2 - Creación del primer proyecto empleando React

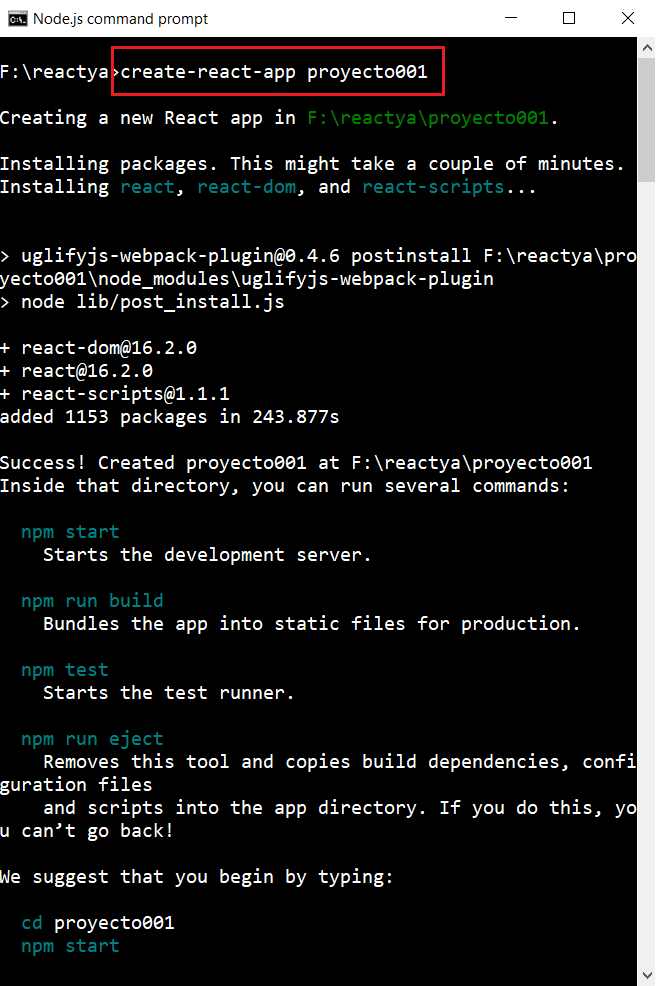
En el concepto anterior presentamos que es React y la instalación de herramientas básicas, veamos ahora los pasos que debemos dar para crear el primer proyecto.

La aplicación create-react-app nos automatiza la generación de todos los archivos y carpetas básicas del proyecto y nos instala Webpack, Babel, EsLint etc. más adelante veremos el objetivo de cada uno de estos programas.

Creación del proyecto.

Debemos abrir la consola de comandos de Node y proceder a llamar al programa create-react-app pasando como parámetro el nombre del proyecto a crear:

create-react-app proyecto001



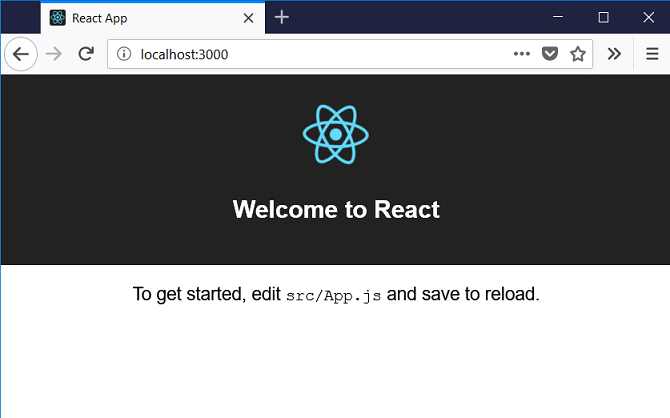
Requiere varios segundos para que se creen las carpetas, archivos básicos y descarguen los programas que necesita luego la aplicación:

Luego de esto podemos probar la aplicación mínima sin hacer cambios descendiendo desde la línea de comandos a la carpeta 'proyecto001' y ejecutando npm start:

cd proyecto001

npm start

Al cabo de unos segundos se abre el navegador y aparece la aplicación funcionando:



Antes de analizar las carpetas y archivos generados probemos de hacer unos cambios mínimos en el archivo 'App.js' de la carpeta 'src':

import React, { Component } from 'react';

import './App.css';

class App extends Component {

render() {

return (

<h1>Hola Mundo</h1>

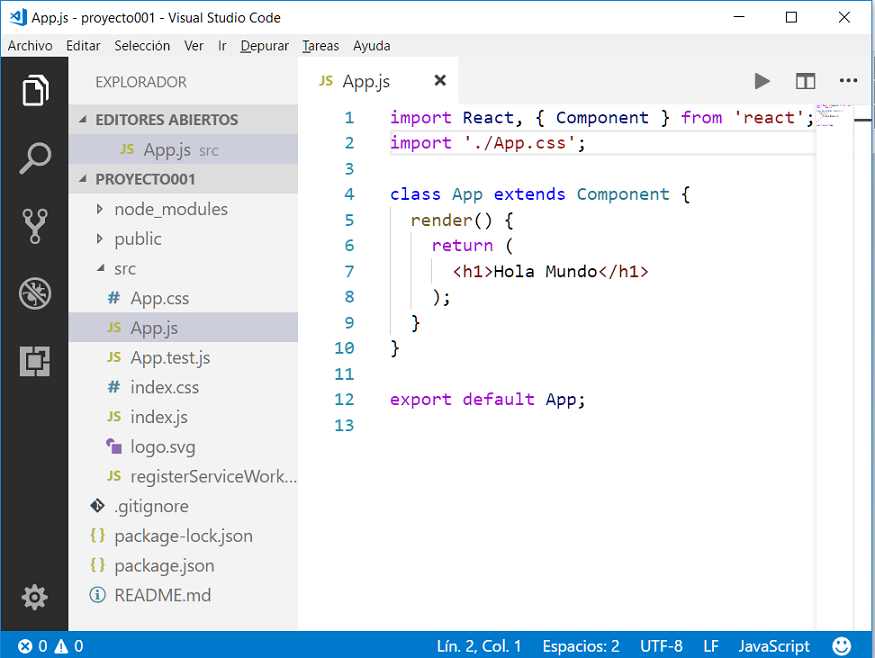
);

}

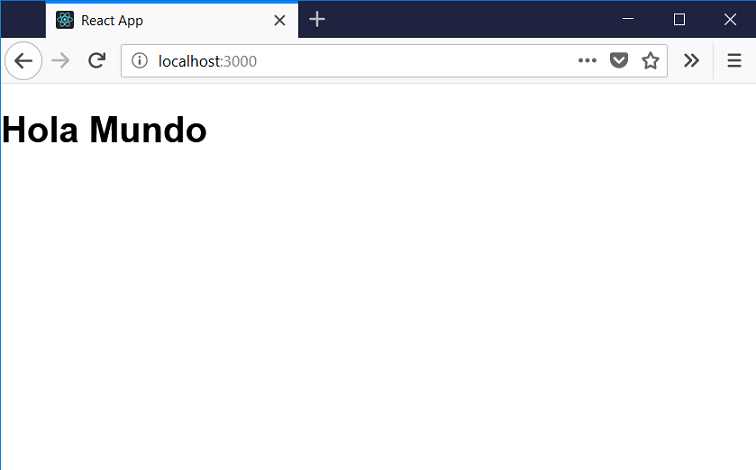
}

export default App;

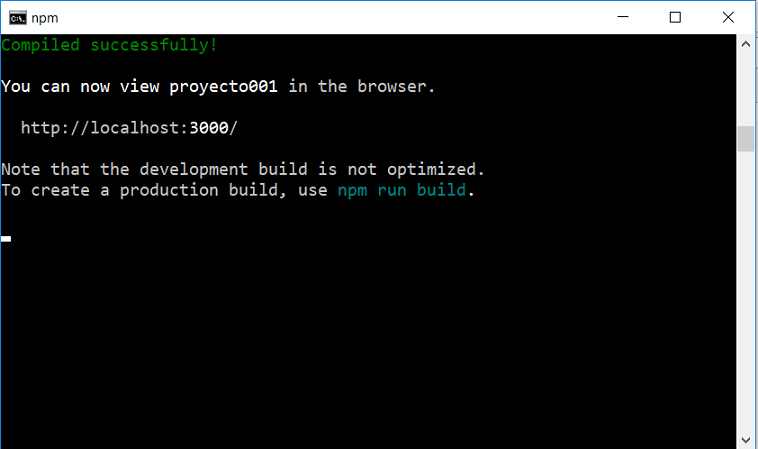
Para hacer los cambios podemos utilizar cualquier editor de texto, yo recomiendo [Visual Studio Code](https://code.visualstudio.com/download)



Luego de grabar el archivo, volvemos a ver el navegador y comprobar que se actualizó en forma automática la pantalla:

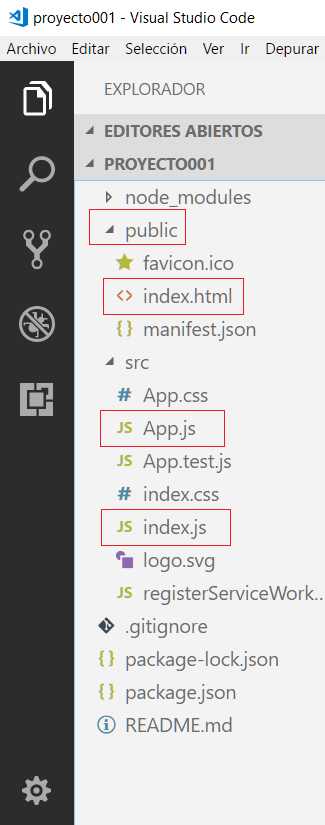


Es importante tener en cuenta de NO cerrar la consola de Node.js donde está en ejecución la aplicación (si la cerramos por error la debemos volver a abrir y ejecutar 'npm start' en la carpeta del proyecto):



Archivos y carpetas principales

Veremos algunos archivos y carpetas esenciales de todo proyecto:



El archivo 'index.html' que se encuentra en la carpeta 'public' tiene un elemento div donde en forma dinámica y mediante Javascript se agregaran en su interior los elementos HTML de nuestro programa:

<div id="root"></div>

El segundo archivo importante es el 'index.js' que se encuentra en la carpeta raíz del proyecto:

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import registerServiceWorker from './registerServiceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

registerServiceWorker();

En los próximos conceptos comenzaremos con el código de React en este momento solo quiero que veamos como se relacionan estos archivos. En la llamada al método render del objeto ReactDOM mediante getElementById obtenemos la referencia del div del otro archivo (index.html) llamado 'root':

ReactDOM.render(<App />, document.getElementById('root'));

El método render es el responsable de generar todas las etiquetas HTML en forma dinámica. El primer parámetro del método render es la referencia de la componente 'App' que se encuentra en el archivo 'App.js':

import React, { Component } from 'react';

import './App.css';

class App extends Component {

render() {

return (

<h1>Hola Mundo</h1>

);

}

}

export default App;

Hemos visto los tres archivos que siempre existirán en todo proyecto que desarrollemos con React:



[Retornar](https://www.tutorialesprogramacionya.com/reactya/index.php?inicio=0)