



## DWEC 2º DAW DOM, EVENTOS, FORMULARIOS

Curso 2.024-2025

### UDAD 5 EVENTOS: SIMULADOR ENVÍO EMAILS

#### INTRODUCCIÓN RESUMEN:

Partiendo de los ficheros .html y .css (tailwind) de una página web con un formulario, daremos vida al mismo.

Podemos ver 3 campos o inputs y dos botones: enviar (deshabilitado), y borrar formulario (Reset).

Al clicar sobre el botón “Enviar”, tras unos segundo de espera (spinner), aparecerá una alerta de “El mensaje se ha enviado” (simulación de envío).

Una vez enviado el mensaje, el formulario se reiniciará.

El botón de Reset estará habilitado para borrar el contenido de los inputs en cualquier momento.

A la izquierda, ejemplo inicial del formulario y a la derecha, ejemplo de formulario correctamente llenado.

Enviar mails

Email:

Asunto:

Mensaje:

▷ Enviar ✖️ Reset

Enviar mails

Email:

Asunto:

Mensaje:

▷ Enviar ✖️ Reset

## 1. EJECUCIÓN → VALIDACIÓN DE LOS CAMPOS DEL FORMULARIO

### a) Validación del email:

La validación de mail se hará a través de una expresión regular y se validará a través del método test que devuelve un booleano.

```
const regex = /^[^\\w+([._+]?\\w+)*@\\w+([.-]?\\w+)*(\\.\\w{2,10})+$/;\nconst resultado = regex.test(email);
```

\*\*El método test devuelve un booleano (true si el string cumple con la expresión regular)

### b) Validación de los campos del formulario → ERRORES

Para que el botón de “Enviar” se habilite y recobre su color original deben estar los tres campos rellenados correctamente. Para ello, podremos crear un objeto (**Object Literal**), que lo iremos llenando según vaya pasando la validación.

Podemos seleccionar los botones del formulario según el tipo. Así:

```
const btnReset = formulario.querySelector('button[type="reset"]');
```

Ojo con un campo relleno únicamente con espacios.

Si hay **errores**, deberá aparecer una alerta indicándolos. Para ello, crearemos a través de **DOM Scripting**, los elementos del DOM para los mensajes. Las **clases** que se le aplicarán al mensaje son:

```
'bg-red-600', 'text-white', 'p-2', 'text-center'
```

### Enviar mails

Email:

nescrivan@.com

El campo EMAIL NO es válido

NO haremos cada vez una alerta, creamos una función para todas ellas, así evitamos repetir código y ganamos en reutilización.

Cada mensaje de error deberá aparecer debajo de la caja de texto correspondiente (con la palabra clave en mayúscula).

### Enviar mails

Email:

Email Destino, ej: mkt@empresa.com

El campo EMAIL es obligatorio

Asunto:

Asunto Email

El campo ASUNTO es obligatorio

Mensaje:

Mensaje de Email

El campo MENSAJE es obligatorio

▷ Enviar

Reset

Cuando el usuario rellena bien los campos, el error debe desaparecer.

Cada vez que no se cumplen las condiciones (borramos un campo o escribimos mal el campo Email), el botón se deshabilitará y su color será el de la pantalla inicial.

## 2. EJECUCIÓN → GESTIÓN ERRORES DUPLICADOS

Debemos evitar duplicar los mensajes de error.

A la izquierda, INCORRECTO, a la derecha, gestión CORRECTA

The image shows two side-by-side 'Enviar mails' (Send emails) forms. Both forms have three fields: Email, Asunto (Subject), and Mensaje (Message). They both contain validation errors:

- Email:** The input 'GHFGHDGDFG' is invalid, resulting in two red error messages: 'El campo EMAIL NO es válido'.
- Asunto:** The input 'Asunto Email' is invalid, resulting in one red error message: 'El campo ASUNTO es obligatorio'.
- Mensaje:** The input 'Mensaje de Email' is empty, resulting in one red error message: 'El campo MENSAJE es obligatorio'.

The left form shows redundant error messages for the Email field, while the right form correctly groups the errors by field.

## 3. EJECUCIÓN → ENVÍO DE DATOS Y SPINNER

Una vez los tres campos son correctos y el botón “Enviar” estará habilitado.

Al clicar sobre “Enviar”, debemos simular la espera de un envío de mail. Añadiremos un spinner. Para ello, se ha usado la página <https://tobiasahlin.com/spinkit/>.

En esa página encontramos el código de HTML y de CSS del spinner que elegimos. Debemos:

- Insertar el código HTML a través del fichero JS
- Añadir el código CSS en otro fichero CSS para ello.

The image shows two side-by-side 'Enviar mails' (Send emails) forms. Both forms have three fields: Email, Asunto (Subject), and Mensaje (Message). All fields are filled correctly:

- Email:** 'nescrivan@gmail.com'
- Asunto:** 'Prueba'
- Mensaje:** 'tyertyertyer'

The left form shows the standard button state. The right form shows the 'Enviar' button with a spinner icon (a small circle with a vertical line) and a green success message at the bottom: 'MENSAJE ENVIADO CORRECTAMENTE'.

El spinner desaparecerá en varios segundos. Una vez que desaparezca, el formulario debe vaciarse y el botón “enviar” quedará de nuevo deshabilitado → estado inicial.

### Enviar mails

Email:  
Email Destino, ej: mkt@empresa.com

Asunto:  
Asunto Email

Mensaje:  
Mensaje de Email

 Enviar    Reset

#### 4. AMPLIACIÓN → AÑADIR CAMPO CC

Añadiremos un campo debajo de “Email” llamado “CC” (se puede hacer con DOM Scripting o con Template String).

NO será obligatorio rellenarlo para que se active el botón ENVIAR.

Si se rellena, se tendrá que validar la dirección de email

### Enviar mails

Email:  
Email Destino, ej: mkt@empresa.com

CC:  
Destino copia, opcional

Asunto:  
Asunto Email

Mensaje:  
Mensaje de Email

 Enviar    Reset