

# Componente de tipo clase:

1. Crear un proyecto de react:

<https://create-react-app.dev/>

```
npx create-react-app class-components
```

Las dependencias principales que son instaladas en un proyecto react-app son:

- React
- React-dom
- React-script

```
PS C:\Users\silvi\OneDrive\Escritorio\Agile_innova\Curso_fronter_03_2021\3-React-component-sesion3\clase> npx create-react-app class-components
npx: instaló 67 en 8.575s

Creating a new React app in C:\Users\silvi\OneDrive\Escritorio\Agile_innova\Curso_fronter_03_2021\3-React-component-sesion3\clase\class-components.

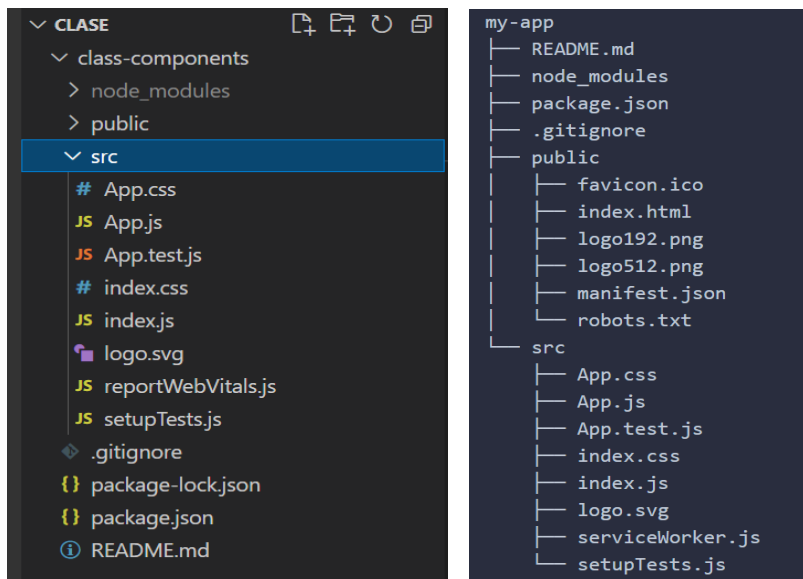
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[ ] ..... ] \ fetchMetadata: sill resolveWithNewModule big.js@5.2.2 checking insta
```

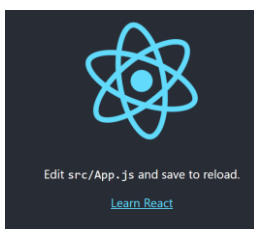
2. Ingresar a la aplicación

```
cd nombre-app
npm start
```

3. Estructura de trabajo de una aplicación de React:



Resultado de la ejecución



4. Limpiar archivos que no son necesarios por el momento

En la carpeta public el archivo index.html queda de esta manera

```
public
  <> index.html

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>App Geek</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

La carpeta src queda de la siguiente manera.

```
src
  JS index.js

class-components > src > JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3
4  ReactDOM.render(
5    <h1>Ejercicio react</h1>,
6    document.getElementById('root')
7  );

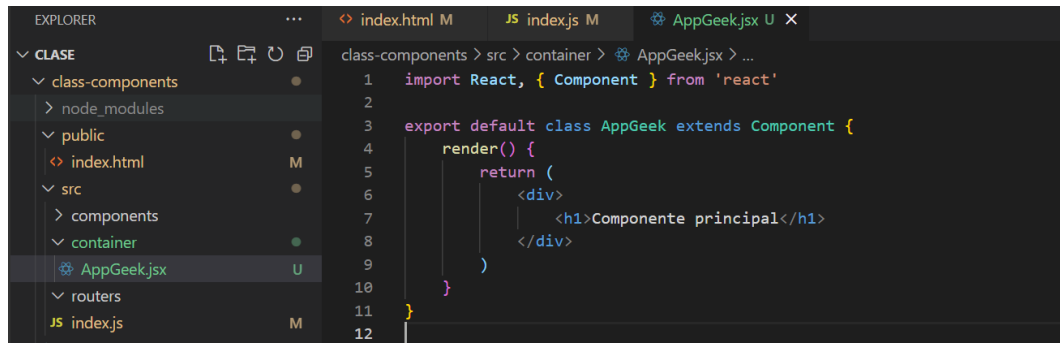
localhost:3001
Nueva pestaña YouTube Sistem.
```

## Ejercicio react

5. Crear las carpetas de trabajo

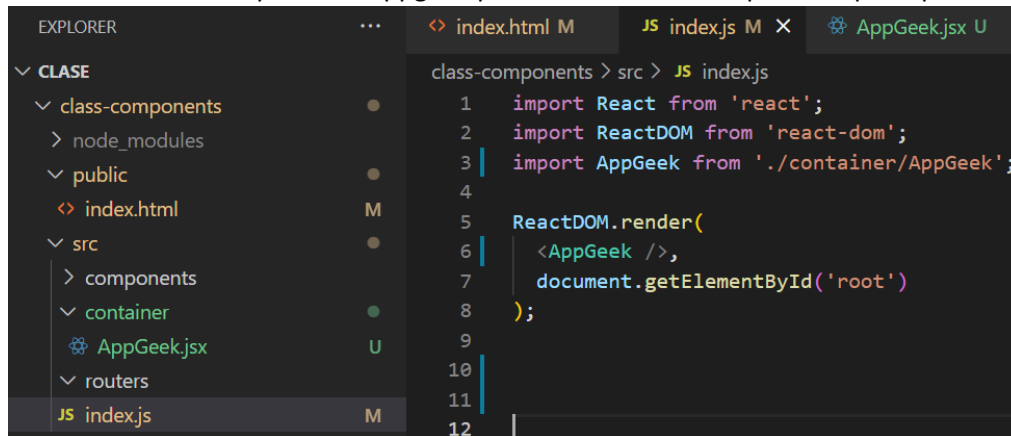
```
src
  > components
  > container
  > routers
  JS index.js
```

6. Crear el componente contenedor de la aplicación



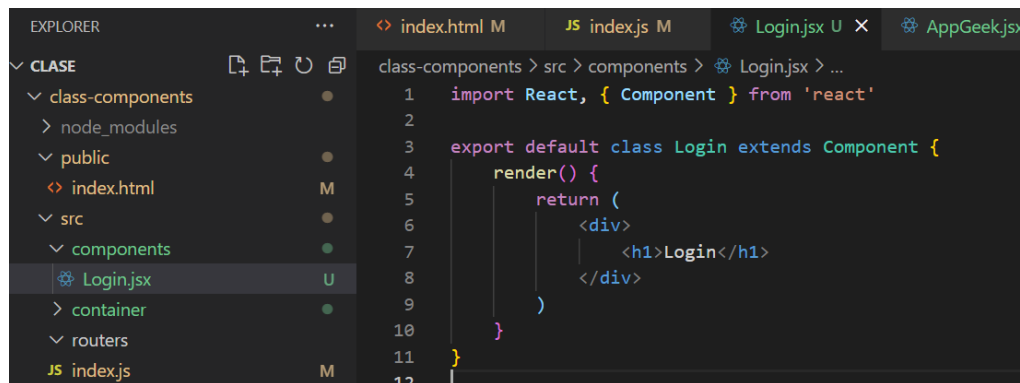
```
1 import React, { Component } from 'react'
2
3 export default class AppGeek extends Component {
4   render() {
5     return (
6       <div>
7         <h1>Componente principal</h1>
8       </div>
9     )
10   }
11 }
12
```

Ir al archivo index importar el Appgeek para renderizar el componente principal



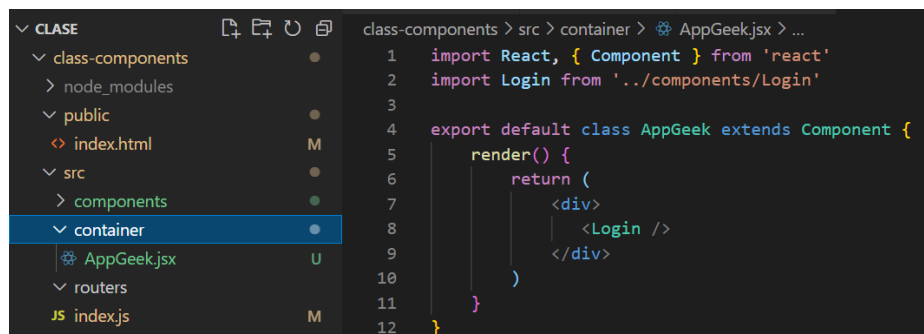
```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import AppGeek from './container/AppGeek';
4
5 ReactDOM.render(
6   <AppGeek />,
7   document.getElementById('root')
8 );
9
10
11
12
```

## 7. Crear un componente para iniciar sesión Login.jsx



```
1 import React, { Component } from 'react'
2
3 export default class Login extends Component {
4   render() {
5     return (
6       <div>
7         <h1>Login</h1>
8       </div>
9     )
10   }
11 }
12
```

Importar en el componente contenedor para renderizar el Login en el navegador



```
1 import React, { Component } from 'react'
2 import Login from '../components/Login'
3
4 export default class AppGeek extends Component {
5   render() {
6     return (
7       <div>
8         <Login />
9       </div>
10     )
11   }
12 }
```

Agregar la estructura de html necesaria:

Login.jsx

<https://gist.github.com/Garcia091/ab030d8bd058f4b78a008850201bc827>

```
class-components > src > components > Login.jsx > Login > render
1  import React, { Component } from 'react'
2
3  export default class Login extends Component {
4    render() {
5      return (
6        <div>
7          <form className="form-signin">
8            <h1 className="h4 mb-3 font-weight-normal">
9              Inicio de sesión
10            </h1>
11
12            <input
13              type="email"
14              id="inputEmail"
15              className="form-control mt-1"
16              placeholder="Email"
17              required=""
18            />
19
```

## Inicio de sesión

Login with social networks



Sign in with google

Agregarle funcionalidad con el state de React en componente en clases

```
constructor(){
  super();
  this.state = {
    form: {
      email: "",
      password: ""
    }
  }
}
```

Crear la función de handleChange para capturar la propiedad del target que está en cada input

```
handleChange = async e => {  
  ...await this.setState({  
    form: {  
      ...this.state.form,  
      [e.target.name]: e.target.value  
    }  
  });  
  console.log(this.state.form) //imprimir todo el estado  
}
```

El setState me permite actualizar la información de formulario que está en el state.

En cada input agregar la función handleChange para poder obtener la información que será guardada en el state, adicional todo input debe tener si atributo name

```
<input  
  type="email"  
  id="inputEmail"  
  name="email"  
  className="form-control mt-1"  
  placeholder="Email"  
  required=""  
  onChange={this.handleChange}  
>
```

En el botón de Login vamos a crear una función para iniciar sesión

```
<button  
  type="submit"  
  className="btn btn-primary btn-block"  
  onClick={() => this.iniciarSesion()}  
>  
  Login  
</button>
```

Crear una función para prevenir por defecto el reinicio de la información del formulario

## Código completo parte 1

```
class-components > src > components > Login.jsx > Login
1  import React, { Component } from 'react'
2
3  export default class Login extends Component {
4    constructor(){
5      super();
6      this.state = {
7        form: {
8          email: "",
9          password: ""
10       }
11    }
12  }
13
14  handleChange = async e => {
15    await this.setState({
16      form: {
17        ...this.state.form,
18        [e.target.name]: e.target.value
19      }
20    });
21    console.log(this.state.form) //imprimir todo el estado
22  }
23
24  iniciarSesion = async () => {
25    console.log(`Bienvenido ${this.state.form.email}`)
26  }
27
28  handleSutmit = (e) => {
29    e.preventDefault()
30  }
31
```

```
render() {
  return (
    <div>
      <form className="form-signin" onSubmit={this.handleSutmit}>
        <h1 className="h4 mb-3 font-weight-normal">
          Inicio de sesión
        </h1>

        <input
          type="email"
          id="inputEmail"
          name="email"
          className="form-control mt-1"
          placeholder="Email"
          required=""
          onChange={this.handleChange}
        />

        <input
          type="Password"
          name="password"
          id="inputPassword"
          className="form-control mt-1"
          placeholder="password"
          required=""
          onChange={this.handleChange}
        />

        <button
          type="submit"
          className="btn btn-primary btn-block"
          onClick={() => this.iniciarSesion()}
        >
          Login
        </button>
      </form>
    </div>
  )
}
```

```

    <div className="">
      <p>Login with social networks</p>

      <div className="google-btn btn-primary">
        <div className="google-icon-wrapper">
          
        </div>
        <p className="btn-text">
          <b>Sign in with google</b>
        </p>
      </div>
    </div>
  </form>
</div>
}
}

```

## Agregar dependencias necesarias

Axios: <https://www.npmjs.com/package/axios>

`npm i axios`

Md5: <https://www.npmjs.com/package/md5>

`npm i md5`

Importar las dependencias instaladas

Crear una constante para capturar la información del API de heruko

```

Registro.jsx M X
src > components > Registro.jsx > ...
1  import React, { Component } from 'react'
2  import axios from 'axios';
3  import md5 from 'md5';
4  import uuid from 'react-uuid' // id aleatorio
5  |
6  const baseUrl = "https://movies-geek.herokuapp.com/usuario";
7

```

Configurar la función iniciar sesión

```

iniciarSesion = async () => {
  await axios.get(baseUrl, { params: { username: this.state.form.username, password: md5(this.state.form.password) } })
    .then(response => {
      return response.data;
    })
    .then(response => {
      if (response.length > 0) {
        var respuesta = response[0];
        alert(`Bienvenido ${respuesta.nombre} ${respuesta.apellido_paterno}`);
      } else {
        alert('El usuario o la contraseña no son correctos');
      }
    })
    .catch(error => {
      console.log(error);
    })
}

```

Registro:

<https://gist.github.com/Garcia091/ab030d8bd058f4b78a008850201bc827>

```
class-components > src > components > Registro.jsx > ...
1  import React, { Component } from 'react'
2  |
3  export default class Registro extends Component {
4  |
5      handleSubmit = (e) => {
6          e.preventDefault()
7      }
8  |
9      render() {
10         return (
11             <div className="Registro py-5 container text-center">
12                 <form className="form-signin" onSubmit={this.handleSubmit}>
13                     <h1 className="h3 mb-3 font-weight-normal">
14                         ¡Regístrate en nuestro sistema!
15                     </h1>
16                     <div className="fadeIn first ">
17                         
22                         <h3>Crea una cuenta</h3>
23                     </div>
24
25                     <input
26                         type="text"
```

Crear el constructor

```
8  export default class Registro extends Component {
9  |
10     constructor() {
11         super();
12         this.state = {
13             data: [],
14             form: {
15                 id: '',
16                 apellido_paterno: '',
17                 apellido_materno: '',
18                 nombre: '',
19                 username: '',
20                 password: ''
21             }
22         }
23     }
24 }
```

Crear la función handleChange

```
handleChange = async e => {
    await this.setState({
        form: {
            ...this.state.form,
            [e.target.name]: e.target.value
        }
    });
    console.log(this.state.form) //imprimir todo el estado
}
```

¡Regístrate en  
nuestro sistema!



Crea una cuenta

Apellido paterno
Apellido materno
nombre
Email
Password

Register

Already registered?



Asignar la función handleChange a cada input

```
<input
  type="text"
  placeholder="Apellido paterno"
  name="apellido_paterno"
  className="form-control"
  autoComplete="off"
  onChange={this.handleChange}
/>
```

Para el registro de usuarios debemos instalar la dependencia de react-uuid para asignar un id aleatorio

Uuid: <https://www.npmjs.com/package/react-uuid>

```
npm i react-uuid
```

importar la librería

```
class-components > src > components > Registro.jsx > Registro > constru
1  import React, { Component } from 'react'
2  import axios from 'axios';
3  import md5 from 'md5';
4  import uuid from 'react-uuid'
```

Realizar la petición de tipo POST

```
ResgistroUsuario = async () => {
  await axios.post(baseUrl, {
    id: uuid,
    apellido_paterno: this.state.form.apellido_paterno,
    apellido_materno: this.state.form.apellido_materno,
    nombre: this.state.form.nombre,
    username: this.state.form.username,
    password: md5(this.state.form.password)
  }).then(response => {
    alert('Usuario Registrado')
  }).catch(error => {
    console.log(error.message);
  })
}
```

Código Completo:

Registro.jsx M X

src > components > Registro.jsx > Registro

```
1  import React, { Component } from 'react'
2  import axios from 'axios';
3  import md5 from 'md5';
4  import uuid from 'react-uuid' // id aleatorio
5
6  const baseUrl = "https://movies-geek.herokuapp.com/usuario";
7
8  export default class Registro extends Component {
9
10   constructor() {
11     super();
12     this.state = {
13       data: [],
14       form: {
15         id: '',
16         apellido_paterno: '',
17         apellido_materno: '',
18         nombre: '',
19         username: '',
20         password: ''
21       }
22     }
23   }
24
25   handleChange = async e => {
26     await this.setState({
27       form: {
28         ...this.state.form,
29         [e.target.name]: e.target.value
30       }
31     });
32     console.log(this.state.form) //imprimir todo el estado
33   }
34
35   handleSubmit = (e) => {
36     e.preventDefault()
37   }
38
39   ResgistroUsuario = async () => {
40     await axios.post(baseUrl, {
41       id: uuid,
42       apellido_paterno: this.state.form.apellido_paterno,
43       apellido_materno: this.state.form.apellido_materno,
44       nombre: this.state.form.nombre,
45       username: this.state.form.username,
46       password: md5(this.state.form.password)
47     }).then(response => {
48       alert('Usuario Registrado')
49     }).catch(error => {
50       console.log(error.message);
51     })
52   }
53
54   render() {
55     //
56   }
```

```

render() {
  return (
    <div className="Registro py-5 container text-center">
      <form className="form-signin" onSubmit={this.handleSubmit}>
        <h1 className="h3 mb-3 font-weight-normal">
          ¡Regístrate en nuestro sistema!
        </h1>
        <div className="fadeIn first ">
          
          <h3>Crea una cuenta</h3>
        </div>

        <input
          type="text"
          placeholder="Apellido paterno"
          name="apellido_paterno"
          className="form-control"
          autoComplete="off"
          onChange={this.handleChange}
        />

        <input
          type="text"
          placeholder="Apellido materno"
          name="apellido_materno"
          className="form-control"
          autoComplete="off"
          onChange={this.handleChange}
          required=""
        />

        <input
          type="text"
          name="nombre"
          className="form-control"
          placeholder="nombre"
          required=""
          onChange={this.handleChange}
        />

        <input
          type="email"
          name="username"
          className="form-control"
          placeholder="Email"
          required=""
          onChange={this.handleChange}
        />
      </form>
    </div>
  );
}

```

```

89
90
91     <input
92       type="text"
93       name="nombre"
94       className="form-control"
95       placeholder="nombre"
96       required=""
97       onChange={this.handleChange}
98     />
99
100    <input
101      type="email"
102      name="username"
103      className="form-control"
104      placeholder="Email"
105      required=""
106      onChange={this.handleChange}
107    />
108
109    <input
110      type="password"
111      name="password"
112      className="form-control"
113      placeholder="Password"
114      required=""
115      onChange={this.handleChange}
116    />
117    <br />
118    <button
119      type="submit"
120      className="btn btn-primary btn-block mb-1"
121      onClick={() => this.RegistroUsuario()}
122    >
123      Register
124    </button>
125    <br />
126    <a
127      to=""
128      className="link"
129    >
130      Already registered?
131    </a>
132  </form>
133 </div>
134 }
135

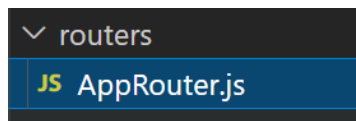
```

## Enrutamiento

Instalar react-router-dom <https://reactrouter.com/>

`npm i react-router-dom`

Crear una carpeta routers y un archivo AppRouter.js



```

src > routers > JS AppRouter.js > ...
1  import React, { Component } from 'react'
2
3  export default class AppRouter extends Component {
4    render() {
5      return (
6        <div>
7
8        </div>
9      )
10    }
11  }

```

## Crear estructura de enrutamiento

```
src > routers > JS AppRouter.js > AppRouter > render
1  import React, { Component } from 'react'
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link
7  } from "react-router-dom";
8
9  export default class AppRouter extends Component {
10    render() {
11      return (
12        <Router>
13          <Switch>
14            <Route path="/" component={}/>
15          </Switch>
16        </Router>
17      )
18    }
19  }
20
```

```
src > routers > JS AppRouter.js > AppRouter > render
1  import React, { Component } from 'react'
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link
7  } from "react-router-dom";
8
9  import Registro from '../components/Registro';
10 import AppGeek from '../container/AppGeek'
11
12 export default class AppRouter extends Component {
13   render() {
14     return (
15       <Router>
16         <Switch>
17           <Route exact path="/registro" component={Registro}/>
18           <Route exact path="/" component={AppGeek}/>
19         </Switch>
20       </Router>
21     )
22   }
23 }
```

En el archivo index.js agregar el archivo AppRouter.js para renderizar la aplicación con enrutamiento

```
src > JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import AppRouter from './routers/AppRouter';
4
5
6  ReactDOM.render(
7    <AppRouter />,
8    document.getElementById('root')
9  );
```

En Login y en Registro agregar enrutamiento

Importar react-router-dom

```
import {Link} from 'react-router-dom'
```

Todas las etiquetas <a> cambiarlas por <Link>

<pre>    &lt;/div&gt;   &lt;/div&gt;   &lt;a     to=""     className="Link"   &gt;     Create new account   &lt;/a&gt; &lt;/form&gt; &lt;/div&gt;</pre>	<pre>    &lt;/div&gt;   &lt;/div&gt;   &lt;Link     to="/registro"     className="Link"   &gt;     Create new account   &lt;/Link&gt; &lt;/form&gt;</pre>
---	---

En Registro agregar enrutamiento

```
import {Link} from 'react-router-dom'
```

Todas las etiquetas <a> cambiarlas por <Link>

<pre>    &lt;/button&gt;     &lt;br /&gt;     &lt;a       to=""       className="link"     &gt;       Already registered?     &lt;/a&gt;   &lt;/form&gt; &lt;/div&gt; )</pre>	<pre>    &lt;/button&gt;     &lt;br /&gt;     &lt;Link       to="/"       className="link"     &gt;       Already registered?     &lt;/Link&gt;   &lt;/form&gt;</pre>
---	---

## Resultado


localhost:3000

YouTube Sistema de Informa... Página principal de... Google SQL: Los fundamen... ITMVirtual - Unidad...

### Inicio de sesión

Login

Login with social networks

 Sign in with google

[Create new account](#)

## Recursos:

[https://www.techdiagonal.com/reactjs\\_courses/beginner/reactjs-component-state/](https://www.techdiagonal.com/reactjs_courses/beginner/reactjs-component-state/)

<https://ihatetomatoes.net/react-tutorial-for-beginners/>