

TRABAJO DE FIN DE GRADO



REALIZADO POR MARIO LIBERATO Y JOSÉ ANDRÉS SEGOVIA BAIRD
ADMINISTRACIÓN SISTEMAS INFORMÁTICOS EN RED
2023/2024

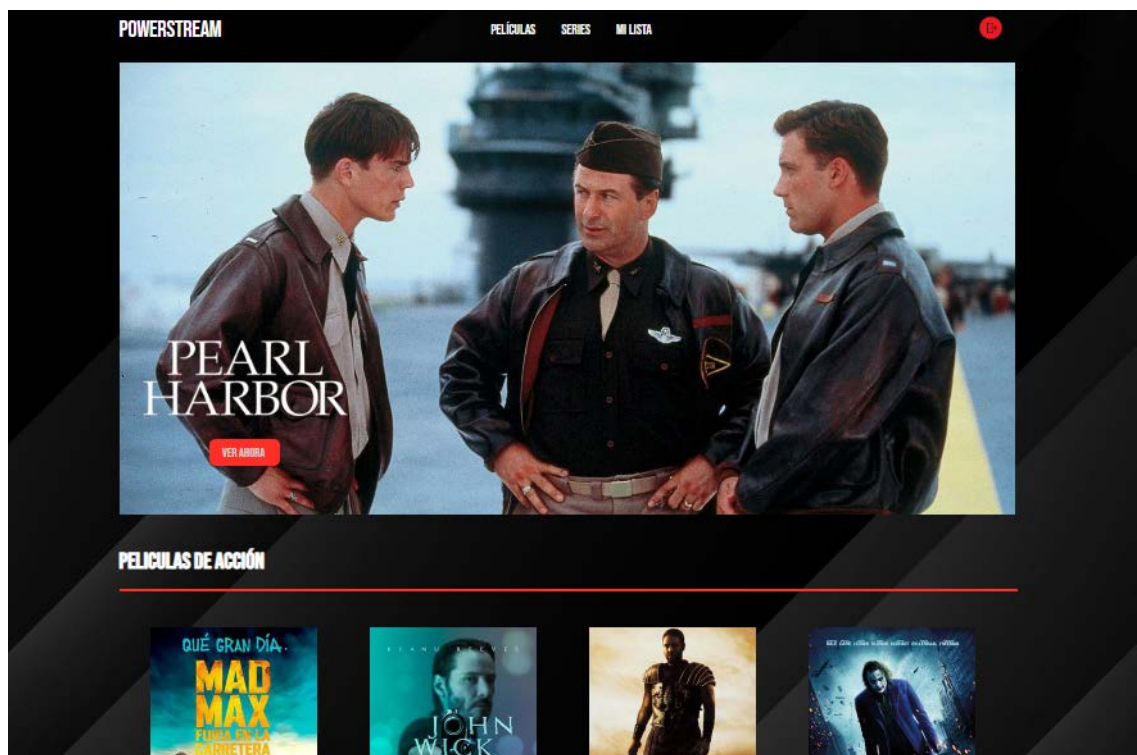
INDICE

ANTEPROYECTO	2
HERRAMIENTAS	3
LENGUAJES.....	4
PROCESO	6
INDEX.....	11
AÑADIR LISTA.....	18
REPRODUCTOR.....	23
REGISTRO	26
PLANES DE PAGO.....	29
PAGO PAYPAL	30
DASHBOARD	38
BIBLIOGRAFÍA	43

ANTEPROYECTO

La idea de nuestro proyecto es crear una plataforma de streaming basándonos en las plataformas de streaming existentes como Amazon Prime Video, Netflix, etc. Añadiéndole nuevas funciones tales como creación de listas de reproducción personalizadas, sugerencias de nuevos contenidos que la plataforma considere que te puedan interesar en función de tus gustos.

Utilizaremos principalmente PHP y CSS para la base de nuestra página web y Javascript para darle interacción a la web, almacenaremos los datos de nuestros usuarios y películas en una base de datos con MySQL donde se añadirá la información que vaya recogiendo la web.



HERRAMIENTAS

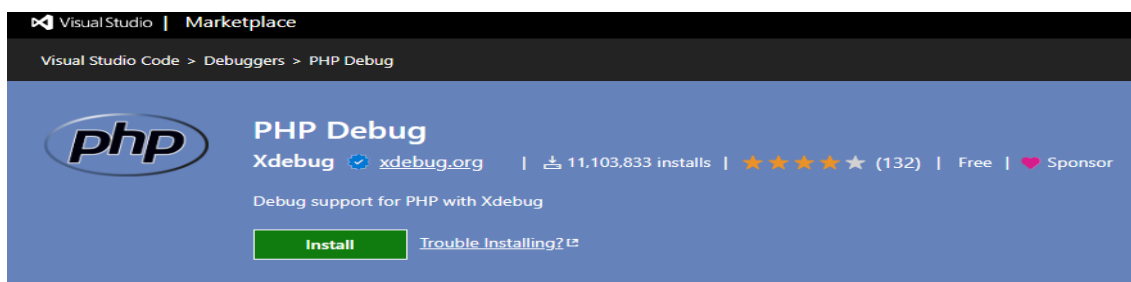
Lo primero de todo tendremos que tener instalado **AppServ**, que nos proporcionará soporte tanto para Apache, PHP y MySQL.

Podemos descargarlo desde la página principal de AppServ <https://www.appserv.org/en/>.

Como editor utilizaremos una herramienta llamada **Visual Studio Code**, que podemos descargar desde el siguiente enlace: <https://code.visualstudio.com/> y donde instalaremos varias extensiones que nos ayudarán a la hora de programar.

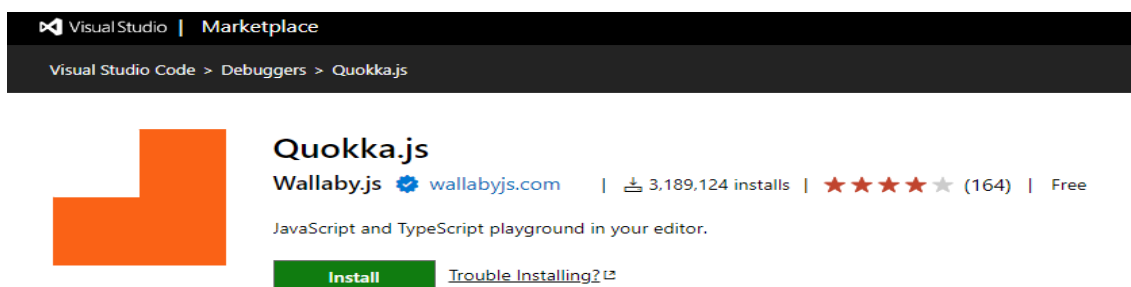
Las **extensiones** que vamos a utilizar son:

- **PHP Debug:** extensión que nos ayuda con la sintaxis de PHP.



<https://marketplace.visualstudio.com/items?itemName=xdebug.php-debug>.

- **Quokka.js:** extensión que nos ayuda con la sintaxis de Javascript.



<https://marketplace.visualstudio.com/items?itemName=WallabyJs.quokka-vscode>

LENGUAJES

Hemos utilizado diferentes lenguajes para poder programar nuestra página web, estos son los lenguajes que hemos utilizado:

- **HTML:** HTML, o Lenguaje de Marcas de Hipertexto, es el componente más fundamental de la web, ya que establece la estructura y el significado del contenido en línea.

Este lenguaje utiliza etiquetas para marcar texto, imágenes y otros elementos, permitiendo que los navegadores web interpreten y muestren el contenido correctamente. El término "hipertexto" se refiere a los enlaces que conectan páginas web entre sí, creando la estructura de la World Wide Web. HTML utiliza una variedad de elementos especiales, como `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, entre otros, para organizar y presentar el contenido de manera semántica.

Además de HTML, otras tecnologías como CSS y JavaScript se utilizan comúnmente para controlar el aspecto visual y el comportamiento interactivo de las páginas web. Con HTML, los usuarios pueden crear y compartir contenido en la web, convirtiéndose así en participantes activos en la Red Informática Mundial.

- **CSS:** CSS, o Hojas de Estilo en Cascada, es el lenguaje utilizado para controlar la presentación y el aspecto visual de documentos HTML o XML. Esto incluye la definición de estilos para la pantalla, papel, habla y otros medios. CSS es esencial en la creación de sitios web y otros documentos en línea, ya que permite establecer el diseño, el formato y la disposición de los elementos estructurados.

- **JAVASCRIPT:** es un lenguaje de programación versátil y dinámico, tanto interpretado como compilado justo-a-tiempo (just-in-time), y con funciones de primera clase. Aunque es ampliamente conocido como un lenguaje de scripting para páginas web, JavaScript también se utiliza en una variedad de entornos fuera del navegador, como Node.js, Apache CouchDB y Adobe Acrobat.

Este lenguaje se basa en prototipos y es multiparadigma, lo que significa que admite varios estilos de programación, incluyendo la programación

orientada a objetos, imperativa y declarativa (como la programación funcional). Además, JavaScript es un lenguaje de un solo hilo, lo que significa que puede realizar una tarea a la vez.

- **PHP:** PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Es ampliamente utilizado en el desarrollo web para crear páginas web dinámicas. Permite la generación de contenido personalizado en función de las interacciones del usuario, como formularios, botones y otras acciones.

Este se integra fácilmente con HTML, lo que permite incrustar código PHP directamente en documentos HTML para generar contenido dinámico.

Es compatible con una variedad de bases de datos, incluyendo MySQL, PostgreSQL y SQLite, lo que permite a los desarrolladores crear aplicaciones web que interactúen con sistemas de gestión de bases de datos.

- **SQL:** El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos.

PROCESO

Ahora vamos a explicar el progreso que hemos seguido hasta construir toda la plataforma de streaming.

Lo primero que tuvimos que pensar fue el logo y nombre que queríamos para nuestra plataforma de streaming, después de estar mirando posibles logos y planteando posibles nombres terminamos decidiéndonos por **PowerStream** y el logo la cara de un león.



Una vez decidido el nombre y logo lo siguiente era decidir el formato que queríamos que tenga nuestra web, después de observar algunas de las plataformas más famosas como youtube, netflix, etc empezamos a desarrollar nuestra página de inicio.

Nuestra web mantiene un mismo formato para las páginas de inicio y las distintas membresías, pero cada una con ciertas características. La página de inicio te permite ver todos nuestros títulos más recientes disponibles en nuestra web clasificados por género y por tipo, sea película o serie. Para mostrar las películas y series disponibles en nuestra web, almacenamos todos los datos de todos los títulos disponibles en una base de datos.

La estructura de nuestra base de datos se divide en 3 tablas diferentes, que son una tabla de usuarios, una de vídeos y una tabla llamada lista que guarda los títulos de los distintos usuarios que ellos elijan guardar en su lista.

```
create table usuarios(  
  id_usuario int auto_increment primary key,  
  email varchar(50) unique not null,  
  password varchar(255) not null,  
  plan varchar(20) not null,  
  fecha_suscripcion date not null,  
  fecha_renovacion date not null,  
  pagado boolean  
);  
  
create table videos(  
  id_video int auto_increment primary key,  
  titulo varchar(255) unique not null,  
  descripcion varchar(500) not null,  
  tipo varchar(40) not null,  
  genero varchar(50) not null,  
  fecha_subida date not null,  
  video varchar(255) not null,  
  img_portada varchar(255) not null,  
  img_publi varchar(255) NOT NULL,  
  nom_publi varchar(255) NOT NULL  
);  
  
create table lista(  
  id_video int references videos(id_video),  
  id_usuario int references usuarios(id_usuario),  
  titulo varchar(40) references videos(titulo),  
  tipo varchar(40) not null,  
  img_portada varchar(40) not null  
);
```


Una vez definida la base de datos, ya empezamos a desarrollar el código de nuestra página de inicio para que muestre los distintos títulos disponibles clasificados por su género, para hacer esto hemos usado bucles para mostrar todas las coincidencias en la base de datos con lo que quiere mostrar el bucle. Lo primero hemos creado 2 archivos php que se encargan de hacer las distintas consultas a la base de datos, vamos a ver ambos archivos a continuación:

Imagenespeliculas.php

Primero empezamos por conectarnos a la base de datos

```
<?php
$servername = "localhost";
$username = "root";
$password = "rootroot";
$dbname = "powerstream";

// Crear una conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Chequea la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
```

Una vez que hemos establecido la conexión con la base de datos, realizamos varias consultas, la primera consulta será para extraer los datos de las películas más recientes, que serán las que mostraremos a modo de cartelera en nuestra página principal.

```
// Consulta SQL para obtener los detalles de las películas para la parte que cambia
$sql = "SELECT * FROM videos WHERE tipo = 'pelicula' ORDER BY id_video DESC LIMIT 5";
$result = $conn->query($sql);
```

Y ahora realizaremos una consulta por cada género, de forma que tengamos una lista de películas por género y las guardaremos en distintas variables según el género.

```
// Consulta SQL para obtener los detalles de las películas para la parte que cambia
$sql = "SELECT * FROM videos WHERE tipo = 'pelicula' ORDER BY rand() LIMIT 5";
$result = $conn->query($sql);

// Añadir las películas de la base de datos

$sqlaccion = "SELECT * FROM videos WHERE genero = 'accion' and tipo = 'pelicula'";
$result_accion = $conn->query($sqlaccion);

$sqlterror = "SELECT * FROM videos WHERE genero = 'terror' and tipo = 'pelicula'";
$result_terror = $conn->query($sqlterror);

$sqlromanticas = "SELECT * FROM videos WHERE genero = 'romantica' and tipo = 'pelicula'";
$result_romanticas = $conn->query($sqlromanticas);

$sqlcomedia = "SELECT * FROM videos WHERE genero = 'comedia' and tipo = 'pelicula'";
$result_comedia = $conn->query($sqlcomedia);
```

Siguiendo la misma estructura, tenemos el segundo archivo:

Imagenesseries.php

Al igual que el archivo php anterior comenzamos por realizar la conexión a la base de datos.

```
<?php
$servername = "localhost";
$username = "root";
$password = "rootroot";
$database = "powerstream";

// Crear una conexión
$conn = new mysqli($servername, $username, $password, $database);

// Chequea la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
```

Una vez realizada la conexión empezamos por hacer una consulta para mostrar las últimas series subidas en la base de datos para luego mostrarlas en la cartelera.

```
// Consulta SQL para obtener los detalles de las películas para la parte que cambia  
$sql = "SELECT * FROM videos WHERE tipo = 'serie' ORDER BY id_video DESC LIMIT 5";  
$result = $conn->query($sql);
```

Y por último hacemos las consultas por cada género y las guardamos en distintas variables por cada género.

```
// Añadir las series de la base de datos  
  
$sqlaccion = "SELECT * FROM videos WHERE genero = 'accion' and tipo = 'serie';"  
$result_accion = $conn->query($sqlaccion);  
  
$sqlterror = "SELECT * FROM videos WHERE genero = 'terror' and tipo = 'serie';"  
$result_terror = $conn->query($sqlterror);  
  
$sqlromanticas = "SELECT * FROM videos WHERE genero = 'romantica' and tipo = 'serie';"  
$result_romanticas = $conn->query($sqlromanticas);  
  
$sqlcomedia = "SELECT * FROM videos WHERE genero = 'comedia' and tipo = 'serie';"  
$result_comedia = $conn->query($sqlcomedia);
```

INDEX

Una vez que ya tenemos definidos ambos archivos podemos seguir con la página principal, arriba del todo en la cabecera añadimos el nombre de la web, 3 pestañas una que muestra las películas, otra las series y una última que muestra la lista personalizada del usuario. Al lado derecho añadimos 2 botones, uno para iniciar sesión y otro para registrarse en nuestra web.

```
<header class="header">
  <div class="menu container">
    <a href="index.php" class="logo">PowerStream</a>
    <input type="checkbox" id="menu">
    <label for="menu">
      
    </label>
    <nav class="navbar">
      <ul>
        <li><a href="index.php">Películas</a></li>
        <li><a href="default_series.php">Series</a></li>
        <li><a href="registrarse.html">Mi lista</a></li>
      </ul>
    </nav>
    <a href="inicio_sesion.html" class="btn-1">Login</a>
    <a href="registrarse.html" class="btn-1">Registro</a>
  </div>
```

La cabecera es común para todas las páginas de las distintas membresías y las de inicio de películas y series.

Debajo de la cabecera tenemos la cartelera donde se muestran las distintas películas o series más nuevas, donde van rotando cada pocos segundos.

Para lograr que nuestra pagina pueda recoger las imágenes de nuestra base de datos tenemos que añadir la línea **include 'php/imagenespeliculas.php'**; esto nos permitirá hacer una llamada php para poder mostrar los datos

```
<div class="header-content container slides">
  <?php
  // Incluir el archivo PHP que realiza la consulta SQL
  include 'php/imagenespeliculas.php';

  // Verificar si hay resultados de la consulta antes de mostrar los datos
  if ($result->num_rows > 0) {
    <div class="header-container">
      <?php
      $i = 0;
      while ($row = $result->fetch_assoc()) {
        <?php if($i == 0) { ?>
          <div class="header-1 active" style="background-image:url(php/img/<?php echo $row['img_publi']; ?>);">
            <?php } else { ?>
              <div class="header-1" style="background-image:url(php/img/<?php echo $row['img_publi']; ?>); "> Unreach
              <?php } ?>
              
              <div>
                <a href="inicio_sesion.html" class="btn-2">Ver ahora</a>
              </div>
            </div>
          <?php
          $i++;
        }
      }
      </div> <!-- Cierre de header-container -->
      <?php
      } else {
        // Si no hay resultados, mostrar un mensaje alternativo
        echo "No se encontraron resultados.";
      }
    }
  }
</div>
```

almacenados en las queries que tenemos en la página **imagenespeliculas.php** que a su vez imprimirá por pantalla los diferentes elementos que necesitamos de la query almacenada en arrays.

Primero comprobamos si hay algún resultado en caso de haber resultados hacemos un bucle while para ir mostrando todos los valores y por cada valor mostramos la imagen **img_publi** de fondo con la imagen **nom_publi** encima. Por último tendremos un botón de ver ahora que en las páginas de inicio nos redirigirá al inicio de sesión ya que sin haber iniciado sesión no podremos ver ninguna película ni serie.

La cartelera va rotando gracias a unas funciones definidas en nuestro javascript, tenemos una primera función llamada *showSlides* que se encarga de la rotación cada 5s de las imágenes.

```
showSlides();

function showSlides() {
  for (var i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }

  slideIndex++;
  if (slideIndex > slides.length - 1) {slideIndex = 0} // Reiniciar índice si se llega al final

  slides[slideIndex].style.display = "block";

  interval = setTimeout(showSlides, 5000); // Cambiar imagen cada 5 segundos
}
```

Y posteriormente tenemos 2 pequeñas funciones para que si el usuario tiene el ratón encima de la imagen que se está mostrando, se detenga la rotación hasta que el usuario quite el ratón de encima de la imagen.

```
// Detener el deslizamiento automático cuando el puntero está sobre una imagen
for (var i = 0; i < slides.length; i++) {
  slides[i].addEventListener("mouseover", stopSlide);
  slides[i].addEventListener("mouseout", startSlide);
}

function stopSlide() {
  clearTimeout(interval);
}

function startSlide() {
  interval = setTimeout(showSlides, 5000);
}
```

Después de la cartelera, tenemos la lista de películas o series dependiendo de en qué página nos encontremos, separadas por género.

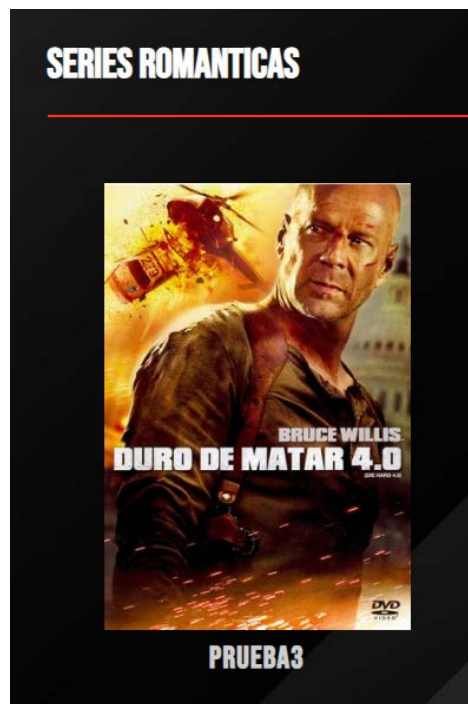
Primero comprobamos si hay alguna coincidencia, para ello heredaremos las variables de cada género que definimos en el archivo php anterior.

```
// Verificar si hay resultados de la consulta antes de mostrar los datos
if ($result_drama->num_rows > 0) {
```

En el caso de que haya coincidencias mediante un bucle while vamos mostrando las distintas películas o series de esa categoría, mostrando los campos de la base de datos `img_portada` y `título`.

```
while ($row = $result_drama->fetch_assoc()) {  
  
    <div class="box-2 <?php if($i == 0) echo 'active'; ?>">  
        <div class="content">  
            ">  
            <h2><?php echo $row['titulo']; ?></h2>  
        </div>  
    </div>  
}
```

De forma que en nuestra web se muestra así:



Esta estructura se repite en cada género, cambiando las variables en función del género que corresponda.

Lo siguiente que diseñamos fueron las distintas páginas de cada membresía, todas tienen el mismo estilo que la página principal pero dependiendo de la membresía tienen algunos cambios.

La membresía más barata llamada **básica**, te permite ver las 3 últimas películas y series subidas a nuestra plataforma, de forma que la página de la membresía básica no tendrá los botones de mostrar más y mostrar menos que tiene la página principal y las otras membresías.

```
<div class="load-more" id="load-more-6">Cargar más</div>  
<div class="load-less" id="load-less-6">Cargar menos</div>
```

La funcionalidad de estos botones está definida en el javascript de cada membresía, tanto la membresía gold como la premium.

Primero definimos una función para el botón de cargar más, que aparte de mostrar más películas o series cambiará a mostrar menos.

```
// Cargar más películas/series  
  
function cargarMasItems(btn, cantidadInicial, claseObjetivo, btnMenosId) {  
  let currentItem = cantidadInicial;  
  let loadLessBtn = document.querySelector('#' + btnMenosId);  
  
  btn.onclick = () => {  
    let contenedores = [...document.querySelectorAll("." + claseObjetivo)];  
    for (let i = currentItem; i < currentItem + 3 && i < contenedores.length; i++) {  
      contenedores[i].style.display = 'inline-block';  
    }  
    currentItem += 3;  
    if (currentItem >= contenedores.length // currentItem >= 6) {  
      btn.style.display = 'none';  
      loadLessBtn.style.display = 'inline-block'; // Mostrar el botón "Cargar menos"  
    }  
  }  
};
```


Y ahora definimos otra función para el botón de cargar menos cuya estructura será muy similar al botón de cargar más.

```
// Cargar menos películas/series

loadLessBtn.onclick = () => {
  let contenedores = [...document.querySelectorAll("." + claseObjetivo)];
  for (let i = currentItem - 3; i < currentItem && i < contenedores.length; i++) {
    contenedores[i].style.display = 'none';
  }
  currentItem -= 3;
  if (currentItem <= cantidadInicial) {
    loadLessBtn.style.display = 'none'; // Ocultar el botón "Cargar menos"
    btn.style.display = 'inline-block'; // Mostrar el botón "Cargar más" nuevamente
  }
};
}
```

Y por último para esta parte de los botones, tenemos definidas todas las variables que utilizamos en dichas funciones, que heredan el id de cada botón que está enumerado del 1 al 6 por cada género.

```
let loadMoreBtn1 = document.querySelector('#load-more-1');
let loadLessBtn1 = document.querySelector('#load-less-1');
let loadMoreBtn2 = document.querySelector('#load-more-2');
let loadLessBtn2 = document.querySelector('#load-less-2');
let loadMoreBtn3 = document.querySelector('#load-more-3');
let loadLessBtn3 = document.querySelector('#load-less-3');
let loadMoreBtn4 = document.querySelector('#load-more-4');
let loadLessBtn4 = document.querySelector('#load-less-4');
let loadMoreBtn5 = document.querySelector('#load-more-5');
let loadLessBtn5 = document.querySelector('#load-less-5');
let loadMoreBtn6 = document.querySelector('#load-more-6');
let loadLessBtn6 = document.querySelector('#load-less-6');

cargarMasItems(loadMoreBtn1, 3, "box-container-1 .box-1", "load-less-1");
cargarMasItems(loadMoreBtn2, 3, "box-container-2 .box-2", "load-less-2");
cargarMasItems(loadMoreBtn3, 3, "box-container-3 .box-3", "load-less-3");
cargarMasItems(loadMoreBtn4, 3, "box-container-4 .box-4", "load-less-4");
cargarMasItems(loadMoreBtn5, 3, "box-container-5 .box-5", "load-less-5");
cargarMasItems(loadMoreBtn6, 3, "box-container-6 .box-6", "load-less-6");
```

Debajo de cada película o serie, una vez iniciada la sesión de dicho usuario, tendrán 2 botones que serán para reproducir dicha película o serie o para añadirla a su lista personalizada.



Ambos botones están definidos en cada género de las distintas películas/series.

```
<div class="opciones">
  <a href="reproductor.php?id_video=<?php echo $row['id_video']; ?>"><div class="reproducir">Reproducir</div></a>
  <div class="reproducir" onclick="añadirALista(<?php echo $row['id_video']; ?>, '<?php echo $row['titulo']; ?>',
    '<?php echo $row['tipo']; ?>', '<?php echo $row['img_portada']; ?>')">Añadir a la lista</div>
</div>
```

El botón de reproducir simplemente redirige al usuario a otra página llamada **reproductor.php** y le pasa el id_video de dicha película o serie.

AÑADIR LISTA

El botón de añadir a lista llama a una función *añadirALista()* y le pasa a esa función varios valores como son el id_video, titulo, tipo e img_portada.

La función está definida en el javascript de cada membresía.

```
function añadirALista(id_video,titulo, tipo, img_portada) {  
    // Realizar una solicitud al servidor para añadir la película a la lista de reproducción  
    var xhr = new XMLHttpRequest();  
    xhr.open("POST", "guardar_pelicula.php", true);  
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
    xhr.onreadystatechange = function() {  
        if (xhr.readyState == 4 && xhr.status == 200) {  
            console.log(xhr.responseText); // Mostrar la respuesta del servidor (opcional)  
        }  
    };  
    // Enviar datos a guardar_pelicula.php  
    xhr.send("id_video=" + id_video + "&titulo=" + titulo + "&tipo=" + tipo + "&img_portada=" + img_portada);  
}
```

Esta función hereda todos esos valores y se los pasa a otra página mediante una llamada AJAX llamada ***guardar_pelicula.php***.

El archivo ***guardar_pelicula.php*** primero hereda los valores que le pasa el javascript y los guarda en distintas variables.

```
$id = $_POST['id_video'];  
$userid = $_SESSION['userid'];  
$titulo = $_POST['titulo'];  
$tipo = $_POST['tipo'];  
$portada = $_POST['img_portada'];
```

Hace una conexión con la base de datos.

```
$conexion = mysqli_connect("localhost", "root", "rootroot") or die ("No se puede conectar con el servidor");  
mysqli_select_db($conexion, "powerstream") or die ("No se puede conectar a la base de datos");
```

Y una vez conectada a la base de datos, hace una primera comprobación para ver si ya está registrada esa película o serie en la tabla lista de la base de datos, en caso de que esté registrada no la añadirá de nuevo.

```
$query_inicial = "SELECT * FROM lista WHERE id_video = '$id' and id_usuario = '$userid'";  
$resultado = mysqli_query($conexion, $query_inicial);  
$coincidencias = mysqli_num_rows($resultado);  
if ($coincidencias > 0) {  
    echo "Esa película ya se encuentra en tu lista";  
    exit();  
}
```

Y si no hay ninguna coincidencia para esa película o serie la añadirá a la tabla de lista con un insert.

```
if ($coincidencias == 0) {  
    $query = "INSERT INTO lista (id_video, id_usuario, titulo, tipo, img_portada) VALUES ('$id', '$userid', '$titulo', '$tipo', '$portada')";  
  
    if (mysqli_query($conexion, $query)) {  
        echo "Película añadida a tu lista";  
    } else {  
        echo "Error al añadir película: " . mysqli_error($conexion);  
    }  
}
```

Ahora que tenemos un botón para agregar las películas o series a la tabla lista, tenemos que crear una página para poder ver las películas y series que tengamos guardadas en nuestra lista personalizada, para ello tenemos el archivo **lista.php**.

Lo primero que tendremos en nuestro archivo de lista, será una comprobación de variables de sesión, ya que la lista es personalizada para cada usuario mostraremos las películas y series guardadas por cada usuario en función de su *id_usuario*.

Heredamos también de la sesión el plan que tiene dicho usuario (básico, gold o premium) y definimos otras variables a las que asignaremos el nombre del archivo de cada membresía.

```
session_start();
$id_usuario = $_SESSION['userid'];
$plan = $_SESSION['plan'];
$basico = 'basico.php';
$gold = 'gold.php';
$premium = 'premium.php';
$series_basico = 'basicoserie.php';
$series_gold = 'goldserie.php';
$series_premium = 'premiumserie.php';
```

Estas otras variables que definimos nos servirán para que cuando el usuario quiera volver a la página de películas o series acceda a la de su membresía, para ello definimos varias condiciones y en función de su plan le asignamos un valor u otro a los enlaces de películas y series, en caso de no tener ningún valor la variable de sesión de plan te redirige automáticamente a la página principal.

La lista está separada en películas y series en vez de por géneros como está separado en las distintas membresías.

```
if ($plan == 'basico')
{
    $enlace = $basico;
    $enlace2 = $series_basico;
}
elseif ($plan == 'gold')
{
    $enlace = $gold;
    $enlace2 = $series_gold;
}
elseif ($plan == 'premium')
{
    $enlace = $premium;
    $enlace2 = $series_premium;
}
else
{
    header("Location: index.php");
}
```

La estructura del código de lista es muy similar a la de las membresías, tiene el botón de reproducir que funciona exactamente igual que en las membresías y luego tiene otro botón que en vez de ser para añadir a la lista es para eliminarla de la lista.

```
<div class="opciones">
  <a href="reproductor.php?id_video=<?php echo $row['id_video']; ?>"><div class="reproducir">Reproducir</div></a>
  <div class="reproducir" onClick="QuitardeLista(<?php echo $row['id_video']; ?>)">Quitar de la lista</div>
</div>
```

El botón de quitar de la lista llama a una función *QuitardeLista()* que está definida en el javascript de lista.

```
function QuitardeLista(id_video) {
  // Realizar una solicitud al servidor para eliminar la película de la lista
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "borrar_lista.php", true);
  xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
      console.log(xhr.responseText); // Mostrar la respuesta del servidor (opcional)
      // Recargar la página después de eliminar la película de la lista
      location.reload();
    }
  };
  // Enviar datos a borrar_pelicula.php
  xhr.send("id_video=" + id_video);
}
```

Esta función es parecida a la que vimos anteriormente de añadir a lista, pero para esta función sólo heredamos el valor del *id_video* y se lo pasamos al archivo **borrar_lista.php**.

En este archivo de borrar lista comprobamos si ha recibido algún valor para *id_video* y en caso afirmativo lo almacenamos en una variable.

```
// Verificar si se recibió el ID de la película
if(isset($_POST['id_video'])) {
    $id = $_POST['id_video'];
}
```

Nos conectamos a la base de datos y hacemos una primera comprobación para ver si ese id está en la tabla lista.

```
// Consulta para verificar si la película está en la lista
$query_inicial = "SELECT * FROM lista WHERE id_video = '$id'";
$resultado = mysqli_query($conexion, $query_inicial);
$coincidencias = mysqli_num_rows($resultado);
```

Si ese id se encuentra en la tabla lista eliminará ese id de la tabla.

```
// Si la película está en la lista, procede con la eliminación
$query = "DELETE FROM lista where id_video = $id";
if (mysqli_query($conexion, $query)) {
    // Si la eliminación es exitosa, devolver "success" como respuesta
    echo "success";
} else {
    // Si hay un error al eliminar, devolver "error" como respuesta
    echo "error";
}
```

REPRODUCTOR

Lo siguiente fue definir el archivo para reproducir las películas o series, el archivo ***reproductor.php***.

Empieza comprobando si ha recibido algún *id_video* y se conecta a la base de datos.

```
// Verificar si se proporcionó un ID de película
if (isset($_GET['id_video'])) {
    // Obtener el ID de la película
    $id_pelicula = $_GET['id_video'];

    $conn = mysqli_connect("localhost", "root", "rootroot")
    or die ("No se puede conectar con el servidor");

    mysqli_select_db ($conn, "powerstream") or die ("No se puede conectar a la base de datos");
```

Una vez conectado a la base de datos, hereda todos los valores de los distintos campos de esa película.

```
// Consulta para obtener los detalles de la película específica
$sql = "SELECT * FROM videos WHERE id_video = $id_pelicula";
$result = $conn->query($sql);
```


Mostramos la imagen de portada del vídeo con su título y descripción de la película y mostramos una imagen de play encima de la imagen de portada de la película que al hacer clic sobre ella llamamos a una función llamada `playVideo()`.

```
// Mostrar Los detalles de la película
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    echo '<header class="header">';
    echo '    <div class="menu container">';
    echo '        <a href="#" class="logo" onclick="goBack()">PowerStream</a>'; // Modificación aquí
    echo '        <input type="checkbox" id="menu">';
    echo '        <label for="menu">';
    echo '            ';
    echo '        </label>';
    echo '        <nav class="navbar">';
    echo '            <ul>';
    echo '                <li><a href="#" onclick="goBack()">Películas</a></li>';
    echo '                <li><a href="#" onclick="goBack()">Series</a></li>';
    echo '            </ul>';
    echo '        </nav>';
    echo '        <a href="index.php"></a>';
    echo '    </div>';

    echo '<div class="header-1" id="movie-image" style="background-image:url(php/img/' . $row['img_publi'] . '); ">';
    echo '';
    echo '<br>';
    echo '';
    echo '</div>';
    echo '<div class="texto" id="movie-details">';
    echo '<h2>' . $row['titulo'] . '</h2>';
    echo '<p>' . $row['descripcion'] . '</p>';
    echo '</div>';
    echo '</header>';
}
```

Y luego buscamos el archivo de vídeo una vez heredado el nombre del archivo desde la base de datos.

```
// Añadir el elemento de video
echo '<div class="video-container" id="video-container">';
echo '    <video id="video" controls>';
echo '        <source src="videos/' . $row['video'] . '" type="video/mp4">';
echo '        Your browser does not support the video tag.';
echo '    </video>';
echo '</div>';
```



La función de *playVideo()* está definida en el javascript, es la encargada de ocultar los detalles de la película o serie y mostrar el vídeo.

```
function playVideo() {  
  document.getElementById('movie-image').classList.add('hidden');  
  document.getElementById('movie-details').classList.add('hidden');  
  const videoContainer = document.getElementById('video-container');  
  videoContainer.style.display = 'block';  
  const video = document.getElementById('video');  
  video.play();  
}
```



REGISTRO

Por otro lado, tenemos las páginas para iniciar sesión y para registrarse, primero empezamos por la página para registrarse, comenzando por el archivo **registrarse.html**, tenemos un sencillo formulario donde pedimos correo y contraseña para registrar.

```
<body>
  <h2>Registrarse</h2>
  <form action="plan.php" method="POST" onsubmit="return validar()">
    <label for="email">Correo:</label><br>
    <input type="text" id="email" name="email" required><br><br>

    <label for="password">Contraseña:</label><br>
    <input type="password" id="password" name="password" required><br><br>

    Ya tienes una cuenta? <a href="inicio_sesion.html">Iniciar sesión</a>
    <br><br>
    <input type="submit" value="Siguiete">
  </form>
</body>
```

El formulario al ser enviado devuelve una función *validar()*, la cual está definida en el javascript y comprueba que el correo sea válido y la contraseña tenga un mínimo de 5 caracteres.

```
function validar(){
    var email = document.getElementById('email').value.trim().length;
    var passwd = document.getElementById('password').value.trim().length;
    var mail = document.getElementById('email').value.trim();
    var arroba = mail.indexOf('@');
    var punto = mail.indexOf('.');
    if (email == 0){
        alert("Debe introducir un correo");
        document.getElementById('email').focus();
        return false;
    }
    if (arroba == -1 || punto == -1 || punto < arroba) {
        alert("Escriba un correo válido");
        document.getElementById('email').focus();
        return false;
    }
    if (passwd < 5){
        alert("La contraseña debe tener mínimo 5 caracteres");
        document.getElementById('password').focus();
        return false;
    }
    return true;
}
```

Este formulario nos generara unas variables que se llevan a la página **plan.php**, el funcionamiento de dicha página la explicamos más adelante en la sección de PayPal.

Al comprobar que los datos son correctos y que el pago ha sido efectuado añade al usuario a la base de datos para que cuando el usuario quiera ingresar haga las comprobaciones, valide los datos y le lleve a sus páginas de suscripción.

```
$SERVER['REQUEST_METHOD'] == 'POST') {
// Verificar que los datos se pasaron correctamente
if (isset($_POST['email']) && isset($_POST['password']) && isset($_POST['plan'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];
    $plan = $_POST['plan'];

    // Generar la fecha actual y la fecha de renovación (30 días después)
    $fecha_actual = date('Y-m-d');
    $fecha_renovacion = date('Y-m-d', strtotime('+30 days'));
    $pago = 1;
    // Hash de la contraseña
    $password_hash = password_hash($password, PASSWORD_BCRYPT);

    // Conexión a la base de datos
    $servername = "localhost";
    $username = "root";
    $password_db = "rootroot";
    $dbname = "powerstream";

    $conn = new mysqli($servername, $username, $password_db, $dbname);

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

    // Insertar datos en la base de datos
    $query = "INSERT INTO usuarios (email, password, plan, fecha_suscripcion, fecha_renovacion,pagado) VALUES
    ('$email','$password_hash','$plan','$fecha_actual','$fecha_renovacion','$pago')";
    $stmt = $conn->prepare($query);
```

PLANES DE PAGO

Una vez hemos comprobado que el correo y contraseña cumplan esas características nos envía a un archivo llamado ***plan.php***, el cual nos hará elegir entre los 3 planes disponibles, mostrando las características de cada uno y el precio de la suscripción.

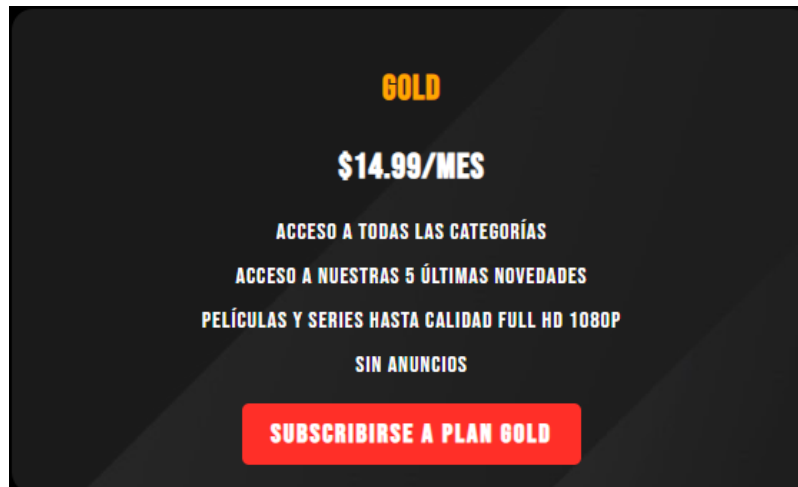
Plan básico

BÁSICO
\$4.99/MES
ACCESO A TODAS LAS CATEGORÍAS
ACCESO A NUESTRAS 3 ÚLTIMAS NOVEDADES
PELÍCULAS Y SERIES HASTA CALIDAD HD 720P
CON ANUNCIOS
SUSCRIBIRSE A PLAN BÁSICO

Plan gold

PREMIUM
\$19.99/MES
ACCESO A TODAS LAS CATEGORÍAS
ACCESO A NUESTRAS 7 ÚLTIMAS NOVEDADES
PELÍCULAS Y SERIES HASTA CALIDAD 4K
SIN ANUNCIOS
SUSCRIBIRSE A PLAN PREMIUM

Plan premium



Una vez elegido el plan que quiera el usuario, le redirigirá a otro archivo llamado **pago.php**, que le dará la opción de pagar mediante paypal o tarjeta de crédito.

PAGO PAYPAL

Para poder realizar pagos mediante la página de **PayPal** primero hemos tenido que registrarnos en dicha plataforma, una vez registrados nos identificamos en el siguiente url:

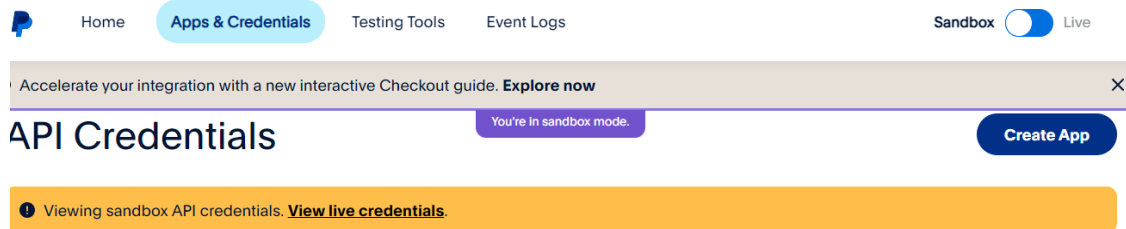
<https://developer.paypal.com/dashboard/accounts>

Una vez aquí comprobaremos que estamos en modo de pruebas sandbox



En este modo tenemos que crear una API la cual va ir relacionada nuestra pagina mediante un **ID** el cual utilizaremos para que desde nuestra página llamemos a **PayPal** y poder registrar los pagos.

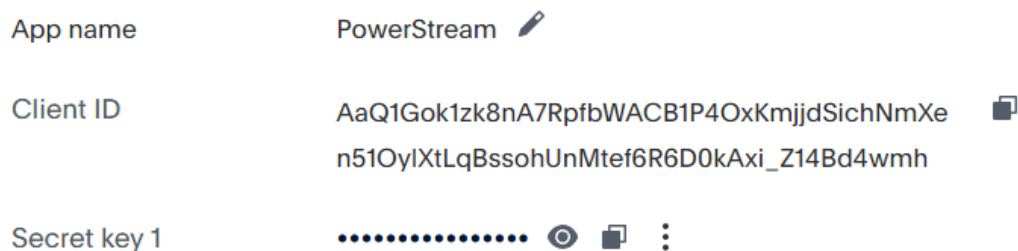
Nos dirigimos a la pestaña “Apps & Credenciales” pulsamos “Create App” y seguimos los pasos para crear nuestra cuenta, cuando la creemos nos aparecerá debajo como se observa en la imagen



REST API apps

App name	Client ID	Secret	Created date	
Default Application	AaQ1Gok1zk8nA7RpfWACB...	22/05/24, 10:32	

Si pulsamos sobre la App creada nos mostrara el nombre de nuestra plataforma y el id el cual necesitamos para editar nuestro formulario.



Este “Client ID” lo copiaremos en el siguiente código el cual te proporciona la propia plataforma para que esta se pueda comunicar con PAY-PAL, a continuación, mostramos la parte del código en la cual se inserta la Id de nuestra App



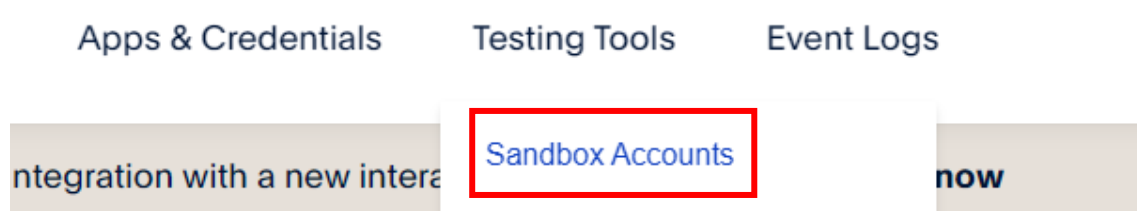
Nosotros modificamos este código para que según la tarifa que elija el usuario cobre el importe correspondiente, en esta parte del código también se define en que tipo de moneda se van a realizar los pagos.

```
createOrder: function(data, actions) {  
  var plan = '<?php echo $plan; ?>';  
  var amount;  
  switch (plan) {  
    case 'basico':  
      amount = 4.99;  
      break;  
    case 'gold':  
      amount = 14.99;  
      break;  
    case 'premium':  
      amount = 19.99;  
      break;  
    default:  
      amount = 0;  
      break;  
  }  
  return actions.order.create({  
    purchase_units: [{ "description": "PAGO" + plan.toUpperCase(), "amount": {"currency_code": "EUR", "value": amount} } ]  
  });  
},
```

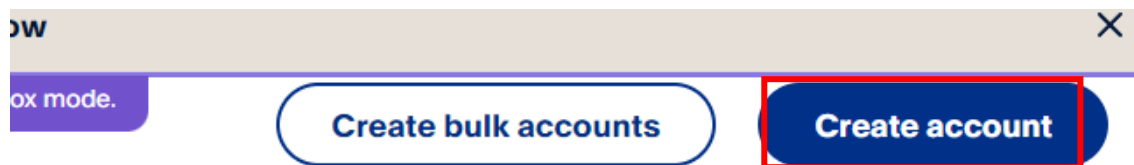
Una vez que se ha efectuado el pago nos redirige a la página de Inicio de sesión.

Para poder hacer pruebas con usuarios ficticios tenemos que crear cuentas de usuario, la plataforma nos permite crear estas cuentas en el modo sandbox.

Nos dirigimos a la pestaña “Testing Tools” en “Sandbox Accounts”



Aquí aparecerá un botón de crear cuentas pulsamos ahí y una vez creado el usuario de pruebas ya podemos utilizar el correo y contraseña aleatorios que ha creado la cuenta para poder hacer pruebas



Se nos abrirá una ventana la cual solo nos hará elegir entre si es una cuenta de negocio o personal y el país e inmediatamente se generará una cuenta con sus credenciales para hacer pruebas.

Create Sandbox Account

Choose your account type and country that will be used for live payments.

Account type:

- ☒ **Personal**
Buyer account
- ☐ **Business**
Merchant account

Country/Region:

Country or region


Spain


▼



Create Account

Una vez creada aparecerá en la pagina principal y si damos clic encima del nombre obtendremos el correo y contraseña de prueba para hacer login en el momento del pago, tambien te genera una tarjeta de crédito para probar el pago en el otro botón que se genera en la página de pago.

<input type="checkbox"/> Account name	Type
<input type="checkbox"/> sb-z00vq31106147@personal.example.com	Personal

Sandbox URL <https://sandbox.paypal.com> 

Email [sb-z00vq31106147@personal.example.com](#)  [Change email](#)

Password   [Change password](#)





Sandbox account info

Name	Andrés Segovia 	Account type	Personal
------	--	--------------	----------

Tarjeta creada para comprobar pagos:

Manage credit cards

Credit cards:

Type	VISA	
Card Number	4020021115403275	
Expiry	07/2029	
CVC Code	Any 3 digits	

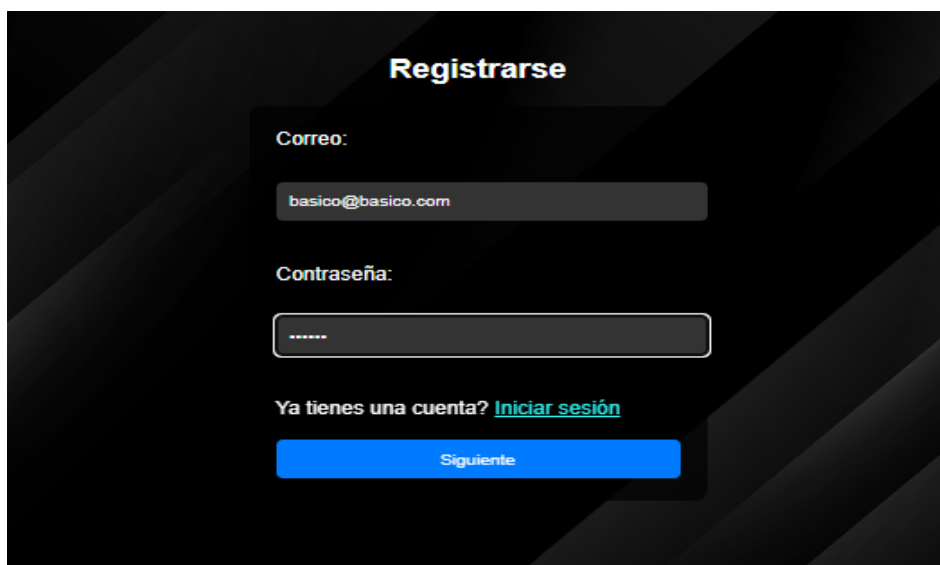
[Generate New Card](#)

Demostración de proceso de registro de usuario:

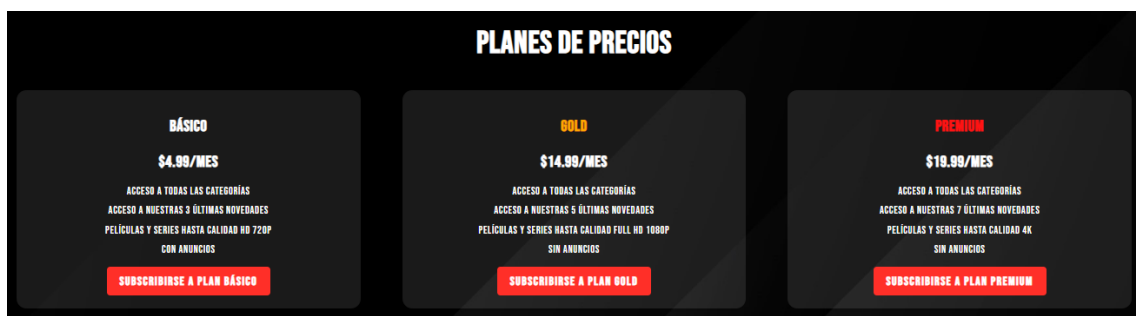
1. Pulsamos botón Registro desde nuestra página principal



2. Introducimos el correo y la contraseña la cual vamos a utilizar para identificarnos en un futuro

A screenshot of the 'Registrarse' (Sign Up) form. The form is titled 'Registrarse' in white text. It has two input fields: 'Correo:' with the email 'basico@basico.com' and 'Contraseña:' with a masked password '*****'. Below the password field is a link that says 'Ya tienes una cuenta? [Iniciar sesión](#)'. At the bottom is a blue button labeled 'Siguiente'.

3. Nos aparecerán las tres opciones para poder elegir tarifa

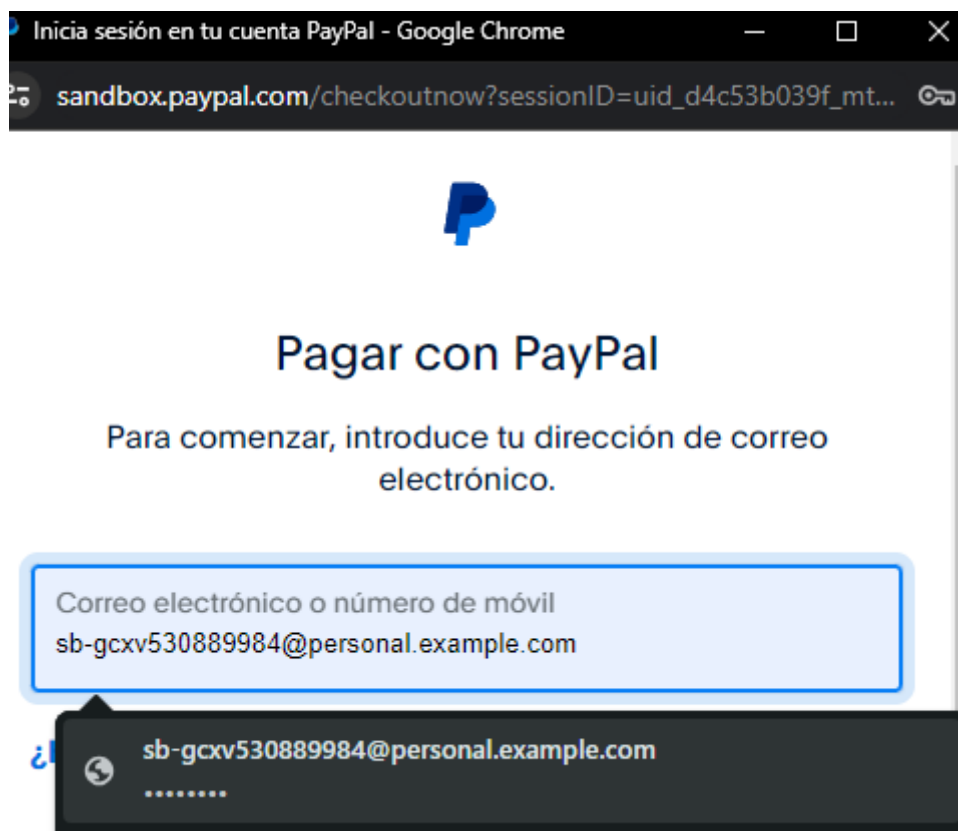
A screenshot of the 'PLANES DE PRECIOS' (Pricing Plans) section. It features three columns representing different subscription plans: 'BÁSICO', 'GOLD', and 'PREMIUM'. Each column lists the price per month, the number of categories, the number of latest releases, the video quality, and whether there are ads. Each column has a red button at the bottom to subscribe to that plan.

PLAN	PRECIO	CATEGORÍAS	ÚLTIMAS NOVEDADES	Calidad	ANUNCIOS
BÁSICO	\$4.99/MES	ACCESO A TODAS LAS CATEGORÍAS	ACCESO A NUESTRAS 3 ÚLTIMAS NOVEDADES	PELÍCULAS Y SERIES HASTA CALIDAD HD 720P	CON ANUNCIOS
GOLD	\$14.99/MES	ACCESO A TODAS LAS CATEGORÍAS	ACCESO A NUESTRAS 6 ÚLTIMAS NOVEDADES	PELÍCULAS Y SERIES HASTA CALIDAD FULL HD 1080P	SIN ANUNCIOS
PREMIUM	\$19.99/MES	ACCESO A TODAS LAS CATEGORÍAS	ACCESO A NUESTRAS 7 ÚLTIMAS NOVEDADES	PELÍCULAS Y SERIES HASTA CALIDAD 4K	SIN ANUNCIOS

4. Una vez elegida la tarifa nos aparecerá el botón de pago de PAY-PAL el cual nos pedirá identificarnos utilizando las cuentas que creamos antes.



5. Al pulsar en “Pagar con PayPal” nos abrirá una ventana para identificarnos con el correo y contraseña que hemos creado



6. Y si todo sale correcto nos redirigirá a la pagina de inicio de sesión y ya podremos ingresar a las páginas según la tarifa que hayamos pagado

/inicio_sesion.html

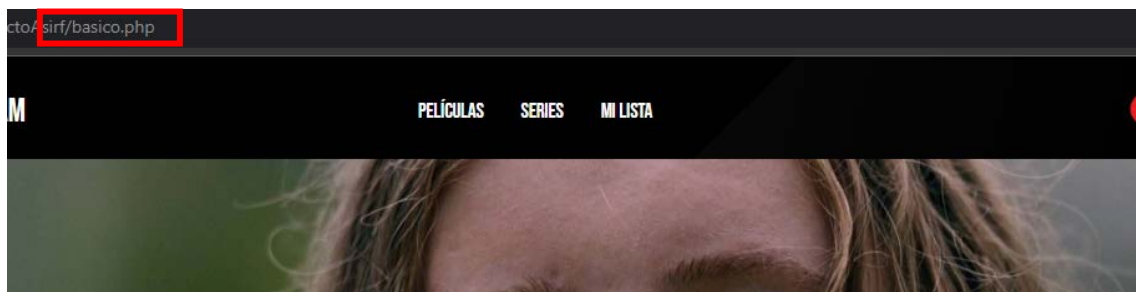
Iniciar sesión

Email:

Contraseña:

No tienes una cuenta? [Registrarse](#)

[Iniciar sesión](#)



DASHBOARD

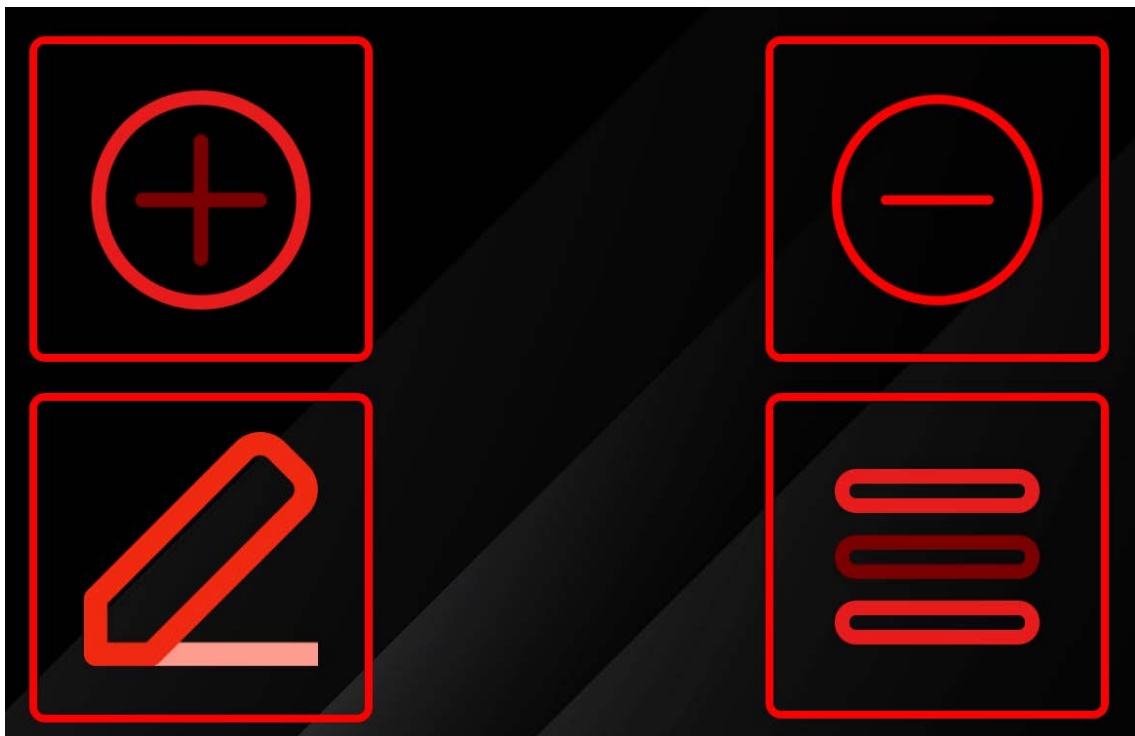
Y por último hemos creado una página llamada **dashboard.php**, que sólo pueden acceder los administradores de la web, en la cual pueden gestionar tanto las películas y series de la web como los usuarios registrados.

La página empieza mostrando 2 pestañas, una de contenido que son los vídeos y otra de usuarios.

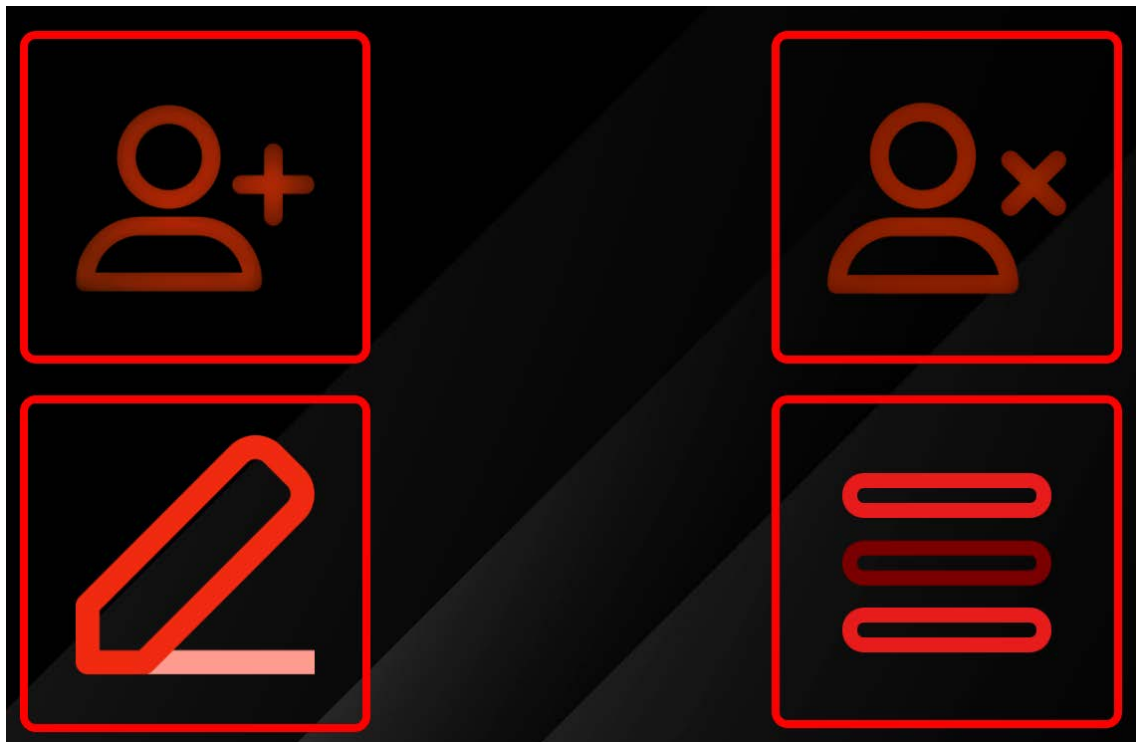
```
<header class="header">
  <div class="menu container">
    <a href="dashboard.php" class="logo">PowerStream</a>
    <nav class="navbar">
      <ul>
        <li><a class="mostrarBotones" data-target="botones" href="#">Contenido</a></li>
        <li><a class="mostrarBotones" data-target="botones2" href="#">Usuarios</a></li>
      </ul>
    </nav>
    
  </div>
</header>
```

Si entramos en cualquiera de las 2 nos aparecerán 4 botones.

Pestaña de Contenido



Pestaña de Usuarios



En ambas pestañas de administración de contenido y usuarios encontramos formularios de subida, eliminación, modificar y listar, este último para llevar un inventario de ambos conjunto de datos.

Estos formularios de se han creado para que la administración de la página sea lo mas sencilla e intuitiva posible para que cualquier persona pueda añadir, eliminar, modificar la base de datos la cual estamos conectados sin tener conocimiento de **SQL** simplemente con rellenar campos o pulsar botones de ejecución.

Mediante los archivos **PHP** heredamos los datos de los contenidos o usuarios que queremos modificar por su **id**, excepto en las paginas de subida los cuales añadiremos desde cero tanto contenido como usuarios.

Como hemos mencionado antes pulsando encima de cada botón se desplegará un formulario, mostramos capturas de ejemplo.

- **Añadir:**

POWERSTREAM

CONTENIDOUSUARIOS

12

SUBIR VIDEO

TÍTULO:

DESCRIPCIÓN:

TIPO:

Película

GÉNERO:

Acción

ARCHIVO DE VIDEO:

Seleccionar archivo

Ningún archivo seleccionado

IMAGEN DE PORTADA:

Seleccionar archivo

Ningún archivo seleccionado

IMAGEN DE PUBLICIDAD:

Seleccionar archivo

Ningún archivo seleccionado

NOMBRE DE PELÍCULA:

Seleccionar archivo

Ningún archivo seleccionado

Subir video

POWERSTREAM

CONTENIDOUSUARIOS

15

AÑADIR USUARIO

EMAIL:

PASSWORD:

PLAN:

Basic

PAGADO:

Si

Añadir Usuario

- **Eliminar:**

POWERSTREAM

CONTENIDOUSUARIOS

ELIMINAR VIDEO

FILTRAR POR:

VALOR PARA FILTRAR:

Buscar

ID VIDEO	TÍTULO	DESCRIPCIÓN	TIPO	GÉNERO	FECHA DE SUBIDA	VIDEOS	IMG PORTADA	IMG PUBLICIDAD	NOMBRE PUBLICIDAD	ACCIONES
2	MAD MAX	"MAD MAX: FURIA EN EL CAMINO" DIRIGIDA POR GEORGE MILLER, ES UNA PELÍCULA POST-APOCALÍPTICA QUE SIGUE A MAX ROCKATANSKY (YON RADY). TRAS LA DISTRUCCIÓN DEL MUNDO, MAX ES CAPTURADO POR LOS NEAN ROYS, SEGUIDORES DEL TIRANO INMORTAL JOE MAX SE ENCUENTRA EN MEDIO DE UNA REBELIÓN LIDERADA POR IMPERATOR FURIOSA (CHARLIZE THERON), QUEN NUNCA DE INMORTAL JOE CON SUS "ESPOSAS". UN GRUPO DE MUJERES QUE EL MANTIENE COMO PRISIONEROS. JUNTOS, MAX Y FURIOSA SE EMBARCAN EN UNA EPICÁNTICA CARRETERA A TRAVÉS DEL DESIERTO, ENFRENTÁNDOSE A PELIGROSOS ENEMIGOS Y OBSTÁCULOS EN SU CAMINO HACIA LA LIBERTAD. A LO LARGO DEL VIAJE, MAX Y FURIOSA FORMAN UNA ALIANZA INESTABLE PERO NECESARIA PARA SOBREVIVIR Y DERROTAR A INMORTAL JOE. LA PELÍCULA DESTACA POR SUS INTENSAS ESCENAS DE ACCIÓN, EFECTOS VISUALES IMPACTANTES Y UNA NARRATIVA QUE EXPLORA TEMAS DE RESISTENCIA Y RESILIENCIA EN UN MUNDO BRUTAL Y DESOLADO. "MAD MAX: FURIA EN EL CAMINO" ES ACLAMADA POR SU DIRECCIÓN, ACTUACIONES Y LA CREACIÓN DE UN UNIVERSO VISUALMENTE IMPRESIONANTE.	PELICULA	ACCION	2024-08-08	VIDEOS.MP4	MADMAY.JPG	MADMAY_PUBL.JPG	MADMAY_NOM.PNG	
3	JOHN WICK	"JOHN WICK", DIRIGIDA POR CHAD STAHLEKIS, SIGUE A JOHN WICK (KEANE REEVES), UN LEVENDARIO EXASSIMO A SUELDO QUE HA DEJADO SU VIDA DE CRIMEN PARA VIVIR EN PAZ. TRAS LA MUERTE DE SU ESPUSA, HELEN, EL TRAMA SE DESARROLA CUANDO UN GRUPO DE MATONES, LIDERADOS POR IOSEF TARASOV (ALFIE ALLEN), EL HIJO DE UN PODEROSO MAGNATE, VIAGO TARASOV (MICHAEL RYNDYGT), IRROMPE EN SU CASA, ROBA SU AUTO Y MATA A SU CACHORRO PAINT. EL ÚLTIMO HERALDO DE HELEN, SEGURO Y FURIOSO, BIEN ARMADO, RETORNA, DESCENTRANDO SU PASADO VIOLENTO. RETOMA SU PAPEL COMO EL TENIDO ASASINO CONOCIDO	PELICULA	ACCION	2024-08-08	VIDEOS.MP4	JHDWICK.JPG	JHDWICK_PUBL.JPG	JHDWICK_NOM.PNG	

POWERSTREAM

CONTENIDO USUARIOS





16

GESTIONAR USUARIOS

FILTRAR POR: ID Usuario

VALOR PARA FILTRAR:

Buscar

ID USUARIO	EMAIL	PLAN	FECHA DE SUSCRIPCIÓN	FECHA DE RENOVACIÓN	PAGADO	ACCIONES
1	ANDRES@EMAIL.COM	BASICO	2024-06-06	2024-07-06	SI	
2	PREMIUM@PREMIUM.COM	PREMIUM	2024-06-06	2024-07-06	SI	
3	ADMIN@ADMIN.COM	ADMIN	2024-06-06	2024-07-06	SI	
4	BASICO@BASICO.COM	BASICO	2024-06-06	2024-07-06	SI	

■ **Modificar:**

POWERSTREAM

CONTENIDO USUARIOS



16

MODIFICAR VIDEO

FILTRAR POR: ID Video

VALOR PARA FILTRAR:

Buscar

ID VIDEO	TÍTULO	DESCRIPCIÓN	TIPO	GÉNERO	FECHA DE SUBIDA	VIDEO	IMG PORTADA	IMG PUBLICIDAD	NOMBRE PUBLICIDAD	ACCIONES
2	MAD MAX	"MAD MAX: FURIA EN EL CAMINO" DIRIGIDA POR GEORGE MILLER, ES UNA PELÍCULA POST-APOLÍPTICA QUE SIGUE A MAX ROCKATSKY (TON HARVEY), TRAS LA DESTRUCCIÓN DEL MUNDO, MAX ES CAPTURADO POR LOS WAR BOTS, SEGUROS DEL TIRANO INMORTAL JOE MAX SE ENCUENTRA EN MEDIO DE UNA REBELIÓN LIDERADA POR IMPERATOR FURIOSA (CHARLIZE THERON), QUIEN HUYE DE INMORTAL JOE CON SUS "ESPOSA", UN GRUPO DE MUJERES QUE EL MANTIENE COMO PRISIONERAS. JUNTO, MAX Y FURIOSA SE ENFANCAN EN UNA FRENÉTICA CAMBIA A TRAVÉS DEL DESERTEO, ENFANTÁNDOSE A PELIGROSOS ENEMIGOS Y OBSTÁCULOS EN SU CAMINO HACIA LA LIBERTAD. A LO LARGO DEL VIAJE, MAX Y FURIOSA FORMAN UNA ALIANZA INESTABLE PERO NECESARIA PARA SOBREVIVIR Y DEBATIR A INMORTAL JOE. LA PELÍCULA DESTACA POR SUS INTENSAS ESCENAS DE ACCIÓN, EFECTOS VISUALES IMPACTANTES Y UNA NARRATIVA QUE EXPLORA TEMAS DE RESISTENCIA Y RESISTENCIA EN UN MUNDO BRUTAL Y DESOLADO. "MAD MAX: FURIA EN EL CAMINO" ES ACLAMADA POR SU DIRECCIÓN, ACTUACIONES Y LA CREACIÓN DE UN UNIVERSO VISUALMENTE IMPRESIONANTE.	PELICULA	ACCION	2024-06-06	VIDEO.MP4	MADMAX.JPG	MADMAX_PUBLI.JPG	MADMAX_NEW.PNG	
3	JOHN WICK	"JOHN WICK", DIRIGIDA POR CHAD STALLERKE, SIGUE A JOHN WICK (KIANU REEVES), UN LEGENDARIO EXASINO A SUELO QUE HA DEJADO SU VIDA DE CRIMEN PARA VIVIR EN PAZ. TRAS LA MUERTE DE SU ESPOSA, HELEN, LA TRAMA SE DESARROLLA CUANDO UN GRUPO DE MATONES, LIDERADOS POR JOSEF TARGOV (ALFIE ALLEN), EL HIJO DE UN PODEROSO MAFIOSO, VIGILADO POR MICHAEL NYSTADT, INVADEN EN SU CASA, ROBAN SU AUTO Y MATAN A SU CACHORRO MUSTY, EL ÚLTIMO REGALO DE HELEN. DEBASTADO Y FURIOSO, WICK DECIDE VENGARSE.	PELICULA	ACCION	2024-06-06	VIDEO.MP4	JOHNWICK.JPG	JOHNWICK_PUBLI.JPG	JOHNWICK_NEW.PNG	

POWERSTREAM

CONTENIDO USUARIOS





16

MODIFICAR USUARIO

FILTRAR POR: ID Usuario

VALOR PARA FILTRAR:

Buscar

ID USUARIO	EMAIL	PLAN	FECHA DE SUSCRIPCIÓN	FECHA DE RENOVACIÓN	PAGADO	ACCIONES
1	ANDRES@EMAIL.COM	BASICO	2024-06-06	2024-07-06	SI	
2	PREMIUM@PREMIUM.COM	PREMIUM	2024-06-06	2024-07-06	SI	
3	ADMIN@ADMIN.COM	ADMIN	2024-06-06	2024-07-06	SI	
4	BASICO@BASICO.COM	BASICO	2024-06-06	2024-07-06	SI	

■ **Inventario:**

POWERSTREAM

CONTENIDO

USUARIOS

15

INVENTARIO

FILTRAR POR:

ID Video

VALOR PARA FILTRAR:

También 'id' para mostrar los

Filtrar

ORDENAR POR:

ID Video ascendente

Ordenar

ID VIDEO	TÍTULO	DESCRIPCIÓN	TIPO	GÉNERO	FECHA DE SUBIDA	VIDEO	IMO PORTADA	IMO PUBLICIDAD	NOMBRE PUBLICIDAD
2	MAD MAX	"MAD MAX: FURIA EN EL CAMINO" DIRIGIDA POR GEORGE MILLER, ES UNA PELÍCULA POST-APOCALÍPTICA QUE SIGUE A MAX ROCKATANSKY (TOM HARDY). TRAS LA DESTRUCCIÓN DEL MUNDO, MAX ES CAPTURADO POR LOS MAX BOYS, SEGUROS DEL TIRANO INMORTAL JOE MAX SE ENCUENTRA EN MEDIO DE UNA REBELIÓN LIDERADA POR IMPERATOR FURIOSA (CHARLIZE THERON), QUIEN HUYE DE INMORTAL JOE CON SUS "ESPOSAS", UN GRUPO DE MUJERES QUE EL MANTIENE COMO PROYECTORAS. JUNTOS, MAX Y FURIOSA SE ENBAZCAN EN UNA FRENETICA CARRERA A TRAVÉS DEL DESERTO, ENFRENTÁNDOSE A PELIGROSOS ENEMIGOS Y OBSTÁCULOS EN SU CAMINO HACIA LA LIBERTAD. A LO LARGO DEL VIAJE, MAX Y FURIOSA FORMAN UNA ALIANZA INESTABLE PERO NECESARIA PARA SOBREVIVIR Y DERROTAR A INMORTAL JOE. LA PELÍCULA DESTACA POR SUS INTENSAS ESCENAS DE ACCIÓN, EFECTOS VISUALES IMPACTANTES Y UNA NARRATIVA QUE EXPLORA TEMAS DE REBELIÓN Y RESISTENCIA EN UN MUNDO HOSTIL Y DESOLADO. "MAD MAX: FURIA EN EL CAMINO" ES AGALANADA POR SU DIRECCIÓN, ACTUACIONES Y LA CREACIÓN DE UN UNIVERSO VISUALMENTE IMPRESIONANTE.	PELICULA	ACCION	2024-06-08	VIDEOS.MP4	MADMAX.JPG	MADMAX_PUBLI.JPG	MADMAX_NOM.PNG
		"IRON TROOP" DIRIGIDA POR JOHN DAHLQUIST, SIGUE A JOHN WICK (KEVIN COSTNER) UN							

POWERSTREAM

CONTENIDO

USUARIOS

INVENTARIO DE USUARIOS

FILTRAR POR:

ID Usuario

VALOR PARA FILTRAR:

Escribe 'id' para mostrar los

Filtrar

ORDENAR POR:

ID Usuario ascendente

Ordenar

ID USUARIO	EMAIL	PLAN	FECHA DE SUSCRIPCIÓN	FECHA DE RENOVACIÓN	PAGADO
1	ANDRES@GMAIL.COM	BASICO	2024-06-08	2024-07-08	SI
2	PREMIUM@PREMIUM.COM	PREMIUM	2024-06-08	2024-07-08	SI
3	ADMIN@ADMIN.COM	ADMIN	2024-06-08	2024-07-08	SI
4	BASICO@BASICO.COM	BASICO	2024-06-08	2024-07-08	SI

BIBLIOGRAFÍA

<https://developer.mozilla.org/es/docs/Web/HTML>

<https://developer.mozilla.org/es/docs/Web/CSS>

<https://developer.mozilla.org/es/docs/Web/JavaScript>

<https://www.php.net/manual/es/intro-what-is.php>

<https://aws.amazon.com/es/what-is/sql/>

<https://developer.paypal.com/dashboard/>

<https://aula.salesianosatocha.es/course/view.php?id=14>

<https://aula.salesianosatocha.es/course/view.php?id=13>