

# Bootcamp de Full Stack

Bienvenidos a la clase N°44

- Formularios en React
  - Método purista
  - Formik
  - Validaciones con Yup
- Proyecto Integrador F2 (i1)

# React

## Formulario Purista

Manejar **formularios** en React de **forma pura** implica utilizar únicamente las herramientas nativas de React, como el estado (useState) y eventos (onChange, onSubmit, etc.), sin depender de librerías externas.

Esta aproximación ofrece máximo control y flexibilidad, ideal para formularios simples o cuando se busca entender a fondo cómo funciona el manejo de inputs, validación y envío en React.

### Reglas y recomendaciones básicas para formularios

- Controlar los inputs con useState para manejar los valores y onChange para actualizarlos.
- Usar onSubmit para enviar el formulario (se debe prevenir el comportamiento por defecto)
- Realizar validaciones personalizadas dentro de handleSubmit.
- Guardar errores en un estado (errors) y mostrarlos de forma condicional.
- Separar funciones de validación para mantener el código limpio y reutilizable.

# React

## Formik

**Formik** es una librería de JavaScript diseñada para simplificar el manejo de formularios en aplicaciones React. Proporciona una solución declarativa y estructurada para controlar el estado del formulario, la validación de los campos, el manejo de errores y el envío de datos, reduciendo el código repetitivo y los errores comunes asociados a formularios complejos.

Formik es ampliamente utilizado en proyectos React donde se requiere una gestión eficiente y escalable de formularios con múltiples campos y validaciones personalizadas.

Se instala por medio de `npm install formik --save`

Documentación oficial: <https://formik.org/docs/overview>

# React

## Yup

Yup es una biblioteca de validación de esquemas para JavaScript que permite definir reglas de validación de forma declarativa, clara y reutilizable. Es especialmente útil en proyectos React y suele utilizarse junto a Formik para validar formularios de manera robusta y eficiente.

Permite construir objetos de validación que definen la forma y restricciones de los datos esperados, verificando tipos, rangos, formatos, requerimientos, estructuras anidadas, entre otros.

Se instala mediante `npm install yup --save`

Documentación oficial: <https://github.com/jquense/yup?tab=readme-ov-file>

# BREAK

Descansemos 10 minutos



# Proyecto Integrador F2

## Iteración N°1



Para llevar a cabo el proyecto integrador F2, accede al documento de la consigna a través del siguiente enlace:

[https://docs.google.com/document/d/12Lk6Fy73oRCjgNo5CUntmZvL\\_Dw7aqRT7-ZSYw6lOxY/edit?usp=sharing](https://docs.google.com/document/d/12Lk6Fy73oRCjgNo5CUntmZvL_Dw7aqRT7-ZSYw6lOxY/edit?usp=sharing)

Backlog:

<https://docs.google.com/spreadsheets/d/11GYDhodvQ5vulqkWI4zBbzojagbgepKybdyQRehjXE4/edit?usp=sharing>

# CIERRE DE CLASE

Continuaremos en la próxima clase

