



Front End III

## Clase 11: Primer entregable

### \_ Presentación de la actividad

Vamos a poner en práctica los conocimientos que hemos adquirido acerca de la creación y manejo de formularios utilizando React. Para ello, te pediremos que crees un pequeño formulario, el cual deberá permitir a cualquier persona interactuar con el mismo ingresando los datos que se solicitan. Una vez ingresados dichos datos, al hacer clic en el botón de Enviar, se deberán llevar a cabo ciertas validaciones para corroborar que la información ingresada coincida con los datos esperados. Si ello es así, mostraremos en pantalla un componente que contendrá toda la información recibida. Caso contrario, deberás mostrar un mensaje de error.

Como punto de partida, utilizaremos el template que creamos en la clase 10 para adelantar el trabajo. En caso de que no lo tengamos, podemos utilizar el que se encuentra en el siguiente [repositorio](#) inicial.

### \_ Acerca del formato de entrega

1. El trabajo deberá ser realizado en forma **individual**.
2. El plazo máximo de entrega será la **hora de finalización de la clase**.
3. Para realizar la entrega deberás **subir tu código a un repositorio** y **completar el siguiente [formulario](#)**.

## — Consignas y requisitos de entrega

1. La temática del formulario puede ser cualquiera de nuestra preferencia. Podemos solicitar información sobre música, animales, libros, autos, personajes o lo que deseemos.
2. El formulario deberá contener, al menos, dos inputs de texto y un botón de tipo **submit**. Si lo deseamos, podemos agregar más inputs para hacer más completo nuestro formulario, pero esto es opcional.
3. Al hacer clic en el botón de Enviar, debemos realizar las siguientes validaciones:
  - a) Para el caso del **primer input**, la longitud mínima del texto ingresado deberá ser de 3 caracteres y no deberá contener espacios en blanco al comienzo.
  - b) Para el **segundo input**, debemos validar que contenga al menos al menos 6 caracteres.
4. En caso de que alguna de las validaciones sea incorrecta, debemos mostrar el siguiente mensaje de error: **“Por favor chequea que la información sea correcta”**.
5. En caso de que los valores ingresados superen las validaciones en forma exitosa, debemos renderizar el componente llamado **Card** que contenga dicha información. Podemos darle el estilo y/o forma que deseemos, en tanto y en cuanto contenga al menos los mismos valores que se hayan ingresado en el formulario.

## Ejemplos de pantallas

Para que puedas orientarte mejor, te dejamos unos ejemplos de cómo deberían verse las distintas partes del entregable. Dentro de App vamos a tener el formulario y el respectivo mensaje de error o el cartel( Card) con la información ingresada.

Los estilos no son evaluados, luego de cumplir con el funcionamiento pueden agregarle los estilos que deseen.

En la siguiente hoja pueden ver los ejemplos de un formulario que pregunta sobre el color favorito del usuario. 📌



## Formulario

### Elige un color

ENVIAR



## Mensaje de Error

### Elige un color

ENVIAR

Por favor chequea que la información sea correcta.



## Card con los datos ingresados

### Elige un color

ENVIAR

Hola Lucas!

Sabemos que tu color favorito es:

#37BF21

## **\_ Recomendación de pasos a seguir**

Para ayudar a que puedas organizarte y realizar la entrega a tiempo, te dejamos una propuesta de flujo de trabajo. De todas maneras, puedes organizarte como lo creas más conveniente para lograr el objetivo buscado.

### **Paso 1: crear los componentes.**

En primer lugar, nos ocuparemos de crear nuestros componentes sin pensar en la funcionalidad. Para ello, comenzaremos agregando los dos inputs y el botón que nos permitirá enviar el formulario. Acto seguido, deberás asegurarte que tu componente `<Card/>` reciba la información ingresada como "props" y la muestre en la pantalla.

### **Paso 2: controlar los inputs.**

Ahora que ya tenemos todos los componentes, es momento de controlar los inputs para poder almacenar y actualizar sus valores a medida que se ingresan los datos en el formulario.

### **Paso 3: agregar el event handler para el submit.**

Una vez que sea posible ingresar valores en el formulario, el siguiente paso es poder responder al envío del formulario. Para ello, debemos crear el **event handler** necesario.

### **Paso 4: validaciones.**

Dentro del event handler deberás ocuparte de llevar a cabo las validaciones que se solicitan en la consigna. Recordemos que en caso de que alguna de ellas no arroje un resultado satisfactorio, deberás mostrar el mensaje de error. En este caso, los datos ingresados no deberán renderizarse dentro del componente `<Card/>`.

### **Paso 5: resultado final.**

Si las validaciones son exitosas, debemos ocuparte de que la información ingresada aparezca dentro del componente `<Card/>`. Si tenemos tiempo podemos agregar estilos a los componentes.