

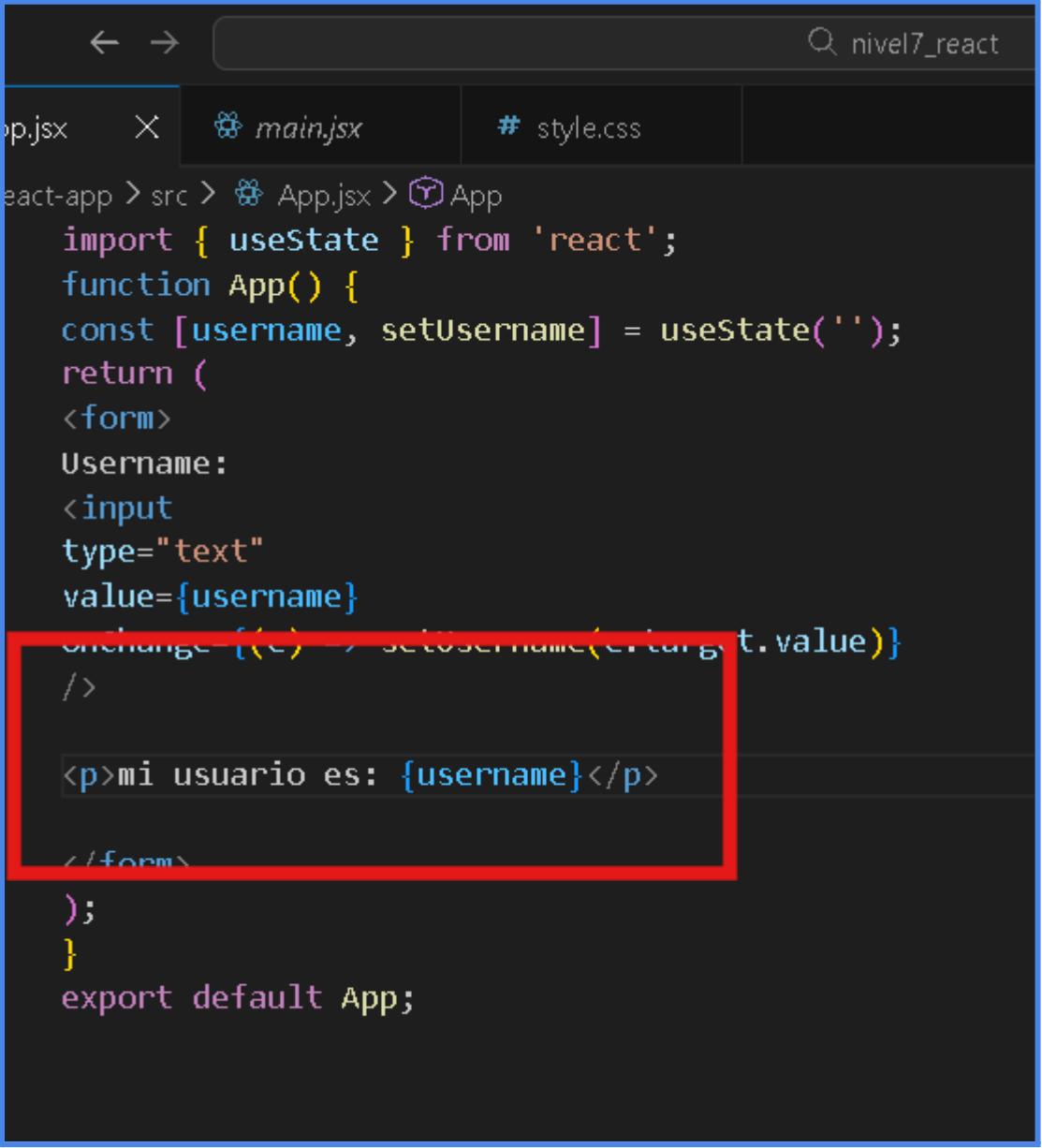
Nivel 8 React

DAW2-DWEC

Índice

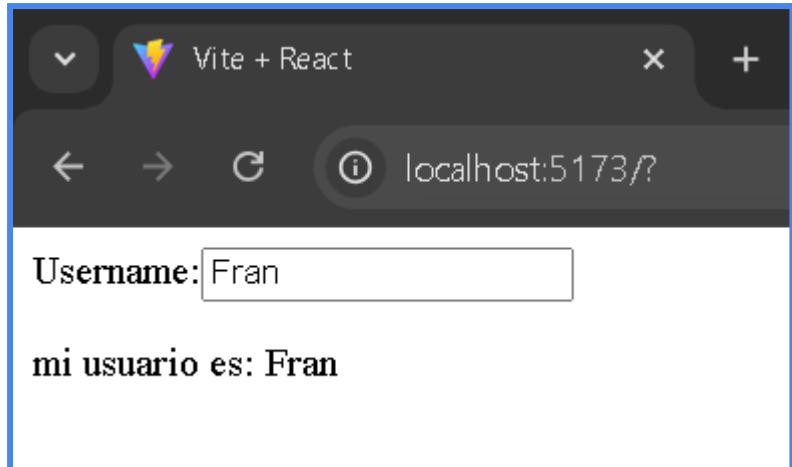
Parte A: Input controlado (Controlled Component).....	2
Parte B: Envío del formulario (onSubmit).....	4
Parte C: Validación básica.....	6
Parte D: Código completo de referencia.....	10
Miniretillo (subimos el nivel chavales).....	11

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.currentTarget.value)} />
      <p>mi usuario es: {username}</p>
    </form>
  );
}
export default App;
```

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



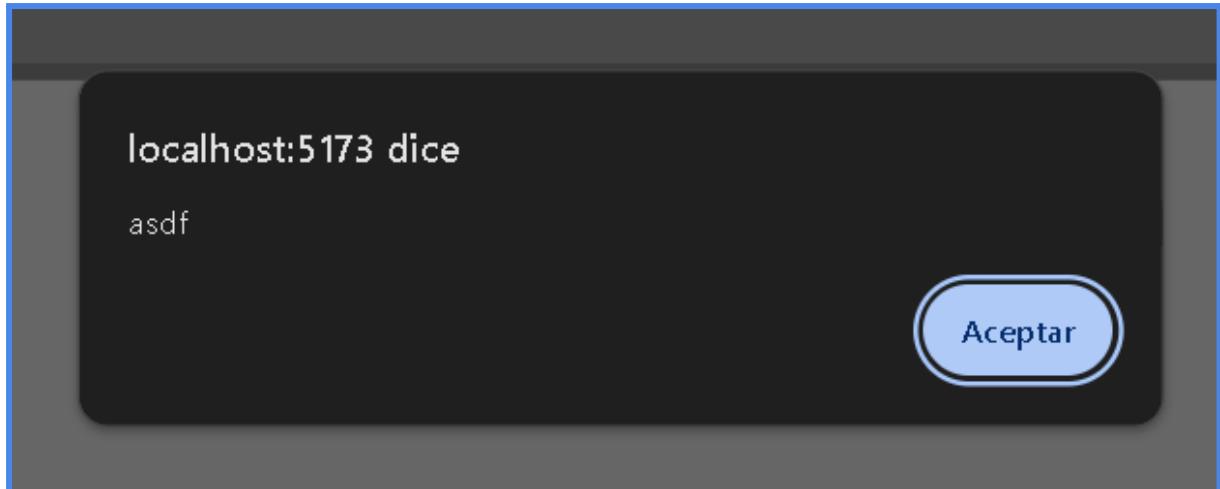
"Figura 02, muestra en pantalla, elaboración propia"

Parte B: Envío del formulario (onSubmit)

The screenshot shows a code editor window with the title bar "nive17_react". The tabs at the top are "App.jsx" (selected), "main.jsx", and "style.css". The code in "App.jsx" is as follows:

```
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('');
    }
  };
}
```

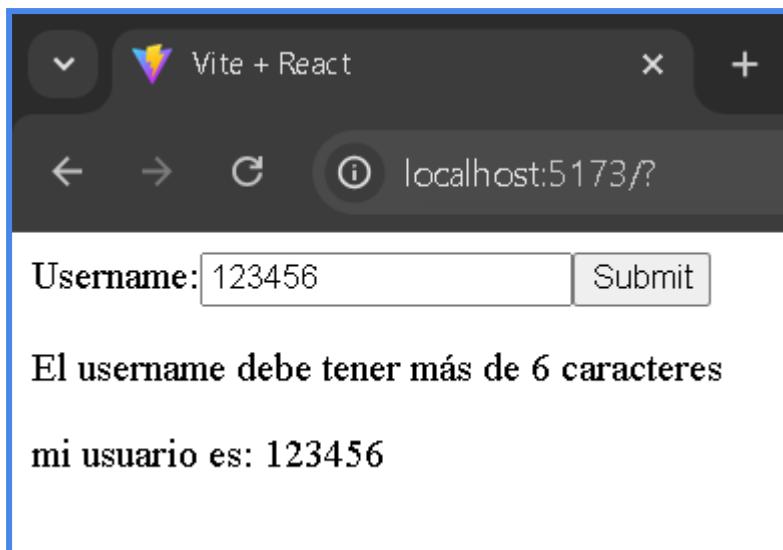
“Figura 06, función handleUsername para que no haya menos de 7 caracteres, elaboración propia”

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

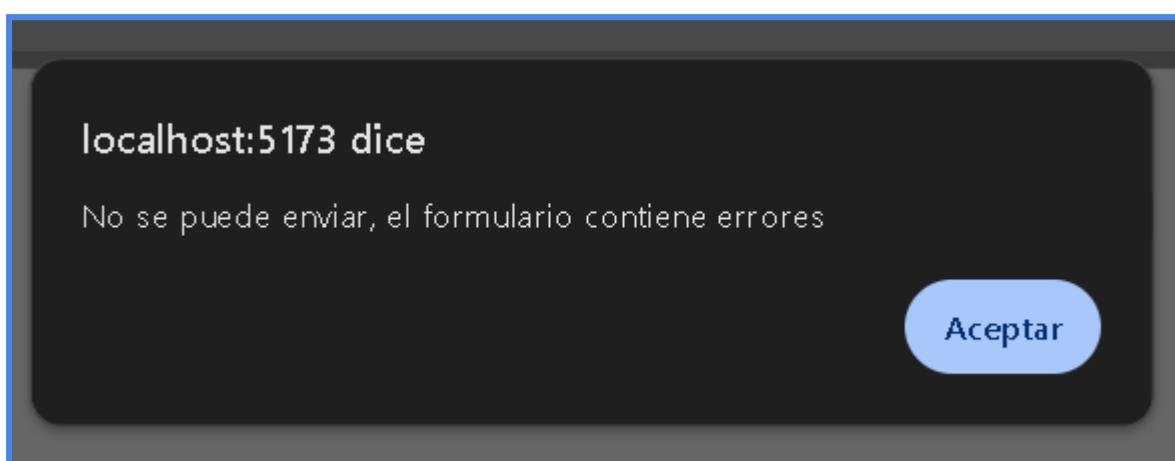
“Figura 07, conexión de handleUsername al input y muestra del error, elaboración propia”

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

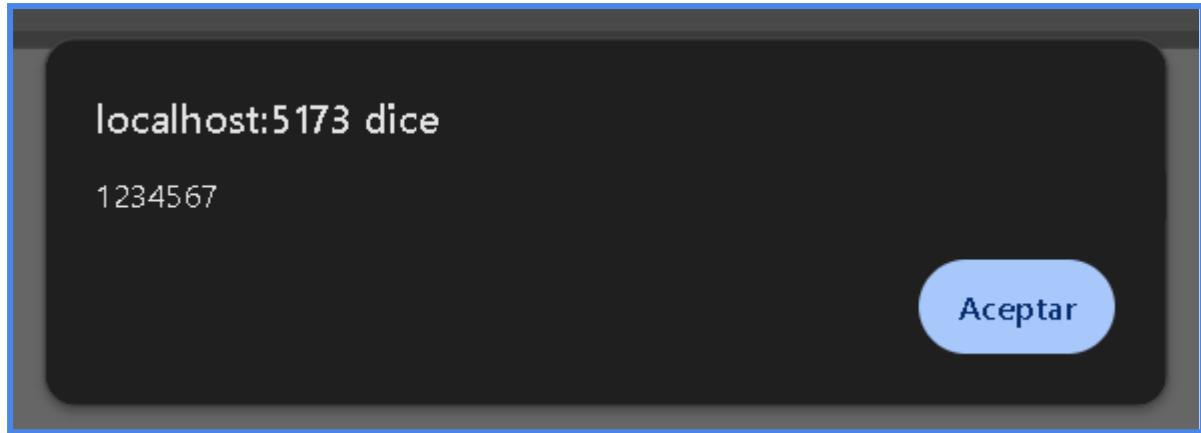
“Figura 08, impedimento del envío cuando hay un error, elaboración propia”



“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:
<input
  type="text"
  value={username}
  onChange={handleUsername}
/>
<button>Submit</button>

<p>{usernameError}</p>

<p>mi usuario es: {username}</p>

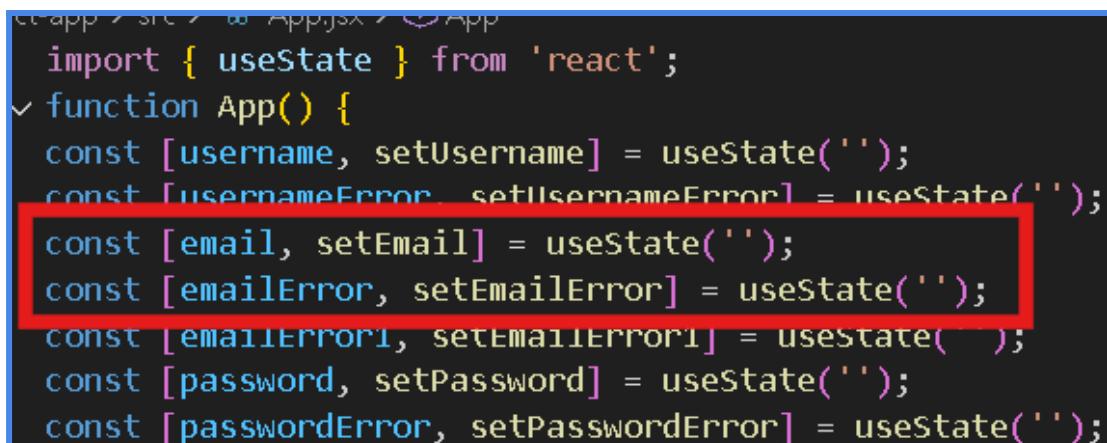
</form>
);
}

export default App;

```

- 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}>

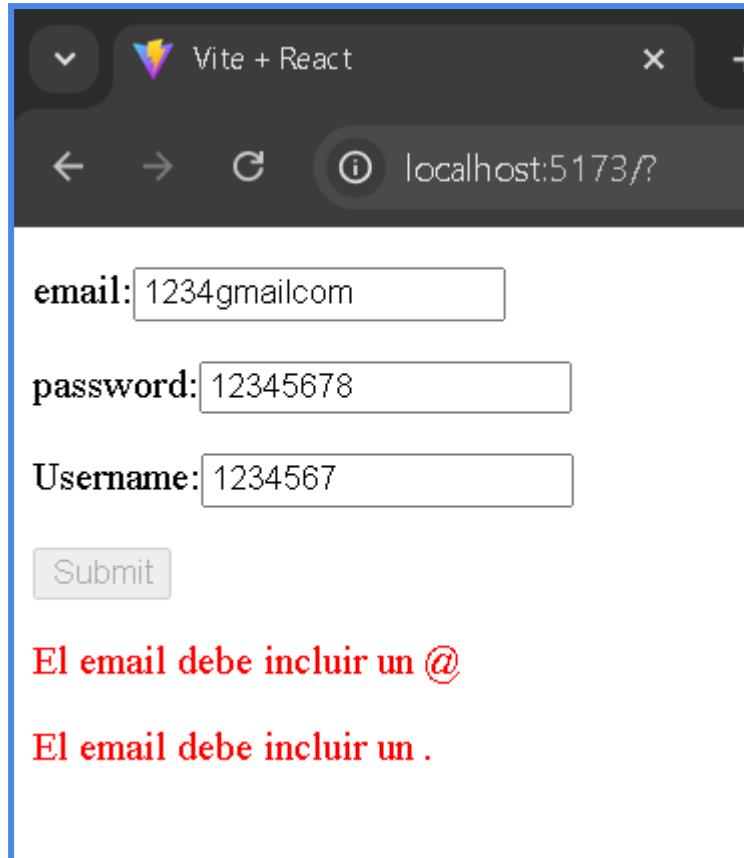
<p>email:
<input
  type="text"
  value={email}
  onChange={handleEmail}
/></p>
```

“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”



"Figura 15, muestra en pantalla, elaboración propia"

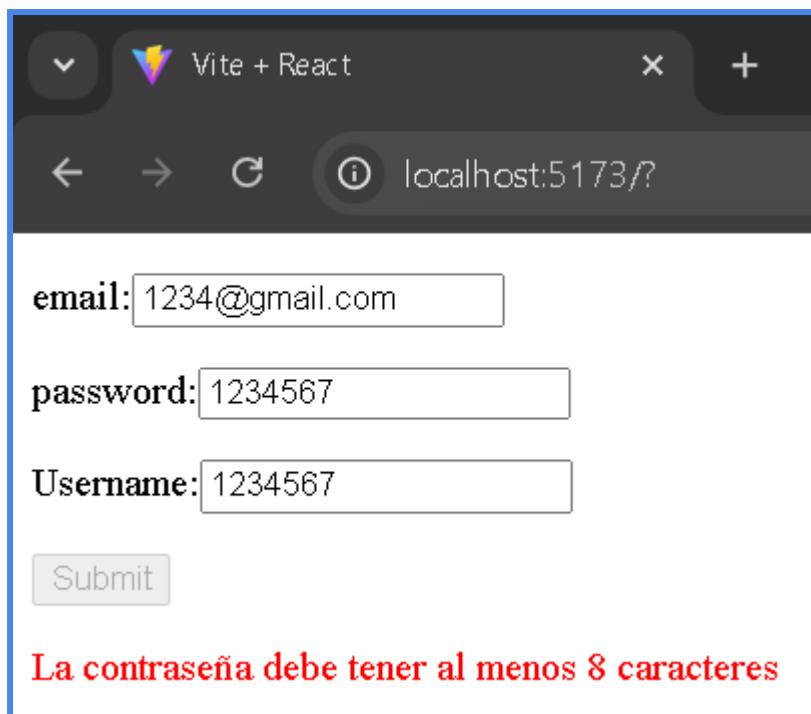
```
my-react-app > src > App.jsx > App
1 import { useState } from 'react';
2 function App() {
3   const [username, setUsername] = useState('');
4   const [usernameError, setUsernameError] = useState('');
5   const [email, setEmail] = useState('');
6   const [emailError, setEmailError] = useState('');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

```
<p>password:<br/><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”



“Figura 19, muestra en pantalla, elaboración propia”

```
git-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:<br/>
<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”

A screenshot of a web browser window titled "Vite + React". The address bar shows "localhost:5173/?". The page contains a login form with three input fields and a submit button. The first field is labeled "email:" with the value "1234@gmail.com". The second field is labeled "password:" with the value "12345678". The third field is labeled "Username:" with the value "123456". Below the form is a red error message: "El username debe tener más de 6 caracteres".

“Figura 23, muestra en pantalla, elaboración propia”

```
<button disabled={isValid}>Submit</button>
```

A screenshot of a code editor showing the following code snippet:

```
<button disabled={isValid}>Submit</button>
```

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

A screenshot of a web browser window showing a login form. The form has one input field labeled "Username:" with the value "123456". Below the form is a red error message: "El username debe tener más de 6 caracteres".

“Figura 25, muestra en pantalla, elaboración propia”

```
<p class="error">{usernameError}</p>
<p class="error">{emailError}</p>
<p class="error">{emailError1}</p>
<p class="error">{passwordError}</p></form>
);
```

A screenshot of a code editor showing the following code snippet:

```
<p class="error">{usernameError}</p>
<p class="error">{emailError}</p>
<p class="error">{emailError1}</p>
<p class="error">{passwordError}</p></form>
);
```

```
my-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 browser window titled "Vite + React" at "localhost:5173/?". The page contains a login form with three fields: "email:" containing "1234gmailcom", "password:" containing "1234567", and "Username:" containing "123456". Below the form is a "Submit" button. To the right of the form, four error messages are displayed in red text:
1. "El username debe tener más de 6 caracteres"
2. "El email debe incluir un @"
3. "El email debe incluir un ."
4. "La contraseña debe tener al menos 8 caracteres"

“Figura 28, muestra en pantalla, elaboración propia”