($stmt->num_rows > 0) {
19        $stmt->bind_result($id, $hashed_password);
20        $stmt->fetch();
21
22        if (password_verify($password, $hashed_password)) {
23            // Generar un token de sesión
24            $token = bin2hex(random_bytes(32));
25
26            $_SESSION['user_id'] = $id;
27            $_SESSION['token'] = $token;
28
29            // Redirigir al dashboard
30            header("location: controllers/dashboard.php");
31            exit();
32        } else {
33            echo "<p style='color:red;'>Contraseña incorrecta</p>";
34        }
35    } else {
36        echo "<p style='color:red;'>Usuario no encontrado</p>";
37    }
38 }
```

registro.php: será nuestra interfaz para registrar nuestros usuarios. Facilitará el acceso de nuevos usuarios al sistema.



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <title>Registro</title>
6     <link rel="stylesheet" href="style.css">
7 </head>
8 <script>
9     async function registrarUsuario(event) {
10         event.preventDefault();
11         const datos = {
12             nombre: document.getElementById('nombre').value,
13             email: document.getElementById('email').value,
14             password: document.getElementById('password').value,
15             telefono: document.getElementById('telefono').value
16         };
17
18         try {
19             const respuesta = await fetch('/investigacionAplicada/controllers/usuarios.php', {
20                 method: 'POST',
21                 headers: { 'Content-Type': 'application/json' },
22                 body: JSON.stringify(datos)
23             });
24
25             const resultado = await respuesta.json();
26             alert(resultado.mensaje || resultado.error);
27
28             if (!resultado.error) {
29                 window.location.href = 'login.php';
30             }
31         } catch (error) {
32             console.error("Error en la solicitud:", error);
33             alert("Hubo un problema al conectar con el servidor.");
34         }
35     }
36 </script>
37 </body>
```

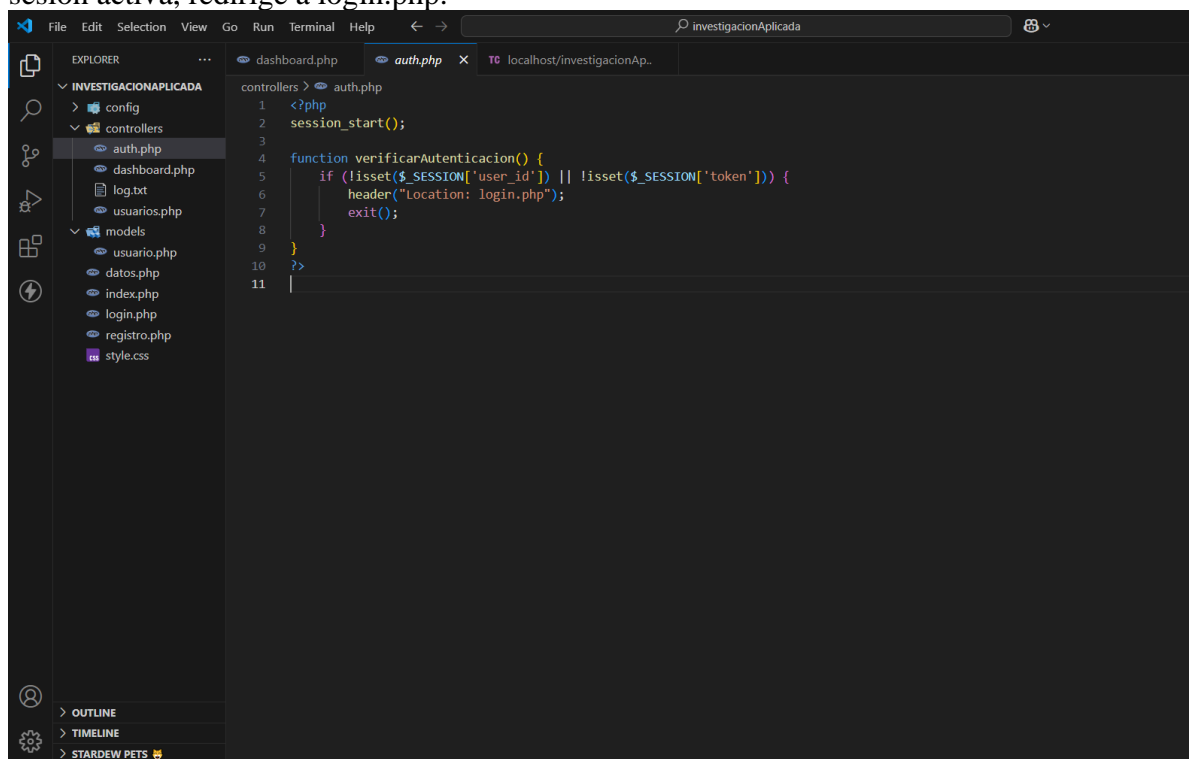
datos.php: recupera y devuelve la lista de usuarios en formato JSON de forma estructurada.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure with folders 'config', 'controllers', and 'models'. The 'controllers' folder is expanded, showing files 'auth.php', 'dashboard.php', 'log.txt', 'usuarios.php', and 'datos.php'. The 'datos.php' file is selected and its code is visible in the editor. The code is a PHP script that connects to a database, queries a table named 'usuarios', and returns the results as a JSON array.

```
1 <?php
2 header('Content-Type: application/json');
3 require 'config/database.php';
4
5 $sql = "SELECT id, nombre, email, telefono FROM usuarios";
6 $result = $conn->query($sql);
7
8 $usuarios = [];
9 if ($result->num_rows > 0) {
10     while ($row = $result->fetch_assoc()) {
11         $usuarios[] = $row;
12     }
13 }
14
15 $conn->close();
16 echo json_encode($usuarios, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
17 ?>
```

- Crearemos una nueva carpeta llamada “Controllers” que manejará parte de la lógica del sistema, autenticación, aparte de relacionarse con otros archivos del sistema.
auth.php: se encargará de comprobar si el usuario ha iniciado sesión, si no hay una sesión activa, redirige a login.php.

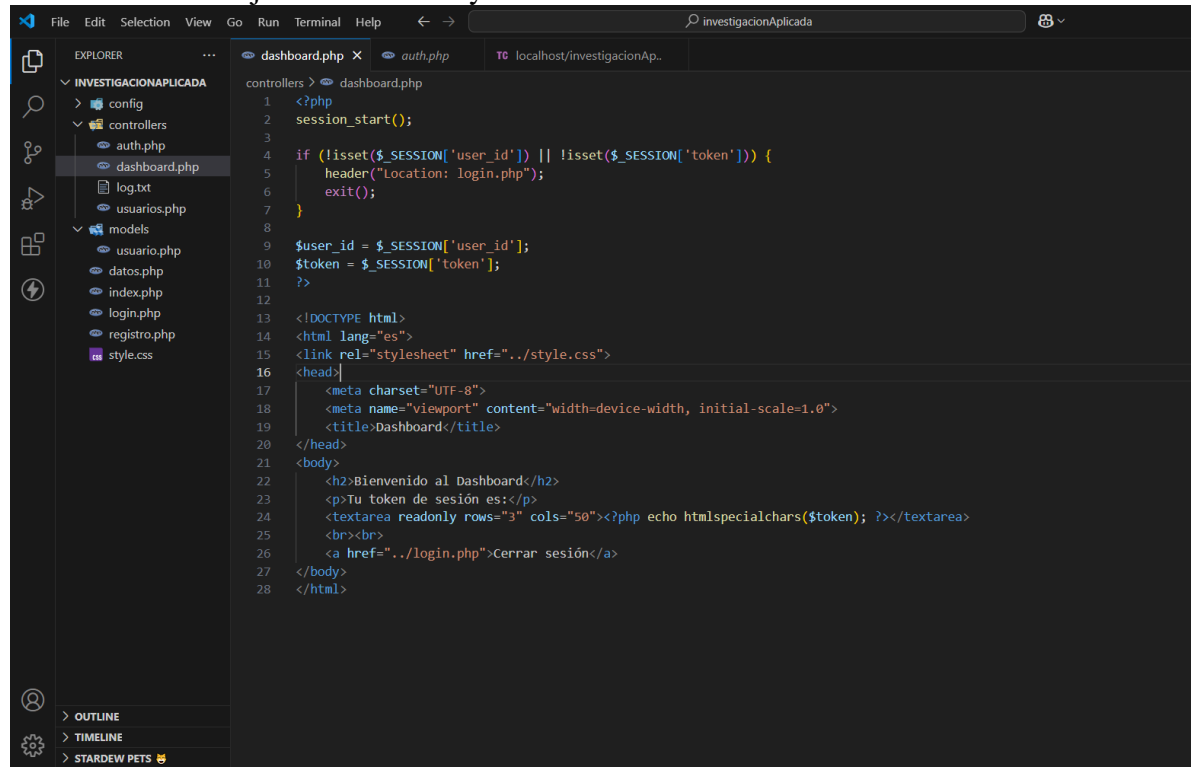


The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the same project structure as the previous image. The 'controllers' folder is expanded, and the 'auth.php' file is selected. The code in the editor shows the start of a session and a function to verify authentication. If the session is not active, it redirects the user to login.php.

```
1 <?php
2 session_start();
3
4 function verificarAutenticacion() {
5     if (!isset($_SESSION['user_id']) || !isset($_SESSION['token'])) {
6         header("Location: login.php");
7         exit();
8     }
9 }
10 ?>
11
```

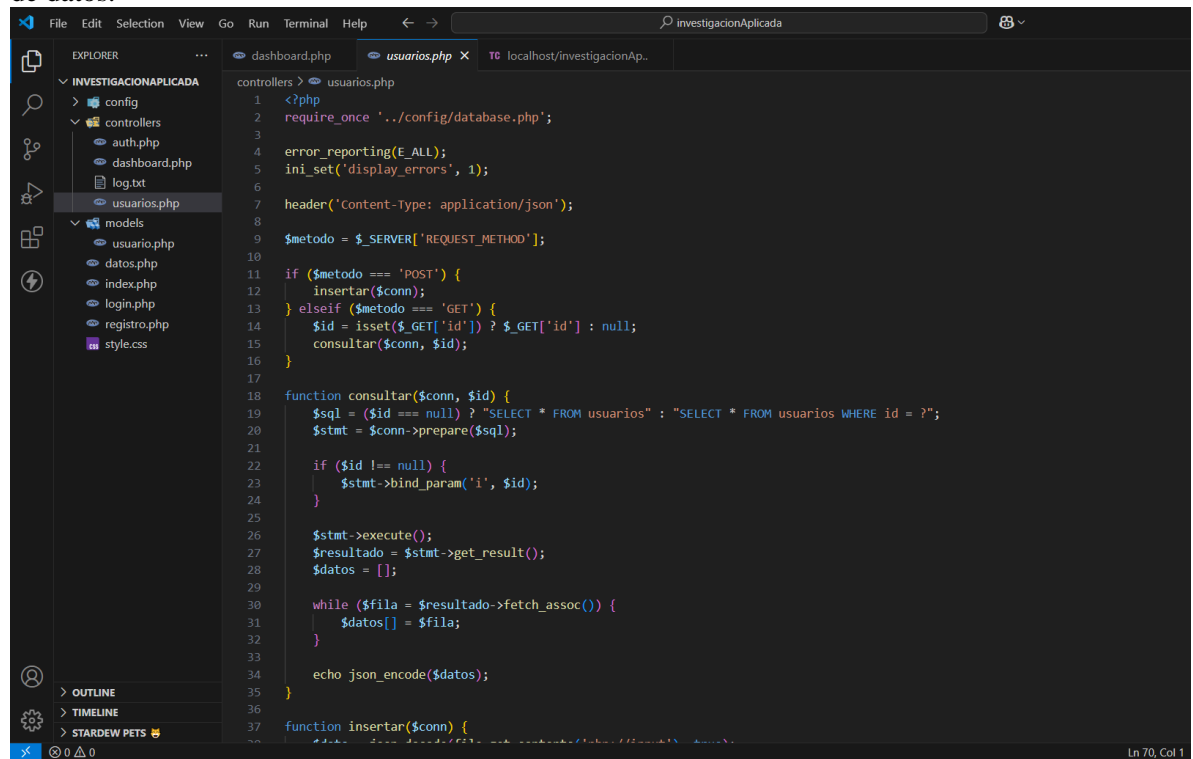
dashboard.php: para esto, se requiere que el usuario esté autenticado, si es así,

mostrará un mensaje de bienvenida y el token de sesión del usuario.



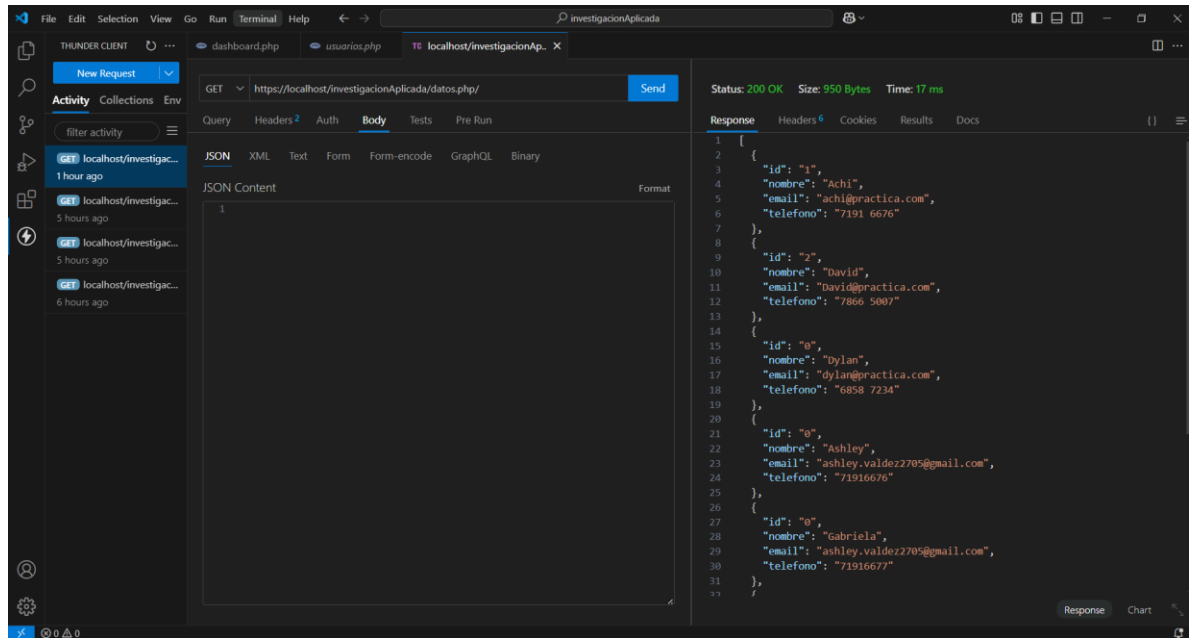
```
1 <?php
2 session_start();
3
4 if (!isset($_SESSION['user_id']) || !isset($_SESSION['token'])) {
5     header("Location: login.php");
6     exit();
7 }
8
9 $user_id = $_SESSION['user_id'];
10 $token = $_SESSION['token'];
11 ?>
12
13 <!DOCTYPE html>
14 <html lang="es">
15 <link rel="stylesheet" href="../../style.css">
16 <head>
17     <meta charset="UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0">
19     <title>Dashboard</title>
20 </head>
21 <body>
22     <h2>Bienvenido al Dashboard</h2>
23     <p>Tu token de sesión es:</p>
24     <textarea readonly rows="3" cols="50"><?php echo htmlspecialchars($token); ?></textarea>
25     <br><br>
26     <a href="../../login.php">Cerrar sesión</a>
27 </body>
28 </html>
```

usuarios.php: manejará peticiones HTTP para interactuar con la tabla “usuarios” en la base de datos.



```
1 <?php
2 require_once '../config/database.php';
3
4 error_reporting(E_ALL);
5 ini_set('display_errors', 1);
6
7 header('Content-Type: application/json');
8
9 $metodo = $_SERVER['REQUEST_METHOD'];
10
11 if ($metodo === 'POST') {
12     insertar($conn);
13 } elseif ($metodo === 'GET') {
14     $id = isset($_GET['id']) ? $_GET['id'] : null;
15     consultar($conn, $id);
16 }
17
18 function consultar($conn, $id) {
19     $sql = ($id === null) ? "SELECT * FROM usuarios" : "SELECT * FROM usuarios WHERE id = ?";
20     $stmt = $conn->prepare($sql);
21
22     if ($id !== null) {
23         $stmt->bind_param('i', $id);
24     }
25
26     $stmt->execute();
27     $resultado = $stmt->get_result();
28     $datos = [];
29
30     while ($fila = $resultado->fetch_assoc()) {
31         $datos[] = $fila;
32     }
33
34     echo json_encode($datos);
35 }
36
37 function insertar($conn) {
38     // ...
39 }
```

Haciendo uno de la extensión de Thunder Client, seleccionaremos New Request y en el apartado de URL, pondremos: <http://localhost/investigacionAplicada/archivo.php>.



- Finalmente crearemos una carpeta llamada “Models” quien definirá la función para obtener un usuario de la base de datos a partir de su correo electrónico.

usuario.php: realizará una consulta SQL para buscar un usuario en la tabla “usuarios” con el email proporcionado.

