



GUÍA 2.2.2:

Configuración de React en un entorno de desarrollo en EC2 de AWS

Sigla	Asignatura	Experiencia de Aprendizaje
DSY1104	Desarrollo Full Stack II	EA Creación y Configuración de Sitios Web
Tiempo	Modalidad de Trabajo	Indicadores de logro
2 h	Individual	IL 1.1

다. Antecedentes generales

Esta guía tiene como objetivo enumerar las acciones necesarias para dar solución a los problemas planteados.



Requerimientos para esta actividad

Para el desarrollo de esta actividad deberás disponer de:

- Computador
- AWS EC2



Esta actividad consiste en enumerar las acciones necesarias para dar solución a los casos que se verán a continuación, para ello los estudiantes deberán realizar la actividad de forma individual.

Sigue las Instrucciones

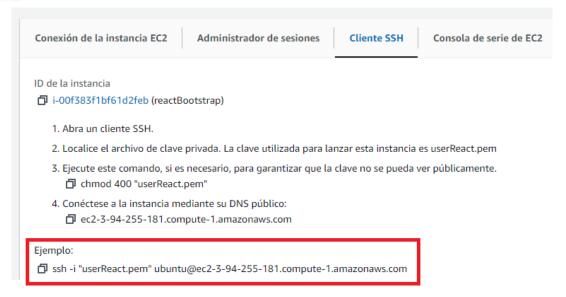
1. Ingresar a EC2 > Instancias y presionar Conectar



2. Se abrirá una nueva pestaña de conexión y copia el código que esta enmarcado en roio:







3. Abre un CMD y ejecútalo como administrador. Debes dirigirte a la dirección donde esta userReact.pem y copiar el ejemplo anterior y colocar yes

```
C:\Users\Donacion\Downloads>ssh -i "userReact.pem" ubuntu@ec2-3-94-255-181.compute-1.amazonaws.com
The authenticity of host 'ec2-3-94-255-181.compute-1.amazonaws.com (3.94.255.181)' can't be establishe
d.
ED25519 key fingerprint is SHA256:bmvTooIOd1CwTMq7n4/JSzHQODJk165Pa+k9UDOezaw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes_
```

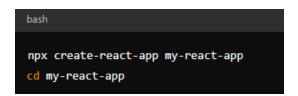
4. Creación de un nuevo proyecto React

Instala create-react-app globalmente: npm install -g create-react-app

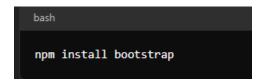
```
npm install -g create-react-app
```

5. Crea un nuevo proyecto React

npx create-react-app my-react-app cd my-react-app



6. Instala Bootstrap: npm install Bootstrap



7. Importa Bootstrap en el archivo principal (src/index.js o src/App.js):

Usa el comando find para buscar los archivos:

find . -name "index.js"





find . -name "App.js"

Esto te mostrará las rutas a index.js y App.js. Deberías ver algo como:

```
./src/index.js
./src/App.js
```

Usa un editor de texto como nano o vim para editar el archivo. Por ejemplo, si quieres editar index.js:

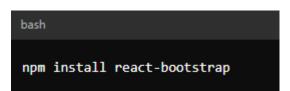
nano src/index.js
O si quieres editar App.js:
nano src/App.js

Agrega la siguiente línea al principio del archivo index.js o App.js:



Si estás usando nano, presiona Ctrl + O para guardar y Ctrl + X para salir. Si estás usando vim, presiona Esc, escribe :wq y presiona Enter para guardar y salir.

8. Instala React Bootstrap: npm install react-bootstrap



9. Usa un componente de React Bootstrap en tu aplicación (src/App.js):

10. Crea un formulario controlado en src/LoginForm.js:





```
import React, { useState } from 'react';
   function LoginForm() {
     const [username, setUsername] = useState(");
     const [password, setPassword] = useState(");
     const handleSubmit = (event) => {
      event.preventDefault();
      console.log('Login:', username, password);
    };
     return (
      <form onSubmit={handleSubmit}>
       <input
        type="text"
        value={username}
        onChange={(e) => setUsername(e.target.value)}
       />
       <input
        type="password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
       />
       <button type="submit">Login</button>
      </form>
    );
   }
   export default LoginForm;
    ,,,
11. Importa y usa el formulario en src/App.js:
    ′′′jsx
   import React from 'react';
   import { Button } from 'react-bootstrap';
   import LoginForm from './LoginForm';
```





```
function App() {
     return (
      <div className="App">
       <Button variant="primary">Clic aquí</Button>
       <LoginForm />
      </div>
     );
    }
    export default App;
12. Crea un formulario con validación básica en src/ValidatedForm.js:
    ′′′jsx
    import React, { useState } from 'react';
    function ValidatedForm() {
     const [email, setEmail] = useState(");
     const [error, setError] = useState(");
     const handleSubmit = (event) => {
      event.preventDefault();
      if (!email.includes('@')) {
       setError('Email inválido');
      } else {
       setError(");
       // Procesar el formulario
      }
     };
     return (
      <form onSubmit={handleSubmit}>
       <input
        type="email"
        value={email}
         onChange={(e) => setEmail(e.target.value)}
       />
```





```
{error && {error}}
       <button type="submit">Enviar</button>
      </form>
    );
   }
   export default ValidatedForm;
13. Importa y usa el formulario en src/App.js:
   ′′′jsx
   import React from 'react';
   import { Button } from 'react-bootstrap';
   import LoginForm from './LoginForm';
   import ValidatedForm from './ValidatedForm';
   function App() {
    return (
      <div className="App">
       <Button variant="primary">Clic aquí</Button>
       <LoginForm />
       <ValidatedForm />
      </div>
    );
   }
   export default App;
```

14. Instala react-hook-form

```
npm install react-hook-form
```

15. Crea un formulario usando *react-hook-form* en *src/HookForm.js*:

```
''jsx
import React from 'react';
import { useForm } from 'react-hook-form';

function HookForm() {
  const { register, handleSubmit, errors } = useForm();
  const onSubmit = data => console.log(data);
```





```
return (
      <form onSubmit={handleSubmit(onSubmit)}>
       <input
        name="email"
        ref={register({ required: true, pattern: /^\S+@\S+$/i })}
       {errors.email && <span>Este campo es requerido y debe ser un email válido</span>}
       <input type="submit" />
      </form>
  );
}
   export default HookForm;
16. Importa y usa el formulario en src/App.js:
   ′′′jsx
   import React from 'react';
   import { Button } from 'react-bootstrap';
   import LoginForm from './LoginForm';
   import ValidatedForm from './ValidatedForm';
   import HookForm from './HookForm';
   function App() {
    return (
      <div className="App">
       <Button variant="primary">Clic aquí</Button>
       <LoginForm />
       <ValidatedForm />
       <HookForm />
      </div>
    );
   }
   export default App;
17. Configuración de seguridad en EC2
   - Asegúrate de que el puerto 3000 esté abierto en el grupo de seguridad de tu
   instancia EC2.
    ```bash
 curl http://tu-ip-publica:3000
```

18. Ejecutar la aplicación





- Inicia la aplicación:
- ```bash npm start
- Accede a tu aplicación desde un navegador usando la IP pública de tu instancia EC2: `http://tu-ip-publica:3000`

Nota: Este método es para desarrollo. Para producción, se recomienda construir la aplicación y servirla con un servidor web como Nginx.

19. Debería levantar la App

#### Recursos de apoyo

Documentación Create React App: <a href="https://cra.link/deployment">https://cra.link/deployment</a>