

# Resumen de Handling React Form Operations with Formik and Yup

El artículo explica cómo manejar formularios en **React** de manera eficiente usando **Formik** (para gestión de estado y eventos) y **Yup** (para validación de datos).

## 1. Problemas con Formularios en React

- Los formularios en React puro requieren mucho código repetitivo (manejo de estado, validación, errores, etc.).
- Soluciones manuales pueden volverse complejas rápidamente.

## 2. Introducción a Formik

- **Formik** simplifica:
  - Gestión del estado del formulario.
  - Manejo de eventos (onSubmit, onChange, etc.).
  - Validación y visualización de errores.

- Ejemplo básico de uso:

```
import { Formik, Form, Field } from 'formik';

<Formik initialValues={{ email: '' }} onSubmit={(values) => console.log(values)}>

  <Form>

    <Field name="email" type="email" />

    <button type="submit">Submit</button>

  </Form>

</Formik>
```

## 3. Validación con Yup

- **Yup** permite definir esquemas de validación declarativos:

```
import * as Yup from 'yup';

const validationSchema = Yup.object().shape({

  email: Yup.string().email('Invalid email').required('Required'),

});
```

- Se integra con Formik mediante la prop validationSchema.

## 4. Ventajas Clave

- **Menos código:** Elimina la necesidad de escribir lógica repetitiva.
- **Validación robusta:** Yup ofrece validaciones predefinidas (email, contraseñas, etc.).
- **Mejor experiencia de desarrollo:** Formik maneja automáticamente estados como `isSubmitting` o `errors`.

## Conclusión

Formik + Yup es una combinación poderosa para formularios en React, reduciendo código boilerplate y mejorando la mantenibilidad.