



wawiwa

React

2022 © Wawiwa Tech | Confidential

**wawiwa**

# 16- Formularios en React

2022 © Wawiwa Tech | Confidential



## Introducción

Los formularios son una parte esencial de muchas aplicaciones web.

En React, puedes construir formularios usando una combinación de elementos “form” de HTML, Javascript y componentes de React.

Hay dos enfoques principales para manejar formularios en React:

- Usando componentes controlados
- Usando componentes no controlados



# Componentes no controlados

Los componentes no controlados son elementos “form” de HTML regulares que mantiene su propio estado. En este enfoque los elementos “form” se actualizan cuando el usuario interactúa con ellos, pero React no controla su estado. Para devolver los valores de los elementos “form” puedes usar el DOM API, por ejemplo, refs en React.

```
const emailRef = useRef();
const handleSubmit = (event) => {
  alert(emailRef.current.value);
};
```

```
return (
  <form onSubmit={handleSubmit}>
    Email:
    <input type="email" ref={emailRef} />
    <button type="submit">Submit</button>
  </form>
);
```





## Componentes controlados

Son elementos “form” controlados por el estado de React. En este enfoque, React controla el estado del elemento “form”, y se actualiza en respuesta a las interacciones del usuario. Esto se logra pasándole el valor del estado en el prop “value” del elemento y actualizando el estado en respuesta al evento “onChange”. Aquí hay un ejemplo de un formulario simple controlado en React:

```
const [email, setEmail] = useState('');
const handleSubmit = () => {
  alert(email);
};
return (
  <form onSubmit={handleSubmit}>
    Email:
    <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} />
    <button type="submit">Submit</button>
  </form>
);
```



## Ejercicio 18

Crea el siguiente formulario:

### Order a hamburger:

\* Name:

\* Email:

\* --Do you want to order chips?--   
Special requests?

☐ \* I have read and agreed to the terms and conditions



- Haciendo click en el botón submit, checa si existen “labels” vacías que deben ser completadas.
- Si no falta información, alerta la orden, si no, alerta un mensaje al usuario.