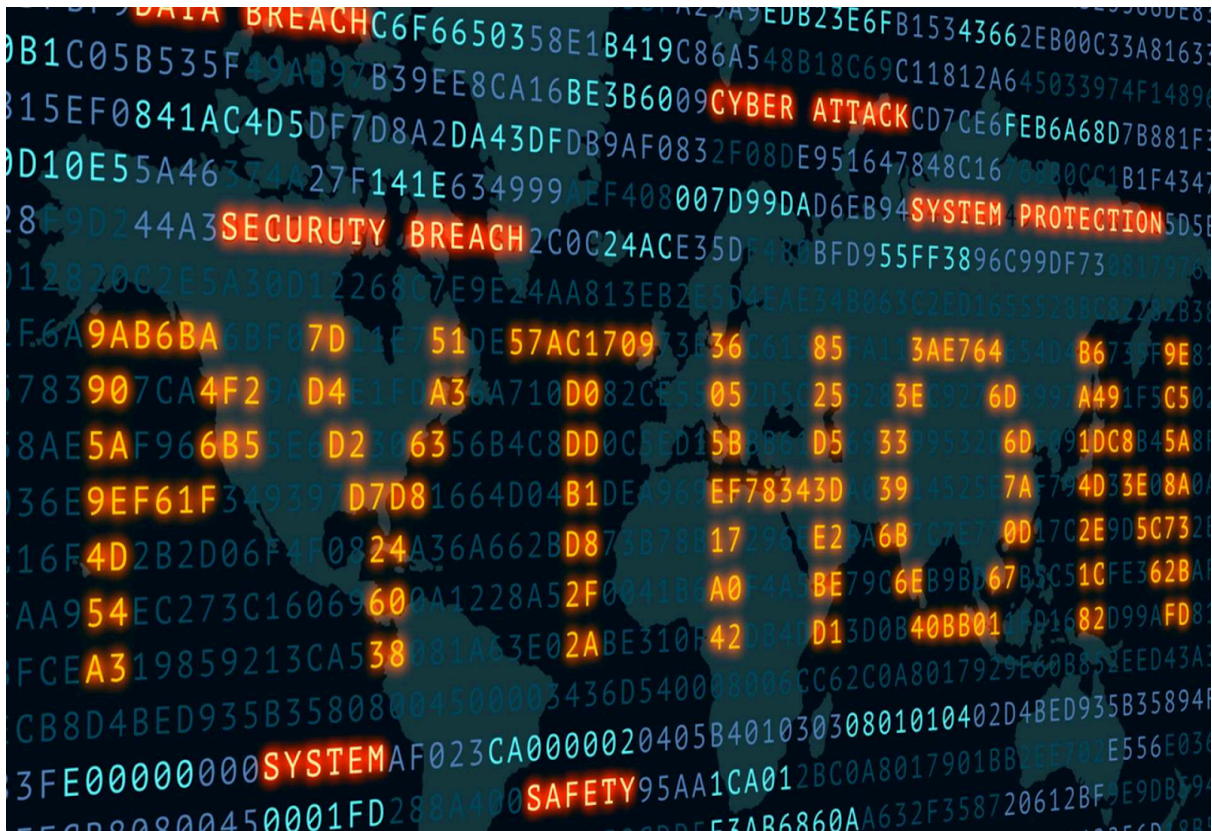


HITO 1 DEL 2º TRIMESTRE DE Programación

ÁLVARO ORGAZ 1DAM

FECHA : 04/02/2025



Aquí dejo en el enlace a mi repositorio de Git Hub:

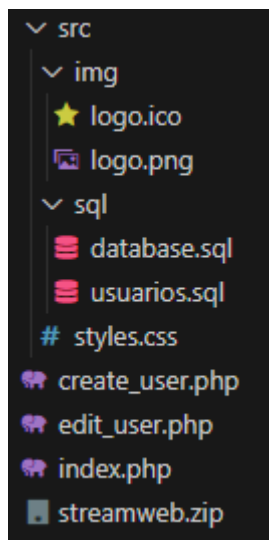
https://github.com/alvaroorgaz/Programacion_H1_2T_AlvaroO

Índice

Índice.....	2
ESTRUCTURA.....	3
Index.php.....	4
create_user.php.....	8
edit_user.php.....	11
styles.css.....	14
database.sql.....	17
Alta de nuevos usuarios.....	20
Eliminación de usuarios.....	22
Se muestra página con todos los usuarios registrados.....	22
Modificar usuario.....	23
Mostrar el coste mensual desglosado de cada usuario.....	25
BIBLIOGRAFÍA.....	26

A continuación, tenemos mi estructura y los códigos que he utilizado para mi página web.

ESTRUCTURA



La estructura de la web cuenta con los archivos PHP que generan la web y se conectan con la base de datos en la raíz del proyecto.

La carpeta src contiene los recursos usados por la web:

- img: almacena las imágenes de la web, en este caso, su logo
- sql: contiene el archivo para crear la base de datos y un archivo con una consulta para mostrar los usuarios
- styles.css: es la hoja de estilos que da formato a la web con CSS

Como podemos ver aquí, tenemos la estructura de nuestra página web, con diferentes archivos: create_user.php, edit_user.php, index.php, plan_info.php y por último tenemos user_database.php.

A continuación explicaremos paso a paso cada código.

Index.php

Aquí tenemos el index.php, que lanza el contenido principal de la web, conectando con la base de datos, mostrando una tabla con los usuarios, permitiendo crear un nuevo usuario con un enlace a la página para hacer esta acción, y dos tablas con los precios de los planes y packs.

Análisis del código:

Explicación de cada bloque de código del archivo

Bloque PHP que instancia las variables para conectar con la BBDD, realiza la conexión, y cuenta con el método POST para eliminar un usuario:

```

<?php
$host = "localhost";
$port = 3306;
$socket = "";
$user = "root";
$password = "root";
$dbname = "StreamWeb";

$con = new mysqli($host, $user, $password, $dbname, $port, $socket)
    or die('Could not connect to the database server' .
mysqli_connect_error());

if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST['delete_user_id'])) {
    $user_id = $_POST['delete_user_id'];
    $sql_delete = "DELETE FROM Usuarios WHERE id = $user_id";
    if ($con->query($sql_delete) === TRUE) {
        echo "<script>alert('Usuario eliminado
exitosamente');</script>";
    } else {
        echo "<script>alert('Error al eliminar el usuario: " .
addslashes($con->error) . "');</script>";
    }
}
?>

```

Cabecera HTML: define la estructura básica de la web, añadiendo la ruta del archivo CSS para los estilos de la web y contiene un Script de JavaScript para redirigir al usuario a la página de creación de usuario al pulsar el botón:

```

<!DOCTYPE html>
<html lang="es" styles="font-family: sans-serif;">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="icon" type="image/x-icon" href="/src/img/logo.ico">
    <link rel="stylesheet" href="src/styles.css">
    <title>StreamWeb</title>
    <script>
        function redirectToCreateUser() {

```

```

        window.location.href = 'create_user.php';
    }

```

Tabla de usuarios: se define una tabla en HTML con una cabecera fija para mostrar todos los campos de los usuarios, se extraen mediante PHP los datos de los usuarios almacenados en la BBDD:

El bloque PHP hace una consulta Select sobre las tablas usuarios, paquetes y planes, uniendo estas tres mediante LEFT JOIN.

Se conecta a la variable \$con previamente creada en la cabecera del archivo para lanzar la consulta y alimentar los datos de la tabla, que se van añadiendo como línea tr mediante un bucle While a ésta, en caso de no haber datos, se muestra un aviso en la tabla:

```

<table>
    <tr>
        <th>ID</th>
        <th>Nombre</th>
        <th>Apellidos</th>
        <th>Correo</th>
        <th>Edad</th>
        <th>Plan Base</th>
        <th>Paquete Adicional</th>
        <th>Duración Suscripción</th>
        <th>Precio Paquete</th>
        <th>Precio Plan</th>
        <th>Coste Mensual</th>
        <th>Acciones</th>
    </tr>
    <?php
        $sql = "SELECT usuarios.id, usuarios.nombre,
usuarios.apellidos, usuarios.correo, usuarios.edad,
                        usuarios.plan_base, paquetes.nombre AS
paquete_adicional, usuarios.duracion_suscripcion,
                        paquetes.precio AS precio_paquete, planes.precio
AS precio_plan,
                        (paquetes.precio + planes.precio) AS
coste_mensual
                FROM usuarios
                LEFT JOIN suscripciones ON usuarios.id =
suscripciones.usuario_id
                LEFT JOIN paquetes ON suscripciones.paquete_id =
paquetes.id
                LEFT JOIN planes ON usuarios.plan_base =
planes.nombre";

```

```

$result = $con->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<tr>
            <td>" . $row["id"] . "</td>
            <td>" . $row["nombre"] . "</td>
            <td>" . $row["apellidos"] . "</td>
            <td>" . $row["correo"] . "</td>
            <td>" . $row["edad"] . "</td>
            <td>" . $row["plan_base"] . "</td>
            <td>" . $row["paquete_adicional"] . "</td>
            <td>" . $row["duracion_suscripcion"] . "</td>
            <td>" . $row["precio_paquete"] . "</td>
            <td>" . $row["precio_plan"] . "</td>
            <td>" . $row["coste_mensual"] . "</td>
            <td>
                <form method='get' action='edit_user.php'>
                    <input type='hidden'
name='edit_user_id' value='" . $row["id"] . "'>
                    <input type='submit' value='Editar'
id='editar'>
                </form>
                <form method='post' action='index.php'
onsubmit='return confirm(\"¿Estás seguro de que deseas eliminar este
usuario?\");'>
                    <input type='hidden'
name='delete_user_id' value='" . $row["id"] . "'>
                    <input type='submit' value='Eliminar'
id='eliminar'>
                </form>
            </td>
        </tr>";
    }
} else {
    echo "<tr><td colspan='12'>No hay datos
disponibles</td></tr>";
}

?>

```

Línea adicional en la tabla con el botón que redirige a la página para crear un usuario mediante una función de JavaScript previamente definida en la cabecera del HTML:

```
<tr><td colspan='12'><input id="crear" type="submit"
value="Crear Usuario" onclick="redirectToCreateUser()"></td></tr>
```

Tablas adicionales con la información de Planes y Packs:

Como apartado auxiliar, se crean dos tablas que siguen el mismo modelo que la de los usuarios, haciendo una consulta mediante PHP por SQL a la tabla paquetes, para mostrar los datos en la página:

```
<h2>Información de Planes</h2>
<br>
<table>
  <tr>
    <th>Pack</th>
    <th>Precio Mensual €</th>
  </tr>
  <?php
    $sql = "SELECT nombre, precio FROM Paquetes";
    $result = $con->query($sql);

    if ($result->num_rows > 0) {
      while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["nombre"] . "</td><td>" .
$row["precio"] . "€</td></tr>";
      }
    } else {
      echo "<tr><td colspan='2'>No hay datos
disponibles</td></tr>";
    }
  ?>
```

Se realiza otra consulta mediante PHP por SQL a la tabla planes, para mostrar los datos en la página:

```
</table>
<br>
<table>
  <tr>
    <th>Plan</th>
    <th>Precio Mensual €</th>
  </tr>
  <?php
    $sql = "SELECT nombre, precio FROM Planes";
    $result = $con->query($sql);

    if ($result->num_rows > 0) {
```

```

        while($row = $result->fetch_assoc()) {
            echo "<tr><td>" . $row["nombre"] . "</td><td>" .
$row["precio"] . "€</td></tr>";
        }
    } else {
        echo "<tr><td colspan='2'>No hay datos
disponibles</td></tr>";
    }
    ?>
</table>
<br>
</body>
</html>

```

Ambas tablas son cargadas con los datos de cada consulta mediante un bucle While que recoge los datos, en caso de no haber resultados, se muestra una línea avisando de esto.

create_user.php

Este archivo contiene la página con el formulario para crear usuarios, siguiendo una estructura similar a la de index.html:

De nuevo, conectamos a la BBDD mediante PHP, con las variables y la conexión.

```

<?php
$host = "localhost";
$port = 3306;
$socket = "";
$user = "root";
$password = "1234";
$dbname = "StreamWeb";

$con = new mysqli($host, $user, $password, $dbname, $port, $socket)
    or die('Could not connect to the database server' .
mysqli_connect_error());

$error_message = "";

```

Se define el método POST que se ejecuta al pulsar el botón de crear el usuario, recogiendo los datos del formulario, añadiendo a las variables de PHP y comprobando mediante una secuencia de condicionales if-else para controlar las condiciones de los datos propuestas en los requerimientos. Si es correcto, se insertan los datos:


```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $email = $_POST['correo'];
    $edad = $_POST['edad'];
    $plan_base = $_POST['plan_base'];
    $paquetes_adicionales = $_POST['paquetes_adicionales'];
    $duracion_suscripcion = $_POST['duracion_suscripcion'];

    // Validaciones
    if ($edad < 18 && $paquetes_adicionales != 3) { // Assuming 3 is
the ID for 'Infantil'
        $error_message = "Los usuarios menores de 18 años solo pueden
contratar el Pack Infantil.";
    } elseif ($plan_base == 'Básico') {
        $sql_check_paquetes = "SELECT COUNT(*) AS paquete_count FROM
Suscripciones WHERE usuario_id = (SELECT MAX(id) FROM Usuarios)";
        $result_check_paquetes = $con->query($sql_check_paquetes);
        $row_check_paquetes = $result_check_paquetes->fetch_assoc();
        if ($row_check_paquetes['paquete_count'] > 1) {
            $error_message = "Los usuarios del Plan Básico solo pueden
seleccionar un paquete adicional.";
        }
    } elseif ($paquetes_adicionales == 1 && $duracion_suscripcion !=
'Anual') { // Assuming 1 is the ID for 'Deporte'
        $error_message = "El Pack Deporte solo puede ser contratado si
la duración de la suscripción es de 1 año.";
    }

    if (empty($error_message)) {
        try {
            // Insertar usuario en la tabla Usuarios
            $sql = "INSERT INTO Usuarios (nombre, apellidos, correo,
edad, plan_base, duracion_suscripcion)
                VALUES ('$nombre', '$apellidos', '$email', '$edad',
'$plan_base', '$duracion_suscripcion')";
            if ($con->query($sql) === TRUE) {
                $usuario_id = $con->insert_id; // Obtener el ID del
usuario recién insertado

                // Insertar suscripción en la tabla Suscripciones

```

```

        $sql_suscripcion = "INSERT INTO Suscripciones
(usuario_id, paquete_id)
                                VALUES ('$usuario_id',
'$paquetes_adicionales')";
        if ($con->query($sql_suscripcion) === TRUE) {
            echo "<script>alert('Nuevo usuario y suscripción
creados exitosamente');</script>";
        } else {
            throw new mysqli_sql_exception($con->error);
        }
    } else {
        throw new mysqli_sql_exception($con->error);
    }
} catch (mysqli_sql_exception $e) {
    $error_message = $e->getMessage();
    echo "<script>alert('Error: " . addslashes($error_message)
. "');</script>";
}
} else {
    echo "<script>alert('Error: " . addslashes($error_message) .
"');</script>";
}
}
?>

```

Formulario para introducir los datos del usuario llamando al método POST mencionado previamente, el formulario cuenta con todos los campos correspondientes del usuario, plan y packs que se quiere contratar:

```

<form method="post" action="create_user.php" class="form_usuario">
    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre" name="nombre" required><br>
    <label for="apellidos">Apellidos:</label><br>
    <input type="text" id="apellidos" name="apellidos"
required><br>
    <label for="correo">Correo Electrónico:</label><br>
    <input type="email" id="correo" name="correo" required><br>
    <label for="edad">Edad:</label><br>
    <input type="number" id="edad" name="edad" min="0"
required><br>
    <label for="plan_base">Tipo de Plan Base:</label><br>
    <select id="plan_base" name="plan_base" required>
        <option value="Básico">Básico</option>

```

```

        <option value="Estándar">Estándar</option>
        <option value="Premium">Premium</option>
    </select><br>
    <label for="paquetes_adicionales">Paquetes
Adicionales:</label><br>
    <select id="paquetes_adicionales" name="paquetes_adicionales"
required>
        <option value="1">Deporte</option>
        <option value="2">Cine</option>
        <option value="3">Infantil</option>
    </select><br>
    <label for="duracion_suscripcion">Duración de la
Suscripción:</label><br>
    <select id="duracion_suscripcion" name="duracion_suscripcion"
required>
        <option value="Mensual">Mensual</option>
        <option value="Anual">Anual</option>
    </select><br><br>
    <input type="submit" value="Crear Usuario" id="crear">
    <input type="button" value="Volver" id="volver"
onclick="window.location.href='index.php';">
</form>

```

edit_user.php

Esta página permite modificar el registro del usuario seleccionado en la pantalla principal, mostrando un formulario similar al de la creación de usuario, contando con los datos del usuario seleccionado:

De nuevo, se crean las variables y se conecta a la BBDD por PHP:

```

<?php
$host = "localhost";
$port = 3306;
$socket = "";
$user = "root";
$password = "1234";
$dbname = "StreamWeb";

$con = new mysqli($host, $user, $password, $dbname, $port, $socket)

```

```

        or die('Could not connect to the database server' .
mysqli_connect_error());

```

Se define el método POST ejecutado por el formulario, que recoge los datos del mismo y lanza una consulta INSERT para actualizar los datos del usuario seleccionado con los nuevos introducidos en el formulario.

Al finalizar, se redirige al usuario a la página principal (index) mediante una función de JavaScript, en caso de fallo, se notifica mediante una alerta:

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user_id = $_POST['edit_user_id'];
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $correo = $_POST['correo'];
    $edad = $_POST['edad'];
    $plan_base = $_POST['plan_base'];
    $paquetes_adicionales = $_POST['paquetes_adicionales'];
    $duracion_suscripcion = $_POST['duracion_suscripcion'];

    $sql_update = "UPDATE Usuarios SET nombre='$nombre',
apellidos='$apellidos', correo='$correo', edad='$edad',
plan_base='$plan_base', duracion_suscripcion='$duracion_suscripcion'
WHERE id=$user_id";
    if ($con->query($sql_update) === TRUE) {
        $sql_update_suscripcion = "UPDATE Suscripciones SET
paquete_id='$paquetes_adicionales' WHERE usuario_id=$user_id";
        if ($con->query($sql_update_suscripcion) === TRUE) {
            echo "<script>alert('Usuario actualizado exitosamente');
window.location.href='user_database.php';</script>";
        } else {
            echo "<script>alert('Error al actualizar la suscripción: "
. addslashes($con->error) . "');</script>";
        }
    } else {
        echo "<script>alert('Error al actualizar el usuario: "
. addslashes($con->error) . "');</script>";
    }
} else {
    $user_id = $_GET['edit_user_id'];
    $sql = "SELECT * FROM Usuarios WHERE id = $user_id";
    $result = $con->query($sql);
    $user = $result->fetch_assoc();
}

```

Formulario de editar usuario: contiene los campos previamente rellenos con los datos del usuario seleccionado para editar, y ejecuta mediante POST el método PHP para actualizar los datos:

```
<body>
    <h1>Editar Usuario</h1>
    <form method="post" action="edit_user.php" class="form_usuario">
        <input type="hidden" name="edit_user_id" value="<?php echo
$user['id']; ?>">
        <label for="nombre">Nombre:</label><br>
        <input type="text" id="nombre" name="nombre" value="<?php echo
$user['nombre']; ?>" required><br>
        <label for="apellidos">Apellidos:</label><br>
        <input type="text" id="apellidos" name="apellidos" value="<?php
echo $user['apellidos']; ?>" required><br>
        <label for="correo">Correo Electrónico:</label><br>
        <input type="email" id="correo" name="correo" value="<?php echo
$user['correo']; ?>" required><br>
        <label for="edad">Edad:</label><br>
        <input type="number" id="edad" name="edad" value="<?php echo
$user['edad']; ?>" min="0" required><br>
        <label for="plan_base">Tipo de Plan Base:</label><br>
        <select id="plan_base" name="plan_base" required>
            <option value="Básico" <?php if ($user['plan_base'] ==
'Básico') echo 'selected'; ?>>Básico</option>
            <option value="Estándar" <?php if ($user['plan_base'] ==
'Estándar') echo 'selected'; ?>>Estándar</option>
            <option value="Premium" <?php if ($user['plan_base'] ==
'Premium') echo 'selected'; ?>>Premium</option>
        </select><br>
        <label for="paquetes_adicionales">Paquetes
Adicionales:</label><br>
        <select id="paquetes_adicionales" name="paquetes_adicionales"
required>
            <option value="1">Deporte</option>
            <option value="2">Cine</option>
            <option value="3">Infantil</option>
        </select><br>
        <label for="duracion_suscripcion">Duración de la
Suscripción:</label><br>
```

```

        <select id="duracion_suscripcion" name="duracion_suscripcion"
required>
            <option value="Mensual" <?php if
($user['duracion_suscripcion'] == 'Mensual') echo 'selected';
?>>Mensual</option>
            <option value="Anual" <?php if
($user['duracion_suscripcion'] == 'Anual') echo 'selected';
?>>Anual</option>
        </select><br><br>
        <input type="submit" value="Editar Usuario" id='editar'>
    </form>
</body>
</html>

```

styles.css

Hoja de estilos que da estructura y estilos a las páginas de la web, esta hoja de estilos es común para toda la web y llamada por cada archivo PHP en la cabecera del HTML, definiendo estilos para las tablas, títulos, formularios y botones, asignando dimensiones, colores, formatos de letra y márgenes a todos los elementos mediante sus etiquetas HTML, clases e IDs:

```

/* Estilos para las tablas */
table {
    width: 75%;
    border-collapse: collapse;
    font-family: Arial, sans-serif;
    margin-bottom: 20px;
    margin: 0 auto;
}

table, th, td {
    border: 1px solid black;
}

th, td {
    padding: 15px;
    text-align: left;
}

th {
    background-color: #f2f2f2;
}

```

```

/* Estilos para los textos */
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    margin: 0;
    padding: 0;
}

h1, h2, h3 {
    text-align: center;
    color: #333;
}

form{
    width: 100%;
    margin: 0 auto;
}

label {
    font-weight: bold;
}

input[type="text"], input[type="email"], input[type="number"], select {
    width: 100%;
    padding: 10px;
    margin: 5px 0 20px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

#editar[type="submit"] {
    width: 100%;
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

```

```

#editar[type="submit"]:hover {
    background-color: #45a049;
}

#eliminar[type="submit"] {
    width: 100%;
    background-color: #e42727;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

#eliminar[type="submit"]:hover {
    background-color: #630c0c;
}

#crear[type="submit"] {
    width: 100%;
    background-color: #0d6eff;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

#crear[type="submit"]:hover {
    background-color: #0c4063;
}

.form_usuario{
    width: 50%;
    margin: 0 auto;
}

#volver {
    width: 100%;
    background-color: #962603;
}

```



```

    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

#volver:hover {
    background-color: #630c0c;
}

```

database.sql

Archivo que contiene el script SQL necesario para crear la base de datos, tablas y datos utilizados por la web:

Se crea y accede a la base de datos

```

CREATE DATABASE StreamWeb;
USE StreamWeb;

```

Se crean y alimentan las tablas usuarios, paquetes, suscripciones y planes:

```

-- Tabla de usuarios
CREATE TABLE Usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT NULL,
    correo VARCHAR(100) NOT NULL UNIQUE,
    edad INT NOT NULL,
    plan_base ENUM('Básico', 'Estándar', 'Premium') NOT NULL,
    duracion_suscripcion ENUM('Mensual', 'Anual') NOT NULL
);

-- Tabla de paquetes
CREATE TABLE Paquetes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre ENUM('Deporte', 'Cine', 'Infantil') NOT NULL UNIQUE,
    precio DECIMAL(5,2) NOT NULL
);

-- Insertar precios de paquetes
INSERT INTO Paquetes (nombre, precio) VALUES

```

```

('Deporte', 6.99),
('Cine', 7.99),
('Infantil', 4.99);

-- Tabla de suscripciones de usuarios a paquetes adicionales
CREATE TABLE Suscripciones (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    paquete_id INT NOT NULL,
    FOREIGN KEY (usuario_id) REFERENCES Usuarios(id) ON DELETE CASCADE,
    FOREIGN KEY (paquete_id) REFERENCES Paquetes(id) ON DELETE CASCADE
);

-- Tabla de precios de planes base
CREATE TABLE Planes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre ENUM('Básico', 'Estándar', 'Premium') NOT NULL UNIQUE,
    precio DECIMAL(5,2) NOT NULL
);

-- Insertar precios de planes
INSERT INTO Planes (nombre, precio) VALUES
('Básico', 9.99),
('Estándar', 13.99),
('Premium', 17.99);

```

Implementación Aplicación Web

La empresa *StreamWeb*, se ha puesto en contacto con nuestra empresa, se dedica a ofrecer canales de streaming para diferentes tipos de usuarios. El proyecto actual consiste en desarrollar una aplicación que permita suscribir nuevos usuarios y gestionar sus suscripciones a distintos paquetes de contenido. El sistema debe ser sencillo y flexible, permitiendo dar de alta usuarios eligiendo sus packs preferidos.

Adjunta una imagen de la visión general de la pantalla principal de la web:

ID	Nombre	Apellidos	Correo	Edad	Plan Base	Paquete Adicional	Duración Suscripción	Precio Paquete	Precio Plan	Coste Mensual	Acciones
42	Álvaro	Orgaz	alvaro.orgaz@gmail.com	18	Básico	Deporte	Anual	6.99€	9.99€	16.98€	<div>Editar</div> <div>Eliminar</div>
43	Sergio	García Rodríguez	s.garcia@gmail.com	32	Premium	Cine	Mensual	7.99€	17.99€	25.98€	<div>Editar</div> <div>Eliminar</div>
44	Marta	Perez	marta.p@gmail.com	32	Básico	Cine	Anual	7.99€	9.99€	17.98€	<div>Editar</div> <div>Eliminar</div>

Crear Usuario

Información de Planes

Pack	Precio Mensual €
Deporte	6.99€
Cine	7.99€
Infantil	4.99€

Plan	Precio Mensual €
Básico	9.99€
Estándar	13.99€
Premium	17.99€

La plataforma, permitirá la contratación de diversos paquetes de entretenimiento, y tu rol será implementar una solución que cumpla con los siguientes requisitos:

Alta de nuevos usuarios

Se podrán dar de alta usuarios en el sistema, se pedirán los siguientes datos:

- **Nombre y apellidos del usuario.**
- **Correo electrónico** (debe tener un formato ser válido y único).
- **Edad:** los usuarios menores de 18 años solo pueden acceder a ciertos paquetes.
- **Tipo de plan base:** se puede elegir entre los planes Básico, Estándar, o Premium.
- **Paquetes adicionales:** los usuarios podrán seleccionar entre Deporte, Cine, o infantil.
- **Duración de la suscripción:** se podrá elegir entre mensual o anual.

La web dispone de un formulario, accesible desde la pantalla principal, para crear nuevos usuarios con sus respectivos planes y suscripciones. El formulario cuenta con todos los campos necesarios para alimentar las tablas de la base de datos, además de manejar las excepciones requeridas.

Crear Usuario

Nombre:

Álvaro

Apellidos:

Orgaz

Correo Electrónico:

alvaro.orgaz@gmail.com

Edad:

18

Tipo de Plan Base:

Básico

Paquetes Adicionales:

Deporte

Duración de la Suscripción:

Anual

Crear Usuario

Volver

El usuario se crea exitosamente:

Usuarios

ID	Nombre	Apellidos	Correo	Edad	Plan Base	Paquete Adicional	Duración Suscripción	Precio Paquete	Precio Plan	Coste Mensual	Acciones
41	Álvaro	Orgaz	alvaro.orgaz@gmail.com	18	Básico	Deporte	Anual	6.99€	9.99€	16.98€	<div>Editar</div> <div>Eliminar</div>

Crear Usuario

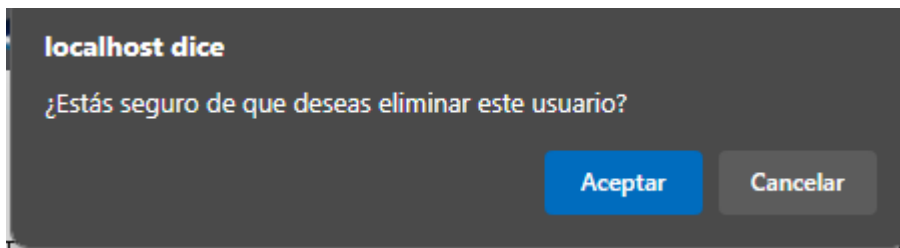
Eliminación de usuarios

- Se podrán eliminar usuarios

Acciones
<div>Editar</div> <div>Eliminar</div>

La página principal con el listado de usuarios dispone de una columna Acciones, que permite editar y eliminar cada uno.

La web notifica de esta eliminación y si esta se acepta, el usuario se elimina de la base de datos:



En este caso, como sólo tenemos un usuario, el listado se actualiza y no hay usuarios a mostrar, pudiendo ver que este usuario ya no está en la base de datos:

Usuarios

ID	Nombre	Apellidos	Correo	Edad	Plan Base	Paquete Adicional	Duración Suscripción	Precio Paquete	Precio Plan	Coste Mensual	Acciones
No hay datos disponibles											
Crear Usuario											

Se muestra página con todos los usuarios registrados

La web cuenta con una pantalla principal en la que se muestran todos los usuarios registrados junto con sus planes y precios contratados:

Usuarios

ID	Nombre	Apellidos	Correo	Edad	Plan Base	Paquete Adicional	Duración Suscripción	Precio Paquete	Precio Plan	Coste Mensual	Acciones
42	Álvaro	Orgaz	alvaro.orgaz@gmail.com	18	Básico	Deporte	Anual	6.99€	9.99€	16.98€	<div>Editar</div> <div>Eliminar</div>
43	Sergio	García Rodríguez	s.garcia@gmail.com	26	Básico	Cine	Mensual	7.99€	9.99€	17.98€	<div>Editar</div> <div>Eliminar</div>
44	Marta	Perez	marta.p@gmail.com	32	Básico	Cine	Anual	7.99€	9.99€	17.98€	<div>Editar</div> <div>Eliminar</div>
Crear Usuario											

Modificar usuario

actualizar su plan base o paquetes adicionales en cualquier momento

Editar

Como se ha mencionado anteriormente, la columna Acciones cuenta con el botón Editar, que abre un formulario con los datos del usuario seleccionado para modificarlos:

Eliminar

En este caso, vamos a modificar la edad y el plan del segundo usuario, ID 42, como ejemplo

Editar Usuario

Nombre:

Sergio

Apellidos:

García Rodríguez

Correo Electrónico:

s.garcia@gmail.com

Edad:

32

Tipo de Plan Base:

Premium

Paquetes Adicionales:

Cine

Duración de la Suscripción:

Mensual

Editar Usuario

La web notifica de la modificación del usuario mediante una alerta, y regresa a la página principal:

localhost dice

Usuario actualizado exitosamente

Aceptar

El usuario se muestra con los datos editados:

43	Sergio	García Rodríguez	s.garcia@gmail.com	32	Premium	Cine	Mensual	7.99€	17.99€	25.98€	<div> <div>Editar</div> <div>Eliminar</div> </div>
----	--------	------------------	--------------------	----	---------	------	---------	-------	--------	--------	--

Mostrar el coste mensual desglosado de cada usuario

- La plataforma debe mostrar el costo total mensual desglosado, según el usuario seleccionado.

La pantalla principal cuenta con la tabla de usuarios registrados como se ha mencionado anteriormente, las tres últimas columnas de datos muestran los precios del paquete, plan y coste total de cada usuario respectivamente:

Paquete Adicional	Duración Suscripción	Precio Paquete	Precio Plan	Coste Mensual
Deporte	Anual	6.99€	9.99€	16.98€
Cine	Mensual	7.99€	17.99€	25.98€
Cine	Anual	7.99€	9.99€	17.98€

Adicionalmente, se han añadido dos tablas con los precios de los paquetes y de los planes como medio informativo, teniendo los datos de la base de datos, de manera que si se modifica un precio en un futuro, se verá reflejado en la web

Información de Planes

Pack	Precio Mensual €
Deporte	6.99€
Cine	7.99€
Infantil	4.99€

Plan	Precio Mensual €
Básico	9.99€
Estándar	13.99€
Premium	17.99€

BIBLIOGRAFÍA:

Para realizar este trabajo he utilizado un poco de ayuda de Chat GPT, para los script de SQL:

[ChatGPT](#)

También he utilizado la documentación de PHP y HTML de w3schools:

<https://www.w3schools.com/>

He consultado errores y dudas en el foro StackOverflow:

<https://stackoverflow.com/>

Herramientas utilizadas:

Visual Studio Code como IDE para desarrollar:

[Visual Studio: IDE y Editor de código para desarrolladores de software y Teams](#)

GitHub y GitHub Desktop para gestionar el repositorio del proyecto

[GitHub · Build and ship software on a single, collaborative platform](#)

[Download GitHub Desktop | GitHub Desktop](#)

