

Desafío 2 - Estado de los componentes y eventos

En este desafío validaremos nuestros conocimientos de la unidad: “Estado de los componentes y eventos”.

Lee todo el documento antes de comenzar para asegurarte de tener el máximo puntaje y enfocar bien los esfuerzos.

El desarrollo de este desafío se realiza **en parejas**, por lo que deberán trabajar de manera colaborativa usando GitHub, y además, cada uno de los integrantes deberá subir el link del repositorio en el LMS.

Descripción

En este desafío pondremos en práctica los conceptos aprendidos para controlar los estados de nuestros componentes en React.

Para ello, un cliente nos solicita una aplicación en React que permita al usuario validar un formulario de registro utilizando los eventos `onChange` y `onSubmit`, similar al que se muestra en la siguiente imagen de referencia:

Una interfaz de usuario para crear una cuenta. El título es "Crea una cuenta". Debajo hay tres íconos circulares para Facebook, GitHub e LinkedIn. Luego, el texto "O usa tu email para registrarte". A continuación, cuatro campos de entrada: "Nombre", "tuemail@ejemplo.com", "Contraseña" y "Confirma tu contraseña". Un botón verde "Registrarse" y un mensaje de error en un recuadro rojo "Completa todos los campos !".

Imagen 1. Imagen de referencia desafío 2.

Fuente: Desafío Latam.

Material de estudio asociado

- Guía 2 - Estado de los componentes y eventos

Requerimientos

1. Crea al menos 4 componentes, `Registro`, `SocialButton`, `Formulario` y `Alert`, donde el componente `Registro` importa todos los demás componentes.
(1 Punto)
2. En el componente `Registro`, se debe importar el componente `SocialButton`, para mostrar los botones de redes sociales.
 - El icono se debe enviar mediante props desde el componente `Registro`.(1 Punto)
3. El componente `Formulario` debe tener un formulario con un campo para el nombre, email, contraseña y otro para la confirmación de la contraseña del usuario.
 - Además, deberás almacenar los estados de los inputs a través de `useState` y relacionarlos con el evento `onChange`.(2 Puntos)
4. Dentro del mismo componente `Formulario` se debe incluir un botón para procesar el formulario.
 - El botón debe validar mediante el evento `onSubmit` que el email tenga su formato correcto, y que las contraseñas sean iguales.(1 Punto)
5. Luego de presionar el botón de iniciar sesión, el componente `Alert` debe mostrar un mensaje indicando al usuario si el registro fue exitoso o si los campos se encuentran vacíos.
 - Utiliza los props para mostrar un mensaje de error o de éxito.
 - Opcionalmente, puedes utilizar otro props para cambiar el color de la alerta (`success` o `danger`) según corresponda.
 - Como recordatorio, debes tener un state en el componente App que almacene el mensaje de error o de éxito, y pasar mediante props al componente `Alert` y `Login`.(2 Puntos)
6. Replicar el estilo de la aplicación usando Bootstrap, ya sea mediante `react-bootstrap` o su [CDN](#).
 - El color de fondo es opcional.
 - Los íconos utilizados son de [Font Awesome](#)(1 Punto)

7. Crear un repositorio en GitHub para alojar el proyecto y trabajar de manera colaborativa.
 - Cada integrante del equipo deberá realizar al menos dos commits con distintas funcionalidades.

(2 Puntos)



¡Mucho éxito!

Consideraciones y recomendaciones

- Los botones de redes sociales no necesariamente deben ser funcionales, pero se debe replicar su estilo.
- Puedes validar los campos de contraseña mediante expresiones regulares para solicitar un mínimo de caracteres o que requiera llevar números y símbolos para más seguridad.