

ciclo: [DAM]

MÓDULO DE [Lenguaje de Marcas y Sistemas de Gestión de Información]

[Tarea Nº 03]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc76_2328992529)

[2. Crear un formulario en HTML 3](#__RefHeading___Toc78_2328992529)

[Formulario final similar al siguiente 3](#__RefHeading___Toc80_2328992529)

[3. RA03\_c) Se han utilizado métodos para la selección y acceso de los diferentes elementos de un documento web. 4](#__RefHeading___Toc82_2328992529)

[Elementos del formulario identificado con un nombre de id o name para poder ser accesible desde las funciones JavaScript. 4](#__RefHeading___Toc84_2328992529)

[4. RA03\_d) Se han creado y modificado elementos de documentos web. 6](#__RefHeading___Toc82_2328992529_Copia_1)

[Se realizarán validaciones al pulsar sobre cada uno de los botones VALIDAR 6](#__RefHeading___Toc93_2328992529)

[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](#__RefHeading___Toc255_588073025)

[5. RA03\_e) Se han eliminado elementos de documentos web. 16](#__RefHeading___Toc82_2328992529_Copia_1_)

[El botón LIMPIAR LOS CAMPOS limpiar los valores y volverá a los estilos iniciales todos los input del formulario 16](#__RefHeading___Toc267_588073025)

[6. RA03\_f) Se han realizado modificaciones sobre los estilos de un documento web. 18](#__RefHeading___Toc82_2328992529_Copia_11)

[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](#__RefHeading___Toc540_3878552419)

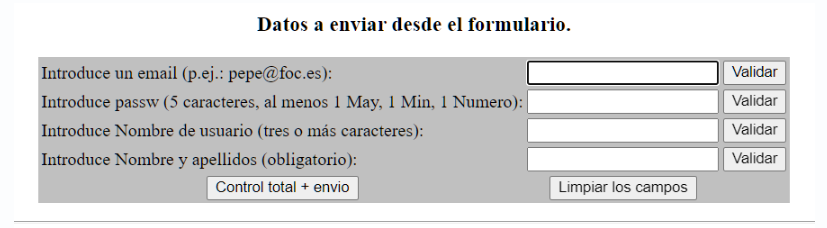
# Documentos que se adjuntan a este informe.

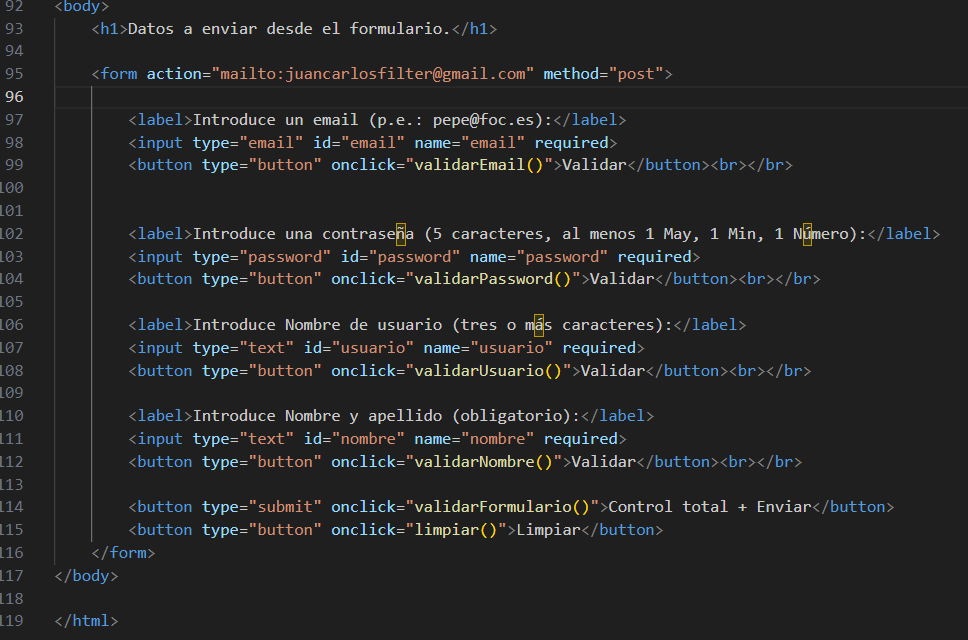
A continuación se detallan los documentos que componen la presente entrega de la tarea:

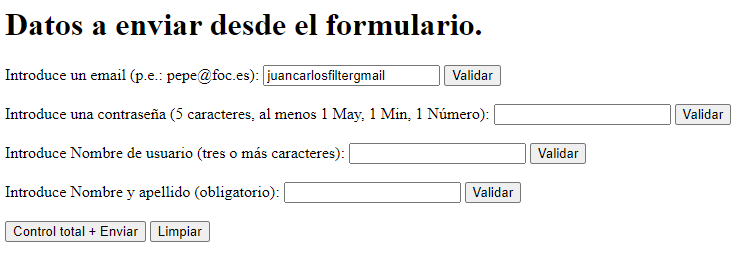
1. Informe de elaboración de la tarea.
2. Fichero HTML, CSS y JS

# Crear un formulario en HTML

## Formulario final similar al siguiente



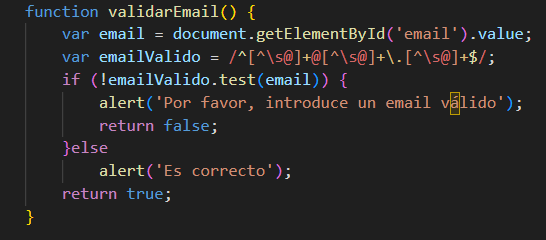
* Agregamos el titulo mediante un h1.
* Introducimos los campos mediante la etiqueta label y seguido del texto
* En la etiqueta input ponemos el tipo, el id que le asignamos, el nombre orientativo y indicándole que es obligatorio “required”.
* Para crear los botones lo indicamos mediante la etiqueta 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.



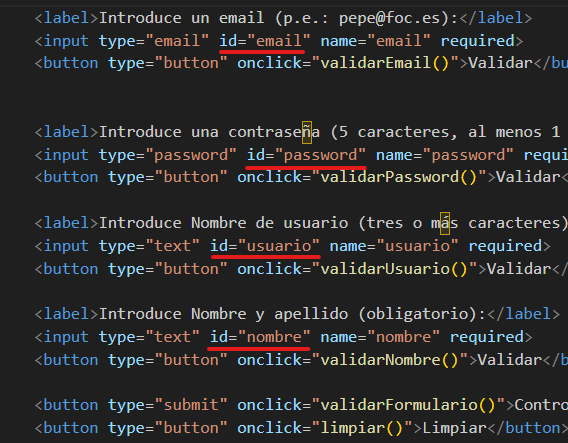
# RA03\_c) Se han utilizado métodos para la selección y acceso de los diferentes elementos de un documento web.

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

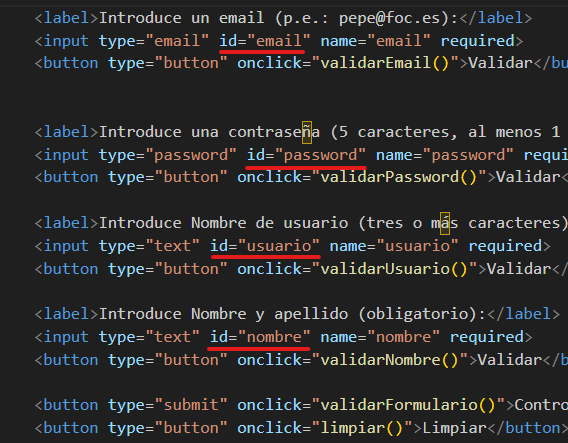
* Agregar un id sirve para que a la hora de acceder a una función de javascript y mediante el método get.ElementById(‘nombreid’) sepa a donde se tiene que dirigir y pueda evaluar dicho contenido.
* Por ejemplo con el email de la tarea, tenemos id =’email’. Entonces cuando pulsemos el botón de validarEmail 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.



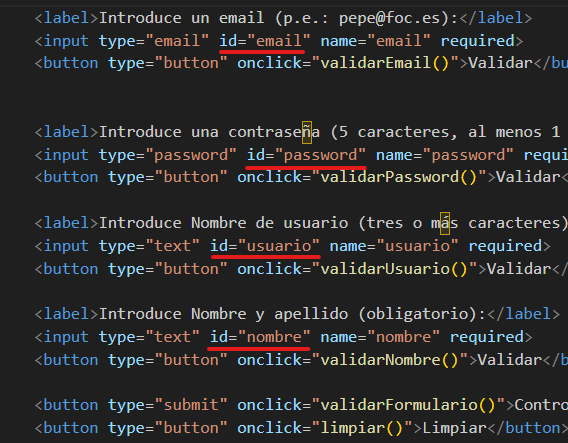
* Ahora voy a indicar los distintos id que he creado en el formulario:
* Para el **email** tenemos el id = **email**



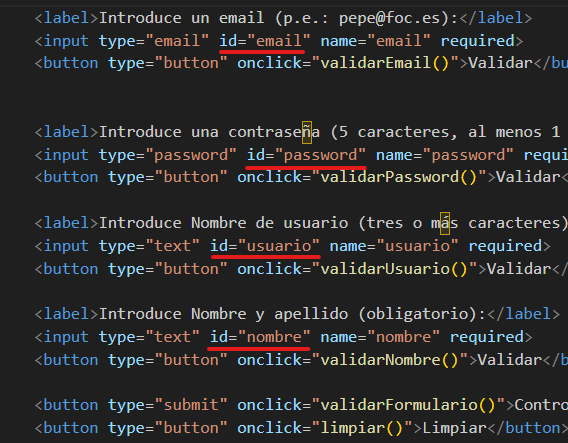
* Para la **contraseña** tenemos el id = **password**



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

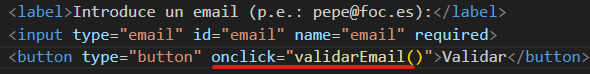


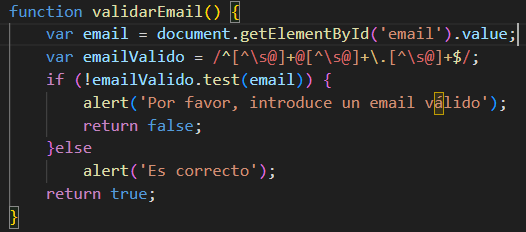
* Para el **nombre y apellido** tenemos el id = **nombre**



# RA03\_d) Se han creado y modificado elementos de documentos web.

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

* Mediante **onclick=”validarEmail”** podemos crear la función
* En javaScript indicamos mediante **function validarEmail()** que el email sea correcto controlando los posibles errores.



Comprueba si la expresión regular

encuentra coincidencias dentro de

una cadena de texto.

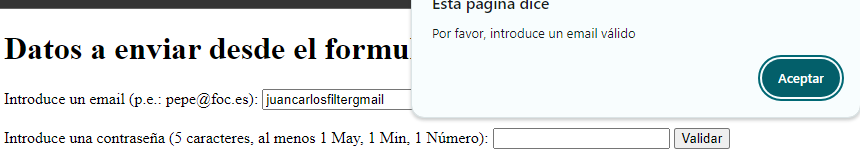
*(true si encuentra una*

*coincidencia y false si no la*

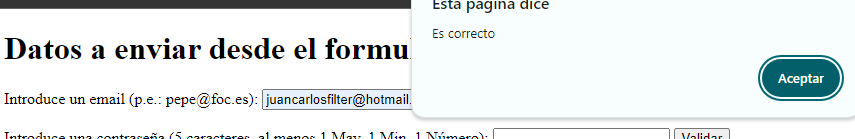
*Encuentra)*

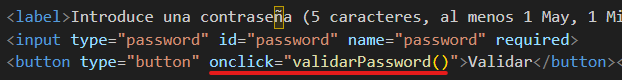
***Al tener ‘!’ está verificando si es false***

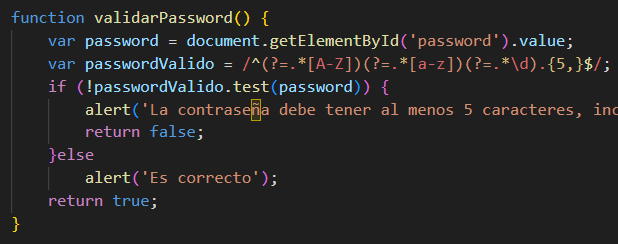
* En la web vamos a comprobar si funciona:
  + - **Con email erróneo**



* + - **Con email correcto**



* Mediante **onclick=”validarPassword**” podemos crear la función
* En javaScript indicamos mediante **function validarPassword()** que el email sea correcto controlando los posibles errores.



Comprueba si la expresión regular

encuentra coincidencias dentro de

una cadena de texto.

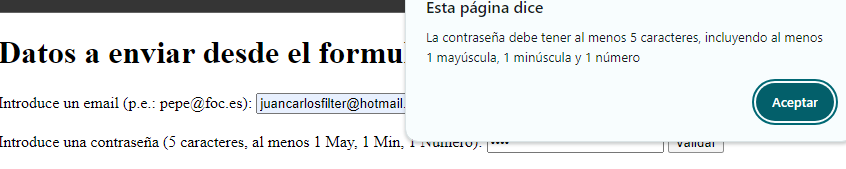
*(true si encuentra una*

*coincidencia y false si no la*

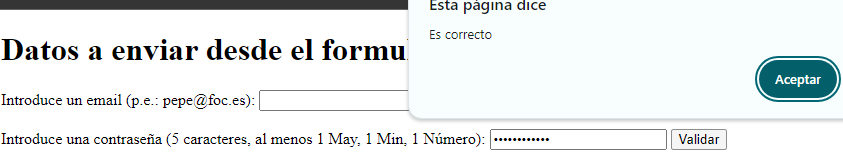
*Encuentra)*

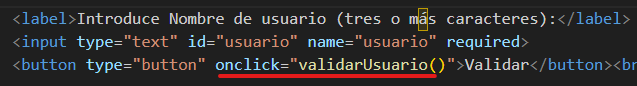
***Al tener ‘!’ está verificando si es false***

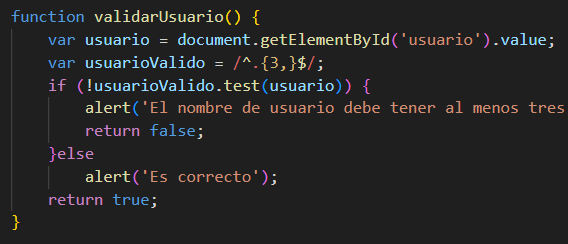
* En la web vamos a comprobar si funciona:
  + - **Con password erróneo**



* + - **Con password correcto**



* Mediante **onclick=”validarUsuario**” podemos crear la función
* En javaScript indicamos mediante **function validarUusario()** que el email sea correcto controlando los posibles errores.



Comprueba si la expresión regular

encuentra coincidencias dentro de

una cadena de texto.

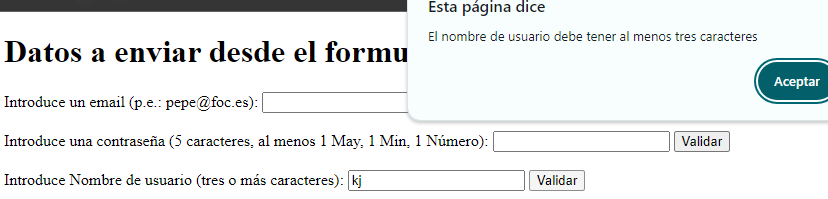
*(true si encuentra una*

*coincidencia y false si no la*

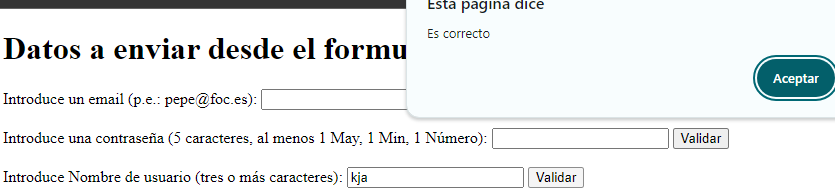
*Encuentra)*

***Al tener ‘!’ está verificando si es false***

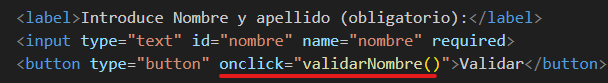
* En la web vamos a comprobar si funciona:
  + - **Con usuario erróneo**



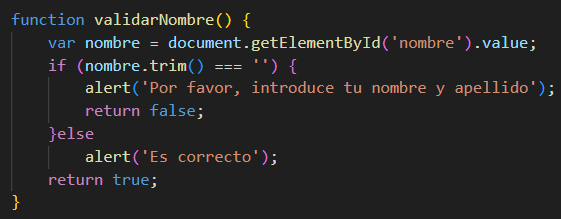
* + - **Con usuario correcto**



* Mediante **onclick=”validarNombre**” podemos crear la función



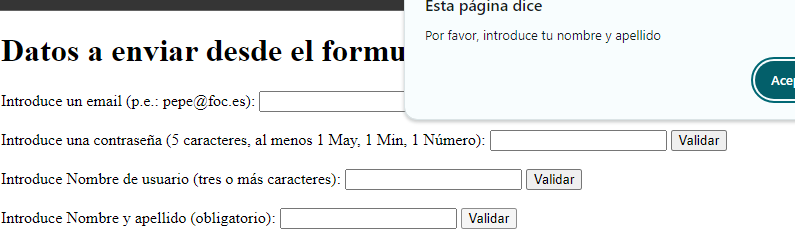
* En javaScript indicamos mediante **function validarNombre()** que el email sea correcto controlando los posibles errores.



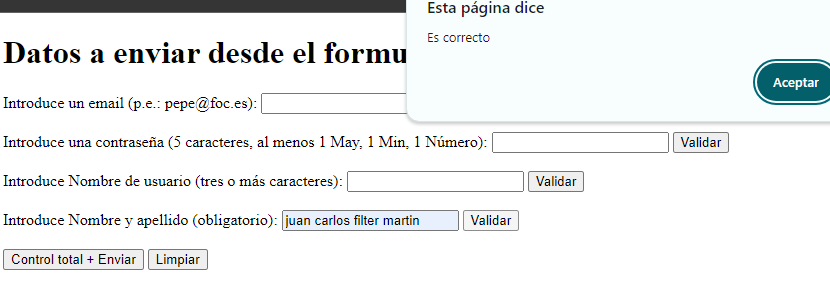
*Elimina los espacios en blanco al*

*principio y al final de la cadena*

* En la web vamos a comprobar si funciona:
  + - **Con nombre y apellidos erroneo**



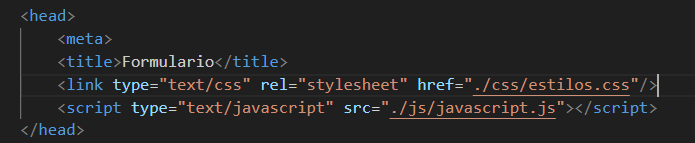
* + - **Con nombre y apellidos correcto**



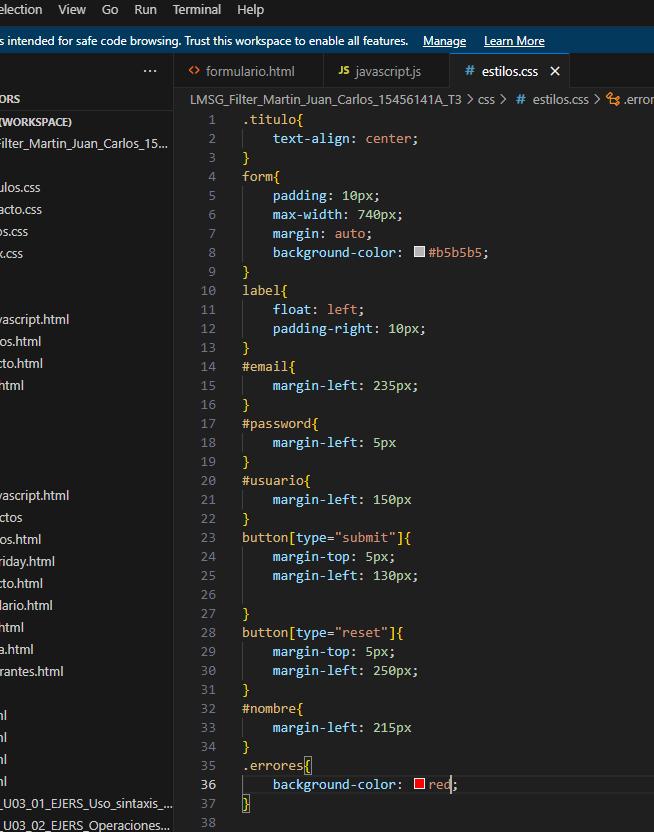
## Pulsar sobre el botón CONTROL TOTAL + ENVIO que validará todos los campos dando un mensaje con todos los errores encontrados y cambiando el estilo de los campos erróneos.

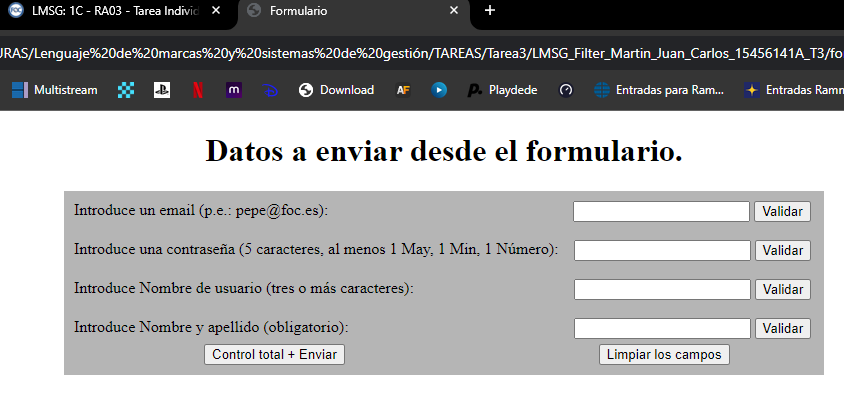
* Empezaré primeramente creando los estilos del formulario y separando tanto **javascript** como **css** fuera del **html**.

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

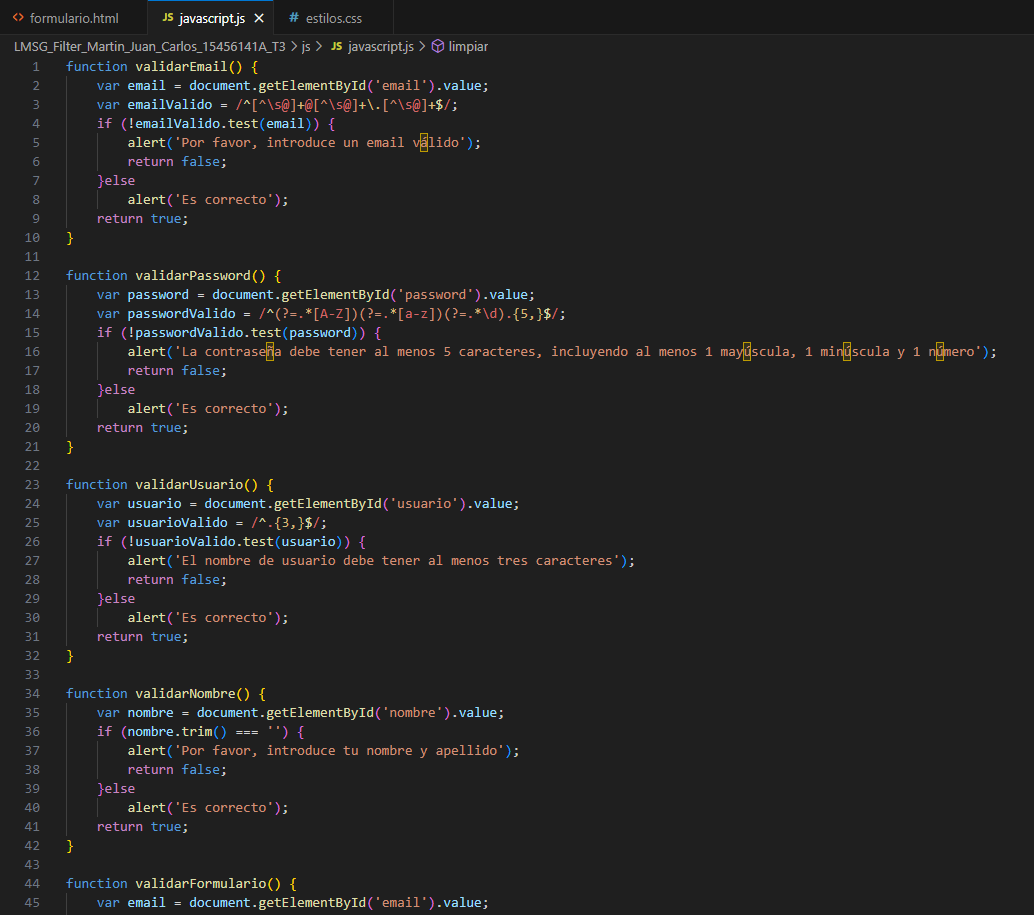
 → Para **javascript** dentro de script indicaremos la ruta mediante **src**

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

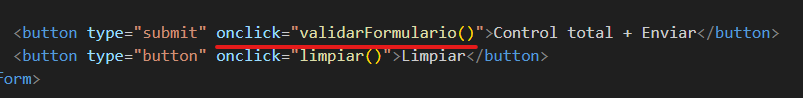




→ Y JavaScript estará aparte del html con los con los script y las condiciones que debe cumplir el formulario al interactuar con los botones

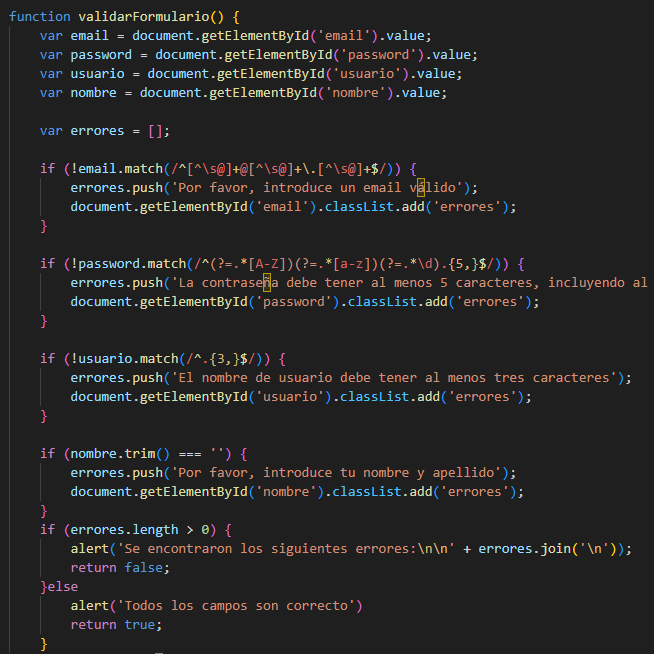


**Botón control total + envio**

* Mediante **onclick=”validarFormulario**” podemos crear la función

**Comprobando posibles errores**

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



*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*

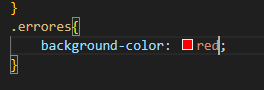
*una cadena de texto*

*Elimina los espacios en blanco al*

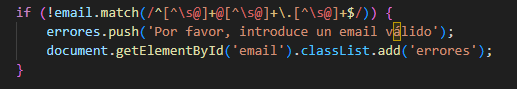
*principio y al final de la cadena*

**Aplicando estilos de error**

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



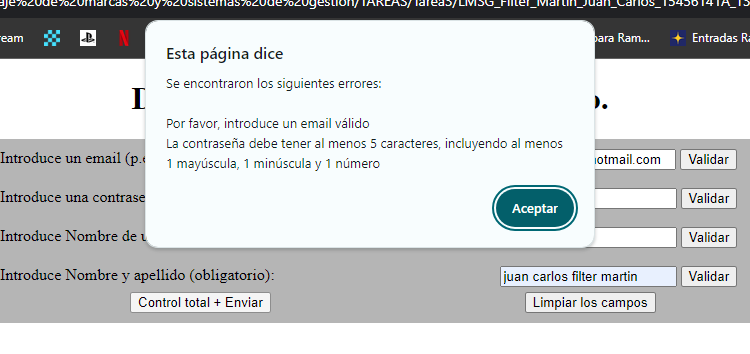
* Ahora en el javaScript dentro de cada if donde se produce dicho error le añadimos dicho estilo de clase **errores**

****

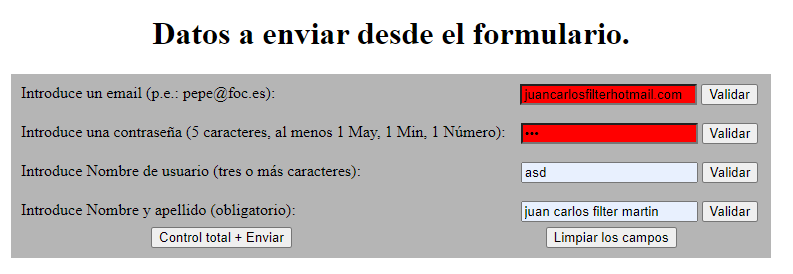
* Por ejemplo en el email si no contiene lo indicado y entra en el if mostrará el mensaje y seguidamente aplicará el estilo de la clase errores

**COMPROBACIÓN EN LA WEB**

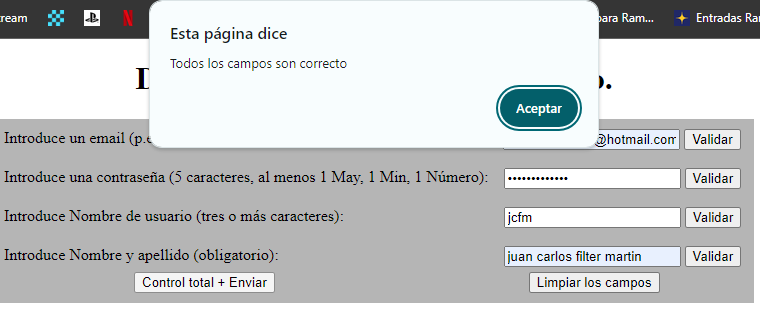
* En la web vamos a comprobar si funciona:
  + - **Con Botón CONTROL TOTAL + ENVIO [datos erróneos ]**



* Y aplica el estilo indicado:



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



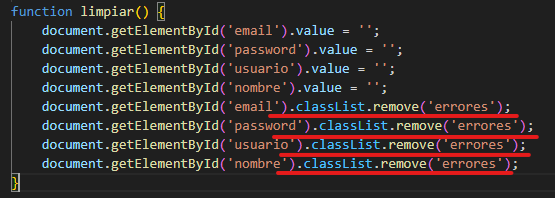
# RA03\_e) Se han eliminado elementos de documentos web.

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

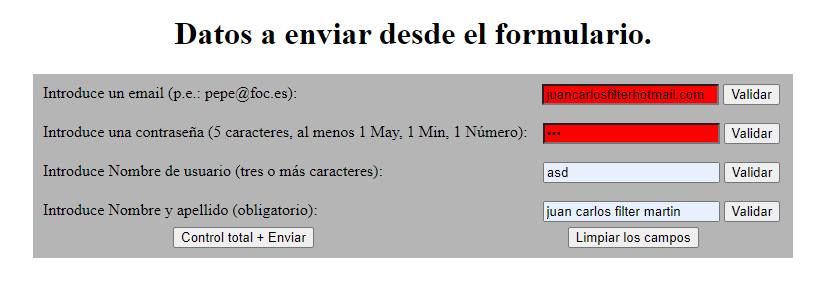
* Mediante **onclick=”limpiar**” podemos crear la función

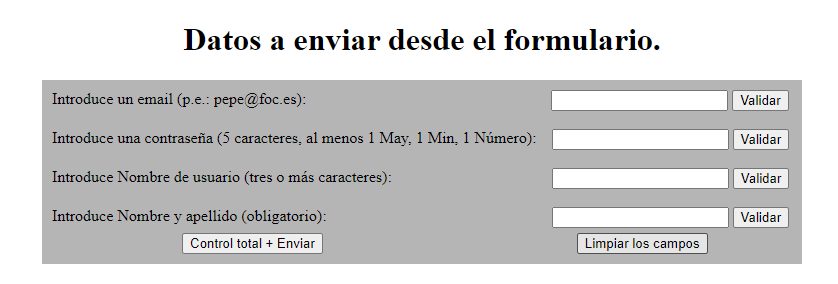


* En javaScript indicamos mediante **function limpiar()** que el email sea correcto controlando los posibles errores.



* Indicandole 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 texto como los estilos.

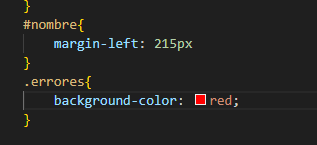




# RA03\_f) Se han realizado modificaciones sobre los estilos de un documento web.

## 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.



* 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)

