

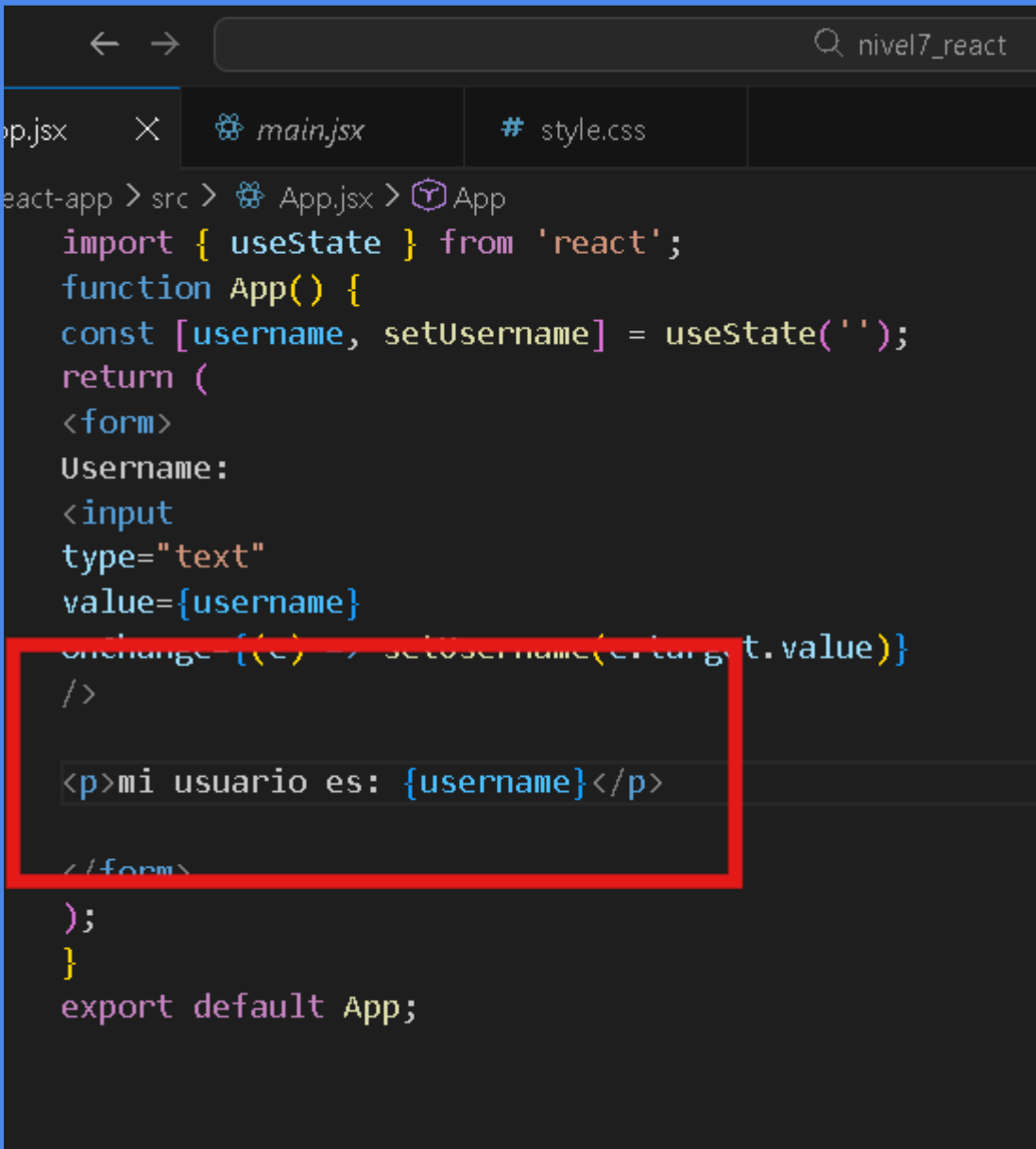
Nivel 8 React

DAW2-DWEC

Índice

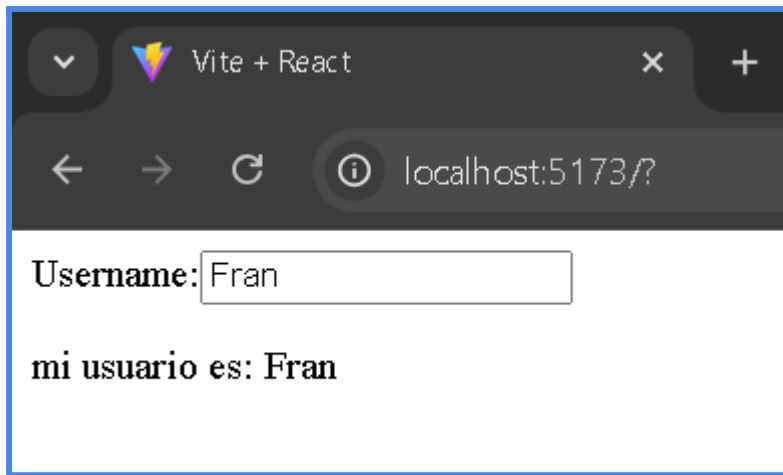
Parte A: Input controlado (Controlled Component)	2
Parte B: Envío del formulario (onSubmit)	4
Parte C: Validación básica	6
Parte D: Código completo de referencia	10
Miniretillo (subimos el nivel chavales)	11

Parte A: Input controlado (Controlled Component)



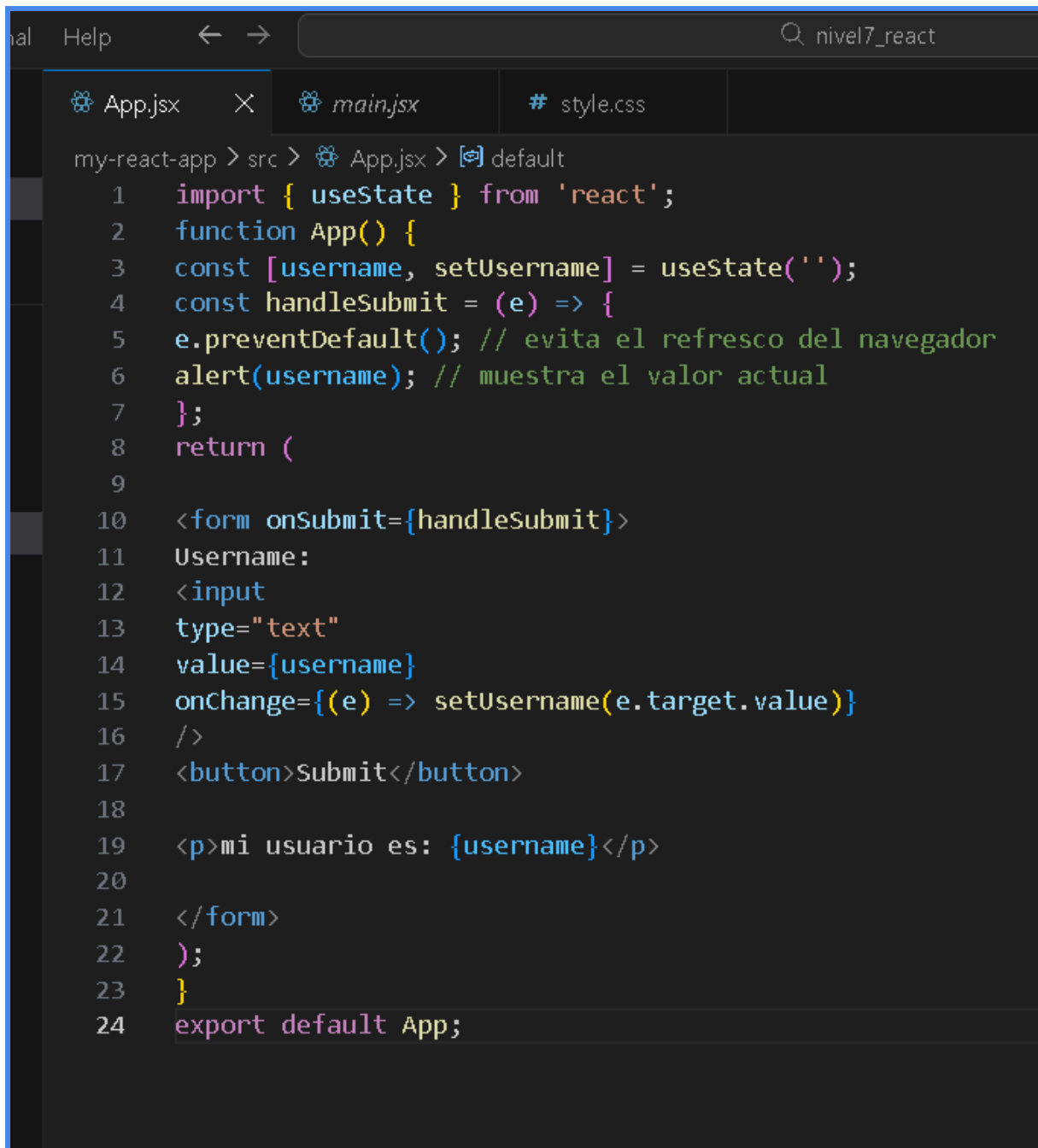
```
App.jsx  X  main.jsx  # style.css
react-app > src > App.jsx > App
import { useState } from 'react';
function App() {
  const [username, setUsername] = useState('');
  return (
    <form>
      Username:
      <input
        type="text"
        value={username}
        onChange={(e) => setUsername(e.target.value)}
      />
      <p>mi usuario es: {username}</p>
    </form>
  );
}
export default App;
```

“Figura 01, mostrar el username del input, elaboración propia”



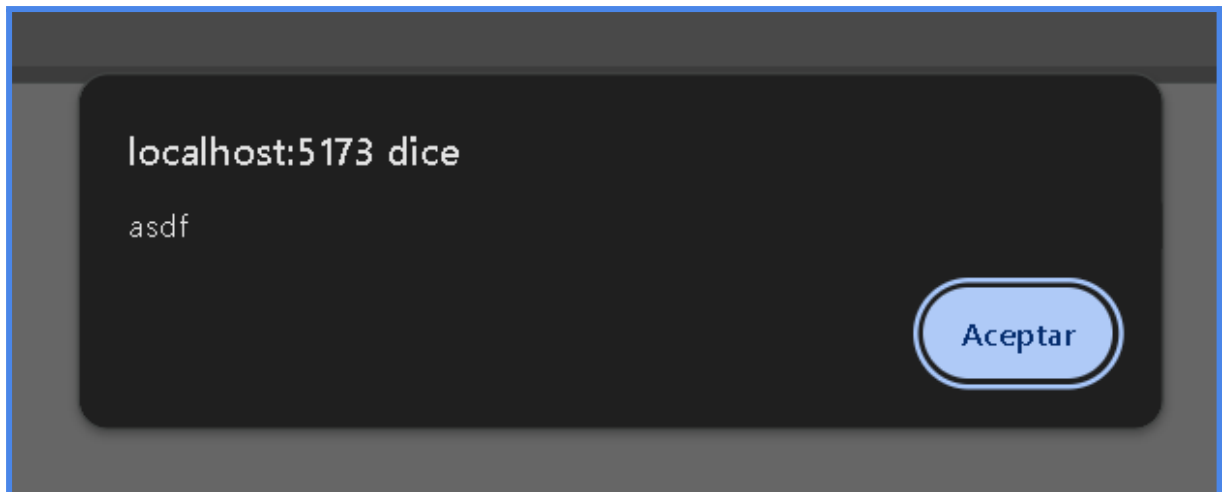
“Figura 02, muestra en pantalla, elaboración propia”

Parte B: Envío del formulario (onSubmit)



```
my-react-app > src > App.jsx > default
1  import { useState } from 'react';
2  function App() {
3    const [username, setUsername] = useState('');
4    const handleSubmit = (e) => {
5      e.preventDefault(); // evita el refresco del navegador
6      alert(username); // muestra el valor actual
7    };
8    return (
9
10     <form onSubmit={handleSubmit}>
11       Username:
12       <input
13         type="text"
14         value={username}
15         onChange={(e) => setUsername(e.target.value)}
16       />
17       <button>Submit</button>
18
19       <p>mi usuario es: {username}</p>
20
21     </form>
22   );
23 }
24 export default App;
```

“Figura 03, botón para el formulario y handlesubmit, elaboración propia”



“Figura 04, muestra en pantalla, elaboración propia”

Parte C: Validación básica

```
-react-app > src > App.jsx > App > handleSubmit
1  import { useState } from 'react';
2  function App() {
3    const [username, setUsername] = useState('');
4    const [usernameError, setUsernameError] = useState('');
5
6    const handleUsername = (e) => {
7      const { value } = e.target;
```

“Figura 05, constante de usernameError, elaboración propia”

```

react-app > src > App.jsx > App > handleSubmit
import { useState } from 'react';
function App() {
  const [username, setUsername] = useState('');
  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('');
    }
  };
};

```

“Figura 06, funcion handleUsername para que no haya menos de 7 caracteres, elaboración propia”

```

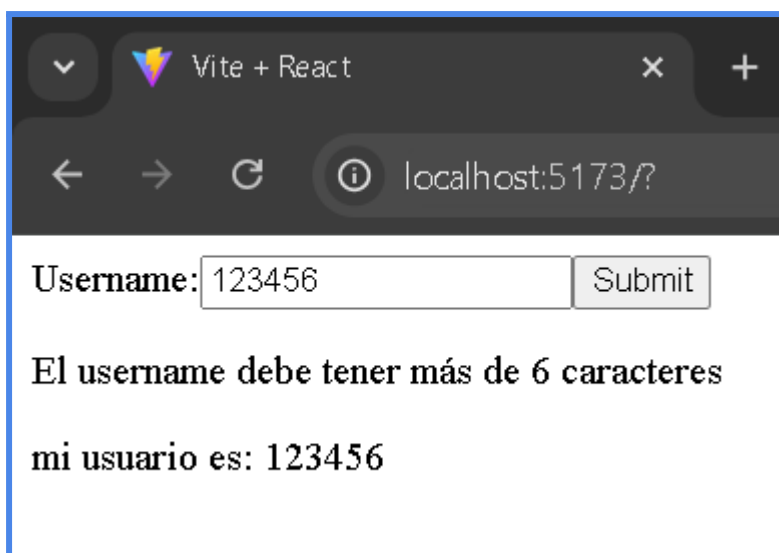
return (
  <form onSubmit={handleSubmit}>
    Username:
    <input
      type="text"
      value={username}
      onChange={handleUsername}
    />
    <button>Submit</button>
  </form>
);

```

“Figura 07, conexión de handleUsername al input y muestra del error, elaboración propia”

```
const handleSubmit = (e) => {
  e.preventDefault(); // evita el refresco del navegador
  if (usernameError) {
    alert("No se puede enviar, el formulario contiene errores");
  }else{
    alert(username);
  }
};
```

“Figura 08, impedimento del envío cuando hay un error, elaboración propia”

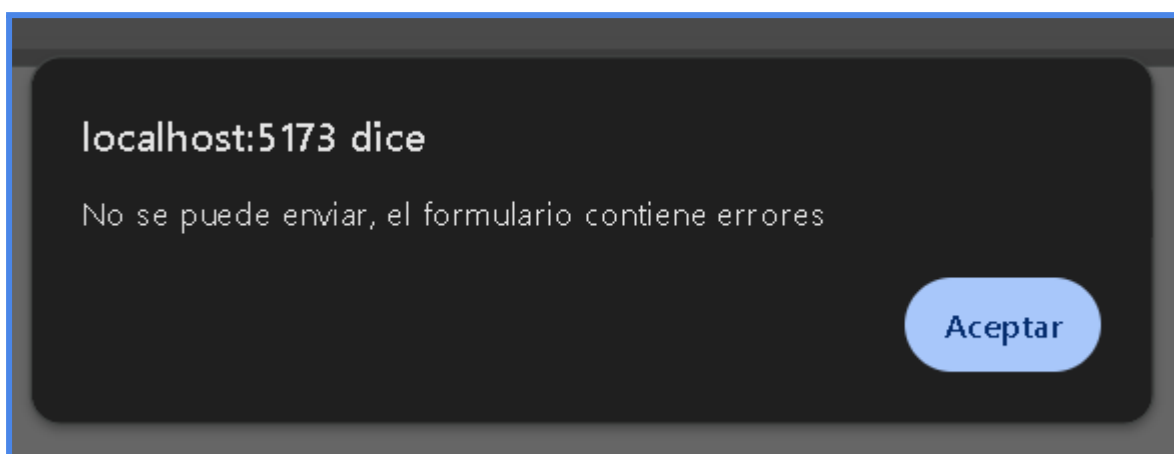


Username:

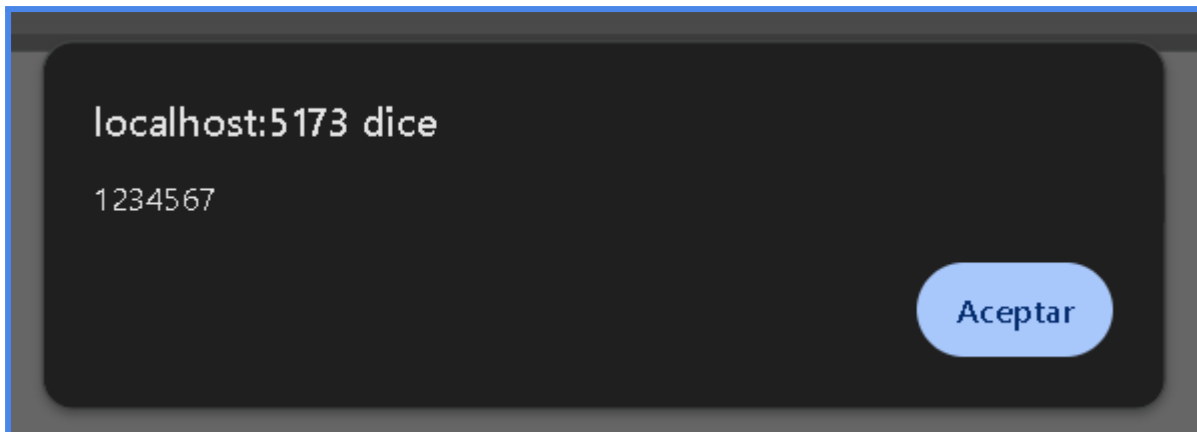
El username debe tener más de 6 caracteres

mi usuario es: 123456

“Figura 09, muestra en pantalla, elaboración propia”



“Figura 10, muestra en pantalla con error, elaboración propia”



“Figura 11, muestra en pantalla sin error, elaboración propia”

Parte D: Código completo de referencia

JavaScript

```
import { useState } from 'react';
function App() {
  const [username, setUsername] = useState('');
  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(); // evita el refresco del navegador
    if (usernameError) {
      alert("No se puede enviar, el formulario contiene errores");
    } else {
      alert(username);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
```



```

Username:


```

- Éste es mi App.jsx entero, en vez de captura prefiero pasarlo con el “modo código” para que sea más sencillo

Miniretillo (subimos el nivel chavales)

```

import { useState } from 'react';
function App() {
  const [username, setUsername] = useState('');
  const [usernameError, setUsernameError] = useState('');
  const [email, setEmail] = useState('');
  const [emailError, setEmailError] = useState('');
  const [emailError1, setEmailError1] = useState('');
  const [password, setPassword] = useState('');
  const [passwordError, setPasswordError] = useState('');

```

```

return (

<form onSubmit={handleSubmit}>

<p>email:
<input
type="text"
value={email}
onChange={handleEmail}
/></p>

```

“Figuras 12 y 13, campos email, elaboración propia”

```

const handleEmail = (e) => {
const { value } = e.target;
setEmail(value);
if (!value.includes('@')) {
  setEmailError('El email debe incluir un @');
}else {
  setEmailError('');
}

if (!value.includes('.')) {
  setEmailError1('El email debe incluir un .');
}else {
  setEmailError1('');
}
};

```

“Figura 14, validaciones, elaboración propia”

A screenshot of a web browser window. The address bar shows 'localhost:5173/'. The page contains a login form with three input fields: 'email:' with the value '1234gmailcom', 'password:' with the value '12345678', and 'Username:' with the value '1234567'. Below the fields is a 'Submit' button. Two red error messages are displayed below the button: 'El email debe incluir un @' and 'El email debe incluir un .'.

“Figura 15, muestra en pantalla, elaboración propia”

```
my-react-app > src > App.jsx > App
1  import { useState } from 'react';
2  function App() {
3    const [username, setUsername] = useState('');
4    const [usernameError, setUsernameError] = useState('');
5    const [email, setEmail] = useState('');
6    const [emailError, setEmailError] = useState('');
7    const [emailError1, setEmailError1] = useState('');
8    const [password, setPassword] = useState('');
9    const [passwordError, setPasswordError] = useState('');
10
```

```

<p>password:
<input
type="text"
value={password}
onChange={handlePassword}
/></p>

```

“Figura 16 y 17, campos password, elaboración propia”

```

const handlePassword = (e) => {
  const { value } = e.target;
  setPassword(value);
  if (value.length < 8) {
    setPasswordError('La contraseña debe tener al menos 8 caracteres');
  } else {
    setPasswordError('');
  }
};

```

“Figura 18, validaciones, elaboración propia”

email: 1234@gmail.com

password: 1234567

Username: 1234567

Submit

La contraseña debe tener al menos 8 caracteres

“Figura 19, muestra en pantalla, elaboración propia”

```

src-app / src / App.jsx / App
import { useState } from 'react';
function App() {
  const [username, setUsername] = useState('');
  const [usernameError, setUsernameError] = useState('');
  const [email, setEmail] = useState('');
  const [emailError, setEmailError] = useState('');
  const [emailError1, setEmailError1] = useState('');
  const [password, setPassword] = useState('');
  const [passwordError, setPasswordError] = useState('');

```

```

<p>Username:
<input
  type="text"
  value={username}
  onChange={handleUsername}
/></p>

```

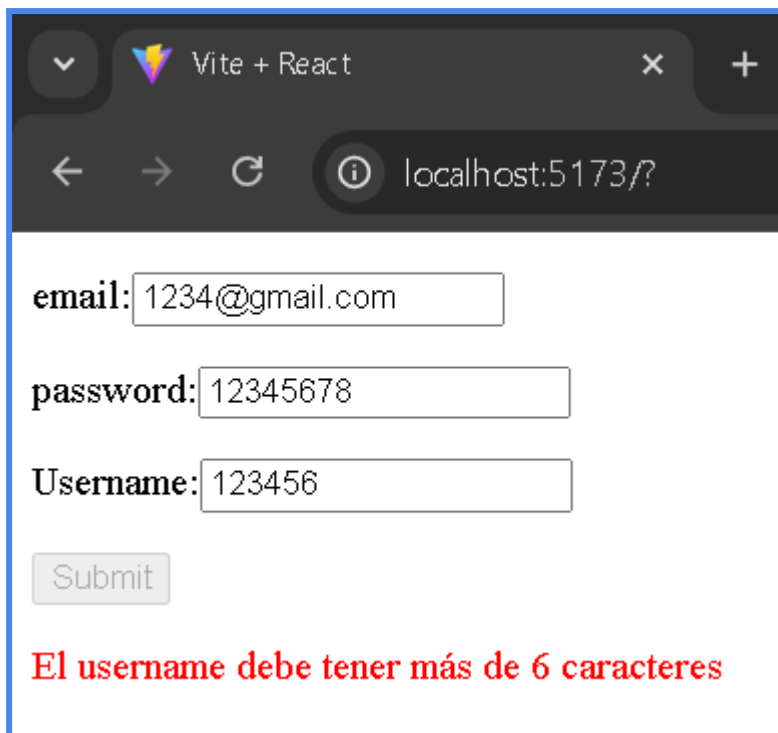
“Figura 20 y 21, campos username, elaboración propia”

```

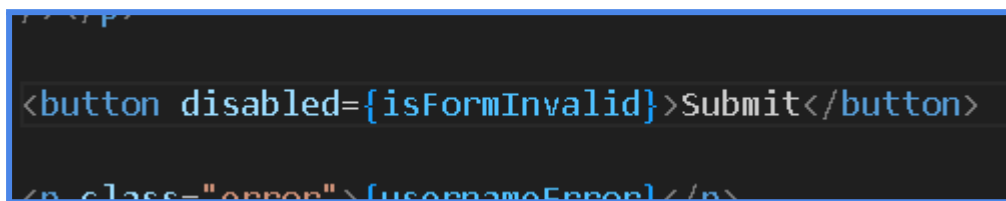
const handleUsername = (e) => {
  const { value } = e.target;
  setUsername(value);
  if (value.length <= 6) {
    setUsernameError('El username debe tener más de 6 caracteres');
  } else {
    setUsernameError('');
  }
};

```

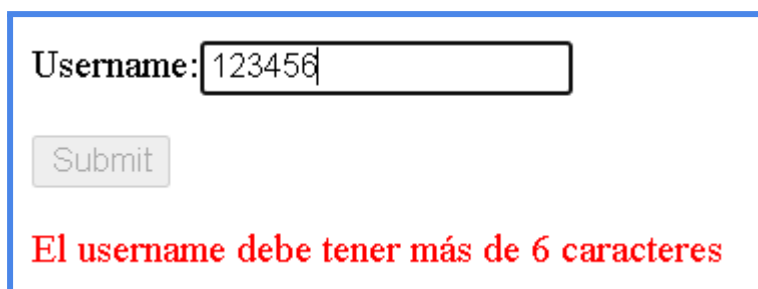
“Figura 22, validaciones, elaboración propia”



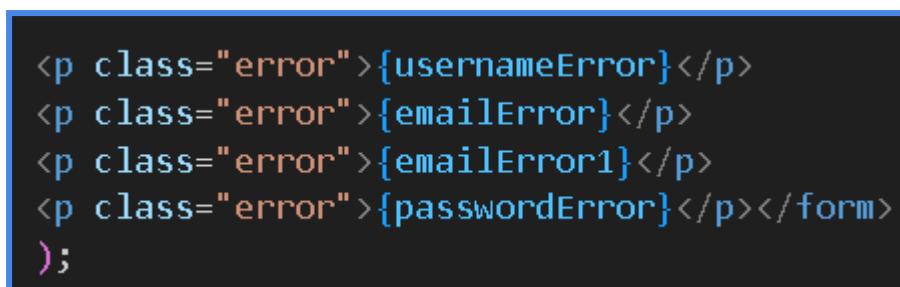
“Figura 23, muestra en pantalla, elaboración propia”



“Figura 24, botón submit si existe algún error, elaboración propia”

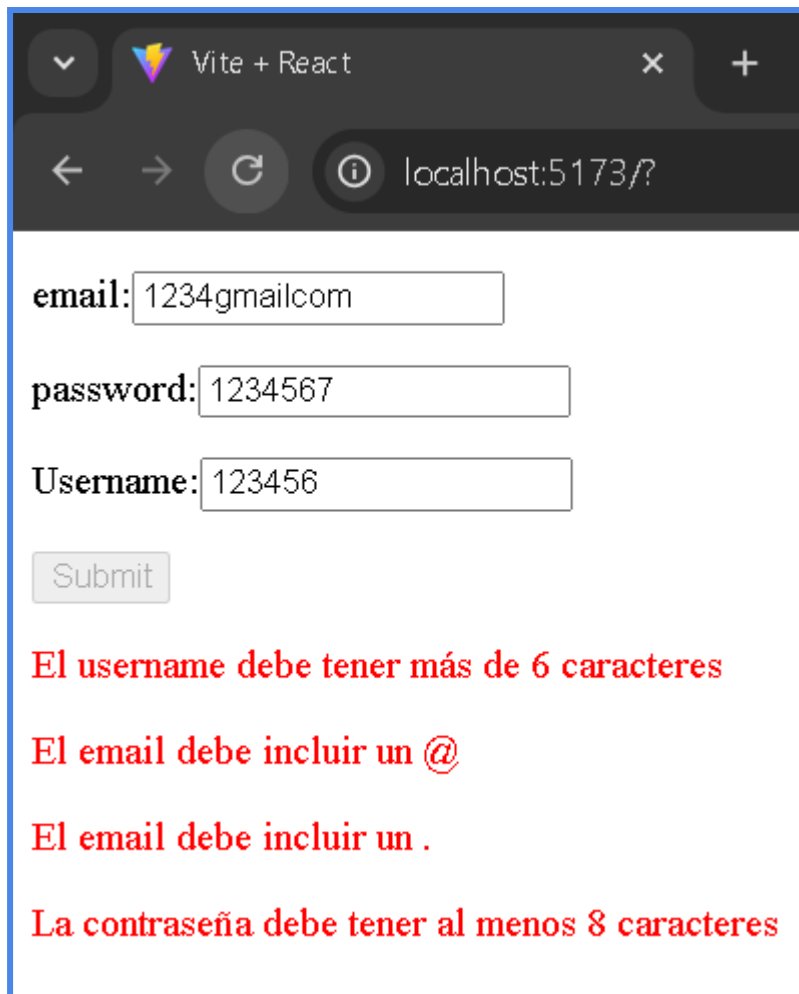


“Figura 25, muestra en pantalla, elaboración propia”



```
ny-react-app > src > # style.css > .err
1  .error {
2    color: red;
3  }
```

“Figura 26 y 27, errores en rojo, elaboración propia”



The screenshot shows a web browser window with the title "Vite + React" and the address bar displaying "localhost:5173/". The page contains a login form with three input fields: "email:" with the value "1234gmailcom", "password:" with the value "1234567", and "Username:" with the value "123456". Below the inputs is a "Submit" button. Underneath the button, there are four lines of red text indicating validation errors: "El username debe tener más de 6 caracteres", "El email debe incluir un @", "El email debe incluir un .", and "La contraseña debe tener al menos 8 caracteres".

“Figura 28, muestra en pantalla, elaboración propia”