

Esquema de funciones del **scriptHome.js**

Nombre de la función	Descripción	Variables locales	Descripción de variable local
goToHome()	Al hacer "click" nos redirige a la sección "home" del header.		
goToMovies()	Al hacer "click" nos redirige a la sección "películas" del header.		
goToAltHome()	Al hacer "click" en el botón correspondiente nos redirige a un home alternativo / indexHomeAlt.html		
IniciatePage()	Marca la inicialización de la ventana modal / recomendaciones de películas con clase date.		
showMovieRecommendations(array)	Renderiza dinámicamente un grupo de tarjetas de películas en HTML, utilizando los datos recibidos en un array de objetos.	let content = ""	Declaramos una variable vacía que se va a llenar con los datos recibidos del array de objetos.
showModal()	Muestra la ventana modal / recomendaciones de películas con clase date.	const hour = new Date().getHours();	Contiene la hora actual, nos sirve para determinar que películas recomendar a lo largo del día.
closeModal()	Cierra la ventana modal.		
showSliderImg()	Cambia la imagen mostrada según el valor del index.		
btnBack()	Hace funcionar el botón izquierdo y nos permite navegar hacia atrás en el slider / carrusel.		
btnForward()	Hace funcionar el botón derecho y nos permite navegar hacia adelante en el slider / carrusel.		
updateImg()	Actualiza el índice de la imagen del slider para mostrar la siguiente imagen.		
startSlider()	Establece un intervalo, la imagen se cambiará de		

	manera automática después de 2 segundos, siempre y cuando el mouse no este sobre la imagen.		
stopSlider()	Limpia el intervalo cuando el mouse se encuentra sobre la imagen, haciendo que el cambio automático de imagen se detenga.		
showSignUp()	Nos muestra el formulario en la ventana modal, cambia el estilo del display a "flex" permitiendo que lo veamos.		
closeSignUp()	Cambia el estilo del display a "None", haciendo que cuando cerremos el formulario ya no lo veamos.		
validateText(input, p, type)	Nos permite validar si el usuario ingreso texto de forma adecuada en el input, y marcarle error en el caso de que no sea así. Utilizamos parámetros para poder condensar esta tarea en una sola función.		
validateNumber(input, p, type)	Nos permite validar si el usuario ingreso número de forma adecuada en el input, y marcarle error en el caso de que no sea así. Utilizamos parámetros para poder condensar esta tarea en una sola función.	let portion = value.split(" ")	Divide el string por espacios, para obtener los números separados.
input1Name()	Si el texto ingresado por el usuario cumple con los requerimientos de la función "validateText", devuelve "true".		
input2Surname()	Si el texto ingresado por el usuario cumple con los requerimientos de la función "validateText", devuelve "true".		
input3Age()	Si el "texto" ingresado por el usuario cumple con los requerimientos de la función "validateNumber", devuelve "true".		

validateEmail(input, p)	Verifica si un campo de correo electrónico es válido: comprueba que no esté vacío y que contenga un "@". Si hay errores, muestra un mensaje en el elemento <p>; si es válido, lo limpia y devuelve "true".		
input5Number()	Si el "texto" ingresado por el usuario cumple con los requerimientos de la función "validateNumber", devuelve "true".		
validatePassword(input, p)	Verifica si la contraseña ingresada por el usuario es correcta: comprueba que el campo no esté vacío y que tenga por lo menos 6 caracteres. Si hay errores, muestra un mensaje en el elemento <p>; si es válido, lo limpia y devuelve "true".		
validateForm(event)	Valida los campos de un formulario antes de permitir su envío. Previene el comportamiento por defecto (submit), verifica cada campo con funciones específicas, y si todos son válidos, cierra el formulario y muestra un mensaje de éxito. Si hay errores, alerta al usuario.	Let valido = true;	Verifica si todos los campos pasaron la validación.

Esquema de funciones del **scriptHomeAlt.js**

Nombre de la función	Descripción	Variables locales	Descripción de la variable local
goToHome()	Al hacer "click" nos dirige a la sección "home" del header.		
goToMovies()	Al hacer "click" nos dirige a la sección "películas" del header.		
returnHome()	Al hacer "click" en el botón correspondiente nos dirige al home		

	principal indexHome.html.		
makeDraggable(img)	Permite que una imagen (img) pueda ser arrastrada con el mouse. Maneja eventos del mouse para actualizar la posición de la imagen en pantalla mientras se arrastra.	1- let offsetX 2- let offsetY 3- let isDragging	1- Guarda la distancia horizontal entre el borde izquierdo de la imagen y el punto donde se hizo clic. 2- Igual que offsetX, pero para la distancia vertical (desde el borde superior de la imagen). 3- Indica si el usuario está actualmente arrastrando la imagen.
onMouseDown(e)	Inicia el arrastre, calcula la posición del click dentro de la imagen, la trae al frente visualmente y cambia el cursor.		
onMouseMove(e)	Se ejecuta cuando el mouse se mueve. Solo actúa si el usuario está arrastrando. Calcula la nueva posición de la imagen para que siga al cursor sin saltos, y actualiza su posición en pantalla.	1- const X = e.clientX - offsetX 2- const y = e.clientY - offsetY	1- Nueva posición horizontal de la imagen. 2- Nueva posición vertical de la imagen.
onMouseUp()	Se ejecuta cuando se suelta el botón del mouse. Finaliza el arrastre de la imagen y actualiza el estilo del cursor para reflejar que ya no se está moviendo.		

Esquema de funciones del **scriptCatalogo.js**

Nombre de la función	Descripción	Variables locales	Descripción de la variable
goToHome()	Al hacer "click" nos redirige a la sección "home" del header.		

goToMovies()	Al hacer "click" nos redirige a la sección "películas" del header.		
showMovies(array)	Muestra un conjunto de películas en la página web. Toma un array de objetos de películas y genera dinámicamente el HTML para mostrar cada una como una tarjeta con su imagen, nombre y año.	let content	Acumula el HTML generado para cada película. Se inicializa como string vacío y se va completando dentro del bucle.
generateGenres()	Extrae todos los géneros únicos del array moviesSC, los ordena alfabéticamente y los inserta como opciones en un menú desplegable para que el usuario pueda filtrar películas por género.	1- let uniqueGenres 2- let actualGenre	1- Array donde se guardan los géneros sin repetir, evitando duplicados. 2- Género de la película actual en el bucle.
quitarTildes(text)	Elimina los tildes de un texto recibido como parámetro, reemplazando las vocales con tilde por sus equivalentes sin tilde, tanto en mayúsculas como minúsculas.		
filters()	<p>Filtra películas según texto escrito por el usuario o género seleccionado.</p> <p>Luego, muestra solo las películas que cumplen ambos requerimientos, o un mensaje si no hay coincidencias.</p>	1- searchByName 2- selectedGenre 3- FilteredMovies 4- movieName	1- El texto ingresado por el usuario en el input de búsqueda, sin tildes, espacios ni mayúsculas. 2- El género actualmente seleccionado en el select. 3- Array donde se guardan las películas que pasan ambos filtros. 4- Primera letra del nombre de la película, procesada

			como texto normalizado.
--	--	--	-------------------------

Esquema de funciones del **scriptAmpliacion.js**

Nombre de la función	Descripción	Variables locales	Descripción de la variable
goToHome()	Al hacer "click" nos redirige a la sección "home" del header.		
goToMovies()	Al hacer "click" nos redirige a la sección "películas" del header.		
relatedClick(e)	<p>Maneja el click en una tarjeta de película relacionada.</p> <p>Detecta si el click fue dentro de una tarjeta con clase .related-card.</p> <p>En ese caso, obtiene su data-id y redirige al usuario a la página de ampliación pasando ese id por URL.</p>	<p>1- const clickedCard</p> <p>2- const selectedId</p>	<p>1- El elemento .related-card más cercano al click.</p> <p>2- El valor del atributo data-id del elemento clickeado. Se usa para armar el link de redirección.</p>
showRelated()	Muestra en el contenedor .co-related hasta 3 películas relacionadas a la película.	<p>1- const coRelatedContainer</p> <p>2- let ActualMovie</p> <p>3- const RelatedMovies</p> <p>4- const selected</p> <p>5- const randomCard</p> <p>6- const card</p> <p>7- const img</p> <p>8- const title</p> <p>9- const peli</p>	<p>1- Elemento DOM donde se mostrarán las películas relacionadas.</p> <p>2- Objeto de la película actual buscada por id.</p> <p>3- Array con películas del mismo género que la actual, sin incluirla.</p> <p>4- Array con hasta 3 películas elegidas</p>

			<p>aleatoriamente para mostrar.</p> <p>5- Índice aleatorio para seleccionar una película relacionada.</p> <p>6- Elemento div creado para cada tarjeta relacionada.</p> <p>7- Elemento img con la portada de la película.</p> <p>8- Elemento h4 con el nombre de la película.</p> <p>9- Objeto de película en el array selected para construir la tarjeta.</p>
--	--	--	---