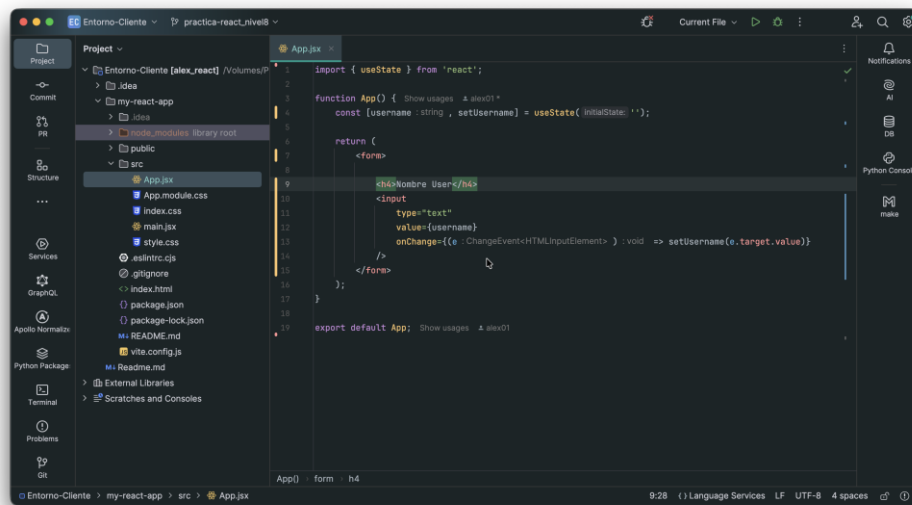
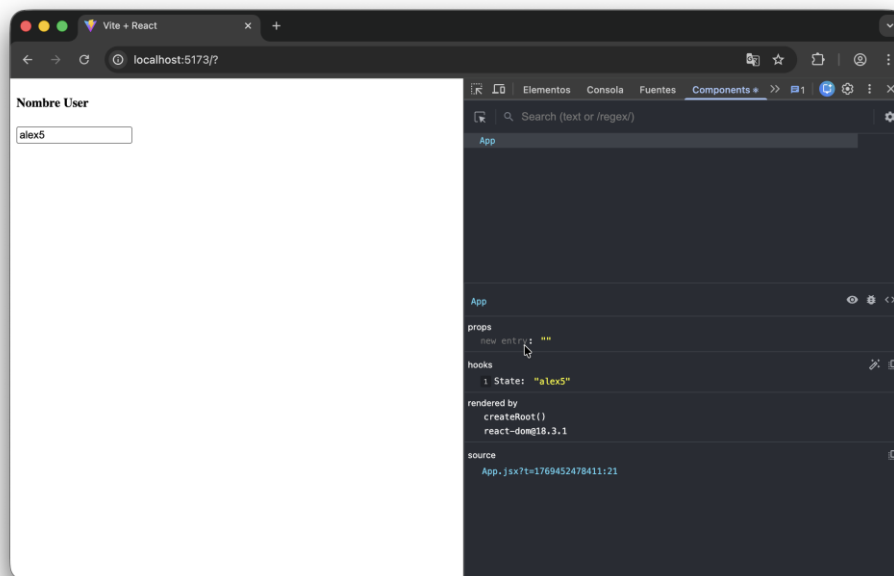


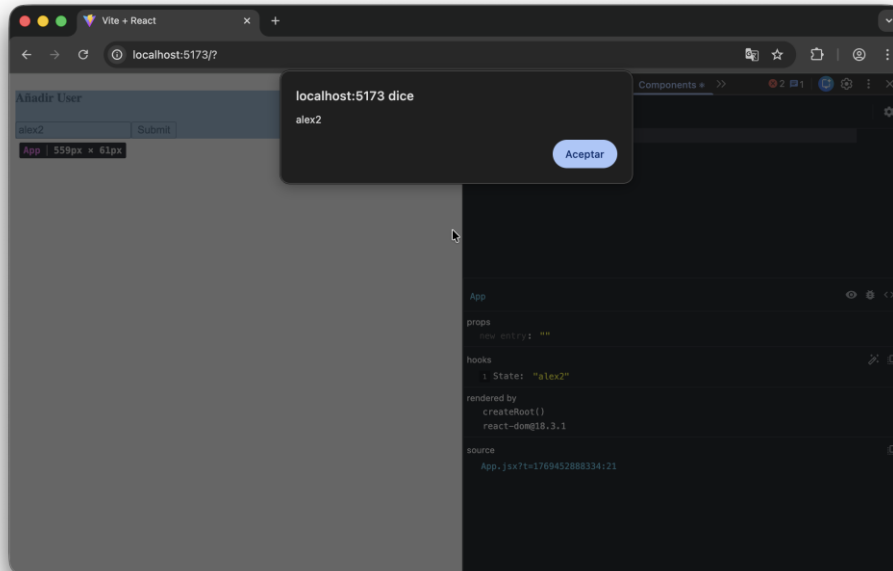
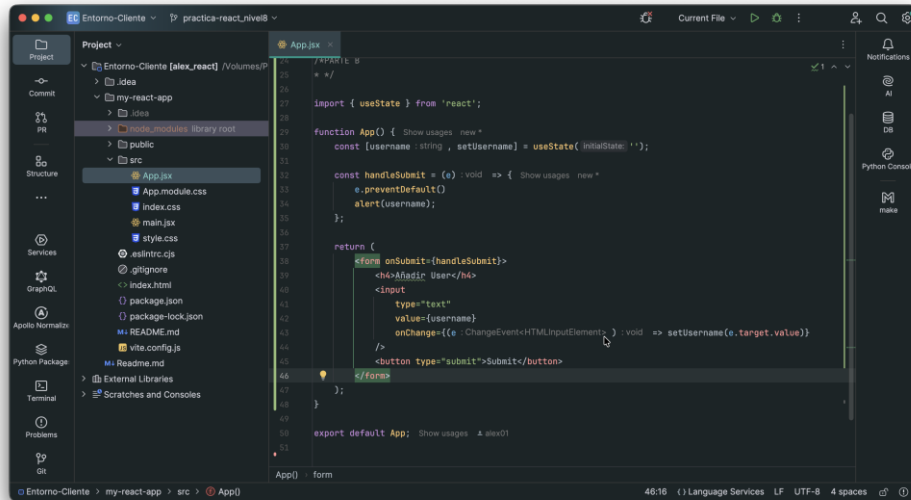
## 1- Parte A: Input controlado (Controlled Component)



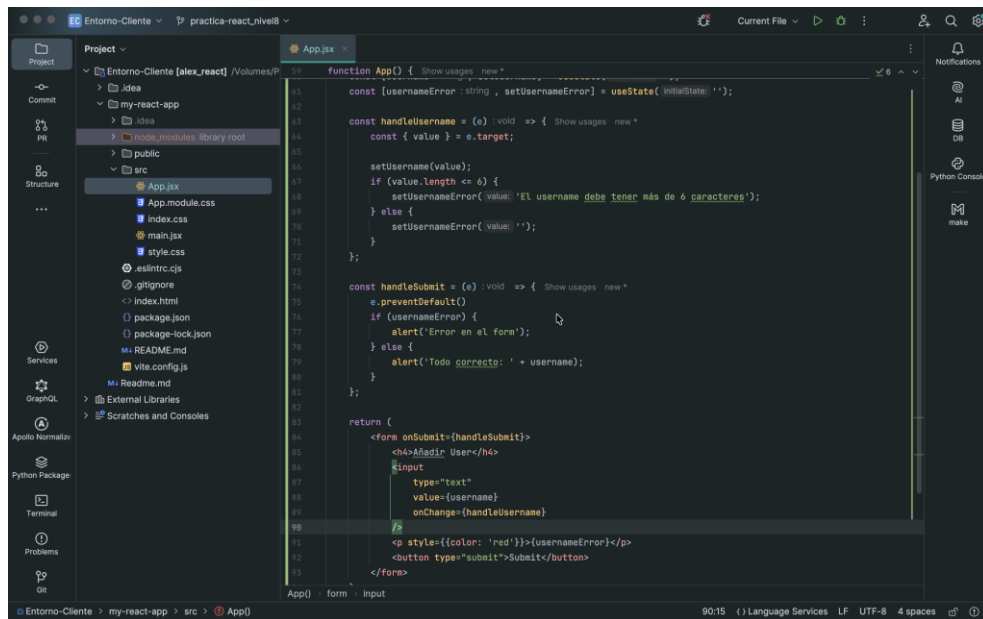
```
1 import { useState } from 'react';
2
3 function App() {
4   const [username, setUsername] = useState('');
5
6   return (
7     <form>
8       <h3>Nombre User</h3>
9       <input
10         type="text"
11         value={username}
12         onChange={(e) => setUsername(e.target.value)}
13       />
14     </form>
15   );
16 }
17
18 export default App;
```



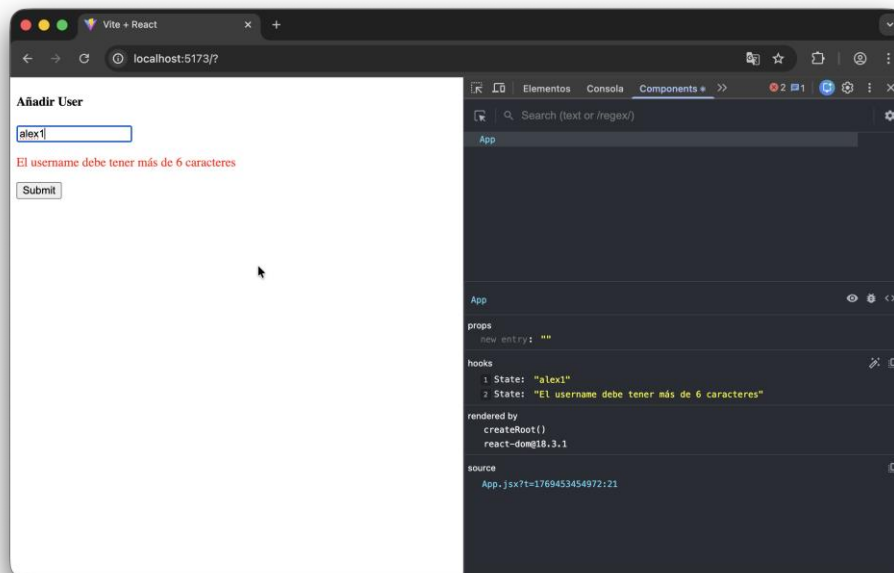
## 2- Parte B: Envío del formulario (onSubmit)



### 3- Parte C: Validación básica



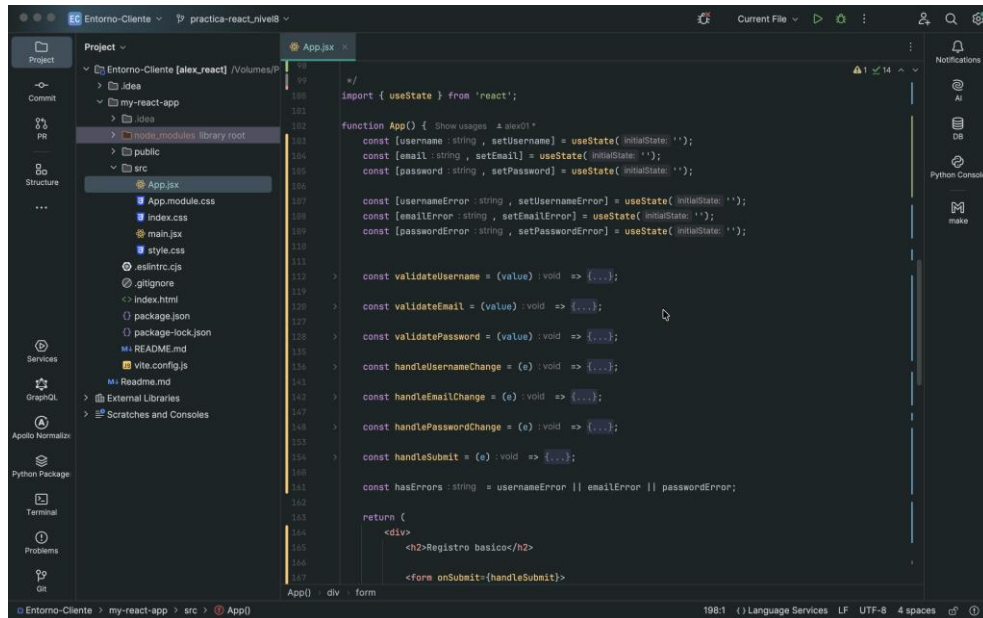
```
function App() {  
  const [usernameError, setUsernameError] = useState('');  
  
  const handleUsername = (e) => {  
    const { value } = e.target;  
    setUsername(value);  
    if (value.length <= 6) {  
      setUsernameError('El username debe tener más de 6 caracteres');  
    } else {  
      setUsernameError('');  
    }  
  };  
  
  const handleSubmit = (e) => {  
    e.preventDefault();  
    if (usernameError) {  
      alert('Error en el form');  
    } else {  
      alert('Todo correcto: ' + username);  
    }  
  };  
  
  return (  
    <form onSubmit={handleSubmit}>  
      <h4>Añadir User</h4>  
      <input  
        type="text"  
        value={username}  
        onChange={handleUsername}/>  
      <p style={{color: 'red'}}>{usernameError}</p>  
      <button type="submit">Submit</button>  
    </form>  
  );  
}
```



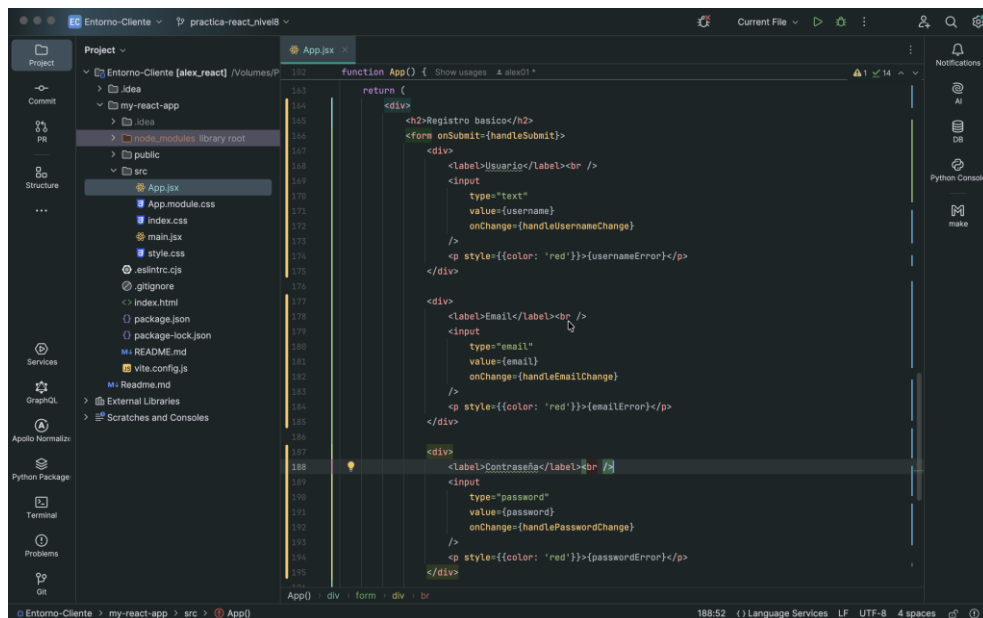
#### 4- Mini Reto

- Con el método del ejercicio anterior, he replicado lo mismo, pero con las validaciones necesarias y he añadido las variables de errores para cada error

- También he replicado el método que me muestra los errores mientras el usuario va rellenando los campos
- Para luego ser utilizados de forma independiente y validar que todo el formulario este correcto, en el caso de que no el botón de registrar se desactiva y no permite enviarlo



```
98
99
100 import { useState } from 'react';
101
102 function App() {
103   const [username, setUsername] = useState('');
104   const [email, setEmail] = useState('');
105   const [password, setPassword] = useState('');
106
107   const [usernameError, setUsernameError] = useState('');
108   const [emailError, setEmailError] = useState('');
109   const [passwordError, setPasswordError] = useState('');
110
111   const validateUsername = (value) => {
112     // ...
113   };
114   const validateEmail = (value) => {
115     // ...
116   };
117   const validatePassword = (value) => {
118     // ...
119   };
120   const handleUsernameChange = (e) => {
121     // ...
122   };
123   const handleEmailChange = (e) => {
124     // ...
125   };
126   const handlePasswordChange = (e) => {
127     // ...
128   };
129   const handleSubmit = (e) => {
130     // ...
131   };
132
133   const hasErrors = usernameError || emailError || passwordError;
134
135   return (
136     <div>
137       <h2>Registro basico</h2>
138       <form onSubmit={handleSubmit}>
139         <div>
140           <input type="text" value={username} onChange={handleUsernameChange} />
141           <p style={{color: 'red'}}>{usernameError}</p>
142         </div>
143         <div>
144           <input type="email" value={email} onChange={handleEmailChange} />
145           <p style={{color: 'red'}}>{emailError}</p>
146         </div>
147         <div>
148           <input type="password" value={password} onChange={handlePasswordChange} />
149           <p style={{color: 'red'}}>{passwordError}</p>
150         </div>
151       </form>
152     </div>
153   );
154 }
```



```
155
156
157 return (
158   <div>
159     <h2>Registro basico</h2>
160     <form onSubmit={handleSubmit}>
161       <div>
162         <label>Usuario</label><br />
163         <input
164           type="text"
165           value={username}
166           onChange={handleUsernameChange}
167         />
168         <p style={{color: 'red'}}>{usernameError}</p>
169       </div>
170       <div>
171         <label>Email</label><br />
172         <input
173           type="email"
174           value={email}
175           onChange={handleEmailChange}
176         />
177         <p style={{color: 'red'}}>{emailError}</p>
178       </div>
179       <div>
180         <label>Contraseña</label><br />
181         <input
182           type="password"
183           value={password}
184           onChange={handlePasswordChange}
185         />
186         <p style={{color: 'red'}}>{passwordError}</p>
187       </div>
188     </form>
189   </div>
190 );
191 }
```

- Resultado

