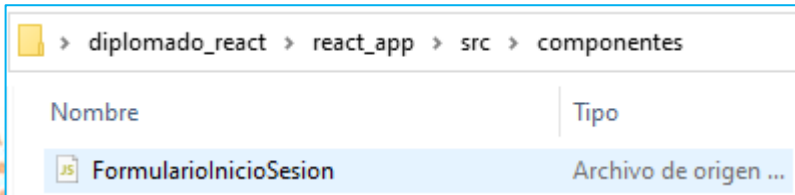


Formularios en React

Vamos a complementar nuestro "Inicio de Sesión" con un formulario usando React. Para ello haremos un archivo nuevo dentro de la carpeta "componentes" llamado FormularioInicioSesion.js



Luego, dentro del archivo creado, importamos useState, creamos el componente, hacemos un formulario con dos párrafos, dos campos de entrada, un botón de envío y lo exportamos:

```
1  import React, {useState} from 'react';
2  const FormularioInicioSesion = () => {
3    return (
4      <form action="">
5        <p>Usuario:</p>
6        <p>Contraseña:</p>
7        <div>
8          <label htmlFor="usuario">Usuario</label>
9          <input type="text" name="usuario" id="usuario"/>
10        </div>
11        <div>
12          <label htmlFor="clave">Clave</label>
13          <input type="password" name="clave" id="clave"/>
14        </div>
15        <button type="submit">Iniciar Sesión</button>
16      </form>
17    );
18  }
19  export default FormularioInicioSesion;
```

Formularios en React

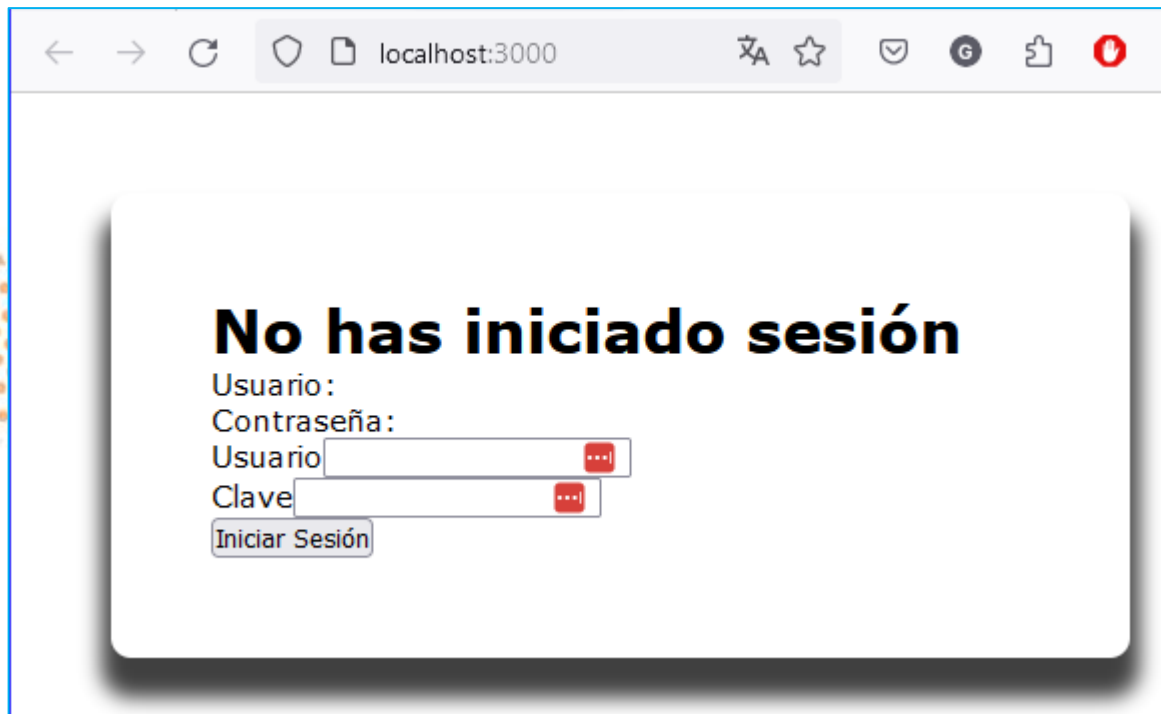
Seguidamente dentro del index.js importaremos el componente, cambiaremos el estado por defecto de la sesión a "false", mostraremos el componente del formulario debajo del encabezado que indica que no se ha iniciado sesión y dejaremos comentado el componente botón que habíamos creado anteriormente, para que no haya dos botones:

```
6 import FormularioInicioSesion from './componentes/FormularioInicioSesion';
7
8 const App = () =>{
9   const [sesion, cambiarEstadoSesion] = useState(false);
```

```
18   <>
19     <h1>No has iniciado sesión</h1>
20     <FormularioInicioSesion/>
21     {/*<Boton verde anchocompleto onClick={() => cambiarEstadoSesion(true)}>Iniciar Sesión</Boton>*/}
22   </>
```

Formularios en React

Una vez realizado todo lo anterior nuestra aplicación se verá así:



← → ↻ 🛡️ 📄 localhost:3000 🔍 ☆ 📧 📁 🔴

No has iniciado sesión

Usuario:
Contraseña:

Usuario

Clave

Iniciar Sesión

A continuación con nuestro formulario fabricado, vamos a practicar el uso de useState, haciendo que al lado de los párrafos salga todo lo que vayamos escribiendo dentro de los campos de entrada de nuestro formulario

Formularios en React

Para lograr el objetivo anteriormente dicho, debemos ingresar en el archivo del componente FormularioInicioSesion y primeramente, crear dos (2) estados con valor vacío: El estado que guardará lo que se escribe en el input de usuario y el estado que guardará lo que se escribe en el input de clave:

```
2  const FormularioInicioSesion = () => {  
3    const [usuario, cambiarUsuario] = useState('');  
4    const [clave, cambiarClave] = useState('');
```

Seguidamente, dentro del formulario, al lado de nuestros párrafos, vamos a mostrar el valor de nuestros dos (2) estados:

```
7    <p>Usuario: {usuario}</p>  
8    <p>Contraseña: {clave}</p>
```

Ahora dentro de cada uno de los inputs de entrada, agregaremos un value igual al valor del estado correspondiente:

```
10   <label htmlFor="usuario">Usuario</label>  
11   <input type="text" name="usuario" id="usuario" value={usuario}/>  
14   <label htmlFor="clave">Clave</label>  
15   <input type="password" name="clave" id="clave" value={clave}/>
```

Formularios en React

Hasta este punto, lo único que hemos hecho es que el input tenga el mismo valor de nuestro estado, sin embargo, cómo nuestro estado es igual a vacío, nuestros inputs también quedan con ese valor y eso no es lo que queremos.

En React, el atributo `onChange` se utiliza para manejar eventos de cambio en elementos de entrada, como campos de texto y casillas de verificación. Cuando ocurre un cambio en el valor de un elemento, se invoca una función que se pasa como valor del atributo `onChange`.

Formularios en React

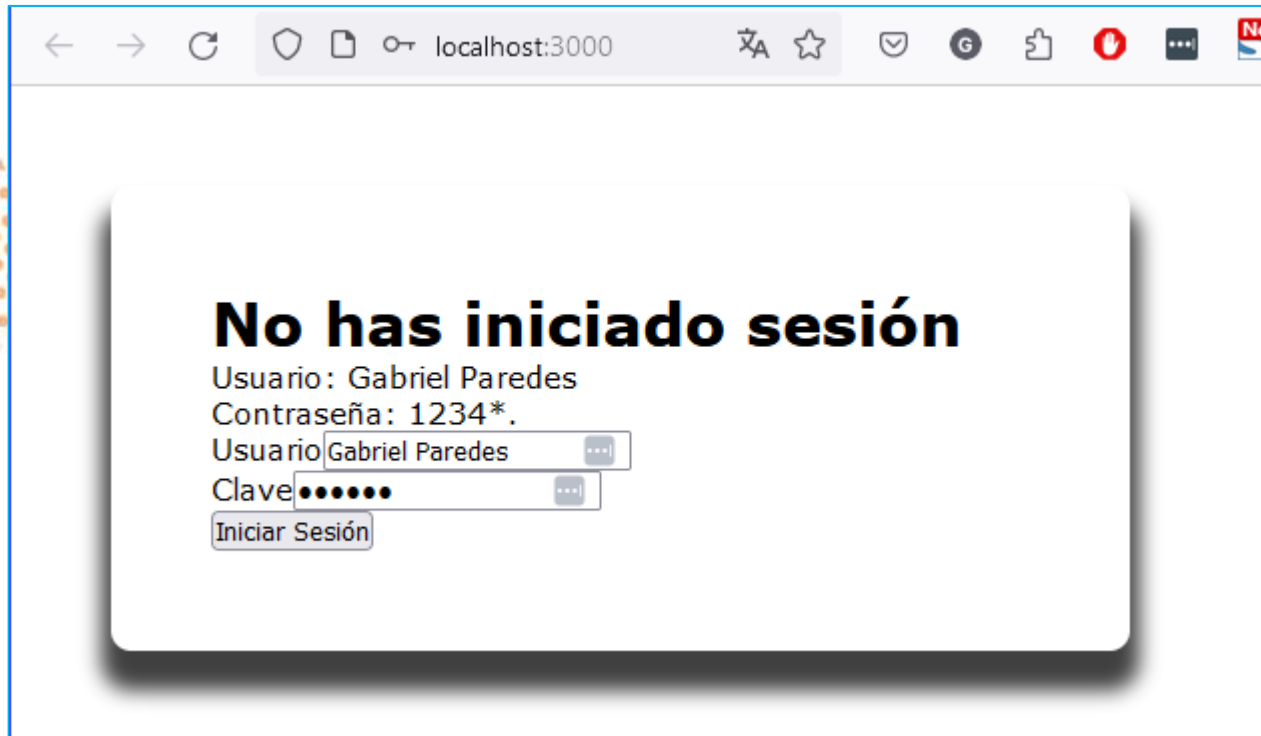
Para poder otorgar a nuestros inputs el valor de lo que escribimos en ellos, debemos agregar a cada uno un `onChange` y dentro de la función invocada por el atributo `onChange`, accederemos al evento que desencadenó el cambio utilizando el parámetro de evento, haremos que esa función ejecute la función de cambio del estado correspondiente y como nuevo valor para ese estado enviaremos el valor del objetivo del evento, es decir, lo que se escribió en el input:

```
9      <div>
10        <label htmlFor="usuario">Usuario</label>
11        <input
12          type="text"
13          name="usuario"
14          id="usuario"
15          value={usuario}
16          onChange={(evento) => {
17            cambiarUsuario(evento.target.value)
18          }}
19        />
20      </div>
```

```
21      <div>
22        <label htmlFor="clave">Clave</label>
23        <input
24          type="password"
25          name="clave"
26          id="clave"
27          value={clave}
28          onChange={(evento) => {
29            cambiarClave(evento.target.value)
30          }}
31        />
32      </div>
```

Formularios en React

De esta forma, todo lo que vamos escribiendo en nuestros campos de entrada, se va colocando al lado de nuestros párrafos, gracias al uso de los formularios y useState con React:



A screenshot of a web browser window displaying a login form. The browser's address bar shows 'localhost:3000'. The form is centered on the page and has a white background with a subtle shadow. It features the heading 'No has iniciado sesión' in bold black text. Below the heading, there are two lines of text: 'Usuario: Gabriel Paredes' and 'Contraseña: 1234*.'. Underneath these are two input fields: the first is labeled 'Usuario' and contains the text 'Gabriel Paredes'; the second is labeled 'Clave' and contains seven dots, indicating a password. To the right of each input field is a small square button with three dots. At the bottom of the form is a button labeled 'Iniciar Sesión'.

Formularios en React

Una vez cumplido nuestro primero objetivo, vamos a optimizar un poco nuestro código haciendo una función genérica llamada CambioEstado que servirá para cambiar el estado del usuario y de la clave en función del input desde el cuál sea invocada:

```
3     const [usuario, cambiarUsuario] = useState('');
4     const [clave, cambiarClave] = useState('');
5     const CambioEstado = (evento) => {
6         if(evento.target.name === "usuario"){
7             cambiarUsuario(evento.target.value)
8         } else if(evento.target.name === "clave"){
9             cambiarClave(evento.target.value)
10        }
11    }
```

De esta forma, dentro de nuestros inputs, en el valor del onChange llamaremos a esta función genérica que se encargará de llamar a la función de Cambio de Estado correspondiente y le asignará el nuevo valor según lo escrito en el input desde el cuál fue llamada:

```
17     <label htmlFor="usuario">Usuario</label>
18     <input type="text" name="usuario" id="usuario" value={usuario} onChange={CambioEstado}/>

21     <label htmlFor="clave">Clave</label>
22     <input type="password" name="clave" id="clave" value={clave} onChange={CambioEstado}/>
```


Validación de Formularios en React

Ahora vamos a ejecutar una validación cuándo el usuario presione el botón de "Iniciar Sesión", para ello, debemos colocar el atributo onSubmit en la etiqueta de apertura del formulario, de esta forma invocaremos una función para manejar el evento de envío del formulario que será ejecutada cuando se envía el formulario, ya sea haciendo clic en el botón de enviar o presionando la tecla Enter dentro de un campo de entrada.

```
16      <form action="" onSubmit={validar}>
```

Dentro de la función llamada por el onSubmit, accederemos al evento como parámetro y evitaremos que el formulario se envíe automáticamente y se recargue la página, llamando al método preventDefault() en el evento.

```
12      const validar = (evento) => {  
13          evento.preventDefault();  
14      }
```

Validación de Formularios en React

Seguidamente, dentro de la función "validar" colocamos un condicional en el cuál se comparará la información ingresada en el input con el "usuario" y "clave", en caso de ser incorrectos saldrá un aviso indicándolo y se limpiará el formulario.

```
12  const validar = (evento) => {  
13    evento.preventDefault();  
14    if(usuario === "clave" && clave === "clave"){  
15  
16    }else{  
17      alert("Datos Incorrectos");  
18      cambiarUsuario("");  
19      cambiarClave("");  
20    }  
21  }
```

En caso de que los datos sean correctos, lo que haremos a continuación es cambiar el estado de la sesión que está en el index.js, por lo cuál, debemos aplicar el uso de propiedades para poder enviar el estado de la sesión hasta el archivo de nuestro componente FormularioInicioSesion.js

Validación de Formularios en React

Para comenzar a cumplir con el caso correcto de nuestro condicional, lo primero que haremos entonces es dirigirnos al index.js y estando dentro del mismo, en el llamado de nuestro componente `<FormularioInicioSesion/>` vamos a colocar la propiedad "cambiarEstadoSesion" y la igualaremos a la función "cambiarEstadoSesion":

```
18  <>
19  <h1>No has iniciado sesión</h1>
20  <FormularioInicioSesion cambiarEstadoSesion={cambiarEstadoSesion}/>
```

De esta forma estamos enviando la función de cambio de estado de la sesión hasta nuestro componente de formulario y ahora, recibiremos las propiedades escribiendo "props" dentro de los parámetros del componente:

```
src > componentes > JS FormularioInicioSesion.js > [e] FormularioInicioSesion
1  import React, {useState} from 'react';
2  const FormularioInicioSesion = (props) => {
```

Validación de Formularios en React

De esta manera, ya teniendo las propiedades recibidas en el componente, podemos acceder a la propiedad de cambio del estado de la sesión y cambiarla a verdadero en caso de que los datos del usuario sean correctos:

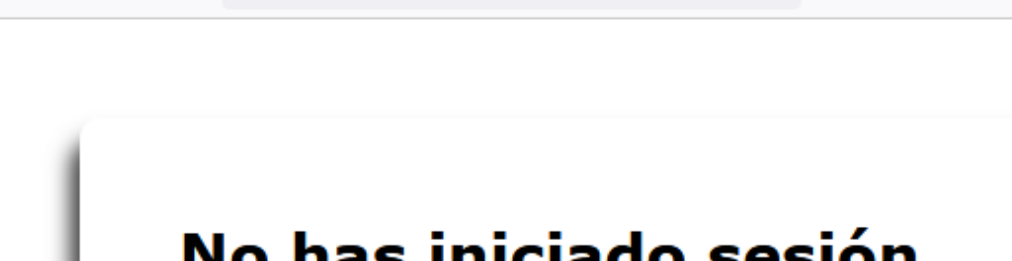
```
12     const validar = (evento) => {  
13         evento.preventDefault();  
14         if(usuario === "clase" && clave === "clase"){  
15             props.cambiarEstadoSesion(true);  
16         }else{  
17             alert("Datos Incorrectos");  
18             cambiarUsuario("");  
19             cambiarClave("");  
20         }  
21     }
```

Una vez hecho esto, en la parte interna de nuestro formulario, borraremos los dos párrafos que hicimos anteriormente para solo dejar nuestro ejemplo de login:

```
23     <form action="" onSubmit={validar}>  
24         <div>  
25             <label htmlFor="usuario">Usuario</label>  
26             <input type="text" name="usuario" id="usuario" value={usuario} onChange={CambiarEstado}/>  
27         </div>
```

Validación de Formularios en React

Como resultado final tendremos nuestro ejemplo de login funcionando efectivamente:




← → ↻ 🔒 📄 localhost:3000 🔍 ☆ 📧 1 🏠

No has iniciado sesión

Usuario

Clave



← → ↻ 🛡️ 📄 localhost:3000 🗑️ ☆ 📧 Ⓜ️ 1 📺 📺 📺

Proyecto 1

[Cerrar Sesión](#)

¡Hola Gabriel!

¡Hola Alejandro!

Te doy la Bienvenida a esta Experiencia de Aprendizaje

Estamos en clase del Diplomado de Programación

Estos son los profesores:

- Pedro
- Maria Eugenia
- Vidermid
- Gabriel

[Cerrar Sesión](#)