2025

CampusFP

Mario del rey

14-2-2025

HITO 2 DEL 2º TRIMESTRE DE Programación

Tabla de contenido

[Estructura 2](#_Toc190021136)

[Código 2](#_Toc190021137)

[Conexión.php 2](#_Toc190021138)

[Controladores.php 2](#_Toc190021139)

[Modelo.php 7](#_Toc190021140)

[Agregar\_tarea.php 9](#_Toc190021141)

[Cerrar\_sesion.php 11](#_Toc190021142)

[Login.php 11](#_Toc190021143)

[Register.php 14](#_Toc190021144)

[Tareas.php 16](#_Toc190021145)

[Index.php 20](#_Toc190021146)

# Estructura

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

# Código

## Conexión.php

<?php

class Conexion {

    private $servidor = 'localhost';

    private $usuario = 'root';

    private $password = 'curso';

    private $base\_datos = 'hito2';

    public $conexion;

    public function \_\_construct() {

        $this->conexion = new mysqli($this->servidor, $this->usuario, $this->password, $this->base\_datos);

        if ($this->conexion->connect\_error) {

            die("Error de conexión: " . $this->conexion->connect\_error);

        }

    }

    // Método para cerrar la conexión

    public function cerrar() {

        $this->conexion->close();

    }

}

?>

## Controladores.php

<?php

ob\_start(); // Activa el almacenamiento en búfer de salida

require\_once '../modulo/modulo.php';

// Función auxiliar para asegurar que la sesión esté activa.

function ensureSession() {

    if (session\_status() !== PHP\_SESSION\_ACTIVE) {

        session\_set\_cookie\_params(0, '/');

        session\_start();

    }

}

class UsuarioControlador {

    private $usuario;

    public function \_\_construct() {

        $this->usuario = new usuario();

    }

    // Función registrar: Registra un nuevo usuario y crea la sesión

    public function registrar() {

        ensureSession();

        $nombre = $\_POST['nombre'];

        $correo = $\_POST['correo'];

        $pass = $\_POST['pass'];

        // Verificar si el correo ya está registrado

        if ($this->usuario->correo\_existe($correo)) {

            header("Location: ../vista/register.php?error=correo\_existe");

            exit();

        }

        $passHashed = password\_hash($pass, PASSWORD\_BCRYPT);

        if ($this->usuario->agregar\_usuario($nombre, $correo, $passHashed)) {

            // Almacena el ID del nuevo usuario en la sesión

            $\_SESSION['usuario'] = $this->usuario->conexion->conexion->insert\_id;

            header("Location: ../vista/tareas.php");

            exit();

        } else {

            header("Location: ../vista/register.php?error=registrar");

            exit();

        }

    }

    // Función iniciar\_sesion: Valida las credenciales del usuario e inicia la sesión

    public function iniciar\_sesion() {

        ensureSession();

        $correo = $\_POST['correo'];

        $pass = $\_POST['pass'];

        $result = $this->usuario->obtener\_usuario($correo);

        if ($result->num\_rows > 0) {

            $user = $result->fetch\_assoc();

            if (password\_verify($pass, $user['pass'])) {

                // Regenera el ID de sesión por seguridad y establece la sesión con el ID del usuario

                session\_regenerate\_id(true);

                $\_SESSION['usuario'] = $user['id\_usuario'];

                header("Location: ../vista/tareas.php");

                exit();

            } else {

                header("Location: ../vista/login.php?error=contraseña");

                exit();

            }

        } else {

            header("Location: ../vista/login.php?error=usuario");

            exit();

        }

    }

    // Función cerrar\_sesion: Finaliza la sesión activa y redirige al inicio

    public function cerrar\_sesion() {

        ensureSession();

        session\_destroy();

        header("Location: ../index.php");

        exit();

    }

}

class TareaControlador {

    private $tarea;

    public function \_\_construct() {

        $this->tarea = new tarea();

    }

    // Función agregar\_tarea: Agrega una nueva tarea para el usuario autenticado

    public function agregar\_tarea() {

        $this->requireLogin();

        $nombre = $\_POST['nombre'];

        $descripcion = $\_POST['descripcion'];

        $estado = 'pendiente';

        $usuario\_id = $\_SESSION['usuario'];

        if ($this->tarea->agregar\_tarea($nombre, $descripcion, $estado, $usuario\_id)) {

            header("Location: ../vista/tareas.php");

            exit();

        } else {

            header("Location: ../vista/tareas.php?error=agregar\_tarea");

            exit();

        }

    }

    // Función listar\_tareas: Retorna las tareas pendientes del usuario autenticado

    public function listar\_tareas() {

        $this->requireLogin();

        $usuario\_id = $\_SESSION['usuario'];

        return $this->tarea->listar\_tareas($usuario\_id);

    }

    // Función eliminar\_tarea: Elimina una tarea específica

    public function eliminar\_tarea() {

        if ($this->tarea->eliminar\_tarea($\_POST['tarea\_id'])) {

            header("Location: ../vista/tareas.php");

            exit();

        } else {

            header("Location: ../vista/tareas.php?error=eliminar\_tarea");

            exit();

        }

    }

    // Función modificar\_tarea: Modifica los datos de una tarea existente

    public function modificar\_tarea() {

        if ($this->tarea->modificar\_tarea($\_POST['tarea\_id'], $\_POST['nombre'], $\_POST['descripcion'], $\_POST['estado'])) {

            header("Location: ../vista/tareas.php");

            exit();

        } else {

            header("Location: ../vista/tareas.php?error=modificar\_tarea");

            exit();

        }

    }

    // Función completar\_tarea: Marca una tarea como completada

    public function completar\_tarea() {

        $this->requireLogin();

        $tarea\_id = $\_POST['tarea\_id'];

        if ($this->tarea->completar\_tarea($tarea\_id)) {

            header("Location: ../vista/tareas.php");

            exit();

        } else {

            header("Location: ../vista/tareas.php?error=completar\_tarea");

            exit();

        }

    }

    // Función requireLogin: Garantiza que el usuario esté autenticado antes de continuar

    private function requireLogin() {

        ensureSession();

        if (!isset($\_SESSION['usuario'])) {

            header("Location: ../index.php");

            exit();

        }

    }

}

// Enrutador: Gestiona las solicitudes en función del parámetro "action"

if (isset($\_GET['action'])) {

    $action = $\_GET['action'];

    if (in\_array($action, ['registrar', 'iniciar\_sesion', 'cerrar\_sesion'])) {

        $controller = new UsuarioControlador();

        switch ($action) {

            case 'registrar':      $controller->registrar();       break;

            case 'iniciar\_sesion': $controller->iniciar\_sesion();   break;

            case 'cerrar\_sesion':  $controller->cerrar\_sesion();    break;

        }

    } elseif (in\_array($action, ['agregar\_tarea', 'listar\_tareas', 'eliminar\_tarea', 'modificar\_tarea', 'completar\_tarea'])) {

        $controller = new TareaControlador();

        switch ($action) {

            case 'agregar\_tarea':   $controller->agregar\_tarea();   break;

            case 'listar\_tareas':   $controller->listar\_tareas();   break;

            case 'eliminar\_tarea':  $controller->eliminar\_tarea();  break;

            case 'modificar\_tarea': $controller->modificar\_tarea(); break;

            case 'completar\_tarea': $controller->completar\_tarea(); break;

        }

    }

}

?>

## Modelo.php

<?php

require\_once '../configuracion/conexion.php';

// Clase para gestionar operaciones sobre la tabla "usuarios"

class usuario {

    public $conexion;

    public function \_\_construct() {

        $this->conexion = new Conexion();

    }

    // Función para crear un usuario: retorna true si se crea correctamente

    public function agregar\_usuario($nombre, $correo, $pass) {

        $query = "INSERT INTO usuarios(nombre, correo, pass) VALUES (?, ?, ?)";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("sss", $nombre, $correo, $pass);

        return $stmt->execute();

    }

    // Nuevo método para verificar si un correo ya existe

    public function correo\_existe($correo) {

        $query = "SELECT id\_usuario FROM usuarios WHERE correo = ?";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("s", $correo);

        $stmt->execute();

        $result = $stmt->get\_result();

        return $result->num\_rows > 0;

    }

    // Función para obtener un usuario (para el login) utilizando el correo

    public function obtener\_usuario($correo) {

        $query = "SELECT \* FROM usuarios WHERE correo = ?";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("s", $correo);

        $stmt->execute();

        return $stmt->get\_result();

    }

}

// Clase para gestionar operaciones sobre la tabla "tareas"

class tarea {

    public $conexion;

    public function \_\_construct() {

        $this->conexion = new Conexion();

    }

    // Función para crear una tarea: retorna true si se crea correctamente

    public function agregar\_tarea($nombre, $descripcion, $estado, $usuario) {

        $query = "INSERT INTO tareas(nombre, descripcion, estado, id\_usuario) VALUES (?, ?, ?, ?)";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("sssi", $nombre, $descripcion, $estado, $usuario);

        return $stmt->execute();

    }

    // Función para listar tareas pendientes de un usuario

    public function listar\_tareas($usuario\_id) {

        // Se añade restricción para que solo muestre tareas "pendiente"

        $query = "SELECT \* FROM tareas WHERE id\_usuario = ? AND estado = 'pendiente'";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("i", $usuario\_id);

        $stmt->execute();

        return $stmt->get\_result();

    }

    // Función para eliminar una tarea: retorna true si se elimina correctamente

    public function eliminar\_tarea($tarea\_id) {

        $query = "DELETE FROM tareas WHERE id = ?";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("i", $tarea\_id);

        return $stmt->execute();

    }

    // Función para modificar una tarea: retorna true si se modifica correctamente

    public function modificar\_tarea($tarea\_id, $nombre, $descripcion, $estado) {

        $query = "UPDATE tareas SET nombre = ?, descripcion = ?, estado = ? WHERE id = ?";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("sssi", $nombre, $descripcion, $estado, $tarea\_id);

        return $stmt->execute();

    }

    // Nueva función para marcar una tarea como completada

    public function completar\_tarea($tarea\_id) {

        // Actualiza la consulta para utilizar el campo correcto: id\_tarea

        $query = "UPDATE tareas SET estado = 'completada' WHERE id\_tarea = ?";

        $stmt = $this->conexion->conexion->prepare($query);

        $stmt->bind\_param("i", $tarea\_id);

        return $stmt->execute();

    }

}

?>

## Agregar\_tarea.php

<?php

// Verificar la sesión activa.

session\_start();

if (!isset($\_SESSION['usuario'])) {

    // Usuario no autenticado, redirigimos a index.php.

    header("Location: ../index.php");

    exit();

}

// Obtener el nombre del usuario activo (copia del encabezado de tareas.php).

$userId = $\_SESSION['usuario'];

require\_once '../modulo/modulo.php';

$usuarioObj = new usuario();

// Consulta para obtener el nombre del usuario.

$query = "SELECT nombre FROM usuarios WHERE id\_usuario = ?";

$stmt = $usuarioObj->conexion->conexion->prepare($query);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

$row = $result->fetch\_assoc();

$nombreUsuario = $row ? $row['nombre'] : 'Desconocido';

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <!-- HEAD: Metadatos, título y enlaces a CSS -->

    <meta charset="UTF-8">

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

    <title>Agregar Tarea</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

    <style>

      /\* Estilos para la navbar y el dropdown \*/

      .navbar {

        box-shadow: 0 2px 4px rgba(0,0,0,0.1);

      }

      .navbar-brand {

        font-weight: bold;

        color: #ffcc00 !important;

      }

      .dropdown-menu {

        border: 1px solid #ffcc00;

        background: #222;

      }

      .dropdown-item {

        color: #ffcc00;

      }

      .dropdown-item:hover {

        background-color: #ffcc00;

        color: #222;

      }

    </style>

</head>

<body>

    <!-- NAVBAR: Barra de navegación superior -->

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

      <a class="navbar-brand" href="#">Tareas</a>

      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

              aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

         <!-- Menú principal -->

         <ul class="navbar-nav mr-auto">

             <li class="nav-item">

                <a class="nav-link" href="tareas.php">Tareas</a>

             </li>

             <li class="nav-item">

                <a class="nav-link" href="agregar\_tarea.php">Agregar Tarea</a>

             </li>

         </ul>

         <!-- Información del usuario y botón para cerrar sesión -->

         <span class="navbar-text mr-3">

            <?php echo htmlspecialchars($nombreUsuario); ?>

         </span>

         <a href="../controladores/Controladores.php?action=cerrar\_sesion" class="btn btn-danger">Cerrar Sesión</a>

      </div>

    </nav>

    <!-- MAIN CONTENT: Sección principal para agregar una tarea -->

    <div class="container mt-4">

        <!-- Título de la sección -->

        <h1 class="mt-3">Agregar Tarea</h1>

        <!-- FORMULARIO: Permite introducir los datos de la tarea -->

        <form action="../controladores/Controladores.php?action=agregar\_tarea" method="post" class="mt-3">

            <!-- Campo: Nombre de la tarea -->

            <div class="form-group">

                <label for="nombre">Nombre de la tarea:</label>

                <input type="text" id="nombre" name="nombre" class="form-control" required>

            </div>

            <!-- Campo: Descripción de la tarea -->

            <div class="form-group">

                <label for="descripcion">Descripción:</label>

                <input type="text" id="descripcion" name="descripcion" class="form-control" required>

            </div>

            <!-- Botones: Enviar el formulario o cancelar -->

            <button type="submit" class="btn btn-primary">Agregar Tarea</button>

            <a href="tareas.php" class="btn btn-secondary">Volver</a>

        </form>

    </div>

    <!-- SCRIPTS: Inclusión de dependencias JavaScript para el correcto funcionamiento de Bootstrap -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

## Cerrar\_sesion.php

<?php

session\_start();

session\_destroy();

header("Location: index.php");

?>

## Login.php

<?php

// Iniciamos la sesión para gestionar la autenticación del usuario.

session\_start();

// Si ya hay un usuario autenticado, lo enviamos directamente a la página de tareas.

if (isset($\_SESSION['usuario'])) {

    header("Location: tareas.php");

    exit();

}

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <!-- HEAD: Contiene metadatos, título y enlaces a estilos -->

    <meta charset="UTF-8">

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

    <title>Iniciar Sesión</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

    <style>

      /\* Estilos para el formulario de inicio \*/

      .form-container {

          background: #fefefe;

          max-width: 400px;

          margin: 70px auto;

          padding: 40px;

          border-radius: 10px;

          border: 1px solid #e0e0e0;

          box-shadow: 0 4px 12px rgba(0,0,0,0.15);

      }

      .form-container h1 {

          text-align: center;

          margin-bottom: 25px;

          color: #2c3e50;

      }

      .form-group label {

          color: #333;

      }

      .btn-primary {

          width: 100%;

          background-color: #27ae60;

          border: none;

      }

      .btn-primary:hover {

          background-color: #2ecc71;

      }

    </style>

</head>

<body>

    <!-- BODY: Contenedor principal del documento -->

    <div class="container">

        <!-- Sección: Contenedor del formulario de inicio de sesión -->

        <div class="form-container">

            <!-- Título principal del formulario -->

            <h1 class="mt-5">Iniciar Sesión</h1>

            <?php

            // Si se pasó un error en la URL, mostramos mensajes específicos.

            if (isset($\_GET['error'])):

                if ($\_GET['error'] === 'usuario'):

                    // Error: usuario no encontrado.

            ?>

            <div class="alert alert-danger">El usuario no existe.</div>

            <?php

                elseif ($\_GET['error'] === 'contraseña'):

                    // Error: contraseña incorrecta.

            ?>

            <div class="alert alert-danger">Contraseña incorrecta.</div>

            <?php

                endif;

            endif;

            ?>

            <!-- Formulario de inicio de sesión -->

            <form action="../controladores/Controladores.php?action=iniciar\_sesion" method="post" class="mt-3">

                <!-- Campo para el correo electrónico -->

                <div class="form-group">

                    <label for="correo">Correo:</label>

                    <input type="email" id="correo" name="correo" class="form-control" required>

                </div>

                <!-- Campo para la contraseña -->

                <div class="form-group">

                    <label for="pass">Contraseña:</label>

                    <input type="password" id="pass" name="pass" class="form-control" required>

                </div>

                <!-- Botón para enviar el formulario -->

                <button type="submit" class="btn btn-primary">Iniciar Sesión</button>

            </form>

            <!-- Enlace para redirigir al registro -->

            <p class="mt-3 text-center">¿No tienes cuenta? <a href="register.php">Regístrate</a></p>

        </div>

        <!-- Fin del contenedor del formulario -->

    </div>

    <!-- Sección: Inclusión de scripts JavaScript -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

## Register.php

<?php

// Iniciamos la sesión para verificar si el usuario ya está autenticado

session\_start();

// Si el usuario ya tiene sesión iniciada, lo redirigimos a tareas.php

if (isset($\_SESSION['usuario'])) {

    header("Location: tareas.php");

    exit();

}

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <!-- Configuración de la cabecera: codificación, viewport y título de la página -->

    <meta charset="UTF-8">

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

    <title>Registrarse</title>

    <!-- Enlace a Bootstrap para estilos -->

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

    <style>

      /\* Estilos básicos para la página de registro \*/

      .form-container {

          background: #fefefe;

          max-width: 400px;

          margin: 70px auto;

          padding: 40px;

          border-radius: 10px;

          border: 1px solid #e0e0e0;

          box-shadow: 0 4px 12px rgba(0,0,0,0.15);

      }

      .form-container h1 {

          text-align: center;

          margin-bottom: 25px;

          color: #2c3e50;

      }

      .form-group label {

          color: #333;

      }

      .btn-primary {

          width: 100%;

          background-color: #27ae60;

          border: none;

      }

      .btn-primary:hover {

          background-color: #2ecc71;

      }

    </style>

    <script>

        // Función para validar que se han aceptado los términos

        function validarFormulario() {

            var aceptarTerminos = document.getElementById('aceptar\_terminos');

            if (!aceptarTerminos.checked) {

                alert('Debe aceptar los términos y condiciones para registrarse.');

                return false;

            }

            return true;

        }

    </script>

</head>

<body>

    <div class="container">

        <!-- Contenedor para el formulario de registro -->

        <div class="form-container">

            <!-- Título principal del formulario de registro -->

            <h1>Registrarse</h1>

            <?php

            // Se muestra un mensaje de error si el correo ya existe

            if (isset($\_GET['error']) && $\_GET['error'] === 'correo\_existe'):

            ?>

            <div class="alert alert-danger">

                El correo ya está registrado. Por favor, utiliza otro correo.

            </div>

            <?php

            endif;

            ?>

            <!-- Formulario para que el usuario ingrese su nombre, correo y contraseña -->

            <form action="../controladores/Controladores.php?action=registrar" method="post" class="mt-3" onsubmit="return validarFormulario()">

                <div class="form-group">

                    <!-- Campo para el nombre del usuario -->

                    <label for="nombre">Nombre:</label>

                    <input type="text" id="nombre" name="nombre" class="form-control" required>

                </div>

                <div class="form-group">

                    <!-- Campo para el correo electrónico -->

                    <label for="correo">Correo:</label>

                    <input type="email" id="correo" name="correo" class="form-control" required>

                </div>

                <div class="form-group">

                    <!-- Campo para la contraseña -->

                    <label for="pass">Contraseña:</label>

                    <input type="password" id="pass" name="pass" class="form-control" required>

                </div>

                <div class="form-group form-check">

                    <input type="checkbox" class="form-check-input" id="aceptar\_terminos" name="aceptar\_terminos" required>

                    <label class="form-check-label" for="aceptar\_terminos">Acepto los <a href="terminos.php">términos y condiciones</a></label>

                </div>

                <!-- Botón para enviar el formulario y registrar al nuevo usuario -->

                <button type="submit" class="btn btn-primary">Registrarse</button>

            </form>

            <!-- Enlace para redirigir al formulario de inicio de sesión en caso de que el usuario ya tenga una cuenta -->

            <p class="mt-3 text-center">¿Ya tienes cuenta? <a href="login.php">Inicia Sesión</a></p>

        </div>

    </div>

    <!-- Inclusión de dependencias JavaScript para Bootstrap -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

## Tareas.php

<?php

// Iniciamos la sesión y comprobamos si el usuario ha iniciado sesión. Si no, se redirige a la página de inicio.

session\_start();

if (!isset($\_SESSION['usuario'])) {

    // Usuario no autenticado, se redirige a index.php.

    header("Location: ../index.php");

    exit();

}

// Obtenemos el ID del usuario autenticado.

$userId = $\_SESSION['usuario'];

// Se incluye la clase usuario para utilizar sus métodos (por ejemplo, obtener el nombre).

require\_once '../modulo/modulo.php';

$usuarioObj = new usuario();

// Preparamos la consulta para obtener el nombre del usuario basado en su ID.

// Esto es importante para mostrar un saludo personalizado en la barra de navegación.

$query = "SELECT nombre FROM usuarios WHERE id\_usuario = ?";

$stmt = $usuarioObj->conexion->conexion->prepare($query);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

$row = $result->fetch\_assoc();

$nombreUsuario = $row ? $row['nombre'] : 'Desconocido';

// Se incluye el controlador de tareas para gestionar las operaciones (listar, agregar, etc.).

require\_once '../controladores/Controladores.php';

$tareaControlador = new TareaControlador();

// Se obtiene la lista de tareas pendientes asociadas al usuario actual.

$tareas = $tareaControlador->listar\_tareas();

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <!-- HEAD: Define la configuración básica, título y enlaces de estilos -->

    <meta charset="UTF-8">

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

    <title>Mis Tareas</title>

    <!-- Enlace a Bootstrap CSS -->

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

    <style>

      /\* CSS: Estilos personalizados para la página \*/

      body {

          background: #f7f7f7;

      }

      .navbar {

          box-shadow: 0 2px 4px rgba(0,0,0,0.1);

      }

      .navbar-brand {

          font-weight: bold;

          color: #ffcc00 !important;

      }

      .container {

          margin-top: 30px;

      }

      h1 {

          color: #2c3e50;

          text-align: center;

          margin-bottom: 20px;

      }

      table {

          background: #fff;

          border-radius: 5px;

          box-shadow: 0 2px 4px rgba(0,0,0,0.1);

      }

      th {

          background: #27ae60;

          color: #fff;

      }

      td, th {

          vertical-align: middle;

      }

      .btn-danger {

          background-color: #e74c3c;

          border: none;

      }

      .btn-danger:hover {

          background-color: #c0392b;

      }

      .btn-success {

          background-color: #27ae60;

          border: none;

      }

      .btn-success:hover {

          background-color: #2ecc71;

      }

    </style>

</head>

<body>

    <!-- NAVBAR: Barra de navegación superior -->

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

      <!-- Marca de la navbar -->

      <a class="navbar-brand" href="#">Tareas</a>

      <!-- Botón de menú para dispositivos móviles -->

      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

              aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <!-- Menú principal y opciones de sesión -->

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

         <!-- Lista de enlaces de navegación -->

         <ul class="navbar-nav mr-auto">

             <li class="nav-item">

                <a class="nav-link" href="tareas.php">Tareas</a>

             </li>

             <li class="nav-item">

                <a class="nav-link" href="agregar\_tarea.php">Agregar Tarea</a>

             </li>

         </ul>

         <!-- Saludo personalizado y botón para cerrar sesión -->

         <span class="navbar-text mr-3">

            <?php echo htmlspecialchars($nombreUsuario); ?>

         </span>

         <a href="../controladores/Controladores.php?action=cerrar\_sesion" class="btn btn-danger">Cerrar Sesión</a>

      </div>

    </nav>

    <!-- MAIN CONTENT: Sección donde se muestran y gestionan las tareas -->

    <div class="container mt-4">

        <!-- Encabezado de la sección -->

        <h1 class="mt-3">Mis Tareas</h1>

        <!-- TABLA: Lista de tareas pendientes -->

        <table class="table table-striped mt-3">

            <thead>

                <tr>

                    <!-- Encabezados de la tabla -->

                    <th>Nombre</th>

                    <th>Descripción</th>

                    <th>Estado</th>

                    <th>Acciones</th>

                </tr>

            </thead>

            <tbody>

                <?php

                // Iteramos sobre cada tarea obtenida y mostramos sus detalles en una fila de la tabla.

                while ($row = $tareas->fetch\_assoc()): ?>

                <tr>

                    <!-- Datos de la tarea -->

                    <td><?php echo $row['nombre']; ?></td>

                    <td><?php echo $row['descripcion']; ?></td>

                    <td><?php echo $row['estado']; ?></td>

                    <td>

                        <?php if ($row['estado'] !== 'completada'): ?>

                        <!-- FORM: Botón para marcar la tarea como completada -->

                        <form method="post" action="../controladores/Controladores.php?action=completar\_tarea" style="display:inline;">

                            <input type="hidden" name="tarea\_id" value="<?php echo $row['id\_tarea']; ?>">

                            <button type="submit" class="btn btn-success btn-sm">Completar</button>

                        </form>

                        <?php endif; ?>

                    </td>

                </tr>

                <?php endwhile; ?>

            </tbody>

        </table>

        <!-- Enlace: Botón para agregar una nueva tarea -->

        <a href="agregar\_tarea.php" class="btn btn-primary mt-3">Agregar Tarea</a>

    </div>

    <!-- SCRIPT: Inclusión de librerías JavaScript para Bootstrap -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

## Index.php

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

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

    <title>Inicio</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

    <style>

      body {

          background: linear-gradient(to right, #f8f9fa, #e9ecef);

          font-family: Arial, sans-serif;

      }

      .container {

          margin-top: 100px;

          text-align: center;

      }

      h1 {

          color: #343a40;

          margin-bottom: 30px;

      }

      .nav-pills .nav-link {

          background-color: #343a40;

          color: #fff;

          margin: 0 5px;

      }

      .nav-pills .nav-link.active {

          background-color: #ffcc00;

          color: #343a40;

      }

    </style>

</head>

<body>

    <div class="container">

        <h1 class="mt-5">Bienvenido</h1>

        <!-- Menú de navegación principal -->

        <nav class="mt-3">

            <ul class="nav nav-pills justify-content-center">

                <li class="nav-item">

                    <a class="nav-link active" href="vista/login.php">Iniciar Sesión</a>

                </li>

                <li class="nav-item">

                    <a class="nav-link" href="vista/register.php">Registrarse</a>

                </li>

            </ul>

        </nav>

    </div>

</body>

</html>