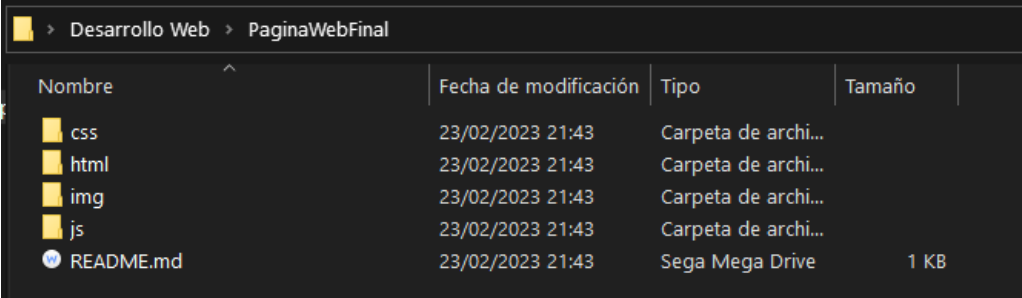


TAREA. Proyecto Multidisciplinario - Javascript.

La idea principal de mi proyecto era hacer un foro sencillo, sin embargo he tomado esta idea y, al pensar que hoy en día los foros están en decadencia y no son muy famosos, la he desarrollado para modernizarla y crear una especie de red o web social donde puedes hacer tus propios “Posts” o publicaciones en formato de pequeñas cartas sencillas e interactivas que podrás publicar de forma pública o privada.

Primero explicaré el orden de carpetas y su orden ya que respecto al trabajo del trimestre pasado ha mejorado y se ha vuelto más sencillo de ver y entender:



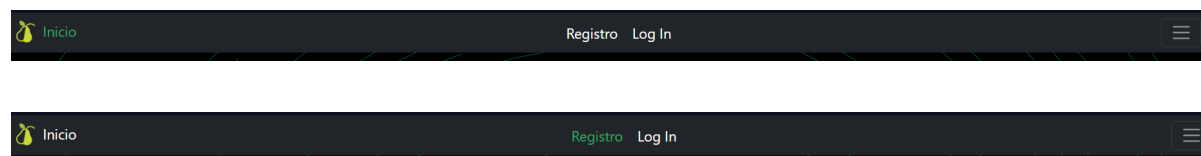
Nombre	Fecha de modificación	Tipo	Tamaño
css	23/02/2023 21:43	Carpeta de archi...	
html	23/02/2023 21:43	Carpeta de archi...	
img	23/02/2023 21:43	Carpeta de archi...	
js	23/02/2023 21:43	Carpeta de archi...	
README.md	23/02/2023 21:43	Sega Mega Drive	1 KB

Es simple, ahora todo está dentro de una carpeta llamada PaginaWebFinal y dentro de esta carpeta están:

La carpeta donde están todos los css, otra donde están todos los html, otra donde están todas las imágenes y por último, y lo que nos interesa a nosotros, otra donde están todos los scripts js.

A continuación explicaré los scripts uno a uno, ubicaré donde se encuentran y a qué js pertenecen.

1. Script iluminación de la barra de navegación .



Es un simple script que marca la barra de navegación según en qué zona estés. El código es el siguiente.

```
//Script para iluminar la barra de navegación según donde te encuentres dentro de la página
$(document).ready(function(){
    $('.active').animate({
        color: "#34AA62"
    }, 1500);
});
```

Está ubicado en un js general para que se inicie en todas las zonas de la página y funciona con un simple document.ready y un animate.

2. Script de desactivación del click derecho.

```
//Script general para deshabilitar el click derecho de todas las páginas
document.addEventListener('contextmenu', noClick);
function noClick(event){
    event.preventDefault();
}
```

Es un script simple ubicado también en un js general que actúa en todas las zonas de la página web y funciona con un addEventListener contextmenu noClick, se ve que poniendo esto todo se hace automáticamente y sin complicaciones.

3. Scripts de validación del formulario de registro.

The image shows two versions of a registration form side-by-side. Both forms have fields for Nombre, Apellidos, Email, Contraseña, Edad, Telefono, and Sexo. The 'Registrarme' button is green in the left form and grey in the right form. The 'Borrar' button is grey in both.

El primer script de la zona de registro es un validador de los datos que introduces en el formulario. Si todos son correctos se ilumina al botón de Registrarme, sino permanece igual. El código es el siguiente:

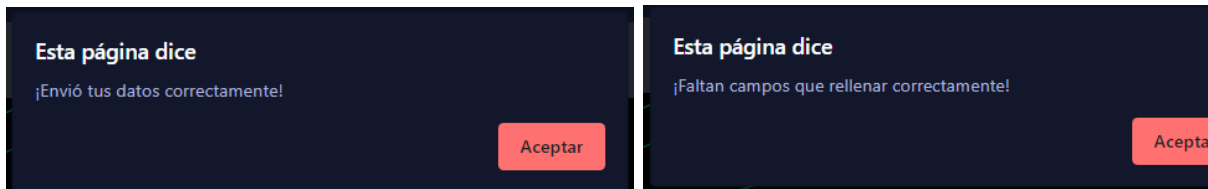
```
//Script que comprueba si se han rellenado correctamente para iluminar el boton de registro en verde
var botonVerde = document.getElementsByClassName("boxcomprobar");
for (let index = 0; index < botonVerde.length; index++) {
    botonVerde[index].addEventListener("mouseout", comprobarVerde);
}
```

```
function comprobarVerde(){
  var input1 = document.getElementById("stringNombre");
  var input2 = document.getElementById("stringApellido");
  var input3 = document.getElementById("integerTelefono");
  var input4 = document.getElementById("integerDni");
  var input5 = document.getElementById("integerEdad");

  var input6 = document.getElementById("pregunta_hombre").checked;
  var input7 = document.getElementById("pregunta_mujer").checked;
  var input8 = document.getElementById("publicidad").checked;

  if(!(input1.value=="") && !(input2.value=="") && (/^\d{9}$/.test(input3.value)) && (/^[^@ \t\r\n]+@gmail\.com$/.test(input4.value)) && (/^\d{2}$/.test(input5.value)) && (input6 || input7) && input8){
    var boton = document.getElementById("botonEnviar");
    boton.style.backgroundColor = "#34AA62";
  }else{
    var boton = document.getElementById("botonEnviar");
    boton.style.backgroundColor = "#6C757D";
  }
}
```

En la primera imagen se agarran los elementos con los que se iniciara la función, en este caso mouseout, y en la segunda imagen se ve el código en profundidad de cómo funciona la validación. Es simple, primero se agarra todo lo que se quiere comprobar con `getElementById` y después se comprueba si es correcto o no mirando si está vacío y utilizando expresiones regulares.



Este segundo script envía un mensaje por pantalla según si se han enviado los campos correctamente o en cambio faltan cosas por rellenar correctamente. El código es el siguiente:

```
//Script que comprueba si se han rellenado bien los campos para enviar un mensaje de correcto o error al enciar el formulario de registro
var formulario = document.getElementById("formulario");
formulario.addEventListener("submit", comprobar)
```

```
function comprobar(event){
  var input1 = document.getElementById("stringNombre");
  var input2 = document.getElementById("stringApellido");
  var input3 = document.getElementById("integerTelefono");
  var input4 = document.getElementById("integerDni");
  var input5 = document.getElementById("integerEdad");

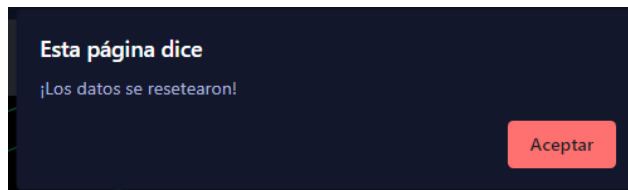
  var input6 = document.getElementById("pregunta_hombre").checked;
  var input7 = document.getElementById("pregunta_mujer").checked;
  var input8 = document.getElementById("publicidad").checked;

  if(input1.value==" " || input2.value==" " || !(/^\d{9}$/.test(input3.value)) || !(/^[^@ \t\r\n]+@gmail\.com/.test(input4.value)) || !(/^\d{2}$/.test(input5.value)) || !(input6 || input7) || !input8){
    alert("¡Faltan campos que rellenar correctamente!");
    event.preventDefault();
  }else{
    alert("¡Envió tus datos correctamente!");
    var boton = document.getElementById("botonEnviar");
    boton.style.backgroundColor = "#6C757D";
  }
}
```

En la primera imagen simplemente se declara que el script se iniciara cuando apretamos el botón de submit del formulario y el segundo script es muy parecido al exterior solo que esta vez envía un mensaje por pantalla según si has rellenado bien o mal los formularios.

Solo faltan dos scripts muy sencillos que se utilizan para resetear el formulario y otro que se utiliza para hacer focus al campo de texto Nombre

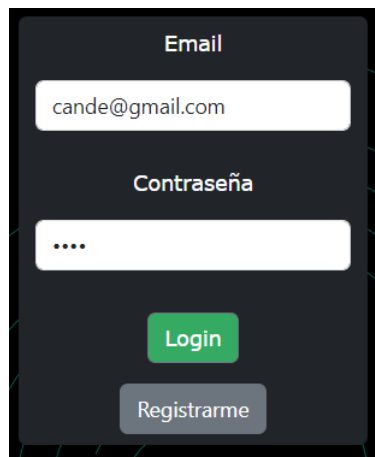
nada más entrar en la página. Su código es muy facil de entender y es el siguiente:



```
function resetear(event){
    document.getElementById("formulario").reset();
    alert("¡Los datos se resetearon!");
    //event.stopPropagation();
}
```

```
//Script para mantener el focus en el input del nombre en el registro
document.getElementById("stringNombre").focus();
```

4. Script de iluminación del botón de login.

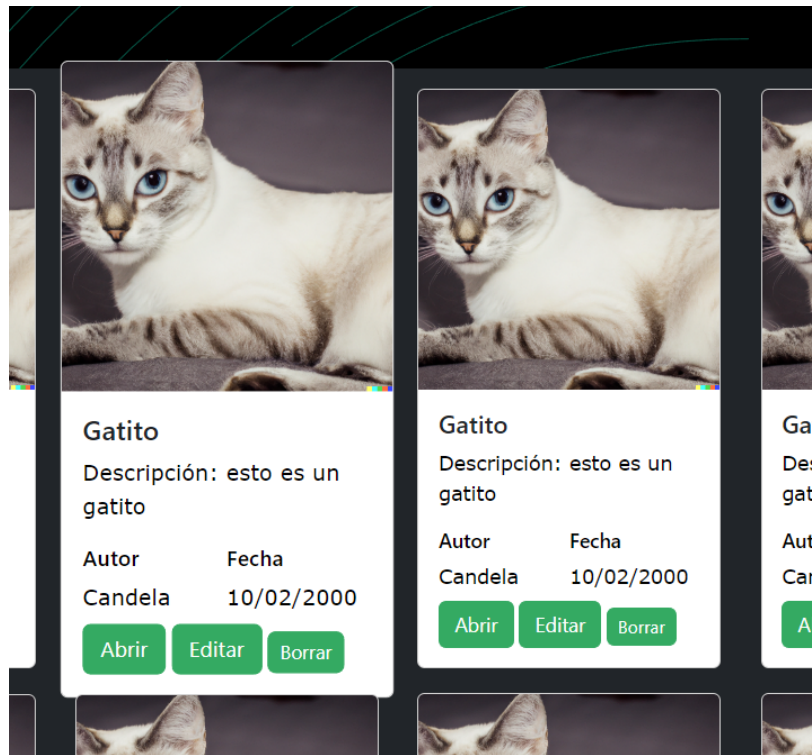


```
$(document).ready(function(){
    //Script para iluminar el boton de login cuando se hayan rellenado ambos campos
    $('#passId').mouseout(function(){
        if(($.trim($('#emailId').val()) != '') && ($.trim($('#passId').val()) != '')){
            $('.botonEnviar').animate({
                backgroundColor : "#34AA62"
            }, 1500);
        }else{
            $('.botonEnviar').animate({
                backgroundColor: "#6C757D"
            }, 1500);
        }
    });
});
```

Es un sencillo script que ilumina el botón de Login una vez los dos campos estén rellenos. Aquí he empezado a utilizar código nuevo respecto al proyecto del primer trimestre utilizando jQuery.

El código es simple, se agarrará el evento mouseout y se aplicará en la zona de abajo de Contraseña. Una vez tu ratón salga de ese elemento de HTML se comprobará si ambos campos están rellenos. Si está relleno se aplicará un .animate de cambio de color y si no esta relleno se aplicará tambien pero del mismo color de base. Esto es así para que vuelva al color original si decidimos subir arriba de nuevo y borrar los datos.

5. Script de aumentar o disminuir el tamaño de las cartas (efecto zoom) si el ratón esta encima o no.



```
//Script para hacer grande y pequeñas las cartas
$(".cartita").mouseover(function(){
    $(this).css('transform', 'scale(1.1)')
});
$(".cartita").mouseout(function(){
    $(this).css('transform', 'scale(1)')
});
```

Es un script de código muy sencillo que simplemente tiene como eventos mouseover y mouseout, utilizándolos para modificar la propiedad de transform de las cartas en el css. Lo más interesante puede ser la utilización apropiada del this para que solo se aumente o disminuya la propia carta a la que apuntas y no todas a la vez. Este script lo he utilizado en el apartado de administracionCartas y la feed.

También añadí un cambio de color a rojo cuando estas sobre el botón de Borrar de la carta que funciona como los anteriores cambios de color de botón de registro y login.



```
//Scripts para iluminar en rojo el boton de Borrar cuando estas encima de el
$(".irBorrar").mouseover(function(){
    $(this).animate({
        backgroundColor: "#CC6666",
        borderColor: "#CC6666"
    }, 0);
});
$(".irBorrar").mouseout(function(){
    $(this).animate({
        backgroundColor: "#34AA62",
        borderColor: "#34AA62"
    }, 0);
});
```

Y finalmente para concluir con este apartado he añadido una funcionalidad muy sencilla que permite mover las cartas a voluntad. No puedo adjuntar imagen de demostración pero se ha mostrado en clase y el código es muy simple ya que funciona con el paquete de ui.

```
//Script para poder mover las cartas a voluntad
$( ".sortable" ).sortable();
```

6. Scripts sin uso real o que se eliminarán en la conversión a Php. (DOM Y AJAX)

```
//Boton para borrar del DOM la carta al clicar Borrar
$(".irBorrar").click(function(){
    $(this).parent().parent().parent().remove();
});

//Boton para crear una carta en DOM
$(".crearCarta").click(function(){
    $.ajax({
        url: "../js/archivoCarta.txt",
        context: document.body,
        success: function(response){
            $('#invisible').after(response);
        }
    });
});
```

Estos dos scripts no tienen funcionalidad real dentro de la página en estos momentos ya que al reiniciar la misma se reinicia y no modifican nada. Sin embargo, he creído oportuno añadirlos para dejar claro cómo funcionará el apartado de Php en forma de pequeña demo.

Los scripts son simples, uno borra el DOM según donde le digamos (borrara las cartas) y el segundo script crea una carta en DOM utilizando ajax y un archivo aparte que mostraré a continuación.

```
js > archivoCarta.txt X
1  <div class=" col-12 col-sm-6 col-lg-3 col-xl-2 cartita">
2    <div class="card mt-3 m-1">
3      
4      <div class="card-body">
5        <h5 class="card-title">Gatito</h5>
6        <p class="card-text">Descripción: esto es un gatito</p>
7        <button class="rounded btn btn-secondary irDescripcion" type="button">Abrir</button>
8        <button class="rounded btn btn-secondary irEdicion" type="button">Editar</button>
9        <button class="rounded btn btn-secondary btn-sm mt-1 irBorrar" type="button">Borrar</button>
10     </div>
11   </div>
12 </div>
```