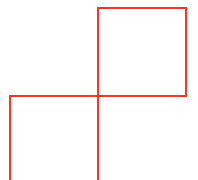


# Documentación

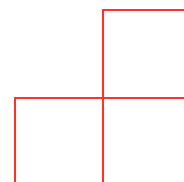
## FilmsNcritics

By: Felipe Giménez, Josías González, Johan Ferreira y Alfonso Chicoma



## Índice

1. Presentación “FilmsNcritics” .....	3
• Que ofrecemos .....	3
• Motivos del proyecto .....	6
• Quienes somos .....	6
2. Front-end .....	7
• ESTRUCTURACIÓN .....	7
• COMÚN .....	11
• INDEX .....	11
• PELÍCULAS .....	12
• COMÚN DIRECTORES, PRODUCTORES Y ACTORES .....	17
• DIRECTORES .....	17
• PRODUCTORES .....	17
• ACTORES .....	17
• PREMIOS .....	18
• LOGIN .....	23
• REGISTRO .....	24
• EJERCICIO DE MEMORIA .....	28
• CONTACTO .....	32
3. Back-end .....	33



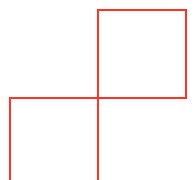
## 1. Presentación “FilmsNcritics”.

FilmsNcritics es una página web dedicada a aquellos fanáticos y amantes del cine, donde esos amantes de las historias puedan compartir sus gustos, encontrar personas con las que compartir sus opiniones, organizar sus películas y encontrar nuevas películas y géneros que les gusten

- *Que ofrecemos*

Nuestro principal objetivo reside en permitir a los usuarios almacenar sus listas de películas, bien organizadas en función de sus gustos y de sus intereses.

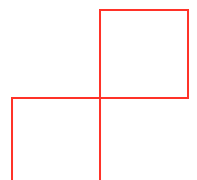
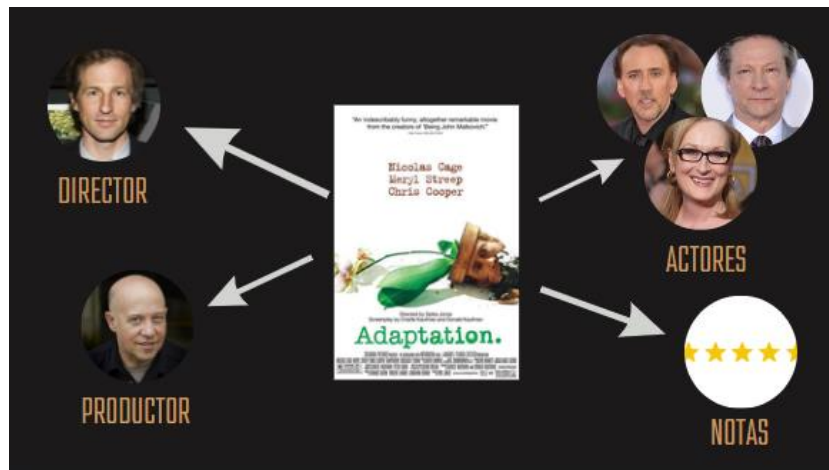
A su vez, estos usuarios pueden guiarse en función de las opiniones que otros usuarios postean en las películas, permitiendo una mayor interactividad y sentimiento de pertenencia a una comunidad con gustos similares.



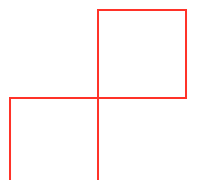
Además de calificar las películas, los usuarios pueden compartir comentarios y críticas con la comunidad, lo que enriquece la experiencia y añade un gran valor a la comunidad. Films N Critics es un espacio donde expresar tus sentimientos y pensamientos sobre el cine, brindando un altavoz a aquellas personas con algo que decir. Además, las críticas más valoradas obtendrán una mayor visibilidad, lo que les permitirá llegar a un público más amplio.



La plataforma cuenta con una base de datos que vincula todas las entidades relacionadas con las películas, lo que permite un sistema de filtrado por una amplia variedad de categorías. Esto facilita a los usuarios la búsqueda de películas según sus actores preferidos, directores, temáticas, y más, ayudándoles a encontrar exactamente lo que desean.



Utilizando las calificaciones de los usuarios, se crearán rankings con diversas temáticas para que estos puedan descubrir las películas mejor valoradas por la comunidad en diferentes géneros o temas. Estos rankings podrían incluir categorías como películas recientes, cine español, obras por género o destacando actores específicos. De esta manera, ofrecemos a los usuarios una manera organizada y fácil de encontrar recomendaciones relevantes dentro de sus intereses cinematográficos.



- *Motivos del proyecto*

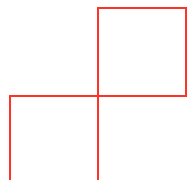
La razón primordial por la que decidimos realizar este proyecto fue porque nos vimos en la necesidad de elaborar una forma de comunicarnos y disfrutar de nuestro hobby, el cine, de una forma distinta a la habitual.

El segundo motivo es la necesidad que teníamos de poder organizar de una forma clara, fácil y rápida nuestras colecciones de películas. Que pudiéramos hacernos una lista personalizable con géneros y películas que planeáramos ver.

Por último, el principal motivo fue crear una comunidad de cinéfilos dispuestos a compartir sus opiniones, valoraciones e ideas sobre sus películas favoritas, ayudando así al resto de comunidad con sus opiniones y recomendaciones

- *Quienes somos*

- Alfonso: Tiene un enfoque más técnico y disfruta desglosando efectos especiales en películas para entender cómo se implementan digitalmente. A menudo busca mejorar sus habilidades de programación para poder desarrollar sus propias herramientas de efectos visuales.
- Johan: Es el creativo del grupo, siempre pensando en nuevas formas de combinar la programación y el cine para crear experiencias interactivas únicas. Sueña con desarrollar videojuegos que fusionen elementos cinematográficos con mecánicas de juego innovadoras.
- Felipe: Es un cinéfilo apasionado que ve cada película como una obra de arte. Le gusta explorar cómo la tecnología influye en la producción cinematográfica y está interesado en desarrollar aplicaciones relacionadas con la experiencia del espectador, como aplicaciones de recomendación de películas o plataformas de análisis de críticas.
- Josías: Es el tipo de persona que encuentra inspiración en las narrativas cinematográficas para aplicar conceptos innovadores en la programación. Le gusta analizar la estructura de las películas y cómo se pueden traducir esos elementos a la lógica de la codificación.



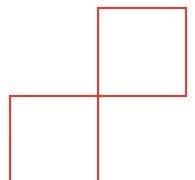
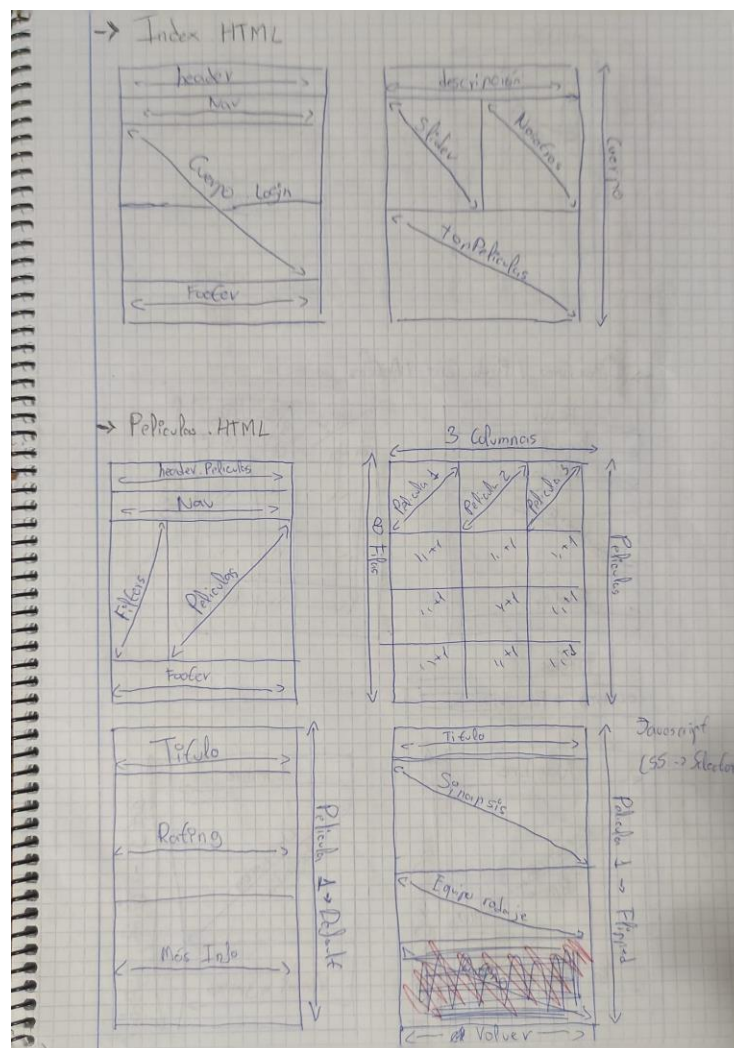
## 2. Front-end.

Nuestra página web está compuesta por diferentes subpáginas o secciones, dedicadas cada una a relacionar las distintas partes del mundo del cine, como películas, sus actores, los directores o que premios han ganado cada una de ellas.

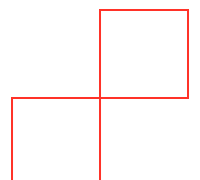
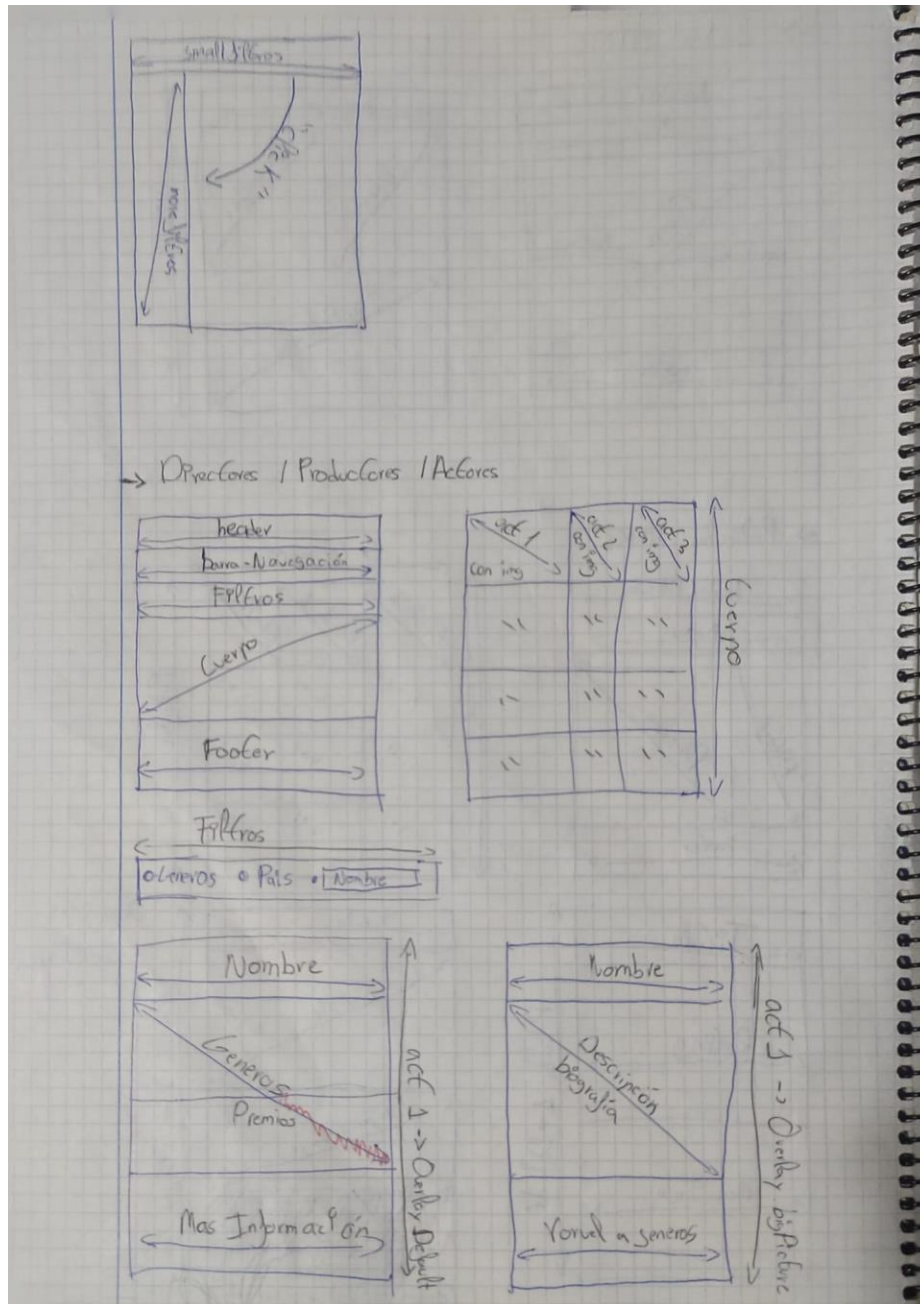
- ESTRUCTURACIÓN

Antes de empezar con el proyecto, realizamos diversas reuniones para llevar a cabo la estructuración de las diferentes páginas que contendrá nuestro sitio web.

Comenzamos con "index.html", seguido de "películas.html"

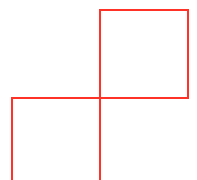
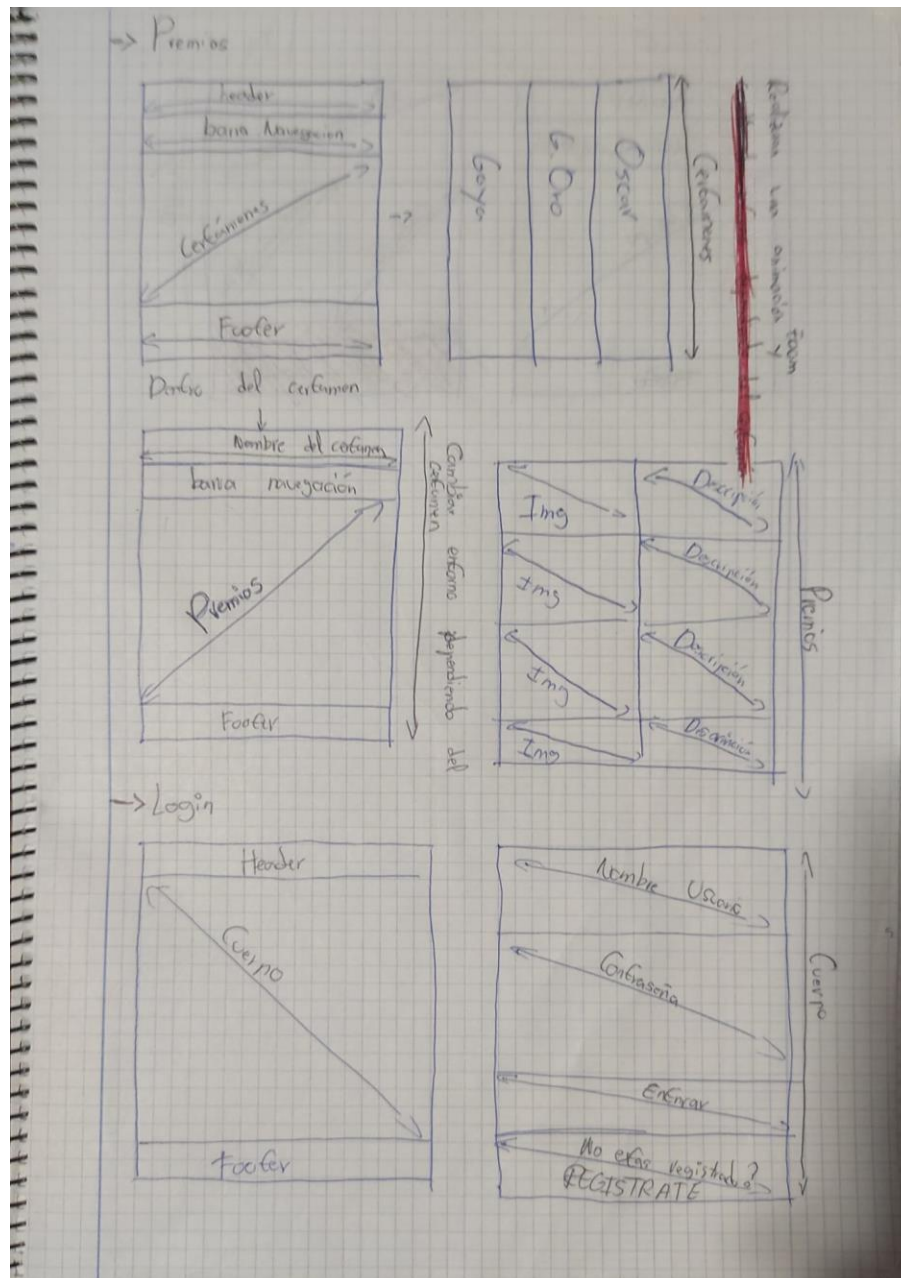


Una vez terminado "películas.html", seguimos con directores, productores y actores, que estos tendrán la misma estructura.

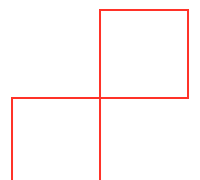
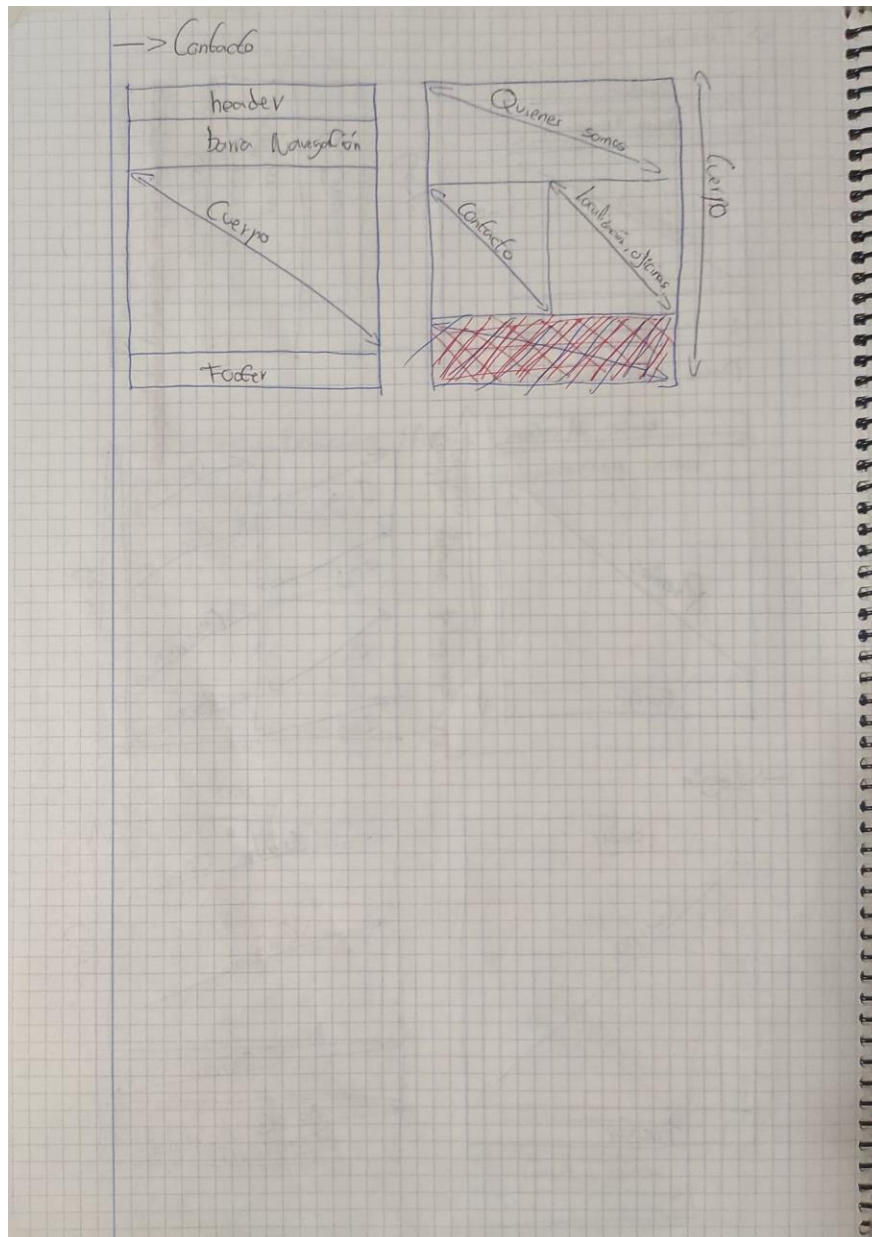




Seguiremos con la página de **"premios"**, esta finalmente sufrió diferentes cambios, puesto que se nos quedaba muy sencillas y decidimos aplicar un minijuego por cada certamen, de esta manera queda más interactivo con el usuario, finalmente la estructura de **"login"** y **registro"**, a ambas tendrán la misma estructura.



Finalmente, la estructura de "contacto", que a medida que la realizábamos, cambiamos la estructura puesto que no queda acorde con nuestras ideas.



- **COMÚN**

Todas las secciones exceptuando la de **registro, login y premios** tienen una sección en común, correspondiente a la parte del header, teniendo la parte principal con el logo y una barra de navegación, con sus respectivos estilos. Estos se mantienen iguales a todas las páginas.

En las páginas de registro, login y premios haremos un giro de tuerca, en ellas nos encontraremos con que la barra de navegación cambia a un **menú “hamburger”**.

- **INDEX**

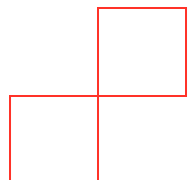
El index.html es la **página inicial** que verás al entrar en nuestra página. Consta de varios elementos, **el slider** de css, la **gráfica** con datos sobre usuarios.

**El slider** funciona mediante una animación llamada “cambio”, que hace que el elemento se desplace horizontalmente fuera del área visible durante el primer 50%, y luego permanezca fuera del área visible para el resto de la animación.

A la derecha vemos una **gráfica** creada con JavaScript usando el api de Google charts, que permite crear gráficos de diferentes tipos. En este caso tenemos un gráfico de barras y un gráfico circular que representan información de usuarios sobre su actividad en la web. Estos datos son leídos de una constante llamada usuarios en el propio JavaScript.

Después las imágenes de las películas tienen un tamaño y posición específicos para que se ajusten bien en la sección. Además, algunas imágenes se desplazan un poco hacia abajo y hacia la derecha para crear un efecto visual interesante. También se añade una sombra sutil a las imágenes para darles más profundidad. En resumen, estos estilos hacen que la sección de películas se vea atractiva y bien organizada en la página web.

Por último, el **pie de página** donde se encuentran enlaces a redes sociales, nuestro contacto y estilos para los derechos de “la empresa”.



- PELÍCULAS

La **sección de películas** consta de varias partes, los filtros y la sección películas donde se muestran las películas.

La **sección de los filtros** primero consta de unos filtros básicos, por nombre de película, y posteriormente haciendo click en “mostrar más filtros” puedes buscar por género de películas y por el nombre de los actores.

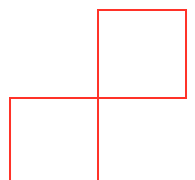
Cuando hacemos click en “mostrar más filtros” cambiará también el formato del cuerpo (incluye todo lo que se muestra al usuario) cambiando la estructura del grid.

En la búsqueda por géneros hemos introducido una animación de **desglosar y englosar** para que cuando queramos filtrar por genero podemos filtrar y en caso contrario no utilizar la herramienta.

```
@keyframes desglose{
  0%{
    transform: translateX(0);
    transform: translateY(-1500px);
    opacity: 0;
  }
  25%{
    transform: translateY(-600px) ;
    opacity: 0.1;
  }
  50%{
    transform: translateY(-400px);
    opacity: 0.3;
  }
  75%{
    transform: translateY(-100px);
    opacity: 0.6;
  }
  100%{
    transform: translateY(0px);
    opacity: 0.8;
  }
}
```

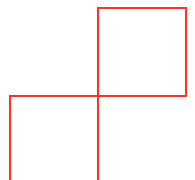
```
@keyframes englosar{
  0%{
    transform: translateX(0);
    transform: translateY(0px);
    opacity: 0.8;
  }
  25%{
    transform: translateY(-300px) ;
    opacity: 0.6;
  }
  50%{
    transform: translateY(-600px);
    opacity: 0.3;
  }
  75%{
    transform: translateY(-800px);
    opacity: 0.1;
  }
  100%{
    transform: translateY(-1500px);
    opacity: 0;
  }
}
```

```
.listaGenerosDesglose{
  animation: 3s desglose ease-in forwards ;
}
.listaGenerosEnglosar{
  animation: 3s englosar ease-out forwards ;
}
```



Con la siguiente función cambiaremos los estilos del cuerpo, para que cuando queramos más filtros, se nos aplicara un grid diferente para dejar espacio a los nuevos filtros.

```
/*-----*/  
| | /*FUNCION QUE NOS PERMITE CAMBIAR EL ESTILO DEL GRID, PARA AÑADIR MAS OPCIONES DE FILTRADO */  
  
mostrarFiltros.forEach(function (showFilter) {  
  botonFiltrar.addEventListener('click', function () {  
    if(showFilter.style.gridTemplateColumns = "4fr"){  
      showFilter.style.gridTemplateColumns = "1fr 4fr";  
      showFilter.style.alignItems = "start";  
      basicFilter.classList.add("filtrosBasicosRemove");  
      advFilter.classList.remove("filtrosAvanzadosRemove");  
  
      botonFiltrar.style.display="none";  
      botonQuitarFiltros.style.display="block";  
  
      botonQuitarFiltros.addEventListener('click', function () {  
        showFilter.style.gridTemplateColumns = "4fr";  
        botonFiltrar.style.display="block";  
        botonQuitarFiltros.style.display="none";  
        basicFilter.classList.remove("filtrosBasicosRemove");  
        advFilter.classList.add("filtrosAvanzadosRemove");  
      });  
    }  
    else if(showFilter.style.gridTemplateColumns = "1fr 4fr"){  
      showFilter.style.gridTemplateColumns = "4fr";  
    }  
  });  
});  
/*-----*/
```



El código js de la funcionalidad de cada filtro, antes de nada, añadiremos una clase específica a los diferentes hijos que contiene cada película.

```
/*AÑADIR LA CLASE tituloPelicula, generoPelicula, actorPelicula A LOS HIJOS DE overlayFlip, overlayDefault PARA SU POSTERIOR FILTRADO EN pelis.html */

let checkOverlayDefault = document.querySelectorAll(".overlayDefault");
let checkOverlayFlip = document.querySelectorAll(".overlayFlip");

document.addEventListener('DOMContentLoaded', function () {
  checkOverlayDefault.forEach(function (checkOverlay) {
    checkOverlay.children[0].classList.add("tituloPelicula");
  });
  checkOverlayFlip.forEach(function (checkOverlayFlip) {
    checkOverlayFlip.children[0].classList.add("tituloPelicula");
  });
});

document.addEventListener('DOMContentLoaded', function () {
  checkOverlayFlip.forEach(function (checkOverlayGenero) {
    if (checkOverlayGenero.children.length >= 11) {
      checkOverlayGenero.children[10].classList.add("generoPelicula");
    }
  });
});

document.addEventListener('DOMContentLoaded', function () {
  checkOverlayFlip.forEach(function (checkOverlayGenero) {
    if (checkOverlayGenero.children.length >= 11) {
      checkOverlayGenero.children[8].classList.add("actorPelicula");
    }
  });
});
```

Ahora realizaremos una función para el filtrado por género, este iguala el valor de un input en el html y lo iguala al valor que tenga género en el contenedor de la película, en caso de que no sean iguales, no se mostrarán.

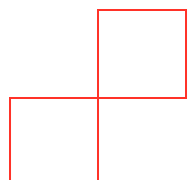
```
/*----- */

/* FILTRADO POR GENERO DE pelis.html */

let inputFiltroGenero = document.querySelectorAll(".genero");

inputFiltroGenero.forEach(function (inputGen) {
  inputGen.addEventListener('click', function () {
    let inputGenValue = inputGen.innerHTML;
    let flipPelículas = document.querySelectorAll(".contenedorPelículas");
    inputGen
    flipPelículas.forEach(function (filtradoGen) {
      let peliculaGenero = filtradoGen.querySelector(".generoPelicula");
      if (peliculaGenero) {
        peliculaGenero = peliculaGenero.innerHTML;
        let belongsToGenre = peliculaGenero.includes(inputGenValue);
        if (belongsToGenre) {
          filtradoGen.style.display = "block";
        } else {
          filtradoGen.style.display = "none";
        }
      }
    });
  });
});

/*----- */
```

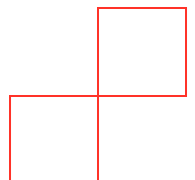


En caso de los filtros por nombres, ya sean por el título o por el actor, la sintaxis es la misma, solo varía la clase que tomamos del HTML, esta función funciona en "KEYDOWN", lo que hace es similar al anterior filtrado, iguala las dos variables (input y el contenedor de cada película), en caso de ser iguales muestra la película y de no serlo no la muestra.

```
/*----- */
| | | | /* FILTRADO POR NOMBRE DE PELICULA DE pelis.html */
let contenidoTituloPelicula = document.querySelectorAll(".tituloPelicula");
let inputFiltroPelicula = document.querySelectorAll(".inputNombrePelicula");

inputFiltroPelicula.forEach(function (inputF) {
  inputF.addEventListener('keydown', function () {
    let inputValue = inputF.value.toLowerCase();
    if (inputValue === ""){
      contenidoTituloPelicula.forEach(function (filtrado) {
        filtrado.style.display = "block";
      });
    }
    let flipPeliculas = document.querySelectorAll(".contenedorPeliculas");
    flipPeliculas.forEach(function (filtrado) {
      let peliculaTitulo = filtrado.querySelector(".tituloPelicula").innerHTML.toLowerCase();

      if (peliculaTitulo.includes(inputValue)) {
        filtrado.style.display = "block";
      } else {
        filtrado.style.display = "none";
      }
    });
  });
});
/*----- */
```



Por ultimo tenemos la sección de películas. Esta organizada por un grid consistente en tres columnas y las rows en auto. Estas películas están seleccionadas con un `querySelector`, ya la función es la misma, las películas cuentan con 2 estados:

- *Frontal:*

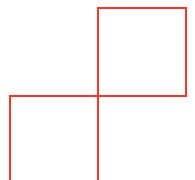
Encontraremos el título de la película y la media de las valoraciones de los diferentes usuarios.

- *Detrás:*

Encontraremos todo lo relacionado con el desarrollo de la película (Directores, actores, productores, etc...) y una breve sinopsis.

Utilizaremos una función para alternar esos dos estados, esta función se ejecuta una vez hagas click dentro de cada película. Cuando pones el ratón encima ("hover"), la animación consiste en un blur y agregar un contenido con la etiqueta `content` de `css`.

```
flipPelículas.forEach(function (elemento) {  
  elemento.addEventListener('click', function voltear () {  
    if (elemento.classList.contains("animacionDefault")) {  
      elemento.classList.remove("animacionDefault");  
      elemento.classList.add("animacionFlip");  
    }  
    else if(elemento.classList.contains("animacionFlip")){  
      elemento.classList.remove("animacionFlip");  
      elemento.classList.add("animacionDefault");  
    }  
    else{  
      elemento.classList.add("animacionDefault");  
    }  
  });  
});
```





- *COMÚN DIRECTORES, PRODUCTORES Y ACTORES*

Estas páginas cuentan con un filtro común, común en sintaxis, cada uno se diferencia por el filtrado correspondiente al nombre de cada director, productor o actor, esto se debe a que nuestra idea está más relacionada con las películas, su clasificación y crítica, por lo que poner más énfasis en estas páginas no tendría mucho sentido.

- *DIRECTORES*

En esta página las funciones son más o menos parecidas. Mediante javascript realizas un cambio en las clases del elemento, cambiando así sus estilos y apariencia.

En este caso, las tarjetas solo tienen el efecto hover, que aplica un blur, oscureciendo la imagen y de nuevo con el elemento content añade información sobre el director elegido. Lo único que cambia es el nombre de las clases en directores y productores.

La parte de los filtros funciona de la misma manera en todas las páginas.

- *PRODUCTORES*

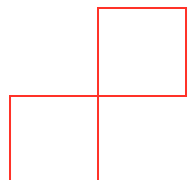
En esta página las funciones son más o menos parecidas. Mediante javascript realizas un cambio en las clases del elemento, cambiando así sus estilos y apariencia.

En este caso, las tarjetas solo tienen el efecto hover, que aplica un blur, oscureciendo la imagen y de nuevo con el elemento content añade información sobre el director elegido.

La parte de los filtros funciona de la misma manera en todas las páginas.

- *ACTORES*

En esta página volvemos a tener un grid para distribuir las imágenes de los actores. En este caso tenemos el efecto hover que permite mostrar cierta información sobre cada actor, una descripción y una sección de sus premios más destacados.



- PREMIOS

En la **sección de premios** hacemos mención a los tres premios más conocidos, con una serie de animaciones y un juego para cada uno de ellos.

Primero encontramos las tres imágenes, juntas para hacer un slider.

La primera parte de la animación consta de un `addventlistener` que permite cambiar entre dos clases, una que que cambia el fondo, cuya clase se llama `fondo`, por otra llamada `fondoactivo`, que realiza un `zoom` y a su vez le añade otro fondo a la pantalla con la clase `zoomFondo`, por ultimo inicia el audio con la función `play()`.

Después de esto añade otra clase al fondo llamada que realiza un `zoom` y hace desaparecer el fondo.

Una vez finalizado esto, has entrado al juego y es cuando empiezan las preguntas. Mediante un `switch` se comprueba que premio has seleccionado, iniciando una animación distinta en la que a su vez se muestra un texto con un narrador. Esto es posible cambiando varias clases y añadiendo otras, como la clase `pantallaCargaOn` para después de pulsar el botón de omitir saltar al juego.

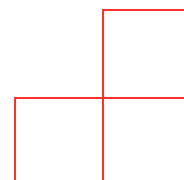
A medida que avanzas en el juego se generan las preguntas dentro de un slider. La primera pregunta tiene unas clases distintas a la segunda, cuando seleccionas respuesta en la primera, se añade una segunda pregunta y así sucesivamente.

Finalmente, aparecen los resultados en función de lo que has seleccionado, acertado o fallado.

Ahora procederemos con la explicación de las diferentes funciones:

En esta función dependiendo de qué premio elijamos realizaremos una animación de `zoom` ( como si estuvieses entrando en el premio)

```
/*-----*/  
  
/* ANIMACIONES DE ZOOM PARA premios.html Y AÑADIR UN SONIDO DE TRANSICIÓN */  
  
document.addEventListener("DOMContentLoaded", function () {  
    var elegirPremio = document.querySelector('.elegirPremio');  
    elegirPremio.addEventListener('click', function (event) {  
        var clickedElement = event.target.closest('.fondo');  
        if (clickedElement) {  
            let fondoPremios = document.querySelector('.background');  
            const aplausosPlay = document.querySelector('.aplausos');  
  
            if (clickedElement.classList.contains("fondo")) {  
                clickedElement.classList.remove("fondo");  
                clickedElement.classList.add("fondoActivado");  
                elegirPremio.classList.add("fondoActivado");  
                fondoPremios.classList.add("zoomFondo");  
                aplausosPlay.play();  
            }  
        }  
    });  
});  
  
/*-----*/
```



Añadiremos un timeout para sincronizar la animación de zoom con la aparición del premio seleccionado.

```
/*-----*/

/*AGREGAR UN TIMEOUT UNA VEZ ELEGIDA LA "ELECCION" NOS MUESTRE EN EL TIPO DE PREMIO QUE ESTAMOS */

    setTimeout(function () {
        fondoPremios.classList.add("zoomFondoDesaparecer");
        let eleccion = clickedElement.querySelector('.tituloPremio').innerHTML;
        let contenidoPremios = document.querySelector('.contenedorPremio');
        console.log(eleccion);
    });

/*-----*/
```

Realizamos un switch para agregar diferentes funciones dependiendo de la elección que hemos seleccionado, a parte de esto tendremos un comentarista que nos presentara el premio y tendrá un apartado de traducción con una función con timeout por cada carácter de la traducción .

```
switch (eleccion) {
    | | | | | /* ELECCION DE LOS PREMIOS GOYA */

    case 'Premios Goya':

        const parrafo0 = document.querySelector('.traduccion0');
        var contenedorTraduccion0 = document.querySelector('.contenedorTraduccion0');
        var nuevoTextoCompleto0 = "";

        botonOmitir=document.querySelector('.contenedorTraduccion0 .omitir');
        setTimeout(function() {
            index0 = 0;
            escribirTexto0(nuevoTextoCompleto0);
        }, 30);

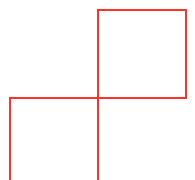
        botonOmitir.style.display="none";
        setTimeout( function () {
            botonOmitir.style.display="block";
        },1000)

        | | | | /* EL TEXTO SE ESCRIBE A MEDIDA DEL COMENTARISTA */

        function escribirTexto0() {
            nuevoTextoCompleto0="Ladies and gentlemen, welcome to the most glamorous
            and distinguished night of Spanish cinema: The Goya Awards! We are gathered
            here to pay tribute to cinematic excellence, celebrate creativity, and
            applaud the talented artists who have brought to life stories that have
            moved and captivated us. Get ready for an evening filled with excitement,
            elegance, and well-deserved recognition for the best of our seventh art!
            Let the Goya Awards ceremony begin!";

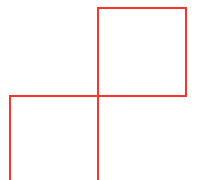
            parrafo0.innerHTML += nuevoTextoCompleto0.charAt(index0);
            index0++;

            if (index0 < nuevoTextoCompleto0.length) {
                const delayTexto = setTimeout(function() {
                    escribirTexto0(nuevoTextoCompleto0);
                }, 47);
            }
        }
    }
}
```



En caso de querer omitir el texto y comentarista quitaremos ese timeout y añadiríamos una pantalla de carga, la pantalla de carga me sirvió como solución provisional puesto que el texto de traducción necesitaba completarse y en vez de quitar el contenedor de traducción y esperar a la siguiente función, optamos por agregar una pantalla de carga.

```
    | | | | | /*BOTON DE OMITIR EL TEXTO Y COMENTARISTA */  
  
    botonOmitir.addEventListener("click",function(){  
        comentaristaGoya.remove();  
        contenedorTraduccion0.style.display="none";  
        botonOmitir.style.display="none";  
        pantallaCarga.classList.remove('pantallaCargaOf')  
        pantallaCarga.classList.add('pantallaCargaOn')  
        const omitirTexto = setTimeout(function() {  
            escribirTexto0(nuevoTextoCompleto0)  
        },1);  
    });
```



La siguiente función nos dará paso al minijuego donde quitaremos la pantalla de carga y añadiremos los diferentes elementos propios del minijuego, contadores (aciertos y fallos en las preguntas) y otros elementos como una canción de fondo, etc...

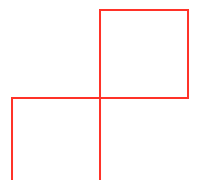
```

/*EMPIECE DEL MINIJUEGO DE LOS GOYA */

setTimeout(() => {
  pantallaCarga.classList.remove('pantallaCargaOn')
  pantallaCarga.classList.add('pantallaCargaOf')
  let musicaAmbiental = document.querySelector('.sonidoAmbiente');
  let sliderPremiosGoya = document.querySelector('.contenedorGoya .sliderPremiosOf');
  let opcionesGoya = document.querySelector('.contenedorGoya .opcionesOf');

  if (sliderPremiosGoya.classList.contains('sliderPremiosOf')) {
    sliderPremiosGoya.classList.remove('sliderPremiosOf');
    sliderPremiosGoya.classList.add('sliderPremiosOn');
    opcionesGoya.classList.remove('opcionesOf');
    opcionesGoya.classList.add('opcionesOn');
  }
  else{
    sliderPremiosGoya.classList.remove('sliderPremiosOn');
    sliderPremiosGoya.classList.add('sliderPremiosOf');
    opcionesGoya.classList.remove('opcionesOn');
    opcionesGoya.classList.add('opcionesOf');
  }
  musicaAmbiental.play();
  contenedorTraduccion0.style.display='none';
  comentaristaGoya.remove();
  var contadorBueno = 0;
  var contadorMalo = 0;
  let peliculasGoya = document.getElementsByClassName('opcionPeliculaGoya');
  let peliculaSeleccionadaGoya = '';
  let revelacionImagenGoya = document.getElementById('revelacionMejorPeliculaGoya');
  let cambioMejorPeliculaGoya = document.getElementById('tituloMejorPeliculaGoya');

```

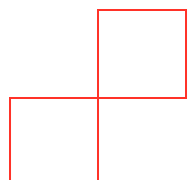


Ahora explicare cómo funciona la función de cada pregunta, recogeremos el valor de cada opción que sea posible en cada pregunta y si el valor es igual a la condición que hemos aplicado, nos parecerá que es correcta si no será incorrecta, después de esto, llamaremos a la siguiente función que nos lleva a la siguiente pregunta.

```
function seleccionMejorActorGoya() {
  let incognita = document.querySelector('.mejorActorGoya .incognita');
  let opcionesGoya = document.querySelector('.contenedorGoya .opcionesOf');
  if (opcionesGoya.classList.contains('opcionesOf')) {
    opcionesGoya.classList.remove('opcionesOf');
    opcionesGoya.classList.add('opcionesOn');
  }
  else{
    opcionesGoya.classList.remove('opcionesOn');
    opcionesGoya.classList.add('opcionesOf');
  }
  let mejorActorGoya = document.getElementById('mejorActorGoya');
  mejorActorGoya.style.display = 'grid';
  let revelacionImagenGoya = document.getElementById('revelacionMejorActorGoya');
  let actoresGoya = document.getElementsByClassName('opcionActorGoya');
  let actorSeleccionadoGoya = '';
  let cambioMejorActorGoya = document.getElementById('tituloMejorActorGoya')

  Array.from(actoresGoya).forEach(actorGoya => {
    actorGoya.addEventListener('click', function() {
      actorSeleccionadoGoya = this.innerText;

      if (actorSeleccionadoGoya === 'Denis Menochet') {
        contadorBueno++
        incognita.src = "../img/premios/correcto.png"
        sonidoCorrecto.play();
        opcionesGoya.classList.remove('opcionesOn')
        opcionesGoya.classList.add('opcionesOf')
        setTimeout(function() {
          incognita.style.opacity = '0';
          revelacionImagenGoya.style.opacity = '1';
          cambioMejorActorGoya.textContent = 'Denis Menochet';
        }, 2000)
      }
      else{
        contadorMalo++
        incognita.src = "../img/premios/incorrecto.png"
        sonidoIncorrecto.play();
        opcionesGoya.classList.remove('opcionesOn')
        opcionesGoya.classList.add('opcionesOf')
        setTimeout(function(){
          incognita.style.opacity = '0';
          revelacionImagenGoya.style.opacity = '1';
          cambioMejorActorGoya.textContent = 'Denis Menochet';
        }, 2000)
      }
      console.log(actorSeleccionadoGoya);
      setTimeout(function(){
        opcionesGoya.classList.remove('opcionesOf')
        opcionesGoya.classList.add('opcionesOn')
        revelacionImagenGoya.style.opacity = '0';
        mejorActorGoya.style.display = 'none';
        seleccionMejorActrizGoya();
      }, 4000)
    });
  });
};
```



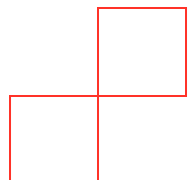
- LOGIN

(Realizado por Felipe Giménez)

La página de inicio de sesión será el primer contacto de los usuarios con nuestra web. Hemos diseñado una estética profesional y atractiva, con un aspecto limpio que deja claro de qué se trata nuestra plataforma y que invita a los usuarios a unirse a nuestra comunidad.

El formulario de inicio de sesión es simple, con solo dos campos: usuario y contraseña, para que los usuarios puedan acceder de manera rápida y sencilla. Si el usuario no tiene una cuenta, encontrará un enlace debajo del formulario que lo dirigirá directamente al formulario de registro para unirse a la comunidad.

En caso de que un usuario tenga una cuenta, pero haya olvidado su contraseña, encontrará un enlace que le ayudará a recuperarla, o algo parecido....

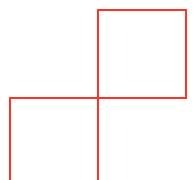
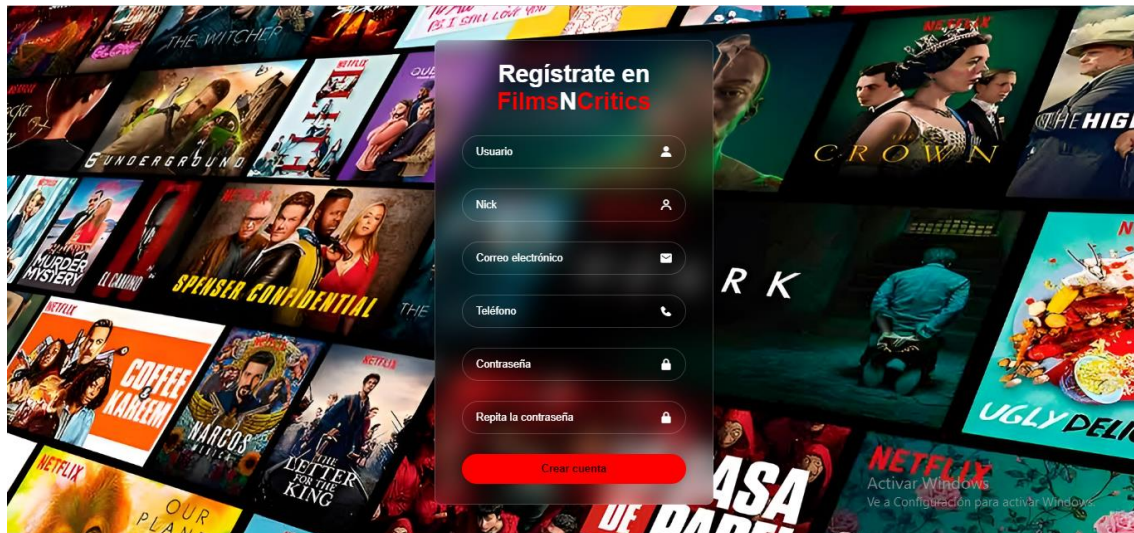




- **REGISTRO**

La página de registro mantiene la misma estética que la página de inicio de sesión, asegurando así una cohesión visual y estética para una mejor experiencia del usuario.

Para crear una cuenta, el usuario deberá elegir un nombre de usuario único, un Nick que deberá de ser diferente, proporcionar una dirección de correo electrónico, un número de teléfono y crear una contraseña sólida y segura. Estos simples pasos permiten a cualquier persona formar parte de nuestra magnífica comunidad en cuestión de segundos.





Entrando más en el tema técnico:

La validación de que los campos cumplen con los requisitos se realiza mediante expresiones regulares:

```
// Expresiones regulares para validar los campos
const usuarioExpreg = /^[a-zA-Z0-9._-]+$;/;
const nickExpreg = /^[a-zA-Z0-9._-]+$;/;
const emailExpreg = /^[^\s@]+@[^\s@]+\.[^\s@]+$;/;
const telefonoExpreg = /^\d{6,12}$/;
const passwordExpreg = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[A-Za-z\d!@#$%^&*()=?~:;]{8,20}$/;

// Obtener los valores de los campos
const usuarioField = form.querySelector('input[placeholder="Usuario"]');
const nickField = form.querySelector('input[placeholder="Nick"]');
const emailField = form.querySelector('input[placeholder="Correo electrónico"]');
const telefonoField = form.querySelector('input[placeholder="Teléfono"]');
const passwordField = form.querySelector('input[placeholder="Contraseña"]');
const confirmPasswordField = form.querySelector('input[placeholder="Repita la contraseña"]');
```

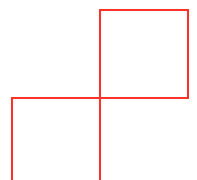
Requisitos:

- Usuario: solo letras y números.
- Nick: igual que el usuario, pero, además, permite los caracteres ".", "\_ y "-".
- Correo electrónico: formato válido (xxx@xxx.xxx). La eñe no es un carácter permitido.
- Teléfono: entre 6 y 12 dígitos.
- Contraseña: al menos una mayúscula, una minúscula y un número para garantizar una contraseña fuerte y segura.
- Repita la contraseña: debe coincidir exactamente con la contraseña y cumplir con los mismos requisitos.

Estos requisitos se explican con una alerta:

```
// Mostrar alerta si hay campos inválidos
if (!valid) {
    let errorMessage = 'Por favor, corrige los siguientes campos incorrectos:\n\n';
    if (!usuarioExpreg.test(usuario)) {
        errorMessage += '- Usuario: solo letras y números\n';
    }
    if (!nickExpreg.test(nick) || nick === usuario) {
        errorMessage += '- Nick: igual que el usuario, pero permite los caracteres ".", "_" y "-\n';
    }
    if (!emailExpreg.test(email)) {
        errorMessage += '- Correo electrónico: formato válido (xxx@xxx.xxx).\n';
    }
    if (!telefonoExpreg.test(telefono)) {
        errorMessage += '- Teléfono: entre 6 y 12 dígitos.\n';
    }
    if (!passwordExpreg.test(password)) {
        errorMessage += '- Contraseña: al menos una mayúscula, una minúscula y un número.\n';
    }
    if (password !== confirmPassword || confirmPassword === '' || !passwordExpreg.test(confirmPassword)) {
        errorMessage += '- Repita la contraseña: debe coincidir exactamente con la contraseña y cumplir con los mismos requisitos.\n';
    }
    alert(errorMessage);
} else {
    alert('¡Formulario enviado correctamente!');
    // Aquí podrás enviar el formulario si todos los campos son válidos
    // form.submit();
}
```

Activar Windows



**Esta página dice**

Por favor, corrige los siguientes campos incorrectos:

- Usuario: solo letras y números
- Correo electrónico: formato válido (xxx@xxx.xxx).
- Teléfono: entre 6 y 12 dígitos.
- Contraseña: al menos una mayúscula, una minúscula y un número.
- Repita la contraseña: debe coincidir exactamente con la contraseña y cumplir con los mismos requisitos.

**Aceptar**

En caso de que un campo no cumpla con los requisitos, el borde del campo se pondrá en rojo para que el usuario sepa que campos no son correctos.

**Regístrate en  
FilmsNCritics**

¿???

a

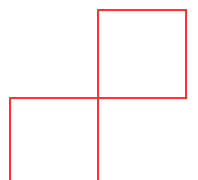
felipe@gmail.com

adq2dq

\*\*\*\*\*

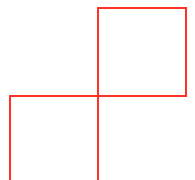
\*\*\*\*\*

**Crear cuenta**



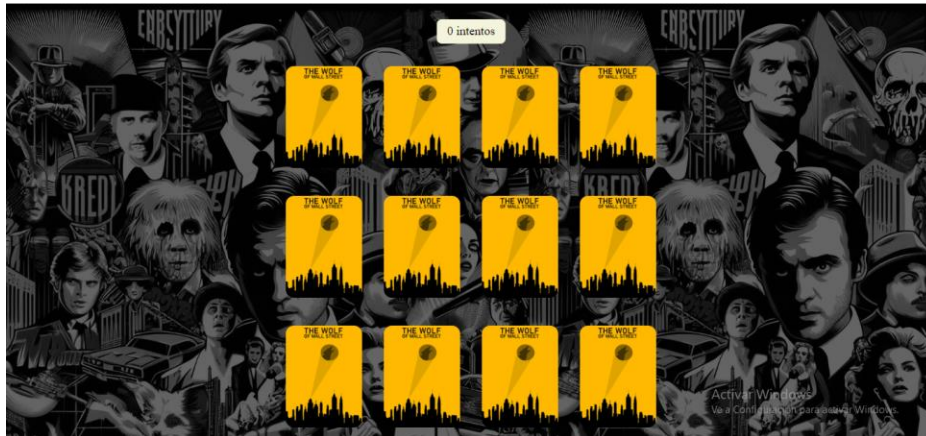
Esto se hace cambiando el estilo de la página mediante java script de la siguiente forma:

```
// Validar los campos y aplicar estilos si es necesario
let valid = true;
if (!usuarioExpreg.test(usuario)) {
    valid = false;
    usuarioField.style.borderColor = 'red';
} else {
    usuarioField.style.borderColor = '';
}
if (!nickExpreg.test(nick) || nick === usuario) {
    valid = false;
    nickField.style.borderColor = 'red';
} else {
    nickField.style.borderColor = '';
}
if (!emailExpreg.test(email)) {
    valid = false;
    emailField.style.borderColor = 'red';
} else {
    emailField.style.borderColor = '';
}
if (!telefonoExpreg.test(telefono)) {
    valid = false;
    telefonoField.style.borderColor = 'red';
} else {
    telefonoField.style.borderColor = '';
}
```

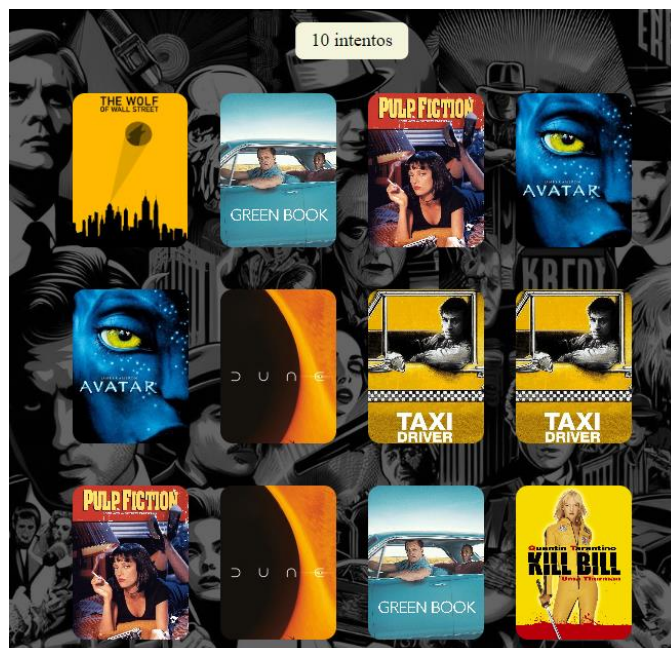


- EJERCICIO DE MEMORIA

Volviendo a la página del Login, si un usuario ha olvidado su contraseña, podrá acceder mediante un enlace a un pequeño juego de memoria a modo de broma por haber olvidado su contraseña.

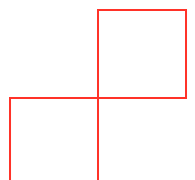


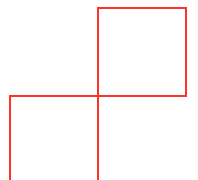
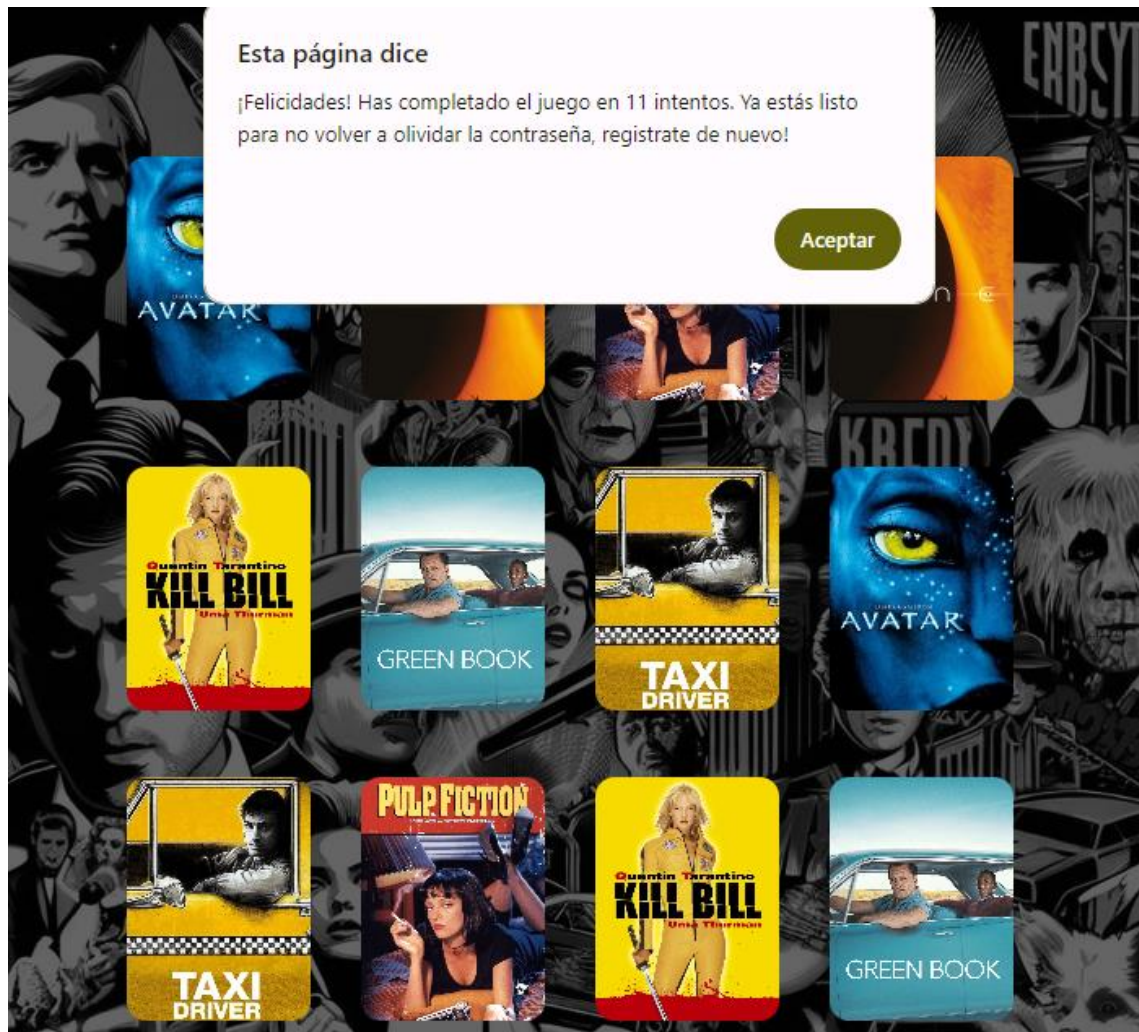
Este mini juego consiste en encontrar las parejas de las tarjetas que contengan la misma portada de película. Los usuarios podrán tener una divertida partida y mejorar su memoria experimentando un juego con una estética basada totalm ente en el contenido de esta web



con un toque retro y cinéfilo, el usuario irá dando la vuelta a las cartas hasta completar todas las parejas y revelar todas las portadas de cine.

Una vez encontradas todas las parejas, el usuario podrá ver el número de intentos que ha necesitado para lograrlo y será redireccionado a la página del formulario de registro para que cree una nueva cuenta y esta vez no vuelva a olvidar la contraseña.







Entrando más en el tema técnico:

Variables declaradas que generarán las cartas, las imágenes, y realizarán las funciones de contar los intentos y cartas seleccionadas.

```
const totalCartas = 12;
const cartasDisponibles = ['pelicula1.jpg', 'pelicula2.jpg', 'pelicula3.jpg', 'pelicula4.jpg', 'pelicula5.jpg', 'pelicula6.jpg'];
let cartas = [];
let cartasSeleccionadas = [];
let movimientoActual = 0;
let intentosActuales = 0;

let plantillaCarta = '<div class="carta"><div class="back"></div><div class="face"></div></div>';
```

Con la siguiente función se barajan las cartas aleatoriamente y se crean las parejas.

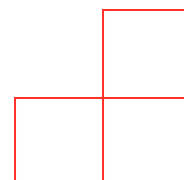
```
function barajarCartas() {
    let cartasAleatorias = cartasDisponibles.concat(cartasDisponibles); // Duplica las cartas para formar parejas
    cartasAleatorias.sort(() => Math.random() - 0.5); // Mezcla las cartas aleatoriamente
    return cartasAleatorias;
}
```

La función activar(e) se encarga de gestionar la interacción del usuario con el juego. Cuando el usuario hace clic en una carta, verifica si se pueden seleccionar más cartas. Si se cumplen ciertas condiciones, como que el número de movimientos actuales sea menor que 2 y la carta no esté ya seleccionada, la función activa la carta seleccionada y la añade al array de cartas seleccionadas mostrado en el primer apartado de este apartado. Luego, si se han seleccionado dos cartas, se aumenta el contador de intentos y se comprueba si las dos cartas seleccionadas son pareja. Si lo son, se reinician las cartas seleccionadas y el contador de movimientos actuales, y se llama a una función para verificar si el juego ha terminado después de un breve retraso (esto es para que dé tiempo a que la última carta se dé la vuelta, si no, salta la alerta antes de que se dé la vuelta y nunca se llega a revelar). Si las cartas no coinciden, se espera un breve período de tiempo antes de desactivar las cartas seleccionadas y reiniciar el contador de movimientos actuales.

```
function activar(e) {
    if (movimientoActual < 2) {
        if (!cartasSeleccionadas[0] || cartasSeleccionadas[0] !== e.target) {
            e.target.classList.add('active');
            cartasSeleccionadas.push(e.target);
        }

        if (++movimientoActual == 2) {
            intentosActuales++;
            document.querySelector('#stats').innerHTML = intentosActuales + ' intentos';

            if (cartasSeleccionadas[0].querySelectorAll('.face')[0].style.backgroundColor ===
                cartasSeleccionadas[1].querySelectorAll('.face')[0].style.backgroundColor) {
                cartasSeleccionadas = [];
                movimientoActual = 0;
                setTimeout(comprobarFinJuego, 1000); // Retraso para mostrar la última tarjeta antes de comprobar el fin del juego
            } else {
                setTimeout(() => {
                    cartasSeleccionadas[0].classList.remove('active');
                    cartasSeleccionadas[1].classList.remove('active');
                    cartasSeleccionadas = [];
                    movimientoActual = 0;
                }, 600);
            }
        }
    }
}
```

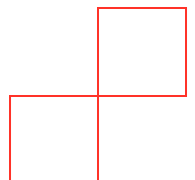


La función “cpmprobarFinJuego” sirve para verificar si el juego ha terminado, es decir, si el usuario ha encontrado todas las parejas. En ese caso, muestra una alerta felicitando al jugador y mostrando los intentos que ha necesitado”

```
function comprobarFinJuego() {  
  if (document.querySelectorAll('.carta:not(.active)').length === 0) {  
    alert('¡Felicidades! Has completado el juego en ' + intentosActuales + ' intentos. Ya estás listo para no volver a olvidar la contraseña');  
    window.location.href = 'registro.html'; // Redireccionar a la página de login  
  }  
}
```

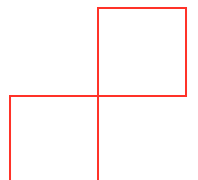
Finalmente, esta función es la encargada de crear las cartas mediante un bucle for y ajustar las imágenes a la tarjeta.

```
for (let i = 0; i < totalCartas; i++) {  
  let div = document.createElement('div');  
  div.innerHTML = plantillaCarta;  
  cartas.push(div);  
  document.querySelector('#juego').append(cartas[i]);  
  cartas[i].querySelectorAll('.face')[0].style.backgroundImage = `url(${cartasDisponibles[i]})`;  
  cartas[i].querySelectorAll('.carta')[0].addEventListener('click', activar);  
}
```



- *CONTACTO*

En esta página explicamos brevemente quienes son los participantes del grupo y cual es nuestra motivación para realizar esta página web





### *3.Back-end.*

# En desarrollo

