Aarón Rodríguez Arias

Segundo trimestre

Fecha: 26/01/2025

Fecha límite: 31/01/2025

CampusFP – GETAFE

HITO 1 DEL 2º TRIMESTRE DE Programación

Logotipo

Descripción generada automáticamente

Índice

[Proyecto subido a GitHub 2](#_Toc189043608)

[Introducción al Proyecto StreamWeb 3](#_Toc189043609)

[Creación de la base de datos: 4](#_Toc189043610)

[Diseño lógico: 5](#_Toc189043611)

[Base de datos en MySQL 5](#_Toc189043612)

[Código en Visual Studio Code 6](#_Toc189043613)

[Estructura: 6](#_Toc189043614)

[Config/conexión.php 7](#_Toc189043615)

[Controlador/UsuarioController.php 8](#_Toc189043616)

[Modelo/usuario.php 9](#_Toc189043617)

[Vista/alta\_usuario.php 11](#_Toc189043618)

[Vista/lista\_usuarios.php 16](#_Toc189043619)

[Vista/editar\_usuario.php 19](#_Toc189043620)

[Vista/eliminar\_usuario.php 23](#_Toc189043621)

[Public/script.js 24](#_Toc189043622)

[Public/styles.css 26](#_Toc189043623)

[Index.php 26](#_Toc189043624)

[Bibliografía: 27](#_Toc189043625)

## Proyecto subido a GitHub

Aquí está el link:

## Introducción al Proyecto StreamWeb

En este trabajo, desarrollaremos una aplicación web para la empresa ficticia **StreamWeb**, dedicada a la gestión de suscripciones a servicios de streaming. La aplicación permitirá registrar usuarios, gestionar sus planes de suscripción, calcular el costo total según las opciones seleccionadas y realizar operaciones básicas de administración como actualización o eliminación de datos.

Para su implementación utilizaremos:

* **PHP**
* **MySQL**
* **JAVASCRIPT**
* **CSS**
* **BOOTSTRAP**

Organizado bajo la arquitectura **MVC**

1. **Diseño de la base de datos**

* **Tabla usuarios:** Almacena información esencial como nombre, correo (único), edad, plan base y duración de la suscripción.
* **Tabla paquetes\_usuario:** Relaciona cada usuario con sus paquetes adicionales contratados, gestionando estas relaciones mediante una clave foránea.
* **Validaciones en PHP y JavaScript:** Las restricciones más complejas, como limitar a los menores de 18 años al paquete Infantil o impedir más de un paquete adicional en el Plan Básico, se implementan directamente en el código.

1. **Creación del formulario de registro**

**Componentes del formulario:**

* **Campos de entrada:** Nombre, correo y edad del usuario.
* **Dropdown:** Selección del plan base (Básico, Estándar o Premium).
* **Checkboxes:** Selección de paquetes adicionales (Deporte, Cine, Infantil).
* **Radio buttons:** Elección de la duración de la suscripción (Mensual o Anual).

1. **Validaciones en PHP y JavaScript**

**Reglas de validación implementadas:**

* **Correo:** Se verifica que sea único en la base de datos y tenga un formato válido.
* **Edad:** Los usuarios menores de 18 años solo pueden seleccionar el paquete Infantil.
* **Plan Básico:** Se permite seleccionar un solo paquete adicional.
* **Paquete Deporte:** Solo puede contratarse si la duración de la suscripción es anual.

1. **Funcionalidades CRUD**

* **Crear:** Registrar nuevos usuarios y sus paquetes adicionales.
* **Leer:** Mostrar la lista de usuarios junto con los paquetes contratados.
* **Editar:** Permitir modificaciones solo en el plan base y los paquetes

adicionales (no se pueden modificar la edad ni el correo del usuario).

* **Eliminar:** Eliminar un usuario y sus paquetes asociados, los cuales se eliminan automáticamente gracias a ON DELETE CASCADE.

1. **Cálculo del Costo Total de la Suscripción**

* **Plan base:** Se asigna el costo correspondiente según la selección del usuario.
* **Paquetes adicionales:** Se suman los costos de los paquetes contratados.
* **Duración:** Si la suscripción es anual, el costo total se divide entre 12 para obtener el valor mensual.

## Creación de la base de datos:

Antes de realizar la base de datos, he hecho un diseño lógico de la base de datos para tener unas directrices claras.

### Diseño lógico:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Una vez marcadas las directrices procedo a crear la base de datos en MySQL

Base de datos en MySQL:

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## Código en Visual Studio Code

Voy a empezar por enseñar la estructura MVC que he implementado en este proyecto:

### Estructura:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Config/conexión.php

conexion.php establece la conexión con la base de datos MySQL para que el resto de la aplicación pueda interactuar con ella.

Se guarda en la carpeta config porque es un archivo de configuración que centraliza la conexión, facilitando su mantenimiento y reutilización en otros archivos.

<?php

//conexión con la base de datos

class Conexion {

    private $servidor = 'localhost';

    private $usuario = 'root';

    private $password = 'curso';

    private $base\_datos = 'streamweb';

    private $port = '3306';

    public $conexion;

    //establecer conexión con la base de datos

    public function \_\_construct() {

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

        //verifica si hay un error en la conexión

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

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

        }

    }

    //cerrar la conexión

    public function cerrar() {

        $this->conexion->close();

    }

}

?>

### Controlador/UsuarioController.php

UsuarioController.php es el controlador que gestiona la lógica de la aplicación.

* Recibe las solicitudes del usuario (como registrar, editar o eliminar un usuario).
* Llama a los métodos del modelo (Usuario.php) para interactuar con la base de datos.
* Devuelve los datos a las vistas para mostrarlos en la interfaz.

<?php

require\_once '../modelo/Usuario.php';

class UsuarioController {

    private $modelo;

    public function \_\_construct() {

        $this->modelo = new Usuario();

    }

    //registrar un usuario

    public function registrarUsuario($nombre, $correo, $edad, $plan\_base, $duracion, $paquetes) {

        $usuario\_id = $this->modelo->agregarUsuario($nombre, $correo, $edad, $plan\_base, $duracion);

        if ($usuario\_id) {

            foreach ($paquetes as $paquete) {

                $this->modelo->agregarPaquete($usuario\_id, $paquete);

            }

            echo "Usuario registrado con éxito.";

        }

    }

    //editar usuario

    public function actualizarUsuario($id, $plan\_base, $duracion) {

        return $this->modelo->actualizarUsuario($id, $plan\_base, $duracion);

    }

    //eliminar usuario

    public function eliminarUsuario($id) {

        return $this->modelo->eliminarUsuario($id);

    }

    //listar usuarios

    public function listarUsuarios() {

        return $this->modelo->obtenerUsuarios();

    }

    //obtener usuario por ID

    public function obtenerUsuarioPorId($id) {

        return $this->modelo->obtenerUsuarioPorId($id);

    }

}

### Modelo/usuario.php

modelo/Usuario.php es el modelo de la aplicación y se encarga de interactuar con la base de datos.

* Contiene los métodos para agregar, editar, eliminar y obtener usuarios.
* Gestiona los paquetes adicionales que el usuario puede contratar.
* Se comunica con la base de datos a través de consultas SQL.

<?php

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

// Clase Usuario para interactuar con la base de datos

class Usuario {

    private $conexion;

    public function \_\_construct() {

        $this->conexion = new Conexion();

    }

    //Agregar un usuario

    public function agregarUsuario($nombre, $correo, $edad, $plan\_base, $duracion) {

        if (!filter\_var($correo, FILTER\_VALIDATE\_EMAIL)) {

            return "Correo inválido.";

        }

        $query = "INSERT INTO usuarios (nombre, correo, edad, plan\_base, duracion\_suscripcion) VALUES (?, ?, ?, ?, ?)";

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

        $stmt->bind\_param("ssiss", $nombre, $correo, $edad, $plan\_base, $duracion);

        if ($stmt->execute()) {

            return $this->conexion->conexion->insert\_id;

        } else {

            return false;

        }

    }

    //Agregar paquetes

    public function agregarPaquete($usuario\_id, $paquete) {

        $query = "INSERT INTO paquetes\_usuario (usuario\_id, paquete) VALUES (?, ?)";

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

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

        return $stmt->execute();

    }

    //Editar usuario

    public function actualizarUsuario($id, $plan\_base, $duracion) {

        $query = "UPDATE usuarios SET plan\_base = ?, duracion\_suscripcion = ? WHERE id = ?";

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

        $stmt->bind\_param("ssi", $plan\_base, $duracion, $id);

        return $stmt->execute();

    }

    //Eliminar usuario y sus paquetes

    public function eliminarUsuario($id) {

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

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

        $stmt->bind\_param("i", $id);

        return $stmt->execute();

    }

    //Obtener lista de usuarios con sus paquetes

    public function obtenerUsuarios() {

        $query = "SELECT u.\*, GROUP\_CONCAT(p.paquete SEPARATOR ', ') AS paquetes

                  FROM usuarios u

                  LEFT JOIN paquetes\_usuario p ON u.id = p.usuario\_id

                  GROUP BY u.id";

        $resultado = $this->conexion->conexion->query($query);

        return $resultado->fetch\_all(MYSQLI\_ASSOC);

    }

    //Obtener usuario por ID

    public function obtenerUsuarioPorId($id) {

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

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

        $stmt->bind\_param("i", $id);

        $stmt->execute();

        return $stmt->get\_result()->fetch\_assoc();

    }

}

?>

### Vista/alta\_usuario.php

vista/alta\_usuario.php es la vista que muestra el formulario de registro para que los usuarios puedan darse de alta en la plataforma.

* Contiene un formulario con Bootstrap donde el usuario introduce su nombre, correo, edad, plan base, paquetes adicionales y duración de la suscripción.
* Incluye validaciones en JavaScript para evitar errores antes de enviar los datos.
* Envía la información al controlador (UsuarioController.php), que se encarga de procesarla y guardarla en la base de datos.

<?php

require\_once '../controlador/UsuarioController.php'; //importa el controlador

$controller = new UsuarioController();

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    //datos del formulario

    $nombre = $\_POST['nombre'];

    $correo = $\_POST['correo'];

    $edad = $\_POST['edad'];

    $plan\_base = $\_POST['plan\_base'];

    $duracion = $\_POST['duracion'];

    $paquetes = isset($\_POST['paquetes']) ? $\_POST['paquetes'] : [];

    //registra el usuario con sus paquetes y redirige

    $controller->registrarUsuario($nombre, $correo, $edad, $plan\_base, $duracion, $paquetes);

    header("Location: lista\_usuarios.php");

}

?>

<!DOCTYPE html>

<html lang="es">

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script> <!-- Sweet alert que utilizaremos más adelnate en el archivo.js !-->

<head>

    <meta charset="UTF-8">

    <title>Registrar Usuario</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="../public/styles.css">

    <script src="../public/script.js" defer></script> <!-- JavaScript para validaciones -->

</head>

<body>

    <div class="container">

        <h1 class="my-4">Registrar Nuevo Usuario</h1>

        <form id="registroUsuario" method="post">

            <!-- Campo para el nombre -->

            <div class="mb-3">

                <label class="form-label">Nombre</label>

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

            </div>

            <!-- Campo para el correo -->

            <div class="mb-3">

                <label class="form-label">Correo</label>

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

            </div>

            <!-- Campo para la edad -->

            <div class="mb-3">

                <label class="form-label">Edad</label>

                <input type="number" class="form-control" name="edad" id="edad" required>

            </div>

            <!-- Selección del plan base -->

            <div class="mb-3">

                <label class="form-label">Plan Base</label>

                <select class="form-control" name="plan\_base" id="plan\_base">

                    <option value="Básico">Básico</option>

                    <option value="Estándar">Estándar</option>

                    <option value="Premium">Premium</option>

                </select>

            </div>

            <!-- Selección de paquetes adicionales -->

            <div class="mb-3">

                <label class="form-label">Paquetes</label><br>

                <input type="checkbox" name="paquetes[]" value="Deporte" id="deporte"> Deporte

                <input type="checkbox" name="paquetes[]" value="Cine"> Cine

                <input type="checkbox" name="paquetes[]" value="Infantil"> Infantil

            </div>

            <!-- Selección de la duración de la suscripción -->

            <div class="mb-3">

                <label class="form-label">Duración</label>

                <select class="form-control" name="duracion" id="duracion">

                    <option value="Mensual">Mensual</option>

                    <option value="Anual">Anual</option>

                </select>

            </div>

            <!-- Botón de registro -->

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

        </form>

    </div>

</body>

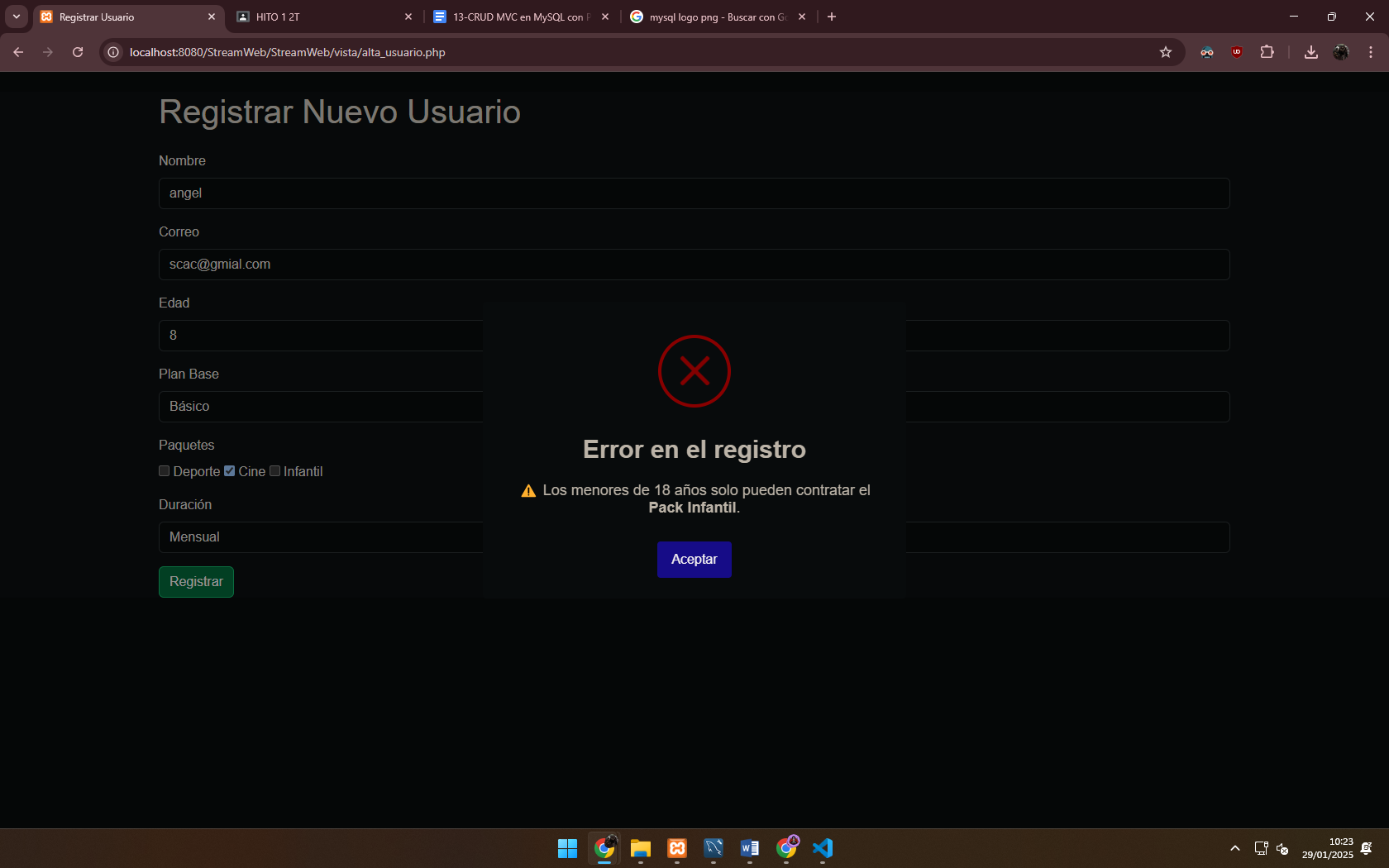
</html>

Este archivo visualmente se vería así:

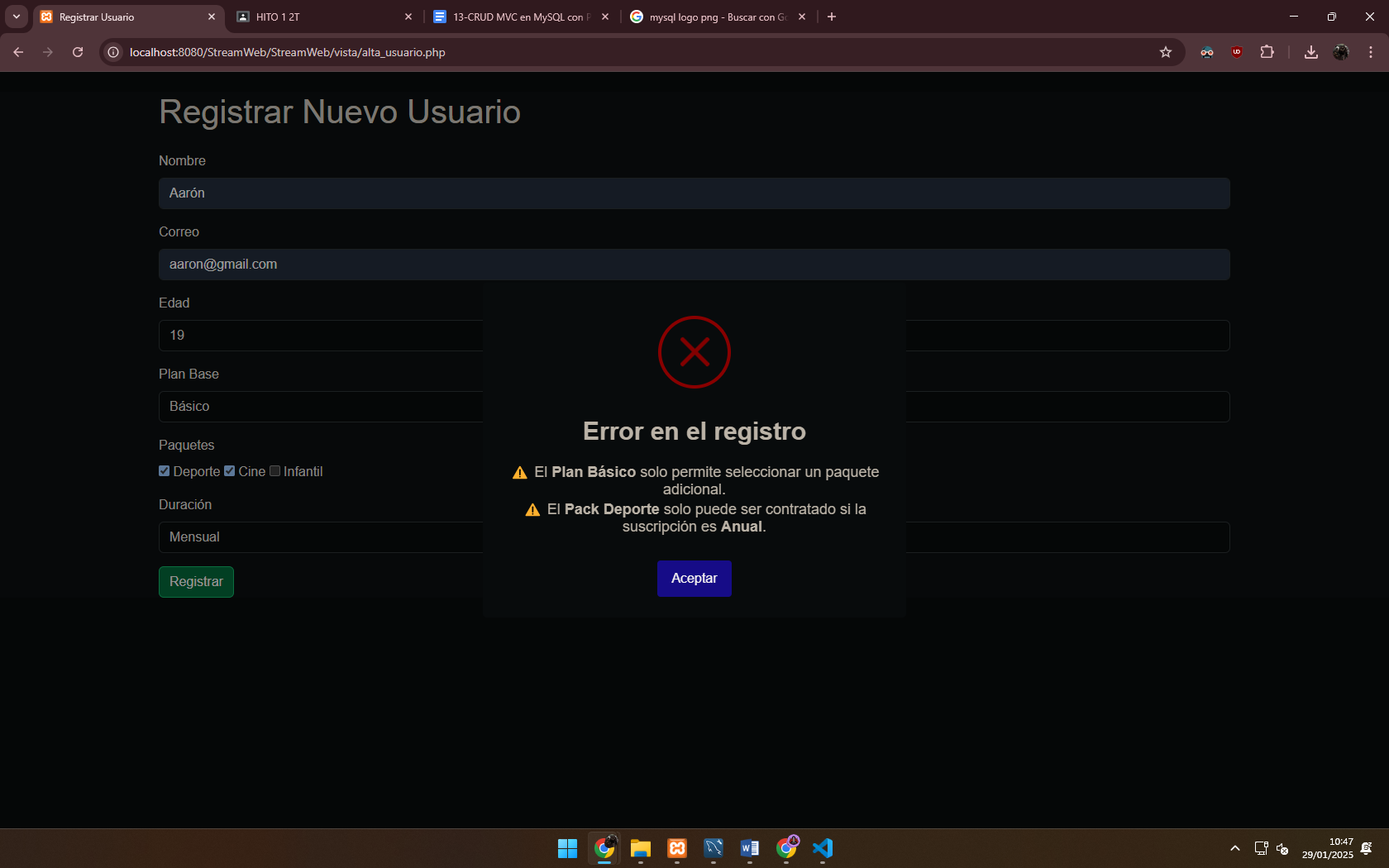
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

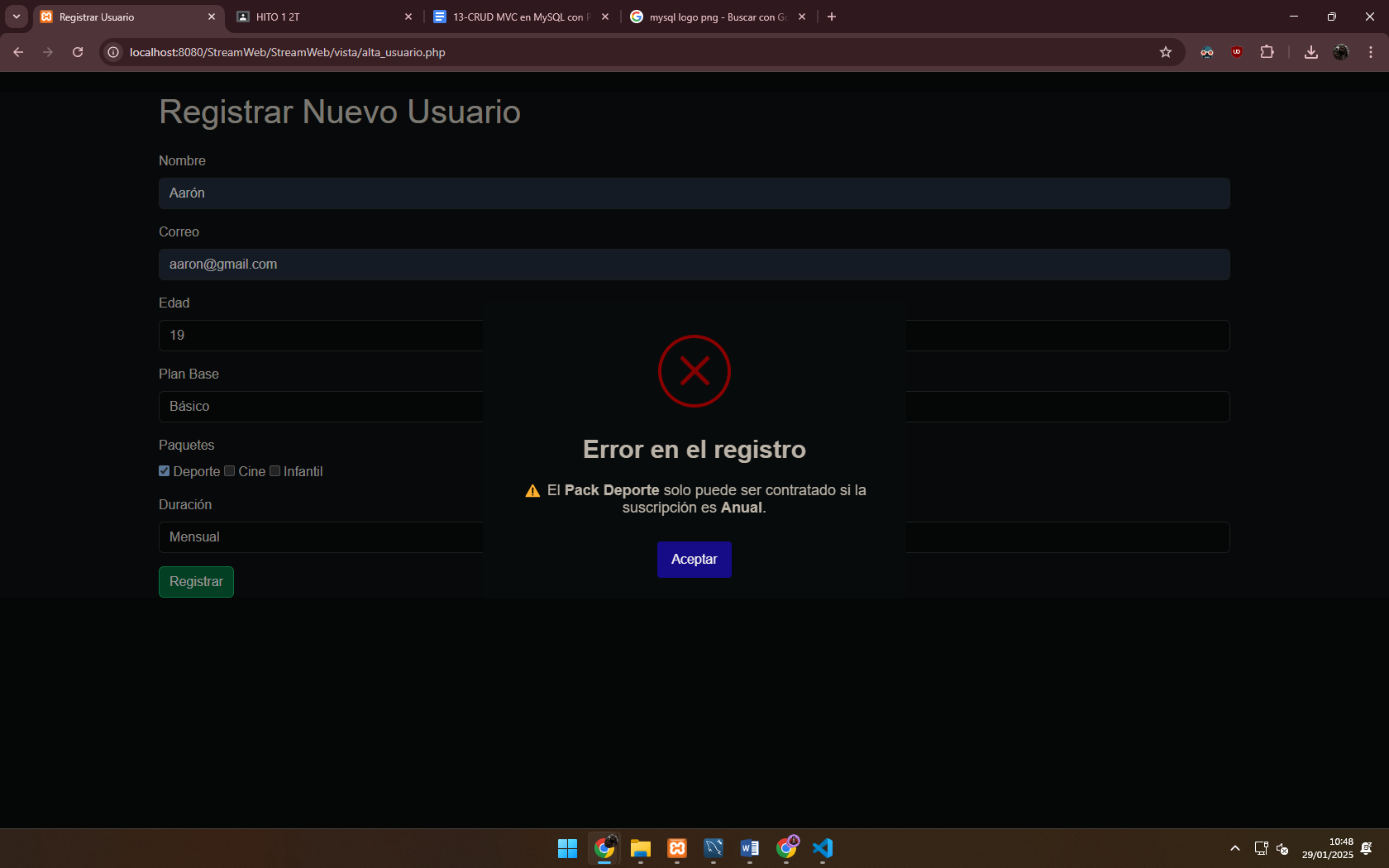
Si un usuario menor de edad quiere registrarse en otro paquete que no sea el infantil saltará un error. Esto es gracias al archivo JavaScript que enseñaré posteriormente



Los usuarios que obtengan el plan básico solo podrán obtener un paquete.



El Pack Deporte solo puede ser contratado si la duración de la suscripción es de 1 año.



Si elegimos los 3 paquetes con el plan Premium (por ejemplo) y con una duración anual vemos que también saldrá en el listado sin problema.



### Vista/lista\_usuarios.php

vista/lista\_usuarios.php es la vista que muestra la lista de todos los usuariosregistrados en la plataforma.

* Recupera los datos desde el controlador (UsuarioController.php).
* Muestra la información en una tabla con Bootstrap, incluyendo nombre, correo, edad, plan base, duración y paquetes y coste mensual
* Incluye botones para editar y eliminar usuarios.
* Facilita la navegación y la gestión de usuarios dentro del sistema.

<?php

require\_once '../controlador/UsuarioController.php'; //importa el controlador

$controller = new UsuarioController();

$usuarios = $controller->listarUsuarios(); //obtiene la lista de usuarios

//definir precios de los planes base

$precios\_planes = [

    "Básico" => 9.99,

    "Estándar" => 13.99,

    "Premium" => 17.99

];

//definir precios de los paquetes adicionales

$precios\_paquetes = [

    "Deporte" => 6.99,

    "Cine" => 7.99,

    "Infantil" => 4.99

];

//función para calcular el costo mensual total del usuario

function calcularCosto($usuario, $precios\_planes, $precios\_paquetes) {

    $costo = $precios\_planes[$usuario['plan\_base']]; // Precio del plan base

    //sumar el costo de los paquetes adicionales si existen

    if (!empty($usuario['paquetes'])) {

        $paquetes = explode(", ", $usuario['paquetes']);

        foreach ($paquetes as $paquete) {

            if (isset($precios\_paquetes[$paquete])) {

                $costo += $precios\_paquetes[$paquete];

            }

        }

    }

    return number\_format($costo, 2) . " €";

}

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Usuarios Registrados</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="../public/styles.css"> <!-- Archivo de estilos -->

</head>

<body>

    <div class="container">

        <h1 class="my-4">Usuarios Registrados</h1>

        <table class="table table-striped"> <!-- Tabla con estilos Bootstrap -->

            <thead>

                <tr>

                    <th>Nombre</th>

                    <th>Correo</th>

                    <th>Edad</th>

                    <th>Plan Base</th>

                    <th>Duración</th>

                    <th>Paquetes</th>

                    <th>Costo Mensual</th>

                    <th>Acciones</th>

                </tr>

            </thead>

            <tbody>

                <?php foreach ($usuarios as $usuario): ?>

                <tr>

                    <td><?= $usuario['nombre'] ?></td>

                    <td><?= $usuario['correo'] ?></td>

                    <td><?= $usuario['edad'] ?></td>

                    <td><?= $usuario['plan\_base'] ?></td>

                    <td><?= $usuario['duracion\_suscripcion'] ?></td>

                    <td><?= $usuario['paquetes'] ?: 'Ninguno' ?></td> <!-- Si no tiene paquetes, muestra Ninguno -->

                    <td><?= calcularCosto($usuario, $precios\_planes, $precios\_paquetes) ?></td> <!-- muestra el costo mensual -->

                    <td>

                        <a href="editar\_usuario.php?id=<?= $usuario['id'] ?>" class="btn btn-warning btn-sm">Editar</a>

                        <a href="eliminar\_usuario.php?id=<?= $usuario['id'] ?>" class="btn btn-danger btn-sm">Eliminar</a>

                    </td>

                </tr>

                <?php endforeach; ?>

            </tbody>

        </table>

        <a href="alta\_usuario.php" class="btn btn-primary">Registrar Usuario</a> <!-- Botón para registrar nuevos usuarios -->

    </div>

</body>

</html>

Visualmente se vería así:

Pantalla de un video juego

Descripción generada automáticamente con confianza media

### Vista/editar\_usuario.php

<?php

require\_once '../controlador/UsuarioController.php'; // Importa el controlador

$controller = new UsuarioController();

$usuario = $controller->obtenerUsuarioPorId($\_GET['id']); // Obtiene el usuario a editar

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    // Actualiza el usuario con los datos del formulario

    $controller->actualizarUsuario($\_POST['id'], $\_POST['plan\_base'], $\_POST['duracion']);

    header("Location: lista\_usuarios.php"); // Redirige a la lista de usuarios

}

?>

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Editar Usuario</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="../public/styles.css"> <!-- Archivo de estilos -->

</head>

<body>

    <div class="container mt-5">

        <div class="card shadow-lg p-4"> <!-- Tarjeta con sombra para mejor presentación -->

            <h2 class="text-center mb-4">Editar Usuario</h2>

            <form method="post">

                <input type="hidden" name="id" value="<?= $usuario['id'] ?>"> <!-- ID oculto -->

                <!-- Selección del Plan Base -->

                <div class="mb-3">

                    <label class="form-label">Plan Base</label>

                    <select class="form-select" name="plan\_base">

                        <option value="Básico" <?= ($usuario['plan\_base'] == 'Básico') ? 'selected' : '' ?>>Básico</option>

                        <option value="Estándar" <?= ($usuario['plan\_base'] == 'Estándar') ? 'selected' : '' ?>>Estándar</option>

                        <option value="Premium" <?= ($usuario['plan\_base'] == 'Premium') ? 'selected' : '' ?>>Premium</option>

                    </select>

                </div>

                <!-- Selección de la Duración -->

                <div class="mb-3">

                    <label class="form-label">Duración</label>

                    <select class="form-select" name="duracion">

                        <option value="Mensual" <?= ($usuario['duracion\_suscripcion'] == 'Mensual') ? 'selected' : '' ?>>Mensual</option>

                        <option value="Anual" <?= ($usuario['duracion\_suscripcion'] == 'Anual') ? 'selected' : '' ?>>Anual</option>

                    </select>

                </div>

                <!-- Botones de acción -->

                <div class="text-center">

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

                    <a href="lista\_usuarios.php" class="btn btn-secondary">Cancelar</a>

                </div>

            </form>

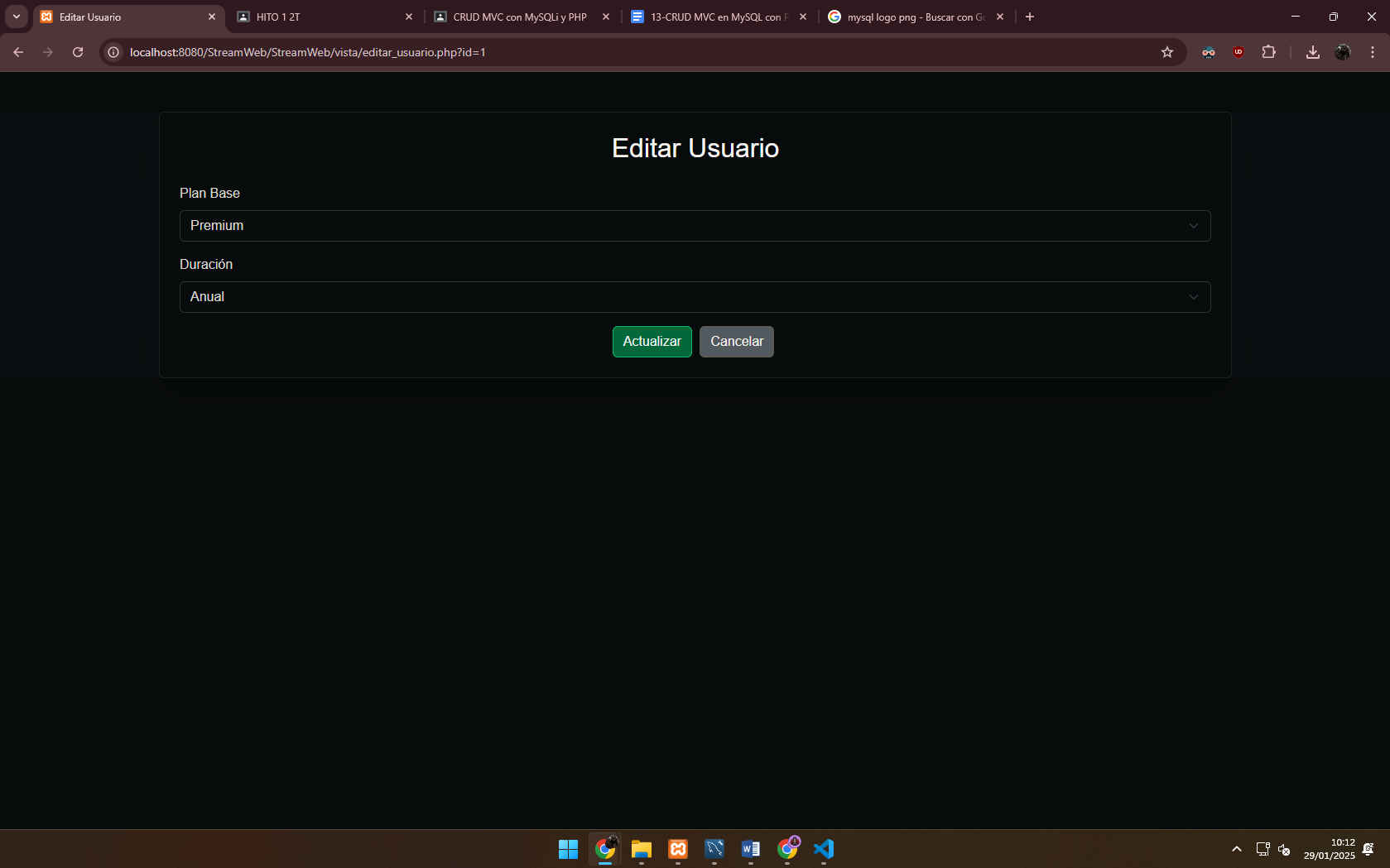
        </div>

    </div>

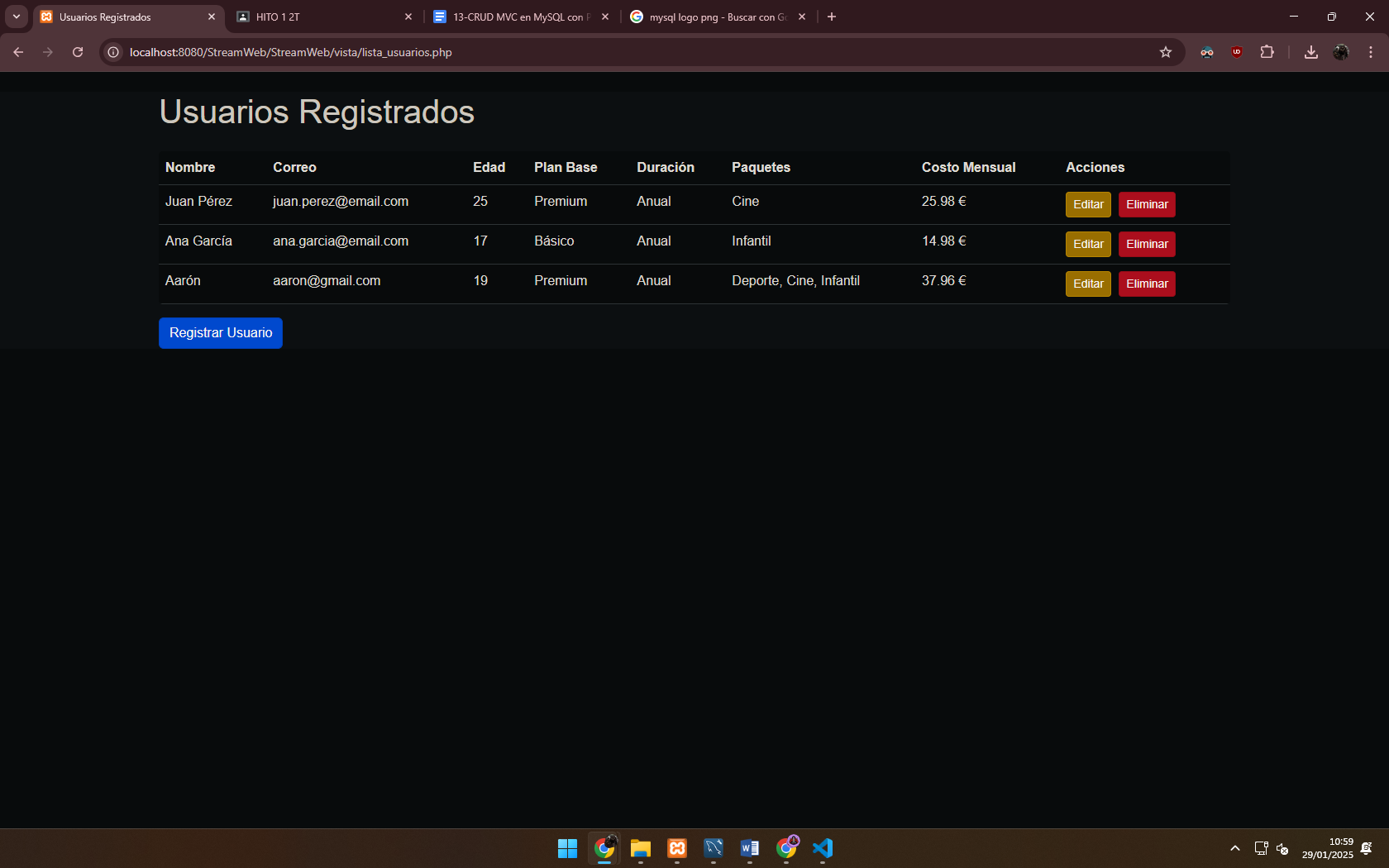
</body>

</html>

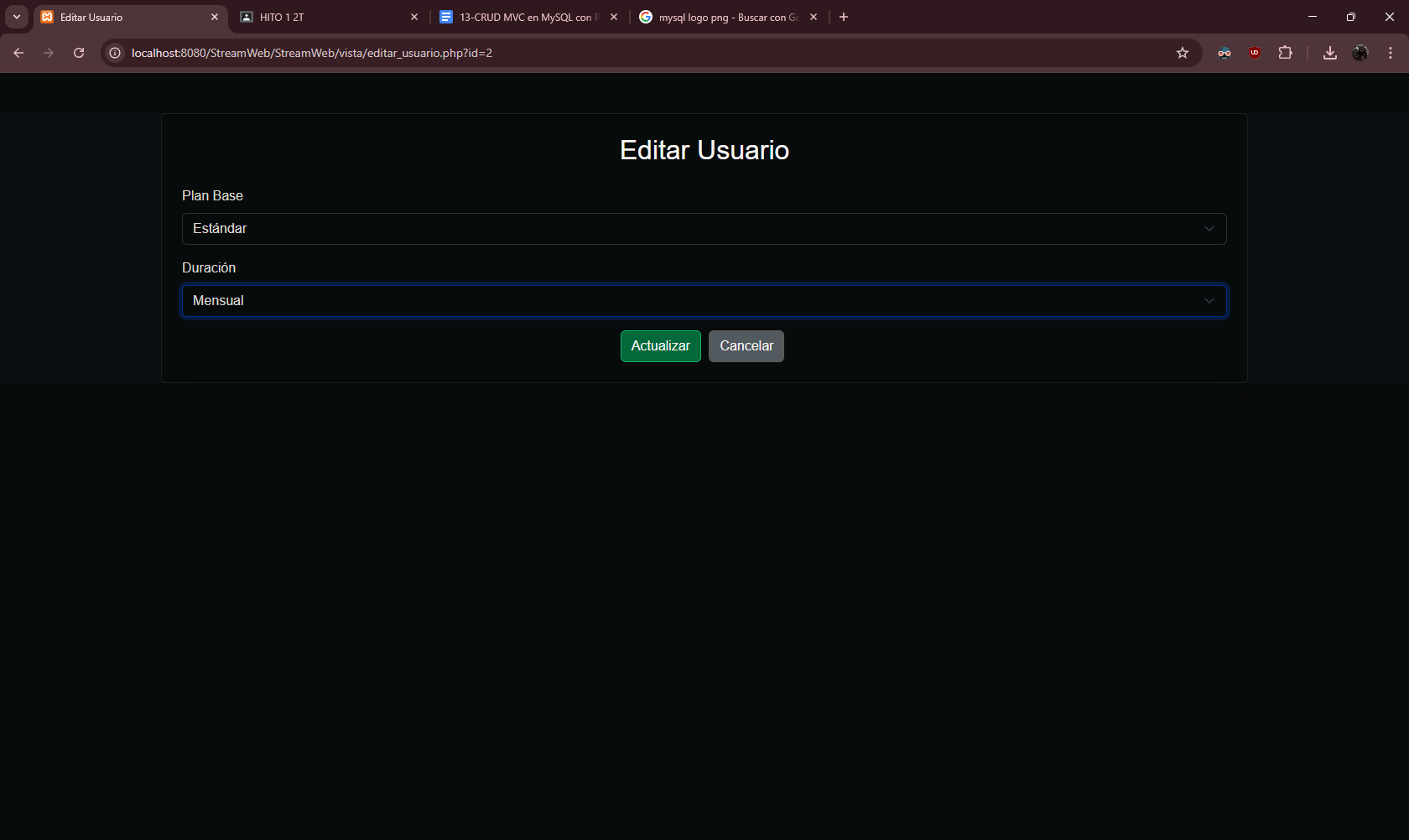
Este archivo visualmente se vería así:



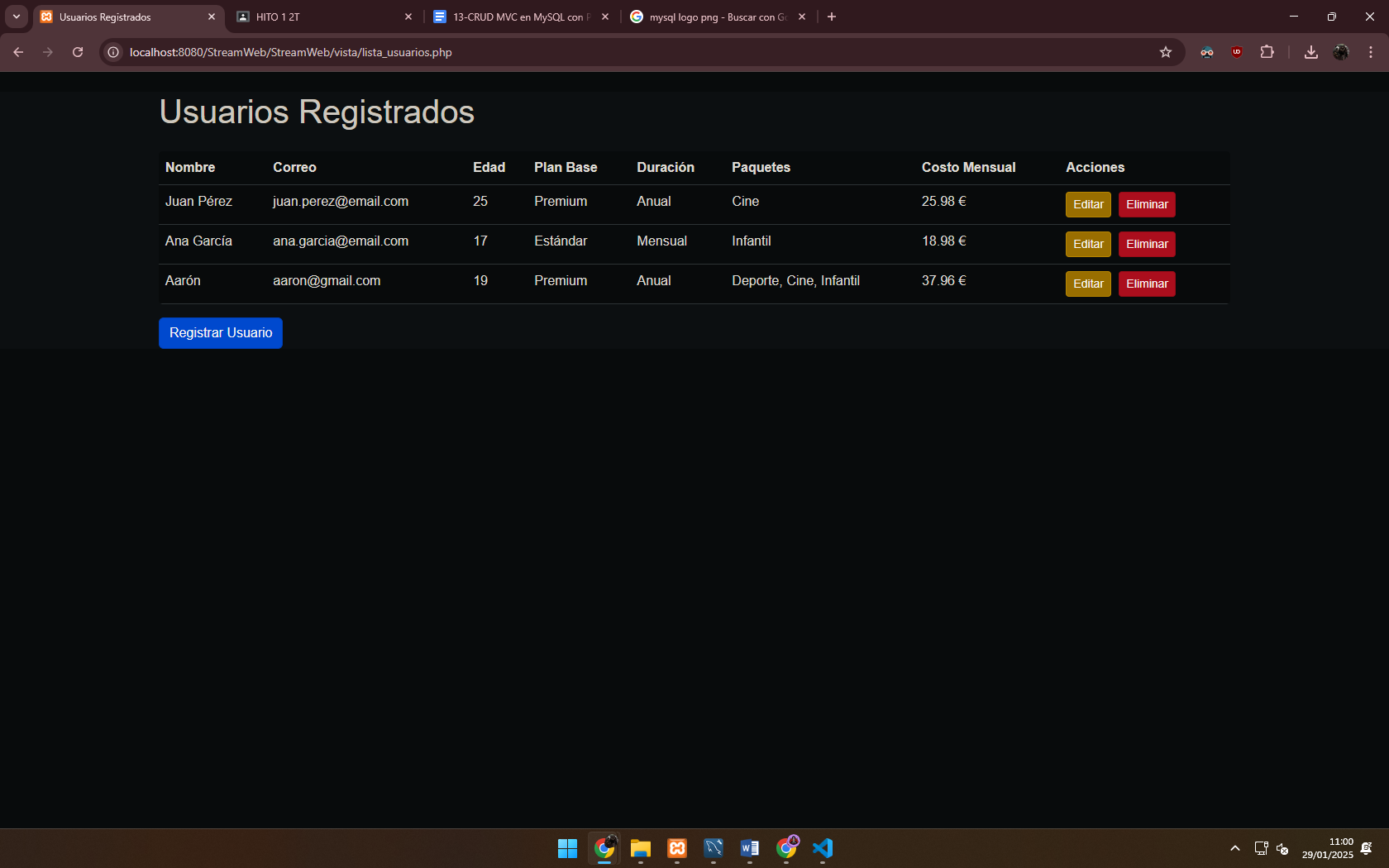
Vamos a cambiar a Ana García para que su duración del plan sea mensual y su plan base cambie a estándar



Lo modificamos



Y se cambiaría:



### Vista/eliminar\_usuario.php

<?php

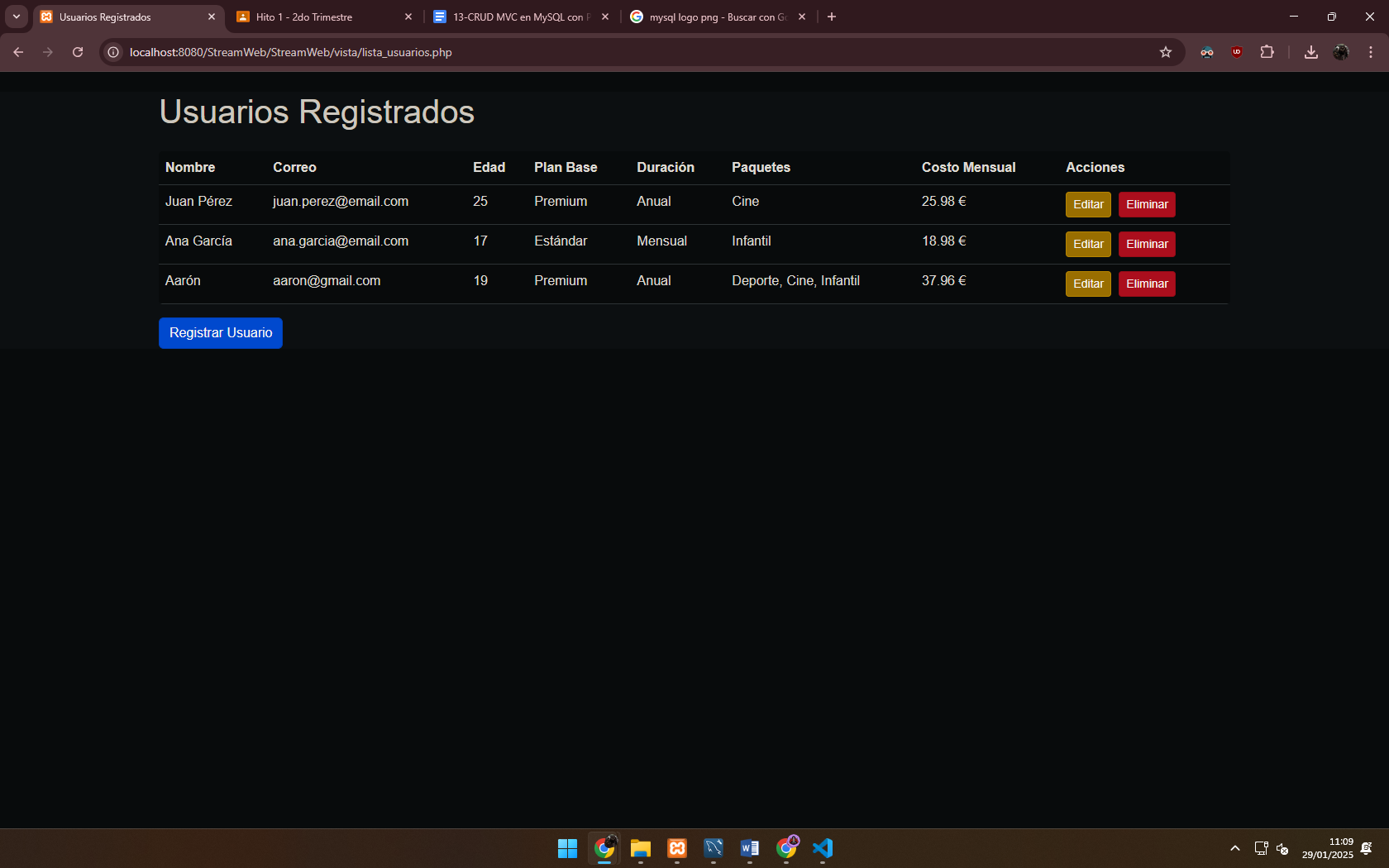
require\_once '../controlador/UsuarioController.php';

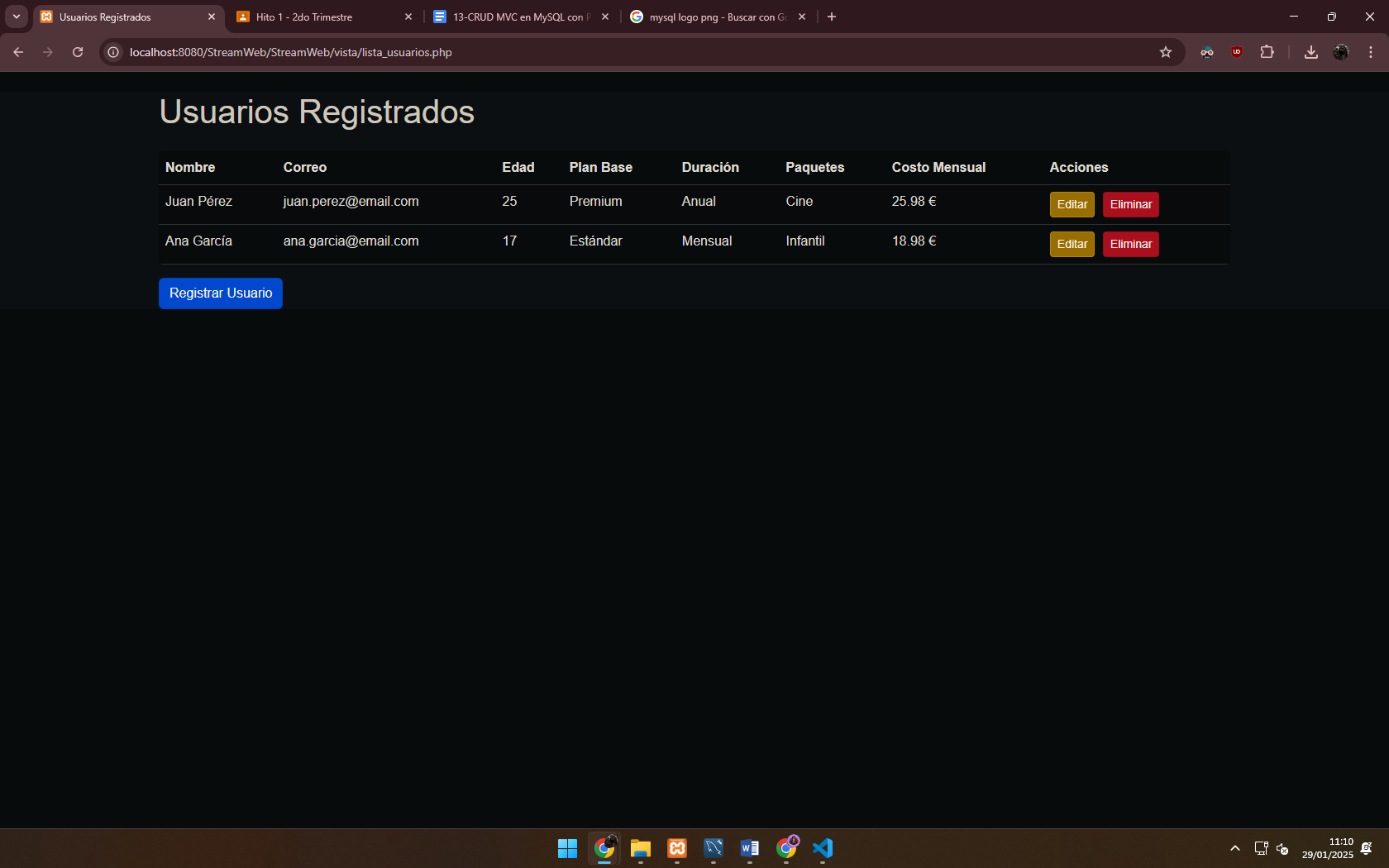
$controller = new UsuarioController();

$controller->eliminarUsuario($\_GET['id']);

header("Location: lista\_usuarios.php");

?>





### Public/script.js

public/script.js valida el formulario de registro antes de enviarlo.

* Captura el formulario y escucha el evento submit.
* Verifica restricciones: edad, plan base y paquetes seleccionados.
* Evita el envío si hay errores y muestra alertas con SweetAlert2.
* Mejora la experiencia del usuario al prevenir errores antes de enviar datos.

document.addEventListener("DOMContentLoaded", function() {

    const form = document.getElementById("registroUsuario");

    form.addEventListener("submit", function(event) {

        const edad = parseInt(document.getElementById("edad").value);

        const paquetes = document.querySelectorAll('input[name="paquetes[]"]:checked');

        const planBase = document.getElementById("plan\_base").value;

        const duracion = document.getElementById("duracion").value;

        let errores = [];

        // Restricción de menores de 18 años

        if (edad < 18) {

            for (let paquete of paquetes) {

                if (paquete.value !== "Infantil") {

                    errores.push("⚠️ Los menores de 18 años solo pueden contratar el <b>Pack Infantil</b>.");

                }

            }

        }

        // Restricción del Plan Básico

        if (planBase === "Básico" && paquetes.length > 1) {

            errores.push("⚠️ El <b>Plan Básico</b> solo permite seleccionar un paquete adicional.");

        }

        // Restricción del Pack Deporte

        for (let paquete of paquetes) {

            if (paquete.value === "Deporte" && duracion !== "Anual") {

                errores.push("⚠️ El <b>Pack Deporte</b> solo puede ser contratado si la suscripción es <b>Anual</b>.");

            }

        }

        if (errores.length > 0) {

            event.preventDefault(); // Evita que el formulario se envíe

            // Mostrar alertas con SweetAlert2

            Swal.fire({

                title: "Error en el registro",

                html: errores.join("<br>"), // Muestra los errores en formato HTML

                icon: "error",

                confirmButtonText: "Aceptar"

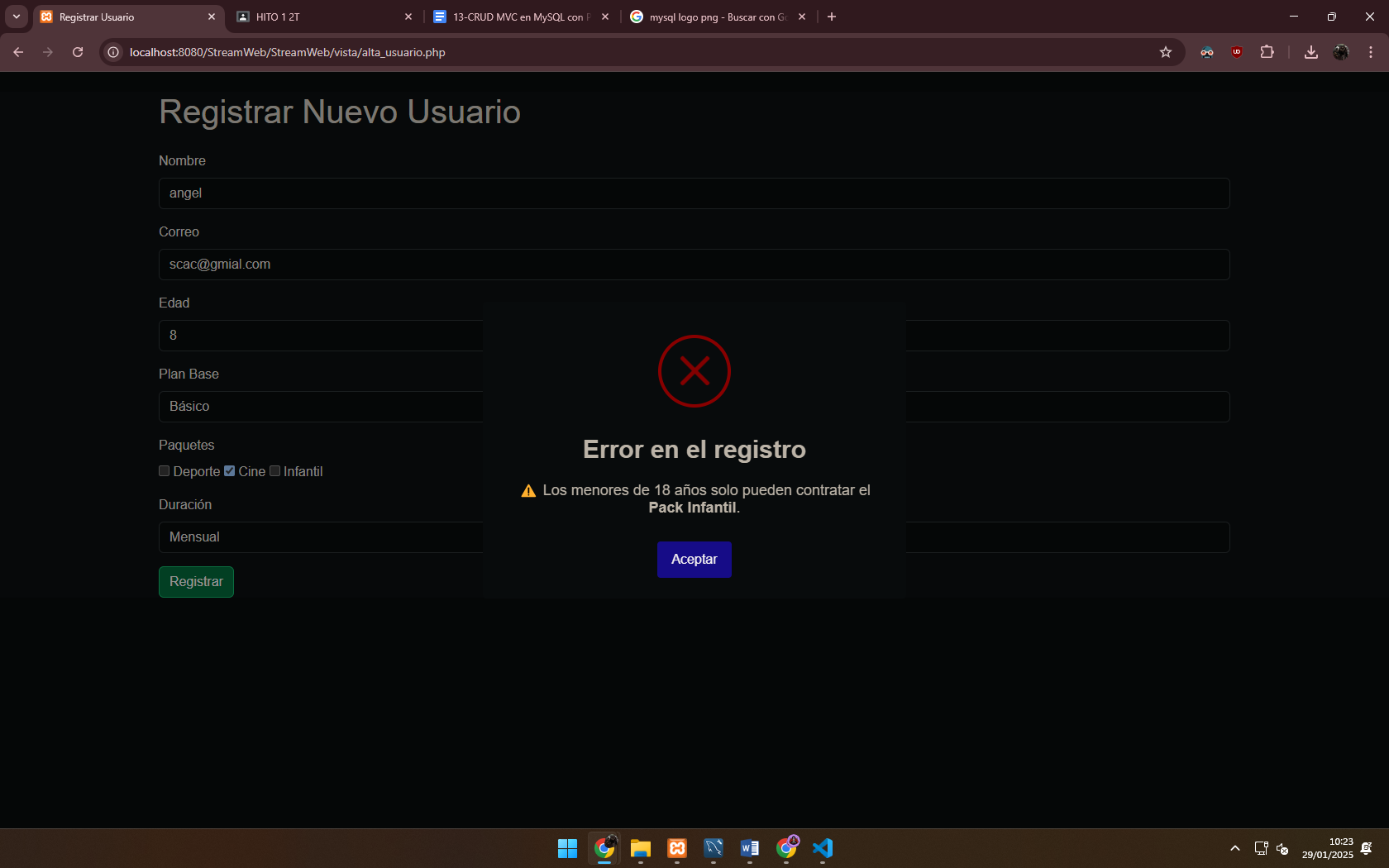
            });

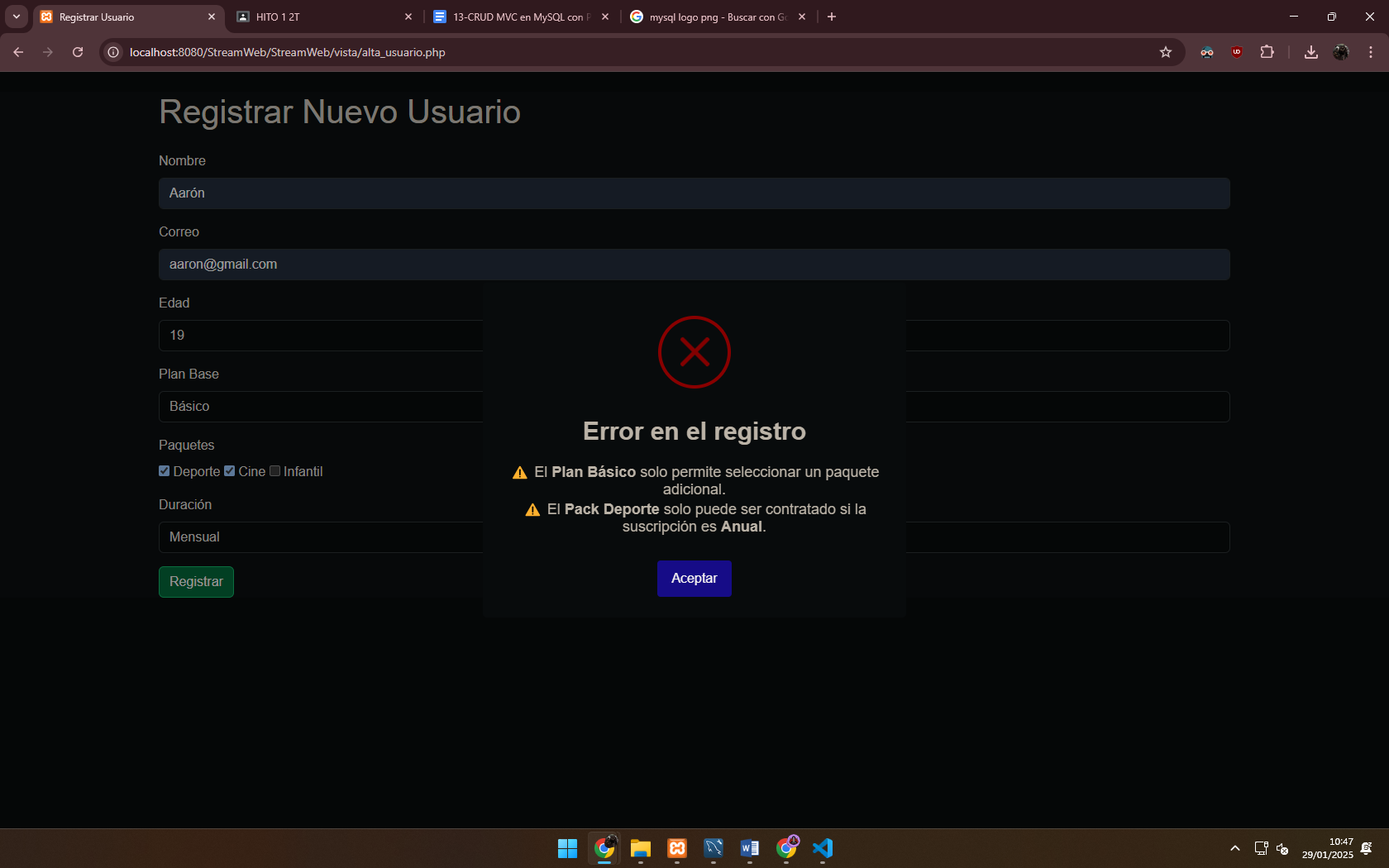
        }

    });

});

Aquí recuerdo los ejemplos en los que aparece el uso de JavaScript





### Public/styles.css

styles.css define el diseño y la apariencia de la aplicación.

* body → Fondo gris claro y fuente Arial para mejorar la legibilidad.
* .container → Añade margen superior para separar el contenido de la parte superior.
* .btn → Espacio entre botones para evitar que se junten.
* .table → Fondo blanco, bordes redondeados y ocultar desbordamiento para una mejor apariencia.
* h1 → Color oscuro para los títulos, haciéndolos más visibles.

Mejora la presentación y hace la interfaz más atractiva.

### Index.php

index.php redirige automáticamente a la página principal de la aplicación.

* header("Location: vista/lista\_usuarios.php"); → Envía al usuario a la lista de usuarios.
* exit(); → Detiene la ejecución para evitar que se procese código adicional.

Facilita el acceso a la aplicación sin escribir rutas largas.

<?php

header("Location: vista/lista\_usuarios.php");

exit();

?>

## Bibliografía:

Referencias:

*ChatGPT*. (n.d.). Chatgpt.com. Retrieved January 29, 2025, from <https://chatgpt.com/>

Gustavo, B. (2018, May 25). *Cómo conectar PHP con Bases de datos MySQL*. Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/conectar-php-mysql>

(N.d.). Fernandoruizrico.com. Retrieved January 29, 2025, from <https://fernandoruizrico.com/programacion-con-javascript-cuadros-de-dialogo-con-sweetalert/>

