

Eventos

Módulo 2 – Unidad 4

Manejo de eventos en formularios y componentes



Objetivos

- Capturar y manejar eventos del navegador en React.
- Diferenciar entre eventos de interacción general y eventos de formularios.
- Entender el flujo de un formulario controlado.



Consigna

1. Crear la interfaz.

- Diseñar un pequeño formulario con:
 - Campo de texto para el nombre.
 - Campo de correo electrónico.
 - Botón de envío.

2. Capturar eventos básicos.

- Detectar el evento `onChange` en ambos campos y registrar los cambios en consola.
- Usar `onFocus` y `onBlur` para mostrar mensajes cuando un campo esté activo o pierda el foco.

3. Controlar el envío.

- Implementar un evento `onSubmit` que:
 - Prevenga el comportamiento por defecto.
 - Muestre en consola los datos ingresados.

4. Interacciones adicionales.

- Detectar con `onKeyDown` si se presiona la tecla Enter dentro del campo de nombre y mostrar un mensaje.
- Cambiar el estilo del botón al pasar el mouse usando `onMouseEnter` y `onMouseLeave`.

5. Opcional.

- Rehacer el formulario usando `react-hook-form` para comparar la diferencia en código y comportamiento.

Formato de presentación

La entrega debe realizarse a través de un **repositorio en GitHub**, el cual debe incluir:

- Proyecto en **React (Vite recomendado)**, limpio y organizado.
- Componente con un **formulario** que contenga:
 - Campo de texto (nombre).
 - Campo de email.
 - Botón de envío.
- Implementación de eventos:
 - `onChange`, `onFocus`, `onBlur`.
 - `onSubmit` con `preventDefault()`.
 - `onKeyDown` para detectar Enter.
 - `onMouseEnter` y `onMouseLeave` para el botón.
- (Opcional) Versión alternativa con **react-hook-form**.
- **Archivo README.md** con:
 - Breve descripción.
 - Instrucciones de instalación y ejecución.

- Capturas de pantalla del formulario funcionando.
- Créditos del autor y citación de fuentes.

Criterios de evaluación

- Uso correcto de los **eventos de React** (`onChange`, `onFocus`, `onBlur`, `onSubmit`).
- Implementación de `preventDefault()` en el envío del formulario.
- Manejo de interacciones adicionales (`onKeyDown`, `onMouseEnter`, `onMouseLeave`).
- Organización y claridad del código.
- Presentación prolíja en el repositorio (estructura clara, README).



Bibliografía utilizada y sugerida

Libros y otros manuscritos

Banks, A. y Porcello, E. *Learning React: Modern Patterns for Developing React Apps.* 2^a ed. O'Reilly Media; 2020.

Freeman, E. y Robson E. *Head First. JavaScript Programming.* 1^a ed. Estados Unidos: O'Reilly Media; 2014.

<https://nibmehub.com/opac-service/pdf/read/Head%20First%20JavaScript%20Programming%20 %20a%20learner's%20guide%20to%20JavaScript%20programming-compressed.pdf>

Artículos y documentación en línea

MDN Web Docs. (s.f.). *DOM events.* Mozilla Corporation.

<https://developer.mozilla.org/en-US/docs/Web/Events>

React. (s.f.). *Responding to Events.* <https://react.dev/learn/responding-to-events>