



CICLO: [DAM]
**MÓDULO DE [LENGUAJE DE MARCAS
Y SISTEMAS DE GESTIÓN DE
INFORMACIÓN]**

[Tarea N° 03]

Alumno:
[Juan Carlos Filter Martín]
[REDACTED]

Contenido

1. Documentos que se adjuntan a este informe.....	3
2. Crear un formulario en HTML.....	3
Formulario final similar al siguiente.....	3
3. RA03_c) Se han utilizado métodos para la selección y acceso de los diferentes elementos de un documento web.....	4
Elementos del formulario identificado con un nombre de id o name para poder ser accesible desde las funciones JavaScript.....	4
4. RA03_d) Se han creado y modificado elementos de documentos web.....	6
Se realizarán validaciones al pulsar sobre cada uno de los botones VALIDAR.....	6
Pulsar sobre el botón CONTROL TOTAL + ENVIO que validará todos los campos dando un mensaje con todos los errores encontrados y cambiando el estilo de los campos erróneos.....	10
5. RA03_e) Se han eliminado elementos de documentos web.....	16
El botón LIMPIAR LOS CAMPOS limpiar los valores y volverá a los estilos iniciales todos los input del formulario.....	16
6. RA03_f) Se han realizado modificaciones sobre los estilos de un documento web.....	18
Cuando se produzca un error en alguno de los datos, el input correspondiente cambiará de estilo, a un estilo genérico que tendremos creado llamado errores.....	18

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Fichero HTML, CSS y JS

2. Crear un formulario en HTML

Formulario final similar al siguiente

Datos a enviar desde el formulario.

Introduce un email (p.ej.: pepe@foc.es):	<input type="text"/>	Validar
Introduce passw (5 caracteres, al menos 1 May, 1 Min, 1 Numero):	<input type="password"/>	Validar
Introduce Nombre de usuario (tres o más caracteres):	<input type="text"/>	Validar
Introduce Nombre y apellidos (obligatorio):	<input type="text"/>	Validar
<input type="button" value="Control total + envío"/>		<input type="button" value="Limpiar los campos"/>

- * Agregamos el titulo mediante un h1.
- * Introducimos los campos mediante la etiqueta label y seguido del texto
- * En la etiqueta input ponemos el tipo, el id que le asignamos, el nombre orientativo y indicándole que es obligatorio "required".
- * Para crear los botones lo indicamos mediante la etiqueta button + el tipo seguido del atributo onclick y el nombre para que una vez pulsemos el botón javascripts sepa a que función tiene que dirigirse.

```
92 <body>
93   <h1>Datos a enviar desde el formulario.</h1>
94
95   <form action="mailto:juancarlosfilter@gmail.com" method="post">
96
97     <label>Introduce un email (p.e.: pepe@foc.es):</label>
98     <input type="email" id="email" name="email" required>
99     <button type="button" onclick="validarEmail()">Validar</button><br></br>
100
101
102     <label>Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Numero):</label>
103     <input type="password" id="password" name="password" required>
104     <button type="button" onclick="validarPassword()">Validar</button><br></br>
105
106     <label>Introduce Nombre de usuario (tres o más caracteres):</label>
107     <input type="text" id="usuario" name="usuario" required>
108     <button type="button" onclick="validarUsuario()">Validar</button><br></br>
109
110     <label>Introduce Nombre y apellido (obligatorio):</label>
111     <input type="text" id="nombre" name="nombre" required>
112     <button type="button" onclick="validarNombre()">Validar</button><br></br>
113
114     <button type="submit" onclick="validarFormulario()">Control total + Enviar</button>
115     <button type="button" onclick="limpiar()">Limpiar</button>
116   </form>
```

Datos a enviar desde el formulario.

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):

Introduce Nombre de usuario (tres o más caracteres):

Introduce Nombre y apellido (obligatorio):

3. RA03_c) Se han utilizado métodos para la selección y acceso de los diferentes elementos de un documento web.

Elementos del formulario identificado con un nombre de id o name para poder ser accesible desde las funciones JavaScript.

- ✖ Agregar un id sirve para que a la hora de acceder a una función de javascript y mediante el método `getElementById('nombreid')` sepa a donde se tiene que dirigir y pueda evaluar dicho contenido.
- ✖ Por ejemplo con el email de la tarea, tenemos `id = 'email'`. Entonces cuando pulsemos el botón de `validarEmail` va a ir al documento donde se encuentra ese id, lo va a almacenar en la variable `email` y seguidamente va a entrar en la estructura de condición que si `email` no contiene ninguno de los caracteres de la variable `emailValido` mande un aviso de que no es correcto y sino indique que si lo es.

```
function validarEmail() {  
    var email = document.getElementById('email').value;  
    var emailValido = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
    if (!emailValido.test(email)) {  
        alert('Por favor, introduce un email válido');  
        return false;  
    }else  
        alert('Es correcto');  
    return true;  
}
```

✖ Ahora voy a indicar los distintos id que he creado en el formulario:

✖ Para el **email** tenemos el id = **email**

```
<label>Introduce un email (p.e.: pepe@foc.es):</label>  
<input type="email" id="email" name="email" required>  
<button type="button" onclick="validarEmail()">Validar</button>
```

✖ Para la **contraseña** tenemos el id = **password**

```
<label>Introduce una contraseña (5 caracteres, al menos 1  
<input type="password" id="password" name="password" required>  
<button type="button" onclick="validarPassword()">Validar</button>
```

✖ Para el **usuario** tenemos el id = **usuario**

```
<label>Introduce Nombre de usuario (tres o más caracteres)  
<input type="text" id="usuario" name="usuario" required>  
<button type="button" onclick="validarUsuario()">Validar</button>
```

✖ Para el **nombre y apellido** tenemos el id = **nombre**

```
<label>Introduce Nombre y apellido (obligatorio):</label>  
<input type="text" id="nombre" name="nombre" required>  
<button type="button" onclick="validarNombre()">Validar</button>
```

4. RA03_d) Se han creado y modificado elementos de documentos web.

Se realizarán validaciones al pulsar sobre cada uno de los botones VALIDAR

EMAIL

- × Mediante **onclick="validarEmail"** podemos crear la función

```
<label>Introduce un email (p.e.: pepe@foc.es):</label>
<input type="email" id="email" name="email" required>
<button type="button" onclick="validarEmail()">Validar</button>
```

- × En javascript indicamos mediante **function validarEmail()** que el email sea correcto controlando los posibles errores.

```
function validarEmail() {
    var email = document.getElementById('email').value;
    var emailValido = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailValido.test(email)) {
        alert('Por favor, introduce un email válido');
        return false;
    } else {
        alert('Es correcto');
        return true;
    }
}
```

Comprueba si la expresión regular encuentra coincidencias dentro de una cadena de texto.
(true si encuentra una coincidencia y false si no la encuentra)
Al tener '!' está verificando si es false

- ✖ En la web vamos a comprobar si funciona:

- ✖ Con email erróneo

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):

Esta pagina dice

Por favor, introduce un email válido

Aceptar

- ✖ Con email correcto

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):

Esta pagina dice

Es correcto

Aceptar

CONTRASEÑA

- ✖ Mediante **onclick="validarPassword"** podemos crear la función

```
<label>Introduce una contraseña (5 caracteres, al menos 1 May, 1 Mi
<input type="password" id="password" name="password" required>
<button type="button" onclick="validarPassword()">Validar</button><
```

- ✖ En javascript indicamos mediante **function validarPassword()** que el email sea correcto controlando los posibles errores.

```
function validarPassword() {
    var password = document.getElementById('password').value;
    var passwordValido = /^(?=.*[A-Z])(?=.*[a-z])(?=.*\d){5,}$/;
    if (!passwordValido.test(password)) {
        alert('La contraseña debe tener al menos 5 caract
        return false;
    }else
        alert('Es correcto');
    return true;
}
```

Comprueba si la expresión regular encuentra coincidencias dentro de una cadena de texto.
(true si encuentra una coincidencia y false si no la encuentra)
Al tener '!' está verificando si es false

- ✖ En la web vamos a comprobar si funciona:

- ✖ Con password erróneo

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):

Validar

Esta página dice

La contraseña debe tener al menos 5 caracteres, incluyendo al menos 1 mayúscula, 1 minúscula y 1 número

Aceptar

- ✖ Con password correcto

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):

Validar

Esta página dice

Es correcto

Aceptar

USUARIO

- ✖ Mediante **onclick="validarUsuario"** podemos crear la función

```
<label>Introduce Nombre de usuario (tres o más caracteres):</label>
<input type="text" id="usuario" name="usuario" required>
<button type="button" onclick="validarUsuario()">Validar</button><br>
```

- ✖ En javascript indicamos mediante **function validarUusario()** que el email sea correcto controlando los posibles errores.

```
function validarUsuario() {
    var usuario = document.getElementById('usuario').value;
    var usuarioValido = /^[3,]+$;
    if (!usuarioValido.test(usuario)) {
        alert('El nombre de usuario debe tener al menos tres
        return false;
    }else
        alert('Es correcto');
    return true;
}
```

Comprueba si la expresión regular encuentra coincidencias dentro de una cadena de texto.
(true si encuentra una coincidencia y false si no la encuentra)
Al tener '!' está verificando si es false

- ✖ En la web vamos a comprobar si funciona:

- ✖ Con usuario erróneo

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número): Validar

Introduce Nombre de usuario (tres o más caracteres): Validar

Esta página dice

El nombre de usuario debe tener al menos tres caracteres

Aceptar

- ✖ Con usuario correcto

Datos a enviar desde el formu

Introduce un email (p.e.: pepe@foc.es):

Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número): Validar

Introduce Nombre de usuario (tres o más caracteres): Validar

Esta pagina dice

Es correcto

Aceptar

NOMBRE Y APELLIDOS

- ✖ Mediante **onclick="validarNombre"** podemos crear la función

```
<label>Introduce Nombre y apellido (obligatorio):</label>
<input type="text" id="nombre" name="nombre" required>
<button type="button" onclick="validarNombre()">Validar</button>
```

- ✖ En javaScript indicamos mediante **function validarNombre()** que el email sea correcto controlando los posibles errores.

```
function validarNombre() {
    var nombre = document.getElementById('nombre').value;
    if (nombre.trim() === '') {
        alert('Por favor, introduce tu nombre y apellido');
        return false;
    } else {
        alert('Es correcto');
        return true;
    }
}
```

Elimina los espacios en blanco al principio y al final de la cadena

✖ En la web vamos a comprobar si funciona:

✖ Con nombre y apellidos erroneo

The screenshot shows a web form titled "Datos a enviar desde el formulario". It contains four input fields: "Introduce un email (p.e.: pepe@foc.es):", "Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):", "Introduce Nombre de usuario (tres o más caracteres):", and "Introduce Nombre y apellido (obligatorio):". Each field has a "Validar" button. A light blue tooltip message is displayed over the form, stating "Esta página dice: Por favor, introduce tu nombre y apellido". The tooltip has a green "Aceptar" button in its top right corner. The "Introduce Nombre y apellido" field is currently empty.

✖ Con nombre y apellidos correcto

This screenshot shows the same web form as above, but with a validation success message. The tooltip now says "Esta página dice: Es correcto". The "Introduce Nombre y apellido" field now contains the text "juan carlos filter martin". At the bottom of the form, there are two additional buttons: "Control total + Enviar" and "Limpiar". The "Validar" button for the name field is also visible.

Pulsar sobre el botón CONTROL TOTAL + ENVIO que validará todos los campos dando un mensaje con todos los errores encontrados y cambiando el estilo de los campos erróneos.

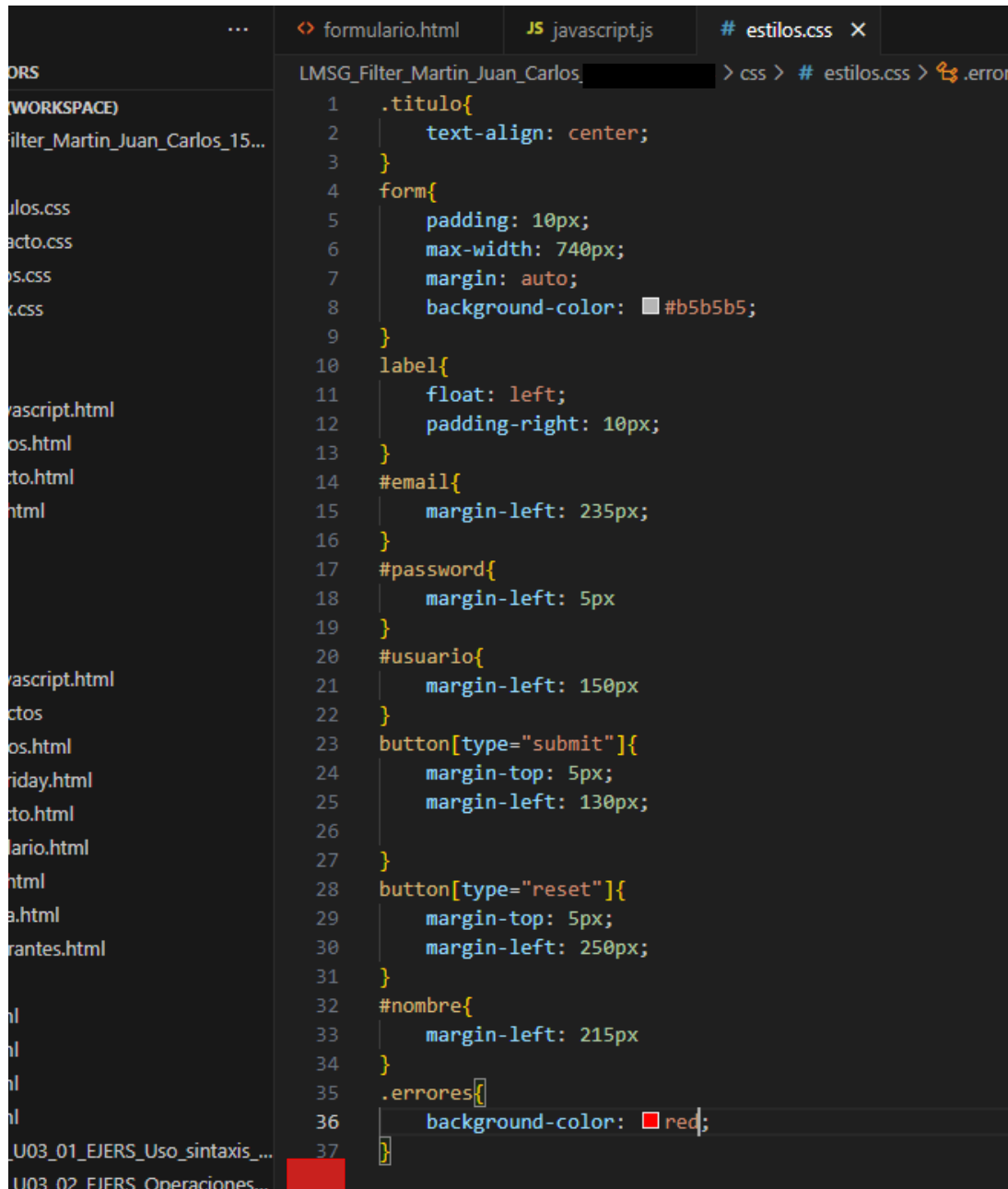
✖ Empezaré primeramente creando los estilos del formulario y separando tanto **javascript** como **css** fuera del html.

→ Para **css** dentro de link aremos referencia con **href** al enlace donde se encuentra nuestro archivo css

→ Para **javascript** dentro de script indicaremos la ruta mediante **src**

```
<head>
  <meta>
  <title>Formulario</title>
  <link type="text/css" rel="stylesheet" href="./css/estilos.css"/>
  <script type="text/javascript" src="./js/javascript.js"></script>
</head>
```

→ El CSS estará aparte del html con los estilos del formulario y el siguiente resultado del mismo:



```
1 .titulo{
2     text-align: center;
3 }
4 form{
5     padding: 10px;
6     max-width: 740px;
7     margin: auto;
8     background-color: #b5b5b5;
9 }
10 label{
11     float: left;
12     padding-right: 10px;
13 }
14 #email{
15     margin-left: 235px;
16 }
17 #password{
18     margin-left: 5px;
19 }
20 #usuario{
21     margin-left: 150px;
22 }
23 button[type="submit"]{
24     margin-top: 5px;
25     margin-left: 130px;
26 }
27 }
28 button[type="reset"]{
29     margin-top: 5px;
30     margin-left: 250px;
31 }
32 #nombre{
33     margin-left: 215px;
34 }
35 .errores{
36     background-color: red;
37 }
```

Datos a enviar desde el formulario.

Introduce un email (p.e.: pepe@foc.es):	<input type="text"/>	Validar
Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):	<input type="password"/>	Validar
Introduce Nombre de usuario (tres o más caracteres):	<input type="text"/>	Validar
Introduce Nombre y apellido (obligatorio):	<input type="text"/>	Validar
<input type="button" value="Control total + Enviar"/>		<input type="button" value="Limpiar los campos"/>

→ Y JavaScript estará aparte del html con los con los script y las condiciones que debe cumplir el formulario al interactuar con los botones

```
formulario.html JS javascript.js X # estilos.css
LMSG_Filter_Martin_Juan_Carlos_ > js > JS javascript.js > limpiar
1 function validarEmail() {
2     var email = document.getElementById('email').value;
3     var emailValido = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
4     if (!emailValido.test(email)) {
5         alert('Por favor, introduce un email válido');
6         return false;
7     }else
8         alert('Es correcto');
9     return true;
10 }
11
12 function validarPassword() {
13     var password = document.getElementById('password').value;
14     var passwordValido = /^(?=.*[A-Z])(?=.*[a-z])(?=.*\d){5,}$/;
15     if (!passwordValido.test(password)) {
16         alert('La contraseña debe tener al menos 5 caracteres, incluyendo al menos 1 mayúscula, 1 minúscula y 1 número');
17         return false;
18     }else
19         alert('Es correcto');
20     return true;
21 }
22
23 function validarUsuario() {
24     var usuario = document.getElementById('usuario').value;
25     var usuarioValido = /^[^{}]{3,}$/;
26     if (!usuarioValido.test(usuario)) {
27         alert('El nombre de usuario debe tener al menos tres caracteres');
28         return false;
29     }else
30         alert('Es correcto');
31     return true;
32 }
33
34 function validarNombre() {
35     var nombre = document.getElementById('nombre').value;
36     if (nombre.trim() === '') {
37         alert('Por favor, introduce tu nombre y apellido');
38         return false;
39     }else
40         alert('Es correcto');
41     return true;
42 }
43
44 function validarFormulario() {
45     var email = document.getElementById('email').value;
```

BOTÓN CONTROL TOTAL + ENVIO

Botón CONTROL TOTAL + ENVIO

- × Mediante **onclick="validarFormulario"** podemos crear la función

```
<button type="submit" onclick="validarFormulario()">Control total + Enviar</button>  
<button type="button" onclick="limpiar()">Limpiar</button>  
</form>
```

COMPROBANDO POSIBLES ERRORES

- × En javaScript indicamos mediante **function validarFormulario()** que el email sea correcto controlando los posibles errores.

```
function validarFormulario() {  
    var email = document.getElementById('email').value;  
    var password = document.getElementById('password').value;  
    var usuario = document.getElementById('usuario').value;  
    var nombre = document.getElementById('nombre').value;  
  
    var errores = [];  
  
    if (!email.match(/^[^\s@]+@[^\s@]+\.[^\s@]+$/)) {  
        errores.push('Por favor, introduce un email válido');  
        document.getElementById('email').classList.add('errores');  
    }  
  
    if (!password.match(/^(?=.*[A-Z])(?=.*[a-z])(?=[^\s]{5,})$/)) {  
        errores.push('La contraseña debe tener al menos 5 caracteres, incluyendo al menos una mayúscula y una minúscula');  
        document.getElementById('password').classList.add('errores');  
    }  
  
    if (!usuario.match(/^[^.\s]{3,}$/)) {  
        errores.push('El nombre de usuario debe tener al menos tres caracteres');  
        document.getElementById('usuario').classList.add('errores');  
    }  
  
    if (nombre.trim() === '') {  
        errores.push('Por favor, introduce tu nombre y apellido');  
        document.getElementById('nombre').classList.add('errores');  
    }  
  
    if (errores.length > 0) {  
        alert('Se encontraron los siguientes errores:\n\n' + errores.join('\n'));  
        return false;  
    } else {  
        alert('Todos los campos son correcto');  
        return true;  
    }  
}
```


Se utiliza en Arrays para agregar uno o más elementos al final del Array

Se usa para buscar una coincidencia entre una expresión regular y una cadena de texto

Elimina los espacios en blanco al principio y al final de la cadena

APLICANDO ESTILOS DE ERROR

- ✖ Para aplicar el estilo errores vamos al CSS y creamos una clase llamada **errores**:

```
}  
.errores{  
    background-color:  red;  
}
```

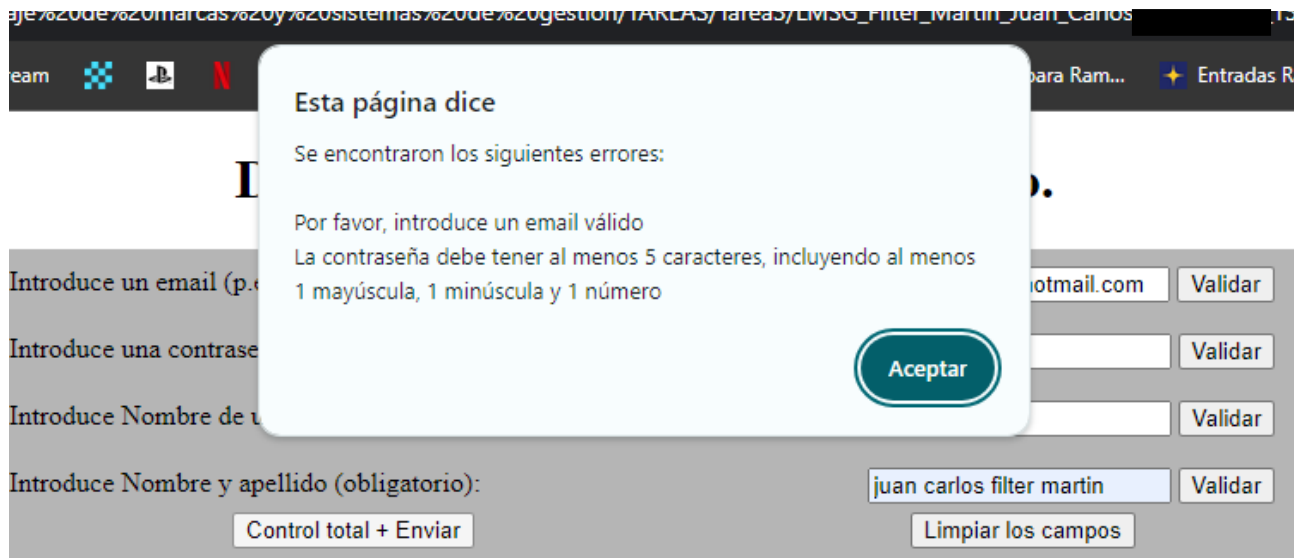
- ✖ Ahora en el JavaScript dentro de cada if donde se produce dicho error le añadimos dicho estilo de clase **errores**

```
if (!email.match(/^[\s@]+@[\s@]+\.[\s@]+$/)) {  
    errores.push('Por favor, introduce un email válido');  
    document.getElementById('email').classList.add('errores');  
}
```

- ✖ Por ejemplo en el email si no contiene lo indicado y entra en el if mostrará el mensaje y seguidamente aplicará el estilo de la clase errores

COMPROBACIÓN EN LA WEB

- ✖ En la web vamos a comprobar si funciona:
- ✖ Con Botón CONTROL TOTAL + ENVIO [datos erróneos]



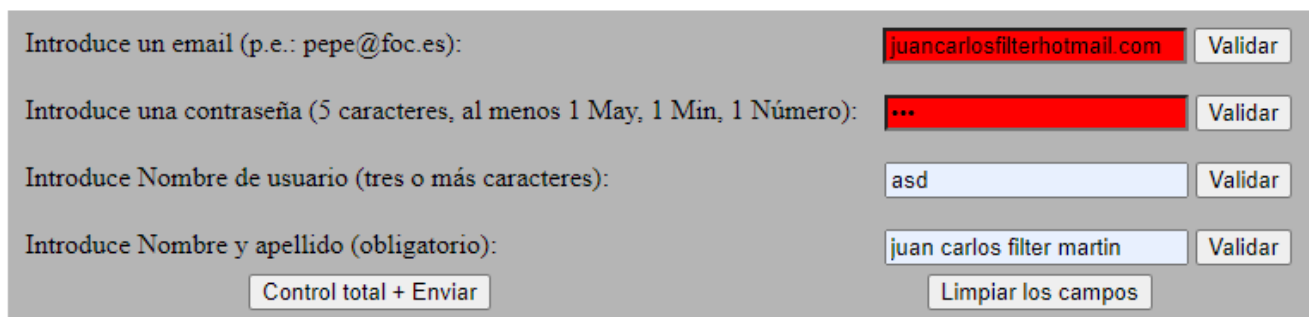
The screenshot shows a web browser window with a form. A modal dialog box is displayed in the center, titled "Esta página dice". The message inside reads: "Se encontraron los siguientes errores: Por favor, introduce un email válido. La contraseña debe tener al menos 5 caracteres, incluyendo al menos 1 mayúscula, 1 minúscula y 1 número". Below the message is a green "Aceptar" button. The form in the background has the following fields and values:

Field Label	Value	Action
Introduce un email (p.e.: pepe@foc.es):	juancarlosfilterhotmail.com	Validar
Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):	[Redacted]	Validar
Introduce Nombre de usuario (tres o más caracteres):	asd	Validar
Introduce Nombre y apellido (obligatorio):	juan carlos filter martin	Validar

At the bottom of the form are two buttons: "Control total + Enviar" and "Limpiar los campos".

- ✖ Y aplica el estilo indicado:

Datos a enviar desde el formulario.



The screenshot shows the same web form as above, but with the data to be sent highlighted in red. The data is as follows:

Field Label	Data to be sent
Introduce un email (p.e.: pepe@foc.es):	juancarlosfilterhotmail.com
Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):	...
Introduce Nombre de usuario (tres o más caracteres):	asd
Introduce Nombre y apellido (obligatorio):	juan carlos filter martin

The buttons "Control total + Enviar" and "Limpiar los campos" are also visible at the bottom.

- * Con Botón CONTROL TOTAL + ENVIO [datos correcto]

A screenshot of a web browser showing a form with a success message. The message box is light blue with the text "Esta página dice" and "Todos los campos son correcto", and a green "Aceptar" button. The form below has four fields: "Introduce un email (p.e. jcfm@hotmail.com)", "Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número)", "Introduce Nombre de usuario (tres o más caracteres)", and "Introduce Nombre y apellido (obligatorio)". Each field has a "Validar" button. At the bottom, there are two buttons: "Control total + Enviar" and "Limpiar los campos".

5. RA03_e) Se han eliminado elementos de documentos web.

El botón LIMPIAR LOS CAMPOS limpiar los valores y volverá a los estilos iniciales todos los input del formulario

Limpiar los campos

- * Mediante **onclick="limpiar"** podemos crear la función

```
<button type="submit" onclick="validarFormulario()">Control total + Enviar</button>
<button type="button" onclick="limpiar()">Limpiar los campos</button>
```

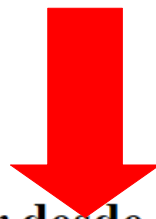
- * En javascript indicamos mediante **function limpiar()** que el email sea correcto controlando los posibles errores.

```
function limpiar() {
    document.getElementById('email').value = '';
    document.getElementById('password').value = '';
    document.getElementById('usuario').value = '';
    document.getElementById('nombre').value = '';
    document.getElementById('email').classList.remove('errores');
    document.getElementById('password').classList.remove('errores');
    document.getElementById('usuario').classList.remove('errores');
    document.getElementById('nombre').classList.remove('errores');
}
```


- ✖ Indicándole mediante `.classList.remove('errores')` que elimine la clase errores de la lista en los id email, password, usuario y nombre del JavaScript
- ✖ Dejo un ejemplo del formulario con errores y pulsando el botón limpiar los campos , elimina tanto los textos como los estilos.

Datos a enviar desde el formulario.

Introduce un email (p.e.: pepe@foc.es):	<input type="text" value="juancarlosfilterhotmail.com"/>	<input type="button" value="Validar"/>
Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):	<input type="password" value="..."/>	<input type="button" value="Validar"/>
Introduce Nombre de usuario (tres o más caracteres):	<input type="text" value="asd"/>	<input type="button" value="Validar"/>
Introduce Nombre y apellido (obligatorio):	<input type="text" value="juan carlos filter martin"/>	<input type="button" value="Validar"/>
<input type="button" value="Control total + Enviar"/>		<input type="button" value="Limpiar los campos"/>



Datos a enviar desde el formulario.

Introduce un email (p.e.: pepe@foc.es):	<input type="text"/>	<input type="button" value="Validar"/>
Introduce una contraseña (5 caracteres, al menos 1 May, 1 Min, 1 Número):	<input type="password"/>	<input type="button" value="Validar"/>
Introduce Nombre de usuario (tres o más caracteres):	<input type="text"/>	<input type="button" value="Validar"/>
Introduce Nombre y apellido (obligatorio):	<input type="text"/>	<input type="button" value="Validar"/>
<input type="button" value="Control total + Enviar"/>		<input type="button" value="Limpiar los campos"/>

6. RA03_f) Se han realizado modificaciones sobre los estilos de un documento web.

Cuando se produzca un error en alguno de los datos, el input correspondiente cambiará de estilo, a un estilo genérico que tendremos creado llamado errores.

- ✖ El estilo genérico llamado errores es una clase dentro del CSS que modifica el color de fondo a rojo cuando se produce un error en el formulario.

```
}  
#nombre{  
    margin-left: 215px  
}  
.errores{  
    background-color: red;  
}
```

- ✖ Y esto es colocado en el JavaScript dentro de la función validarFormulario y en cada comprobación de error de los diferentes id (email, password, usuario y nombre)

```
if (!email.match(/^[^\s@]+@[^\s@]+\.[^\s@]+$/)) {  
    errores.push('Por favor, introduce un email válido');  
    document.getElementById('email').classList.add('errores');  
}  
  
if (!password.match(/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d){5,}$/)) {  
    errores.push('La contraseña debe tener al menos 5 caracteres, incluyendo al menos');  
    document.getElementById('password').classList.add('errores');  
}  
  
if (!usuario.match(/^.{3,}$/)) {  
    errores.push('El nombre de usuario debe tener al menos tres caracteres');  
    document.getElementById('usuario').classList.add('errores');  
}  
  
if (nombre.trim() === '') {  
    errores.push('Por favor, introduce tu nombre y apellido');  
    document.getElementById('nombre').classList.add('errores');  
}  
  
if (errores.length > 0) {  
    alert('Se encontraron los siguientes errores:\n\n' + errores.join('\n'));
```