UAX Universidad Alfonso X el Sabio

Proyecto Watchlist

Camila Valles Lolas

Ingeniería Web I

1. Objetivo

El objetivo de este proyecto es que los estudiantes desarrollen una aplicación web sencilla pero funcional, aplicando las tecnologías y herramientas presentadas durante el curso: HTML, CSS, JavaScript, PHP, MySQL y Node.js (opcional). El proyecto está diseñado para alinearse con el contenido de cada unidad didáctica y clase, permitiendo que los estudiantes avancen de manera progresiva y organizada.

2. Watchlist

En este proyecto, hemos decidido realizar un página web sencilla enfocada a poder crear un perfil de usuario en el que se guarde el registro de la visualización de diferentes películas o series. Ya sean películas o series en curso (pendientes de ver) o ya vistas.

En primer lugar, hemos preparado el entorno que vamos a utilizar, hemos creado una carpeta para nuestro proyecto dentro de la carpeta de XAMPP, htdocs, y lo hemos conectado con nuestro repositorio de GitHub: [Repositorio Proyecto_Watchlist]

En este caso, utilizaremos XAMPP para guardar los datos. En él, hemos activado Apache y MYSQL:

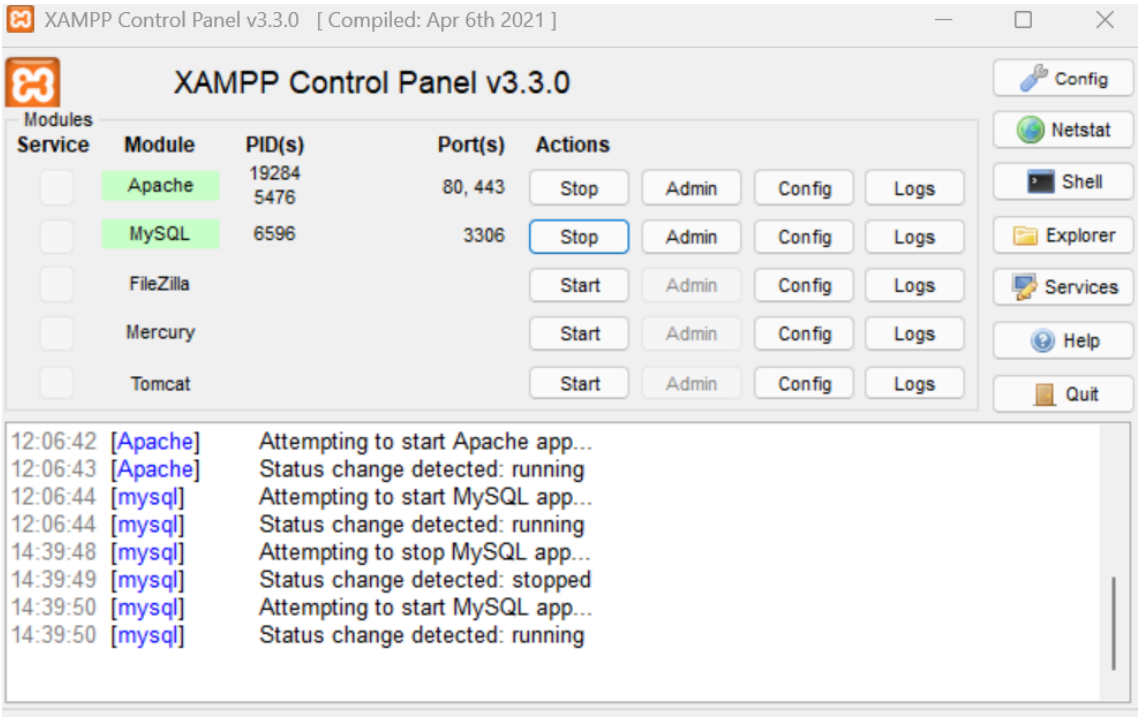


Fig.1 Activación de Apache y MySQL

Una vez activos, hemos creado nuestra base de datos en http://localhost/phpmyadmin :

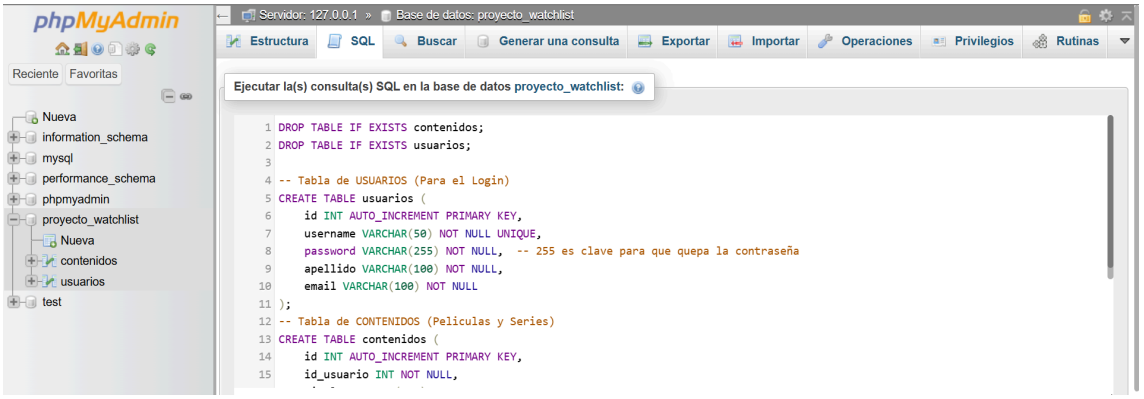


Fig.2 Creación Tablas Usuario y Contenidos SQL

En ella hemos creado dos tablas, una de usuarios y otra de contenidos:

- Tabla usuarios: Para guardar datos personales
- Tabla contenidos: Para guardar las series/pelis. Tiene una clave foránea (id_usuario) para relacionar cada película con su usuario.

A continuación, nos planteamos qué lenguajes o tecnologías vamos a utilizar y a su vez conocer qué archivos vamos a necesitar según la capa de la web:

- Frontend: HTML5, CSS3, JavaScript.
- Backend: PHP (Sin frameworks, código nativo).
- Base de Datos: MySQL.
- Servidor Local: XAMPP (Apache)

Una vez lo conocemos, hemos creado los diferentes archivos para crear la web dentro de nuestra carpeta del proyecto:

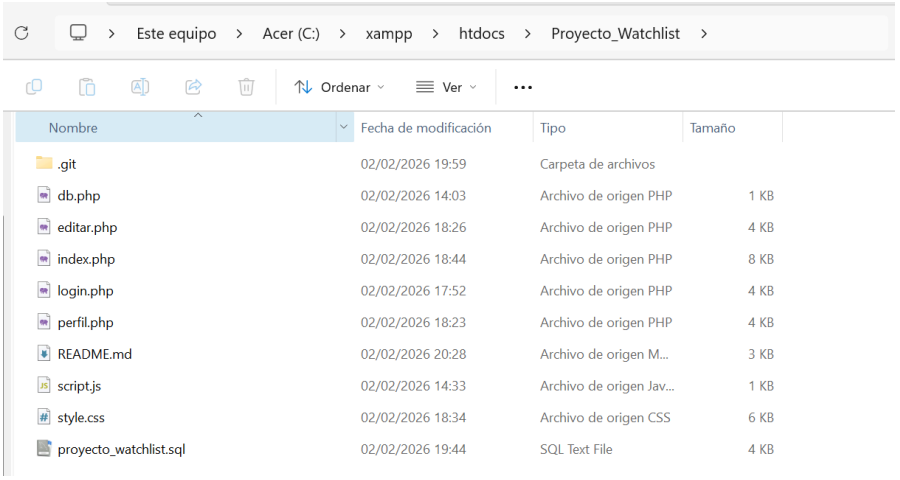


Fig.3 Carpeta Proyecto_Watchlist

Cada uno de los archivos que creamos lo vamos a utilizar para distintas funciones:

- index.php: Dashboard principal (Listas y formularios).
- login.php: Pantalla de acceso y registro.
- editar.php: Pantalla para actualizar el progreso de una serie/película.
- perfil.php: Gestión de cuenta de usuario.
- db.php: Conexión a la base de datos.
- style.css: Estilos visuales.
- script.js: Lógica del frontend (mostrar/ocultar campos).
- proyecto_watchlist.sql: Script de creación de las tablas

Como hemos mencionado sus funciones vamos a tener en cuenta en qué nos van a ayudar en nuestro proyecto:

En primer lugar, mediante el archivo db.php centralizamos la conexión a la base de datos, mientras que con login.php y perfil.php verificamos los datos de usuario y encriptamos contraseñas, las cuales hemos tenido en cuenta que tenga que tener un mínimo de 6 caracteres (password_hash), asegurando que cada usuario acceda únicamente a su información privada .

Por otro lado, para la lógica de negocio utilizamos index.php y editar.php. Estos archivos gestionan el envío de formularios, e implementan la inteligencia necesaria para diferenciar entre películas y series. Gracias a la interacción con script.js en el usuario y las validaciones en PHP en el servidor, conseguimos un sistema que adapta los datos guardados según el tipo de contenido, permitiendo un control de progreso preciso y personalizado.

Finalmente, utilizamos style.css para el estilo visual de la web.

Código:

Index.php

```php
<?php

session_start();

include 'db.php';

if     (!isset($_SESSION['id_usuario']))     {     header("Location: login.php"); exit(); }

$mi_id = $_SESSION['id_usuario'];

// GUARDAR NUEVO

if (isset($_POST['guardar'])) {

    $titulo = $_POST['titulo'];

    $plataforma = $_POST['plataforma'];

    $tipo = $_POST['tipo'];

     $hora = !empty($_POST['hora']) ? $_POST['hora'] : 0;       // Corrección para evitar error si está vacío

    $minuto = !empty($_POST['minuto']) ? $_POST['minuto'] : 0;

    $temporada = ($tipo == 'serie') ? $_POST['temporada'] : 0;

    $episodio = ($tipo == 'serie') ? $_POST['episodio'] : 0;

    $estado = $_POST['estado'];


    $sql = "INSERT INTO contenidos (id_usuario, titulo, plataforma, tipo, hora, minuto, temporada, episodio, estado)

              VALUES ('$mi_id', '$titulo', '$plataforma', '$tipo', '$hora', '$minuto', '$temporada', '$episodio', '$estado')";

    $conn->query($sql);

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

// MOVER A VISTO / BORRAR / RESTAURAR

if (isset($_GET['id']) && isset($_GET['accion'])) {
```

```
    $id = $_GET['id'];
    $accion = $_GET['accion'];
    if ($accion == 'completar') {
          $conn->query("UPDATE contenidos SET estado='vista' WHERE
id=$id AND id_usuario=$mi_id");    }
    if ($accion == 'restaurar') {
           $conn->query("UPDATE contenidos SET estado='pendiente'
WHERE id=$id AND id_usuario=$mi_id");    }
    if ($accion == 'borrar') {
           $conn->query("DELETE FROM contenidos WHERE id=$id AND
id_usuario=$mi_id");    }
    header("Location: index.php");}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
            <meta   name="viewport"   content="width=device-width,
initial-scale=1.0">
    <title>Mi Panel - Watchlist</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <nav>
        <div class="logo">
                                          Hola,   <?php   echo
htmlspecialchars($_SESSION['usuario']); ?>
        </div>
```

```
        <div style="display:flex; gap:10px;">
                        <a  href="perfil.php"  class="btn-logout"
style="border-color:var(--primary);    color:var(--primary);">    Mi
Perfil</a>
                    <a  href="login.php"  class="btn-logout">Cerrar
Sesión</a>
        </div>
    </nav>
    <div class="container">
        <section class="add-section">
                        <form  action="index.php"  method="POST"
class="form-card">
                <h3>Añadir Contenido</h3>
                <div class="form-row">
                                <input  type="text"  name="titulo"
placeholder="Título (ej. Inception)" required>
                    <select name="plataforma">
                                <option>Netflix</option><option>HBO
Max</option><option>Prime</option><option>Disney+</option><option>C
ine</option>
                    </select>
                </div>
                <div class="form-row">
                    <select name="tipo" id="selectorTipo">
                        <option value="pelicula">Película </option>
                        <option value="serie">Serie </option>
                    </select>
                    <div class="time-inputs">
```

```
                                    <input type="number" name="hora"
placeholder="H" min="0"> :
                                <input type="number" name="minuto"
placeholder="Min" min="0" max="59">
                    </div>
                </div>
                <div class="form-row hidden" id="camposSerie">
                                <input type="number" name="temporada"
placeholder="Temp.">
                                 <input type="number" name="episodio"
placeholder="Cap.">
                </div>
                <div class="form-row">
                                            <label style="color:#aaa;
align-self:center;">Estado:</label>
                    <select name="estado">
                                <option value="pendiente"> Para ver
luego</option>
                                            <option value="vista"> Ya
terminada</option>
                    </select>
                </div>
                                <button type="submit" name="guardar"
class="btn-main">Guardar Contenido</button>
            </form>
        </section>
        <section class="dashboard-grid">
            <div class="column">
                <h2 class="col-title"> Pendientes</h2>
```

```
                <div class="scroll-area">
                    <?php
                      $res = $conn->query("SELECT * FROM contenidos
WHERE id_usuario=$mi_id AND estado='pendiente' ORDER BY id DESC");
                    if ($res->num_rows > 0) {
                        while($row = $res->fetch_assoc()) {
                        $icono = ($row['tipo'] == 'serie') ?
                            echo "<div class='card-item'>";
                            echo "<div class='card-info'>";
                              echo "<h4>$icono " . $row['titulo'] .
"</h4>";
                                          echo "<p class='meta'>" .
$row['plataforma'] . " • ";

                             if ($row['tipo'] == 'serie') echo "T" .
$row['temporada'] . ":E" . $row['episodio'] . " • ";

                            $tiempo = "";
                                  if ($row['hora'] > 0) $tiempo .=
$row['hora'] . "h ";
                            $tiempo .= $row['minuto'] . "m";
                            echo "Punto: " . $tiempo;
                            echo "</p></div>";
                            echo "<div class='card-actions'>";
                                  echo "<a href='editar.php?id=" .
$row['id'] . "' class='btn-edit' title='Actualizar progreso'>
Actualizar </a>";
```

```
                                                                    echo  "<a
href='index.php?accion=completar&id="    .    $row['id']    .    "'
class='btn-edit' title='Ya vista'> Visto </a>";

                            echo "</div></div>";

                        }

                    } else {

                            echo "<p class='empty-msg'>¡Estás al día!
Añade algo nuevo.</p>";                    }

                    ?>

                </div>

            </div>

            <div class="column done-column">

                <h2 class="col-title"> Historial Visto</h2>

                <div class="scroll-area">

                    <?php

                      $res = $conn->query("SELECT * FROM contenidos
WHERE id_usuario=$mi_id AND estado='vista' ORDER BY id DESC");

                    while($row = $res->fetch_assoc()) {

                        echo "<div class='card-item done'>";

                        echo "<div style='flex:1;'>";

                                echo "<strong>" . $row['titulo'] .
"</strong>";

                            echo "<br><small style='color:#666'>" .
$row['plataforma'] . "</small>";

                        echo "</div>";

                        echo "<div class='card-actions'>";

                     echo "<a href='index.php?accion=restaurar&id="
.  $row['id']  .  "'  class='btn-edit'  title='Volver  a  Pendientes'>
Pendiente </a>";
```

```
                        echo "<a href='index.php?accion=borrar&id="
. $row['id'] . "' class='btn-mini-trash' title='Eliminar'> Borrar
</a>";

                        echo "</div></div>";                    }

                    ?>

                </div>

            </div>

        </section>

    </div>

    <script src="script.js"></script>

</body>

</html>
```

login.php

```
<?php
session_start();
include 'db.php';
$error = "";
$success = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // --- REGISTRO ---
    if (isset($_POST['registro'])) {
        $user = trim($_POST['username']);
        $apellido = trim($_POST['apellido']);
        $email = trim($_POST['email']);
        $pass = $_POST['password'];
        if (strlen($pass) < 6) {
                    $error = "La contraseña debe tener al menos 6
caracteres.";
        }
             elseif ($conn->query("SELECT id FROM usuarios WHERE
username='$user'")->num_rows > 0) {
            $error = "El usuario '$user' ya existe.";
        }
        else {
            $hash = password_hash($pass, PASSWORD_DEFAULT);
             $stmt = $conn->prepare("INSERT INTO usuarios (username,
apellido, email, password) VALUES (?, ?, ?, ?)");
                $stmt->bind_param("ssss", $user, $apellido, $email,
$hash);
            if ($stmt->execute()) {
```

```
                    $success = "¡Cuenta creada! Sube arriba para
entrar.";
            } else {
                $error = "Error: " . $conn->error;}    }}
    // --- LOGIN ---
    elseif (isset($_POST['login'])) {
        $user = trim($_POST['username']);
        $pass = $_POST['password'];
            $result = $conn->query("SELECT * FROM usuarios WHERE
username='$user'");
        if ($result->num_rows > 0) {
            $row = $result->fetch_assoc();
            if (password_verify($pass, $row['password'])) {
                $_SESSION['usuario'] = $user;
                $_SESSION['id_usuario'] = $row['id'];
                header("Location: index.php");
                exit();
            } else {
                $error = "Contraseña incorrecta.";
            }
        } else {
            $error = "Usuario no encontrado."; }}}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
```

```
            <meta    name="viewport"    content="width=device-width,
initial-scale=1.0">
    <title>Entrar - Watchlist Pro</title>
    <link rel="stylesheet" href="style.css">
</head>
<body class="bg-login">
    <div class="login-container">
        <h1> Watchlist Pro</h1
            <?php  if($error):  ?><div  class="msg  error"><?php  echo
$error; ?></div><?php endif; ?>
          <?php if($success): ?><div class="msg success"><?php echo
$success; ?></div><?php endif; ?>
        <form method="POST">
                                <input   type="text"   name="username"
placeholder="Usuario" required>
                             <input   type="password"   name="password"
placeholder="Contraseña" required>
                                <button   type="submit"   name="login"
class="btn-main">ENTRAR</button>
        </form>
        <div class="separator">
            <span>¿Nuevo por aquí?</span>
        </div>
        <form method="POST">
              <input type="text" name="username" placeholder="Nombre
de Usuario" required>
                                <input   type="text"   name="apellido"
placeholder="Apellidos" required>
```

```
            <input type="email" name="email" placeholder="Correo
Electrónico" required>

                        <input  type="password"  name="password"
placeholder="Crea tu Contraseña" required>

                            <button type="submit" name="registro"
class="btn-sec">CREAR CUENTA</button>

        </form>

    </div>

</body>

</html>
```

db.php

```
<?php

// Configuración de conexión

$conn = new mysqli("localhost", "root", "", "proyecto_watchlist");

if   ($conn->connect_error)   die("Error   de   conexión:   "   .
$conn->connect_error);

// Esto ayuda con las tildes y ñ

$conn->set_charset("utf8");

?>
```

perfil.php

```php
<?php
session_start();
include 'db.php';
if (!isset($_SESSION['id_usuario'])) { header("Location: login.php"); exit(); }
$mi_id = $_SESSION['id_usuario'];
$msg = "";
$tipo_msg = "";
// ACTUALIZAR DATOS PERSONALES
if (isset($_POST['update_info'])) {
    $apellido = $_POST['apellido'];
    $email = $_POST['email'];
    $conn->query("UPDATE usuarios SET apellido='$apellido', email='$email' WHERE id=$mi_id");
    $msg = "Datos actualizados correctamente.";
    $tipo_msg = "success";
}
//  CAMBIAR CONTRASEÑA (Seguridad Alta)
if (isset($_POST['change_pass'])) {
    $old_pass = $_POST['old_pass'];
    $new_pass = $_POST['new_pass'];
    // Buscamos la contraseña actual en la BD
    $res = $conn->query("SELECT password FROM usuarios WHERE id=$mi_id");
    $row = $res->fetch_assoc();
```

```
    // Verificamos si la contraseña vieja es correcta
    if (password_verify($old_pass, $row['password'])) {
        if (strlen($new_pass) >= 6) {
            $new_hash = password_hash($new_pass, PASSWORD_DEFAULT);
            $conn->query("UPDATE usuarios SET password='$new_hash'
WHERE id=$mi_id");
            $msg = "¡Contraseña cambiada con éxito!";
            $tipo_msg = "success";
        } else {
            $msg = "La nueva contraseña debe tener 6 caracteres
mínimo.";
            $tipo_msg = "error";
        }
    } else {
        $msg = "La contraseña actual no es correcta.";
        $tipo_msg = "error";    }}
// Cargar datos actuales para mostrarlos en los inputs
$usuario = $conn->query("SELECT * FROM usuarios WHERE
id=$mi_id")->fetch_assoc();
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Mi Perfil</title>
    <link rel="stylesheet" href="style.css">
</head>
```

```
<body class="bg-login"> <div class="container"
style="max-width:500px;">

        <div class="form-card">

            <div style="display:flex;
justify-content:space-between; align-items:center;">

                <h2> Mi Perfil</h2>

                <a href="index.php" class="btn-edit"> Volver</a>

            </div>

            <?php if($msg): ?>

                <div class="msg <?php echo $tipo_msg; ?>"><?php
echo $msg; ?></div>

            <?php endif; ?>

            <form method="POST" style="margin-bottom:30px;">

                <h3>Mis Datos</h3>

                <div class="form-row">

                    <label style="width:100px;">Usuario:</label>

                    <input type="text" value="<?php echo
$usuario['username']; ?>" disabled style="opacity:0.5;
cursor:not-allowed;">

                </div>

                <div class="form-row">

                    <label style="width:100px;">Apellido:</label>

                    <input type="text" name="apellido" value="<?php
echo $usuario['apellido']; ?>" required>

                </div>

                <div class="form-row">

                    <label style="width:100px;">Email:</label>

                    <input type="email" name="email" value="<?php
echo $usuario['email']; ?>" required>
```

```
                </div>
                <button type="submit" name="update_info"
class="btn-main">Guardar Datos</button>
            </form>
            <hr style="border-color:#444;">
            <form method="POST">
                <h3 style="color:var(--danger);"> Seguridad</h3>
                <div class="input-group"
style="margin-bottom:10px;">
                    <input type="password" name="old_pass"
placeholder="Contraseña Actual" required>
                </div>
                <div class="input-group"
style="margin-bottom:10px;">
                    <input type="password" name="new_pass"
placeholder="Nueva Contraseña (Mín 6)" required>
                </div>
                <button type="submit" name="change_pass"
class="btn-sec" style="border:1px solid var(--primary);
color:var(--primary);">
                    Cambiar Contraseña
                </button>
            </form>
        </div>
    </div>
</body>
</html>
```

script.js

```
document.addEventListener('DOMContentLoaded', () => {
    // Elementos
    const selector = document.getElementById('selectorTipo');
    const camposSerie = document.getElementById('camposSerie');
    // Escuchar cambios en el selector
    selector.addEventListener('change', () => {
        if (selector.value === 'serie') {
            camposSerie.classList.remove('hidden'); // Mostrar
        } else {
            camposSerie.classList.add('hidden'); // Ocultar
        }    });
    // Confirmación al borrar
    const botonesBorrar = document.querySelectorAll('.btn-trash,
.btn-mini-trash');
    botonesBorrar.forEach(btn => {
        btn.addEventListener('click', (e) => {
            if(!confirm("¿Seguro que quieres eliminar esto del
historial?")) {
                e.preventDefault();  }        });    });});
```

style.css

```
:root {
    --bg-color: #121212;
    --card-bg: #1e1e1e;
    --text-color: #e0e0e0;
    --primary: #8a2be2; /* Violeta */
    --accent: #00fa9a;  /* Verde Menta */
    --danger: #ff4757;
    --input-bg: #2c2c2c;}
* {   box-sizing: border-box;}
body {
    font-family: 'Segoe UI', sans-serif;
    background-color: var(--bg-color);
    color: var(--text-color);
    margin: 0; padding: 0;}
/* --- LOGIN ESTILIZADO --- */
.bg-login {
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: linear-gradient(135deg, #0f0c29, #302b63, #24243e);
    padding: 20px;}
.login-container {
    background: rgba(30, 30, 30, 0.95);
    padding: 40px;
```

```
    border-radius: 15px;
    width: 100%;
    max-width: 400px;
    box-shadow: 0 10px 40px rgba(0,0,0,0.5);
    text-align: center;}
h1 { margin-top: 0; color: white; margin-bottom: 30px; }
h3 { margin: 0 0 15px 0; font-weight: normal; color: #aaa;
font-size: 0.9rem; }
/* INPUTS (CAJAS DE TEXTO) - MÁS ESPACIO */
input, select {
    width: 100%;
    padding: 15px;         /* Más gorditos */
    margin-bottom: 20px;   /* MÁS ESPACIO ENTRE ELLOS */
    background: var(--input-bg);
    border: 1px solid #444;
    color: white;
    border-radius: 8px;
    font-size: 1rem;
    transition: 0.3s;}
input:focus {
    border-color: var(--primary);
    outline: none;
    background: #333;}
/* BOTONES GENERALES */
button {
    width: 100%;
    padding: 15px;         /* Botones más altos */
```

```
    border: none;
    border-radius: 8px;
    font-weight: bold;
    cursor: pointer;
    font-size: 1rem;
    transition: 0.3s;
    text-transform: uppercase; /* Letras en mayúscula */
    letter-spacing: 1px;}
/* BOTÓN ENTRAR */
.btn-main {
    background: var(--primary);
    color: white;
    margin-top: 10px;    /* Separado de la contraseña */
    margin-bottom: 20px; /* Separado del texto de abajo */
    box-shadow: 0 4px 15px rgba(138, 43, 226, 0.4);}
.btn-main:hover {
    background: #7022b9;
    transform: translateY(-2px);}
/* BOTÓN CREAR CUENTA */
.btn-sec {
    background: transparent;
    border: 2px solid var(--accent); /* Borde más visible */
    color: var(--accent);
    margin-top: 10px;}
.btn-sec:hover {
    background: var(--accent);
```

```
    color: #121212;
    box-shadow: 0 0 15px var(--accent); }
/* SEPARADOR */
.separator {
    display: flex;
    align-items: center;
    text-align: center;
    margin: 30px 0 20px 0;
    color: #aaa;}
.separator::before, .separator::after {
    content: '';
    flex: 1;
    border-bottom: 1px solid #444;}
.separator span {
    padding: 0 10px;
    font-size: 0.9rem;}
/* MENSAJES DE ERROR */
.msg { padding: 10px; border-radius: 5px; margin-bottom: 15px;
font-size: 0.9rem; }
.error { background: rgba(255, 71, 87, 0.2); color: #ff4757;
border: 1px solid #ff4757; }
.success { background: rgba(0, 250, 154, 0.2); color: #00fa9a;
border: 1px solid #00fa9a; }
/* --- RESTO DE LA WEB --- */
nav { background: var(--card-bg); padding: 15px 30px; display:
flex; justify-content: space-between; align-items: center;
box-shadow: 0 2px 10px rgba(0,0,0,0.5); }
.logo { font-weight: bold; color: var(--accent); }
```

```
.btn-logout { color: var(--danger); text-decoration: none; border: 1px solid var(--danger); padding: 5px 15px; border-radius: 20px; font-size: 0.9rem; }

.container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }

.dashboard-grid { display: flex; gap: 20px; margin-top: 20px; }

.column { flex: 1; background: var(--card-bg); padding: 20px; border-radius: 10px; min-height: 400px; }

.col-title { border-bottom: 2px solid #333; padding-bottom: 10px; color: var(--accent); margin-top:0;}

.card-item { background: #2c2c2c; padding: 15px; margin-bottom: 10px; border-radius: 8px; display: flex; justify-content: space-between; border-left: 3px solid #555; }

.card-item.done { opacity: 0.7; background-color: #1a1f1a; border-left: 3px solid var(--accent); }

.card-item.done span { text-decoration: none; color: #aaa; }

.card-item.done:hover { opacity: 1; }

.card-actions a { text-decoration: none; margin-left: 10px; font-size: 1.2rem; }

.hidden { display: none !important; }

.add-section .form-card { background: var(--card-bg); padding: 20px; border-radius: 10px; }

.form-row { display: flex; gap: 10px; margin-bottom: 10px; }

.time-inputs { display: flex; align-items: center; gap: 5px; color: #aaa; }

.time-inputs input { text-align: center; }


/* --- BOTONES DE ACCIÓN EN LAS TARJETAS --- */

/* Estilo base para todos (Actualizar, Visto, Borrar) */

.btn-edit, .btn-check, .btn-trash, .btn-mini-trash {
```

```
    text-decoration: none;        /* Quita el subrayado */
    color: white !important;
    font-size: 0.8rem;            /* Tamaño de letra discreto */
    padding: 5px 10px;            /* Relleno para que parezca botón
*/
    border: 1px solid #666;       /* Borde gris fino */
    border-radius: 4px;           /* Bordes un poco redondeados */
    margin-left: 5px;             /* Espacio entre botones */
    transition: 0.3s;
    display: inline-block;
    background: transparent;}
.btn-edit:hover {
    background: white;
    color: #121212 !important;
    border-color: white;}
/* Visto: Se vuelve verde */
.btn-check:hover {
    background: var(--accent);
    color: #121212 !important;
    border-color: var(--accent);}
/* Borrar: Se vuelve rojo */
.btn-trash:hover, .btn-mini-trash:hover {
    background: var(--danger);
    color: white !important;
    border-color: var(--danger);}
@media (max-width: 768px) { .dashboard-grid { flex-direction:
column; } }
```

Una vez realizado el código, pasamos a subir/actualizar nuestro archivos en el repositorio de GitHub a través de Git Bash:

```
Cami@LAPTOP-O12GO3O7 MINGW64 /c/xampp/htdocs/Proyecto_Watchlist (main)
$ git add .

Cami@LAPTOP-O12GO3O7 MINGW64 /c/xampp/htdocs/Proyecto_Watchlist (main)
$ git commit -m "Subiendo proyecto completo"
[main 93d9e52] Subiendo proyecto completo
 8 files changed, 750 insertions(+), 1 deletion(-)
 create mode 100644 db.php
 create mode 100644 editar.php
 create mode 100644 index.php
 create mode 100644 login.php
 create mode 100644 perfil.php
 create mode 100644 script.js
 create mode 100644 style.css

Cami@LAPTOP-O12GO3O7 MINGW64 /c/xampp/htdocs/Proyecto_Watchlist (main)
$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 10.47 KiB | 765.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/camivalles99/Proyecto_Watchlist.git
   8637383..93d9e52  main -> main
```

Fig.4 Conexión con Repositorio Proyecto_Watchlist con Git Bash

Finalmente, pasamos a comprobar la web que hemos creado en http://localhost/Proyecto_Watchlist/ :

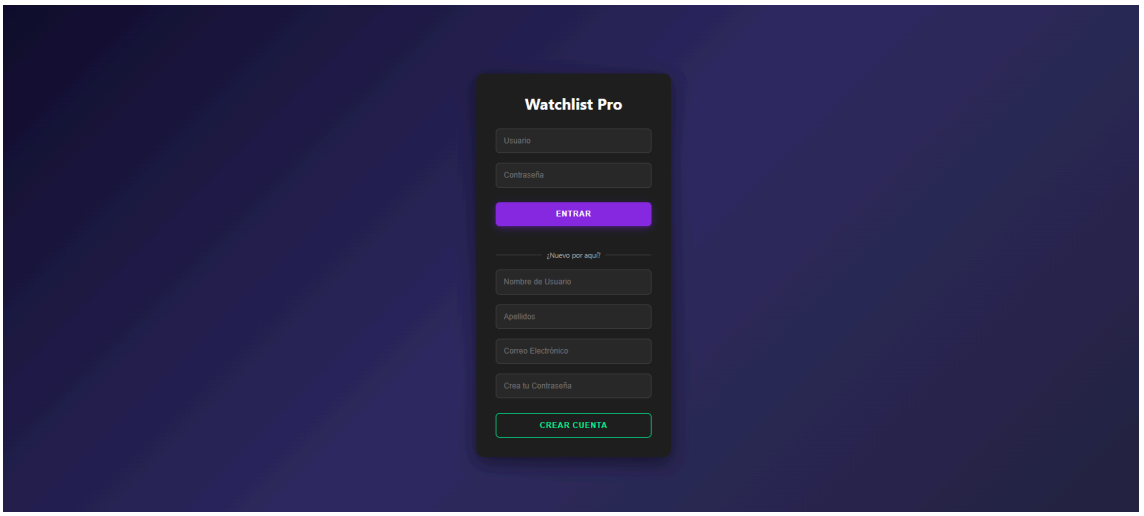


Fig.5 Log in Web Watchlist Pro

Podemos comprobar cómo la web nos pide iniciar sesión con nuestro usuario o bien registrarnos como tal. Para implementar estas funciones, hemos implementado el código correspondiente en login.php, que nos gestiona tanto el formulario de entrada como el de registro, verificando las credenciales contra la base de datos o creando el nuevo usuario.

Una vez dentro, nos permite añadir nombres de series o películas, seleccionar si ya las hemos visto (“Ya terminada”) o si la queremos guardar “Para ver luego”. En este último caso, hemos tenido en cuenta la posibilidad de que la película o serie esté empezada, por lo que si es una película, tendremos la opción de colocar la hora y/o minuto en el que la hemos parado la película, y si es una serie, podemos añadir lo mismo, la temporada y/o episodio en el que estamos.

A su vez hemos añadido la opción de elegir la plataforma dónde ver la película o serie.

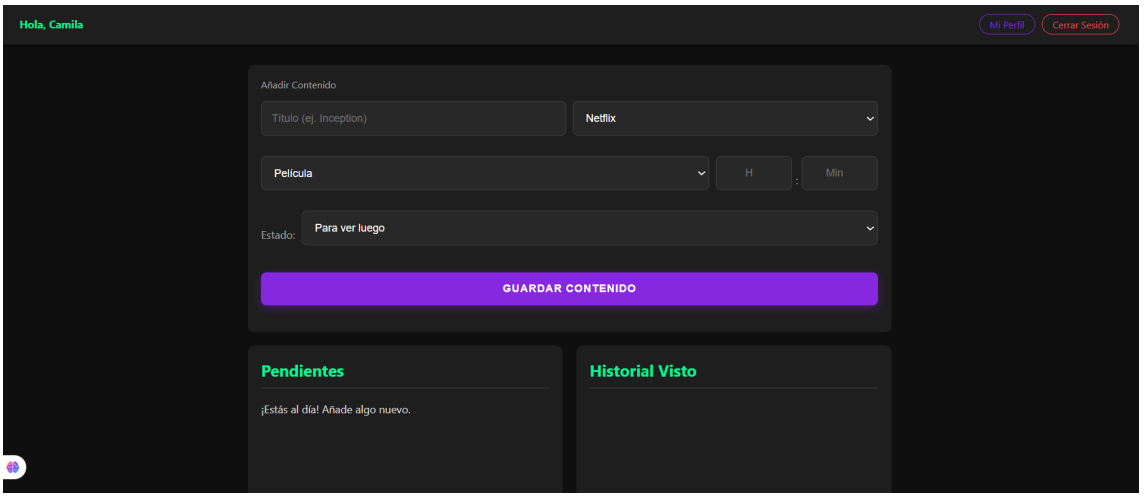


Fig.6 Inicio Web Watchlist Películas

Para estas funciones, hemos implementado el código correspondiente en los siguientes archivos:

- login.php: Gestiona tanto el formulario de entrada como el de registro, verificando las credenciales contra la base de datos o creando el nuevo usuario.
- index.php: aquí incluimos el código HTML del formulario para añadir contenidos (título, plataforma, estado) y la lógica PHP (INSERT) que recibe esos datos y los guarda en la base de datos.
- script.js: es el que comprueba qué tipo hemos seleccionado (serie o película) y nos enseña u ocultar los campos de "hora/minuto"o "temporada/episodio" sin necesidad de recargar la página.

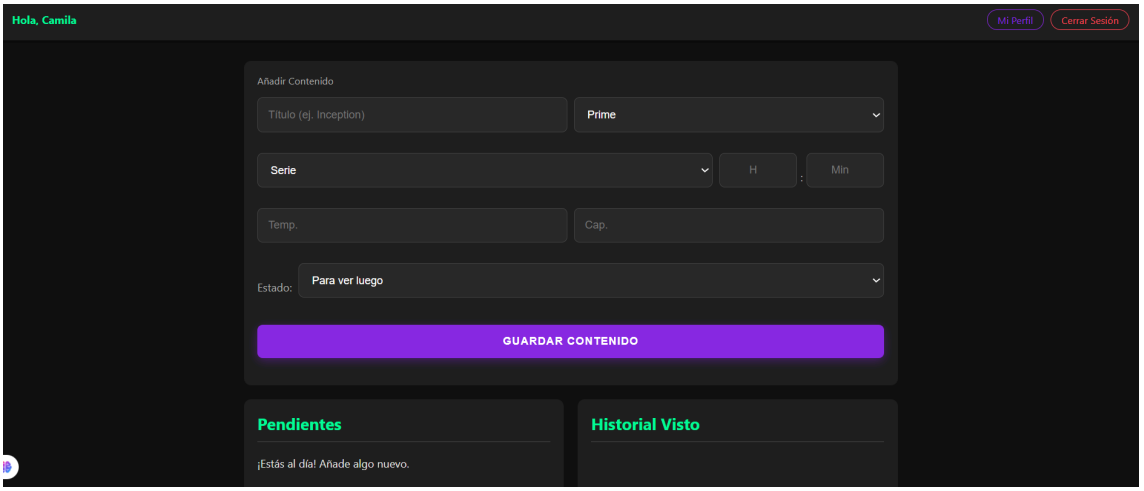


Fig.7 Inicio Web Watchlist Series

Una vez rellenamos el formulario, pasamos a “Guardar contenido”, donde según la opción seleccionada “Para luego/Ya terminada”, lo guardaremos en “Pendientes/Historial Visto”:

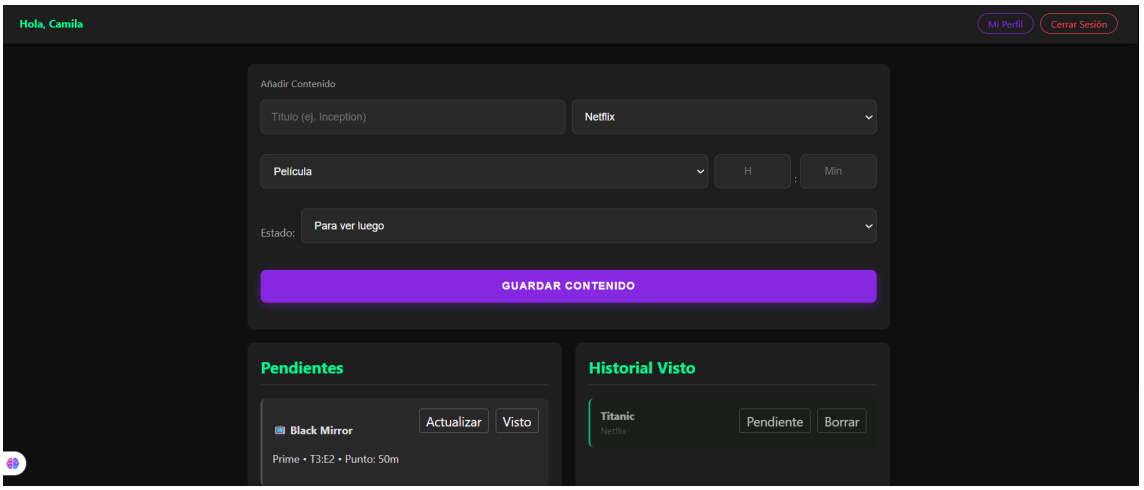


Fig.8 Lista Series/Películas Web Watchlist

Para estas funciones, hemos implementado el código correspondiente en: index.php: contiene el formulario HTML para introducir los datos y, lel código PHP en la parte superior que recibe esa información ($_POST) y ejecuta la sentencia INSERT en la base de datos, guardando el estado correcto ('pendiente' o 'vista') según lo que eligió el usuario.

En “Pendientes”, podemos comprobar que tenemos las opciones “Actualizar”, donde podemos cambiar las horas/minutos o el episodio/temporada (en el caso de las series), y “Visto”, donde si lo seleccionamos pasará automáticamente a “Historial Visto”:

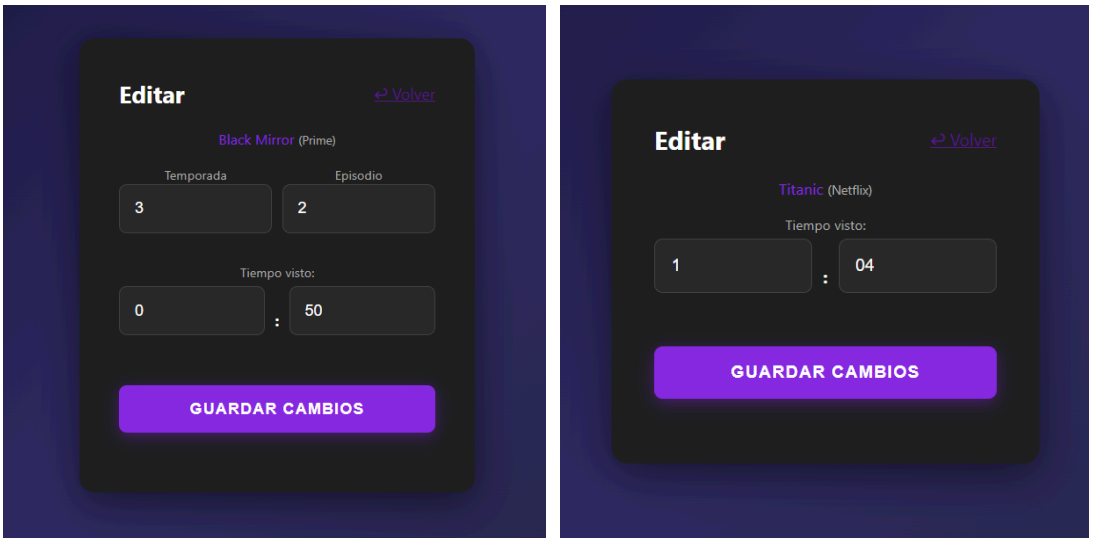


Fig.9 Actualizar Serie // Actualizar Película

Por otro lado, en “Historial ya visto”, tenemos la opción “Pendiente” que nos permite retornar la película o serie a “Pendientes” o simplemente “Borrar” para eliminarlo.

Para estas funciones, hemos implementado el código correspondiente en los siguientes archivos:

- index.php: le encargamos mostrar los botones de acción en cada tarjeta y contiene la lógica para el botón "Visto". Al pulsarlo, este archivo recarga la página, detecta la acción y actualiza el estado en la base de datos para moverlo al historial automáticamente. La gestión del historial también la centralizamos en este archivo. Este script detecta a través de la URL si el usuario ha solicitado una acción específica (como 'restaurar' o 'borrar') y ejecuta las consultas SQL correspondientes (UPDATE o DELETE) para modificar el estado del contenido o eliminarlo definitivamente de la base de datos
- editar.php: con él gestionamos la opción "Actualizar". Cuando el usuario quiere cambiar horas o episodios, este archivo carga los datos actuales de esa entrada específica, muestra el formulario de edición y ejecuta la sentencia UPDATE para guardar los cambios de progreso.

Dentro de la página de inicio de nuestra web, tenemos la opción de entrar en el perfil de usuario “Mi perfil “ o de “Cerrar sesión” arriba a la derecha.

Si entramos en nuestro perfil, comprobamos que tenemos la opción de actualizar la contraseña, nuestro email y/o apellido:

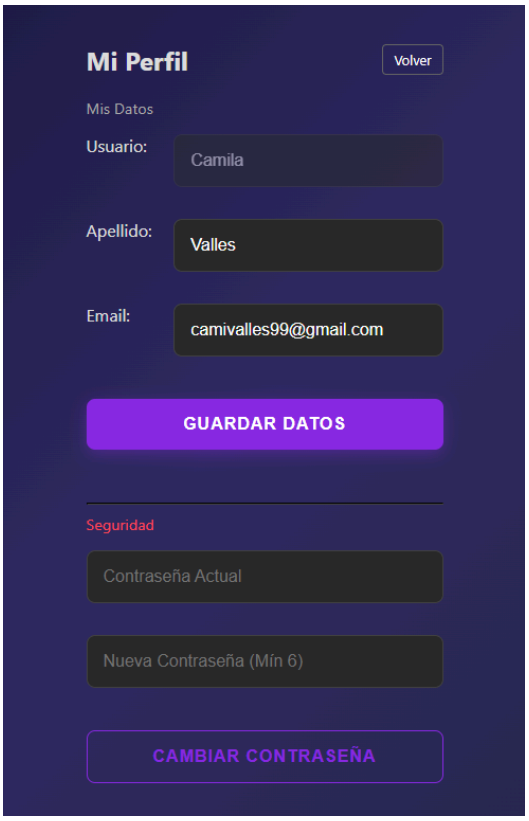


Fig.10 Mi perfil Web Watchlist

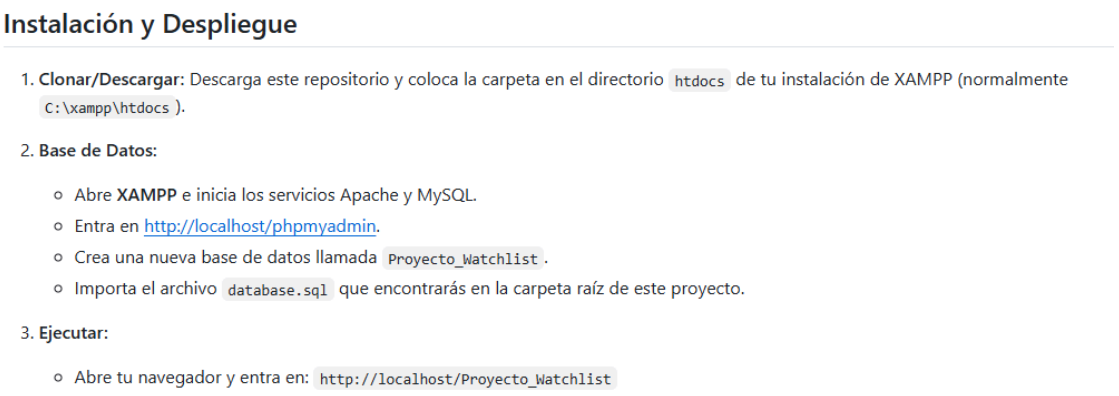
En cambio, si seleccionamos la opción de cerrar sesión nos devolverá al página de inicio de sesión.

El acceso a estas funciones se encuentra en la barra de navegación situada en index.php. Al acceder a 'Mi Perfil', el sistema nos dirige al archivo perfil.php, diseñado específicamente para la gestión de la cuenta. Este script recupera los datos del usuario actual mediante consultas SQL (SELECT) y gestiona de forma segura las peticiones de actualización (UPDATE) de credenciales y datos personales. Incluye medidas de seguridad importantes, como verificar la contraseña antigua antes de permitir crear una nueva (usando password_verify y password_hash).

Finalmente, el botón de 'Cerrar Sesión' nos redirige a login.php, finalizando el flujo de navegación dentro del área privada.

Instrucciones de ejecución

Para ejecutar este proyecto, he redactado en el archivo "README.md" las instrucciones:



Instalación y Despliegue

1. **Clonar/Descargar:** Descarga este repositorio y coloca la carpeta en el directorio `htdocs` de tu instalación de XAMPP (normalmente `C:\xampp\htdocs`).
2. **Base de Datos:**
   - Abre **XAMPP** e inicia los servicios Apache y MySQL.
   - Entra en http://localhost/phpmyadmin.
   - Crea una nueva base de datos llamada `Proyecto_Watchlist`.
   - Importa el archivo `database.sql` que encontrarás en la carpeta raíz de este proyecto.
3. **Ejecutar:**
   - Abre tu navegador y entra en: `http://localhost/Proyecto_Watchlist`

Fig.11 Instalación y despliegue README.md

-Clonar/Descargar: Descarga este repositorio y coloca la carpeta en el directorio htdocs de tu instalación de XAMPP (normalmente C:\xampp\htdocs).

-Base de Datos:

- Abre XAMPP e inicia los servicios Apache y MySQL.
- Entra en http://localhost/phpmyadmin.
- Crea una nueva base de datos llamada Proyecto_Watchlist.
- Importa el archivo database.sql que encontrarás en la carpeta raíz de este proyecto.

-Ejecutar:

- Abre tu navegador y entra en: http://localhost/Proyecto_Watchlist

UAX

Universidad
Alfonso X el Sabio

GRACIAS

UAX.COM