

# Desarrollo en ReactJS

---

Eventos

*Desarrollador Web con React Js*

# Eventos con React

React tiene sus propios eventos, que cuentan con la misma interfaz de los eventos nativos del navegador, con la ventaja que tienen un comportamiento compatible con la mayoría de los navegadores. Reciben una función, o manejadores de eventos. Lo que hacen estas funciones es definir el comportamiento de la aplicación si se corre X o Y evento.

Algo muy común en React, es declarar los manejadores de eventos como funciones dentro de una clase. Cuando hacemos esto, debemos tener cuidado, ya que **this** no se conecta por defecto a la clase. Para resolver esto, tenemos tres opciones: hacer la conexión en el `render()`, en el `constructor()`, o usar arrow functions.

# Cómo declarar los eventos

En JavaScript la declaración de eventos la realizamos de la siguiente manera:

```
<div onclick="handleClick()">click me</div>

function handleClick() {
    alert('clicked');

    return false;
}
```

Como vemos utilizamos el **onclick**, en el cual hacemos referencia a una función declarada.

En react lo haremos de la siguiente manera:

```
<div onClick={handleClick}>click me</div>

function handleClick(event) {
  alert('clicked');

  event.preventDefault();
  event.stopPropagation();
}
```

Como vemos también utilizamos el evento **onClick**, pero a diferencia de lo realizado en Javascript solo declaramos el nombre de la función. 02

La función declarada sólo recibe el argumento evento, sobre el cual podremos realizar la acción de stop propagation (detener propagación o comportamiento por defecto).

Consideraciones:

- El nombre del evento tiene que ser **camelCase** y no minúscula sostenida.
- Al evento se le pasa la función y no una cadena de texto.
- En React si quieres prevenir un comportamiento por defecto o la propagación de un evento debes hacerlo explícitamente llamando los métodos **preventDefault()** y **stopPropagation()** respectivamente.

# Eventos disponibles en React

React soporta los siguientes eventos:

- **Mouse:** onClick, onContextMenu, onDoubleClick, onMouseDown, onMouseEnter, onMouseLeave, onMouseMove, onMouseOut, onMouseOver, onMouseUp,
- **Drag & Drop:** onDrag, onDragEnd, onDragEnter, onDragExit, onDragLeave, onDragOver, onDragStart, onDrop.
- **Focos y formularios:** onFocus, onBlur, onChange, onInput, onSubmit.
- **Touches:** onTouchCancel, onTouchEnd, onTouchMove, onTouchStart.
- **Cortar y pegar:** onCopy, onCut, onPaste.
- **Scrolls:** onScroll, onWheel.



# Cómo recibir parámetros en los eventos

Si queremos pasar un parámetro a la función llamada en la ejecución de un evento en react debemos hacerlo utilizando la función **arrow** de es6. Por ejemplo:

```
<button onClick={{e} => this.deleteRow(id, e)}>Delete Row</button>
```

Aquí vemos que cuando se ejecuta el **onClick**, se llama a la función arrow que tiene como parámetro a e.

Luego esta función llama en su ejecución a la función **deleteRow**, la cual debe estar declarada dentro del componente.

# Gracias!