

**Tipo** : Guía de laboratorio  
**Capítulo** : React JS  
**Duración** : 30 minutos

---

## I. OBJETIVO

Crear una aplicación usando el CLI create-react-app.

## II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Editor Visual Studio Code
- NodeJS

## III. EJECUCIÓN DEL LABORATORIO

### • Ejercicio 2.1: Crear una aplicación usando create-react-app

1. Instalar el CLI create-react-app.

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

2. Abrir la carpeta react-app en Visual Studio Code para revisar la estructura del proyecto creado.
3. Crear los módulos de admin, login.
4. Crear los componentes header, footer, products para la aplicación de admin de productos.

[src/modules/admin/Footer.JS](#)

```
1  import React from 'react'
2
3  const Footer = () => {
4    const date = new Date();
5
6    return (
7      <div>
8        Admin desarrollado en React JS &copy; { date.getFullYear() }
9      </div>
10    );
11  }
12
13  export default Footer;
14
```

5. Mostrar los componentes creados en la aplicación.

src/App.js

```
1  import React from 'react';
2
3  import Footer from './modules/admin/Footer';
4  import Header from './modules/admin/Header';
5  import Products from './modules/admin/Products';
6
7  function App() {
8    return (
9      <div className="App">
10        <Header />
11        <Products />
12        <Footer />
13      </div>
14    );
15  }
16
17  export default App;
18
```

Estructura inicial

```
> node_modules
> public
└─ src
   └─ modules
      └─ admin
         JS Footer.js
         JS Header.js
         JS Products.js
      └─ login
         JS App.js
         # index.css
         JS index.js
         JS serviceWorker.js
         JS setupTests.js
      .gitignore
      {} package-lock.json
      {} package.json
      ⓘ README.md
      ⚡ README.old.md
```

#### IV. EVALUACIÓN

1. ¿Qué es el virtual DOM?

El **virtual DOM** crea una copia del DOM en memoria, luego utiliza un algoritmo para comparar (diff) las propiedades y actualizar solamente las partes que cambiaron y no tener que revisar y repintar el DOM entero de nuestra página.

2. ¿Cuáles son las principales diferencias entre crear componentes de clase o de funcion?

Los componentes de clases manejan estados y se extiende de la clase component, la cual nos permite ejecutar una serie de métodos durante el ciclo de vida del componente. A diferencia que en los componentes de funciones no se podía realizar estas acciones ni tener estados, hasta antes de la llegada de los hooks.