



Florida
Universitària

Formularios

Desarrollo Web en Entorno Cliente

Curso 2025/2026

Paco Segura

Formularios

- Con JavaScript podemos recoger la información recibida en un formulario y tratarla para, por ejemplo, validarla antes de enviar una petición al servidor.
 - Para ello, la función del manejador de eventos debe tomar un parámetro [\(event\)](#). Con **event** y **preventDefault** evitamos el comportamiento por defecto que el navegador asocia a los formularios -realizar una petición POST-.
-

Formularios

- **Eventos del formulario:**
 - **submit**: al hacer pulsar un botón submit.
 - **focus/blur**: obtener/perder el foco de entrada.
-

Evento *submit*

- Es el evento asociado a un formulario -elemento '**form**' HTML-.
 - Se activa cuando se pulsa el botón **submit** de un formulario.
 - Permite capturar **todos** los campos del formulario.
 - Con **reset()** podemos vaciar los campos del formulario.
-

Evento *submit*

```
<form id="form">
  <div>
    <input type="text" name="name" value="" placeholder="Nombre" />
  </div>
  <div>
    <input type="text" name="email" value="" placeholder="Email" />
  </div>
  <div>
    <input type="submit" value="Enviar" class="primary" />
  </div>
</form>
```

```
const form = document.getElementById('form');
const inputs = document.getElementsByTagName('input');
```

```
function validateForm(event){
  event.preventDefault();

  for(let i = 0; i < inputs.length - 1; i++) {
    console.log(inputs[i].value);
  }

  form.reset();
}
```

```
form.addEventListener('submit', (event) => validateForm(event));
```

1 - Seleccionamos el formulario y los inputs del formulario

2 - Aplicamos el evento **'submit'** para recoger todos los campos del formulario. Debemos pasar **'event'** por parámetro a la función que maneja el evento -**validateForm**-.

3 - En la función que maneja el evento, aplicamos **event.preventDefault()** para evitar el comportamiento que por defecto tiene el elemento **form** de HTML -realizar una petición POST a un servidor-.

4 - Con **reset()** formateamos el formulario.

Expresiones regulares

- Para validar los campos de un formulario, empleamos [expresiones regulares](#).
 - Nos permiten comprobar de manera rápida si la información introducida es correcta.
-

Expresiones regulares

- Ejemplos:
 - Para indicar que una expresión estará formada por letras -cualquiera que sea- que estén en minúsculas:

```
const regex = /^[a-z]+$/;
```

- Para indicar que una expresión estará formada por letras que estén en mayúsculas:

```
const regex = /^[A-Z]+$/;
```

Expresiones regulares

- Ejemplos:
 - Para indicar que una expresión estará formada por letras que estén en minúsculas o mayúsculas, indistintamente:

```
const regex = /^[a-zA-Z]+$/;
```

Expresiones regulares

- Ejemplos:
 - Se pueden añadir caracteres especiales, como letras con acento:

```
const regex = /^[áéíóúa-zA-Z]+$/;
```

- También se puede validar que la expresión esté formado por números:

```
const regex = /^[0-9]+$/;
```

Expresiones regulares

- Para validar que la información introducida cumple con lo indicado, empleamos la función **test()** de JavaScript.

```
const form = document.getElementById('form');
const inputs = document.getElementsByTagName('input');

function validateForm(event){
  event.preventDefault();
  for(let i = 0; i < inputs.length - 1; i++) {
    const regex = /^[a-zA-Z]+$/;
    console.log(regex.test(inputs[i].value));
  }
  form.reset();
}

form.addEventListener('submit', (event) => validateForm(event));
```

1- Definimos una **expresión regular** indicando que la información introducida está formada sólo por letras -mayúsculas o minúsculas-.

2- Aplicamos la función **test()** a los campos del formulario para validar. Si cumple con la expresión regular, devolverá **true**. Si no cumple, devolverá **false**.

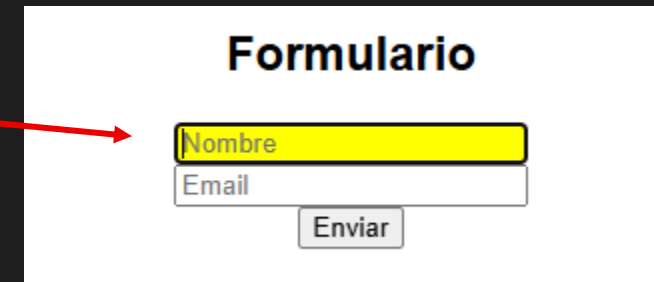
Eventos *focus* y *blur*

- Para obtener/quitar el foco de entrada de los campos del formulario.

```
const inputs = document.getElementsByTagName('input');

inputs[0].addEventListener("focus", (event) => {
  event.target.style.background = "yellow";
});

inputs[0].addEventListener("blur", (event) => {
  event.target.style.background = "";
});
```



The diagram shows a form titled "Formulario". It contains two input fields: "Nombre" and "Email". The "Nombre" field is highlighted in yellow, indicating it is the current focus. Below the "Email" field is a button labeled "Enviar". A red arrow points from the "focus" event listener code in the adjacent block to the "Nombre" input field, illustrating the effect of the event.