



POLITÉCNICO COLOMBIANO  
JAIME ISAZA CADAVID

Educación para  
*vivir mejor*

# Construcción de elementos de software web 1

## Semana 10

### Introducción de React JS - Hola Mundo

---

Media Técnica en Programación de Sistemas de Información  
Politécnico Jaime Isaza Cadavid - 2020



Politécnico Colombiano Jaime Isaza Cadavid



## ¿Qué es un framework Web?

Cuando empleamos el termino framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables y comunes para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

El objetivo principal que persigue un framework es acelerar el proceso de desarrollo de aplicativos de software, reutilizando código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web lo podemos definir como un conjunto de componentes (por ejemplo, Bootstrap está compuesto por diferentes archivos css y javascript) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

## ¿En qué casos y por qué se utilizan los frameworks Web?

Una framework, librería o marco de trabajo web nos soluciona uno de los principales problemas de la programación frontEnd: mantener la interfaz de usuario (UI, del inglés User Interface) en sincronización con el estado nuestra aplicación.

Pero, ¿qué es el estado de una aplicación web?. Una aplicación web, a diferencia de una simple página web, se encarga de gestionar datos. Por ejemplo, en una aplicación como Gmail gestionamos datos de correos (nuevos, leídos, archivados, etc.) desde una interfaz. En conclusión, el estado de una aplicación web, es la manipulación de datos que son presentados en una interfaz, generando una interpretación de estos, por ejemplo:

- En una simple aplicación de una lista de tareas, manejamos datos de tareas, si están completados o las fechas de realización.



Por lo cual, las aplicaciones donde se identifica la manipulación de datos, que genere una variación y cambio en la presentación de dicha información es recomendable el uso de un framework web.

## *Framework JavaScript para FrontEnd*

Adicionalmente de las virtudes mencionadas respecto a los beneficios de un framework, los marcos de trabajo de JavaScript surgen como una respuesta a la necesidad del correcto funcionamiento de las aplicaciones web que conviven en los ambientes de los navegadores, tales como: Chrome, Firefox, Opera, Safari e Internet Explorer.

El uso de estas herramientas acarrea grandes ventajas para el mundo del desarrollo web, pero también es importantes resaltar algunas debilidades que se tienen al momento de usarse.

Entre las principales ventajas se encuentran:

- **Ahorro de tiempo:** Una de las mayores problemáticas que ha tenido el desarrollo de aplicaciones web es su alto tiempo de implementación. Pero la aparición de los marcos de trabajo de JavaScript ha permitido que el tiempo de desarrollo de estas aplicaciones disminuya considerablemente.
- **Mayor velocidad:** Contar con estas herramientas de trabajo ha permitido a los equipos de desarrollo aumentar su velocidad de implementación de las soluciones de software. Indudablemente los marcos de trabajo de JavaScript le permiten al desarrollador concentrar sus esfuerzos en solucionar los retos que demanda el negocio y no, en muchas dificultades técnicas que se presentan durante el desarrollo, las cuales son mitigados con frameworks de JavaScript.
- **Mejor entendimiento de código:** Usar las mejores prácticas y estándares de programación ayudan que los productos codificados por los desarrolladores sean más entendibles y más claros para los demás. De este modo se genera una alta escalabilidad en los productos de software.



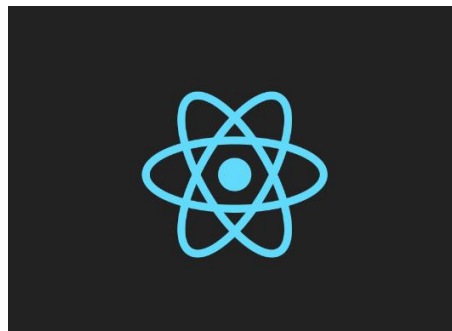
Las mayores dificultades que puede presentar el uso de framework JavaScript son:

- **Dependencia de una librería:** la expansión de las aplicaciones es sin lugar a duda una realidad que se debe tener en cuenta en los proyectos a corto y largo plazo. Esto implica que los marcos de trabajo en JavaScript deben contar con mayores prestaciones que les permitan a las aplicaciones crecer de una manera más rápida y sin mayores traumatismos, por lo cual, la dependencia que se asume en estas librerías es muy alta y es recomendable analizarlas antes de proceder a implementarlas en un proyecto software.
- **Demanda de recursos:** implementar cualquier librería o marco de trabajo, representa un consumo considerable en cualquier aplicación de software, debido a que internamente estas cuentan con procesos y rutinas que demandan recursos a los clientes que están usando estas aplicaciones.

## Los principales framework JavaScript FrontEnd

Hay muchos frameworks de JavaScript y la realidad de esta industria es que día a día irán apareciendo nuevas librerías que den respuesta a restos a los interrogantes que están y estarán en un futuro en la industria del desarrollo de aplicativos web. A continuación, se mencionan algunos de los frameworks más utilizados en la actualidad y cuentan con una mayor demanda en la industria.

### React



Uno de los frameworks web más usados por parte de los desarrolladores. Es una librería de Facebook enfocada en la creación de vistas con particularidades como los patrones de



eventos que permiten actualizar las mencionadas vistas cuando los datos hayan sido modificados, aportando una carga en tiempo real a nuestro desarrollo.

## Angular



Este framework, y el resto de sus versiones fue creado inicialmente y es promovido por Google, entre sus características encontramos que modifica el DOM directamente y alguna que otra característica revolucionaria para su época que hoy en día nos sorprenden poco, habiendo visto el panorama que hay de frameworks y librerías.

## VueJS



Vue es una iniciativa de un ex-empleado de Google, que había estado relacionado con AngularJS y que pretendía coger sus partes favoritas de este framework y crear algo muy ligero. En 2014 fue liberado a la comunidad y se encuentra disponible de manera Open Source.

El curso centrará el esfuerzo en el framework de REACT como tecnología que apoye el desarrollo de la parte FrontEnd de las aplicaciones.



## ¿Qué REACT?

React es una librería JavaScript desarrollada principalmente por el grande las redes sociales Facebook. Se fundamente principalmente como software libre y desde su liberación al público a acaparado una creciente comunidad de desarrolladores que buscan ayudar al crecimiento de esta fantástica tecnología. La creación de este framework se derivada de las diferentes necesidades presentes en la popular red social, en su carrera por brindar una excelente experiencia de usuario a su público. El desarrollo ágil de componentes de interfaces de usuario y su elevado rendimiento fueron las premisas con las cuales fue creada React.

## ¿Por qué usar REACT?

- 1. Alto crecimiento:** Durante mucho tiempo Angular ha sido el framework de JavaScript predominante para la creación de aplicaciones web. Pero REACT ha tenido un alto crecimiento que muy probablemente en algún momento lo coloque como el framework predominante de este lenguaje.

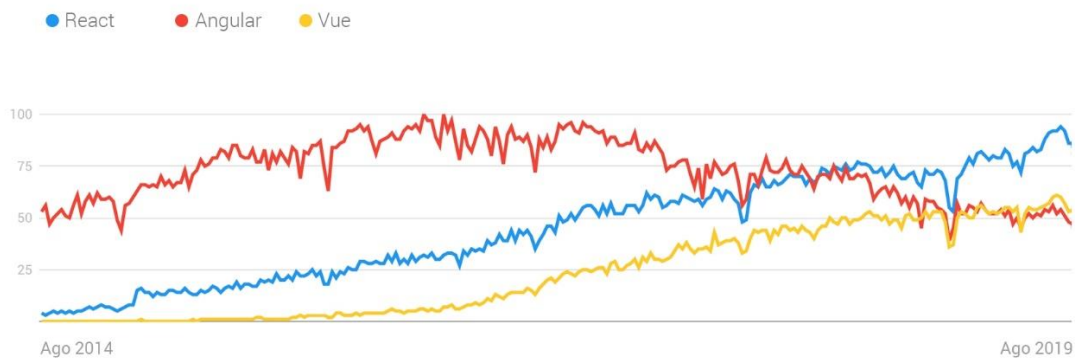


Figura: Grafico que muestra el crecimiento exponencial de REACT durante el periodo 2014 – 2019. Recuperado de <https://platzi.com/blog/react/>.



2. **SPA:** Single Page Application es una aplicación web de una sola página, es decir, la interacción de la aplicación es en una página. Esta característica de REACT carga el HTML, JavaScript y CSS una sola vez en la página y se puede observar en aplicaciones como YouTube, Netflix, Gmail, etc.
3. **DOM virtual:** Uno de los grandes problemas que se ha tenido en las aplicaciones construidos en navegadores es la renderizado de la información, debido a que antiguamente era obligatorio recargar la página completa para poder visualizar un cambio. En la actualidad se puede hacer mejor uso del DOM (Modelo de Objetos del Documento), debido a que solo es necesario actualizar el componente que haya actualizado. Gracias a esta característica mejora: la experiencia de usuario al navegar por la aplicación web, la rapidez en la carga de las páginas y facilita el mantenimiento de la aplicación.
4. **Amplia comunidad:** JavaScript cuenta con una amplia comunidad que apoya el crecimiento de esta tecnología. Por ello REACT se ha expandido tan fácilmente entre las diferentes comunidades de desarrolladores, esto ayuda a que se encuentre disponible una gran variedad de librerías para el desarrollo de aplicaciones.
5. **ECMAScript 6 (ES6):** React trabaja con la última versión de JavaScript y a medida que se van liberando actualizaciones de este lenguaje, este framework las adapta de manera muy rápida y sin generar mayores traumas sobre las aplicaciones que ya vienen trabajando con él. De igual forma, los navegadores que son más obsoletos no se quedan fuera de la cobertura de REACT, para esto se incorpora dentro del core Babel, una herramienta que se encarga de transformar el código JavaScript escrito con la especificación ES6 en código que cualquier navegador puede interpretar.

Estas son algunas de las ventajas que se pueden enumerar con el uso de REACT. En conclusión, este framework en la actualidad es una gran elección cuando se busca implementar una solución de software.



## ¿Cómo instalar REACT?

Antes de iniciar la instalación de cualquier entorno o framework, es importante tener en cuenta la página oficial de este, con el fin de referenciar la documentación que nos permita realizar todas las configuraciones iniciales de forma adecuada y estar más actualizado en torno a su manera de uso. En el caso de REACT su sitio oficial es <https://reactjs.org/> y su sección de documentación es <https://reactjs.org/docs>.

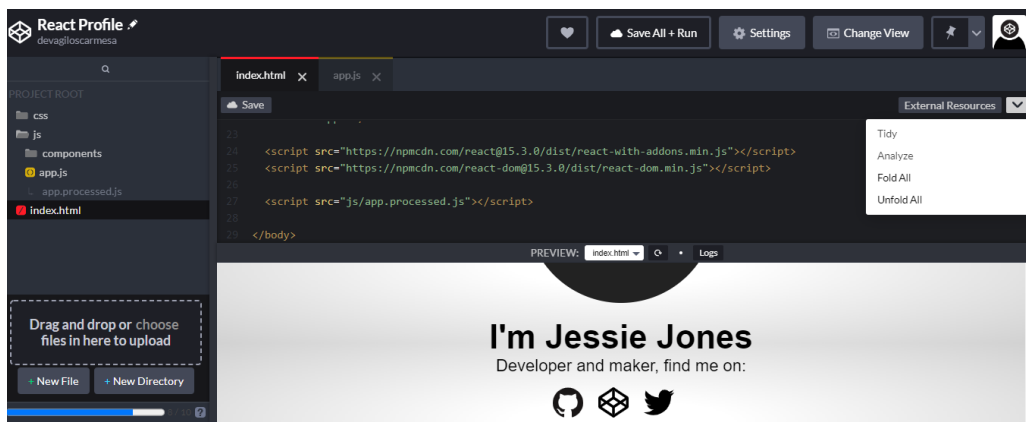
Para instalar REACT en nuestras aplicaciones se puede realizar de dos maneras.

1. Realizar la instalación mediante los CDN es muy sencillo, basta con adicionar la siguiente porción de código dentro de nuestro archivo html.

```
<script crossorigin src="https://unpkg.com/react@16/umd/react.production.min.js"></script>  
<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
```

**CodePen** es una comunidad en línea para probar y mostrar fragmentos de código HTML, CSS y JavaScript creados por el usuario. En esta plataforma se irán dejando algunos enlaces donde se pueda encontrar los ejemplos que se vayan ilustrando.

Al finalizar se podrá tener integrado **react** y **react-dom** para empezar a hacer uso de la librería. El paquete **react-dom** proporciona métodos específicos del DOM que pueden ser utilizados en el nivel más alto de tu aplicación como una vía de escape del modelo de React si así lo necesitas. La mayoría de tus componentes no deberían necesitar usar este módulo.



<https://codepen.io/devagiloscarmesa/project/editor/ZqeBRk>





2. La siguiente forma es la más recomendada, ya que se instala todo el ambiente con las diferentes dependencias que facilitan el trabajo con REACT.

Lo primero que se debe tener en cuenta es instalar los siguientes programas que se listan a continuación para facilitar la creación de una aplicación en REACT.

**Node.js:** es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Gracias a node se podrá gestionar los paquetes necesarios que se necesiten instalar en el proyecto de REACT.



Para instalar node.js, se debe descargar desde el siguiente enlace <https://nodejs.org/es/download/>. Dependiendo de la versión de su sistema operativo.

hoy.

LTS Recomendado para la mayoría	Actual Últimas características	
 Instalador Windows node-v12.18.2-x64.msi	 Instalador macOS node-v12.18.2.pkg	 Código Fuente node-v12.18.2.tar.gz
Instalador Windows (.msi)	32-bit	64-bit
Binario Windows (.zip)	32-bit	64-bit
Instalador macOS (.pkg)	64-bit	
Binario macOS (.tar.gz)	64-bit	
Binario Linux (x64)	64-bit	
Binario Linux (ARM)	ARMv7	ARMv8
Código Fuente	node-v12.18.2.tar.gz	

Plataformas adicionales





Una vez se tenga instalado node, se puede verificar la versión que quedo en la máquina. Por lo cual debemos abrir una **terminal** (la terminal, CMD o símbolo del sistema en Windows, es desde donde se puede ejecutar los comandos que se reconozcan desde el sistema operativo, tales como dir, mkdir, cd, etc) y procedemos a ejecutar el siguiente comando: node -v

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\W1803>node -v
v12.16.3

C:\Users\W1803>
```

**NPM:** Node Package Manager o simplemente npm es un gestor de paquetes, el cual hará más fáciles nuestras vidas al momento de trabajar con Node, ya que gracias a él podremos tener cualquier librería disponible con solo una línea de código, npm nos ayudará a administrar nuestros módulos, distribuir paquetes y agregar dependencias de una manera sencilla.



Con la instalación de node.js, se instala NPM, por ello se puede verificar la versión instalada desde la **terminal**, ejecutando el siguiente comando: npm -v.

```
Símbolo del sistema
C:\Users\W1803>npm -v
6.14.5

C:\Users\W1803>
```



Estas versiones pueden variar según el equipo, no necesariamente deben aparecer la misma versión que aparece en esta guía.

**Yarn:** YARN es un gestor dependencias de JavaScript, que está enfocado en la velocidad y la seguridad, y a diferencia de otros gestores como NPM, YARN es muy rápido y muy fácil de usar.



A continuación, se creará la primera aplicación REACT, para ello se empleará **Create React App**, esta es un ambiente cómodo para aprender React, y es la mejor manera de comenzar a construir una nueva aplicación de página única usando React.

**Create React App** configura tu ambiente de desarrollo de forma que puedas usar las últimas características de Javascript, brindando una buena experiencia de desarrollo, y optimizando tu aplicación para producción. Necesitarás tener Node  $\geq 8.10$  y npm  $\geq 5.6$  instalados en tu máquina. Para crear un proyecto ejecuta:

```
npx create-react-app mi-aplicacion
```

Una vez se completa la ejecución de este comando ya se tendrá creada el scaffolding de la aplicación.



```
Símbolo del sistema
C:\Users\W1803\REACT>npx create-react-app mi-aplicacion

Creating a new React app in C:\Users\W1803\REACT\mi-aplicacion.

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

yarn add v1.19.2
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.12: The platform "win32" is incompatible with this module.
info "fsevents@1.2.12" is an optional dependency and failed compatibility check. Excluding it from installation.
info fsevents@2.1.2: The platform "win32" is incompatible with this module.
info "fsevents@2.1.2" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.17.1" has unmet peer dependency "typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta".
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 14 new dependencies.
info Direct dependencies
├─ cra-template@1.0.3
├─ react-dom@16.13.1
├─ react-scripts@3.4.1
└─ react@16.13.1
info All dependencies
├─ @babel/plugin-syntax-typescript@7.10.4
├─ @babel/plugin-transform-flow-strip-types@7.9.0
├─ @babel/plugin-transform-runtime@7.9.0
├─ @babel/plugin-transform-typescript@7.10.4
├─ @babel/preset-typescript@7.9.0
├─ babel-preset-react-app@9.1.2
└─ cra-template@1.0.3
```

Al finalizar, se ingresa a la carpeta y se abre el proyecto con algún editor, en esta guía se usa Visual Studio Code, como se visualiza en la imagen. También se pueden visualizar algunos comandos que se pueden ejecutar para la ejecución de la aplicación.

```
Símbolo del sistema
Done in 12.74s.

Created git commit.

Success! Created mi-aplicacion at C:\Users\W1803\REACT\mi-aplicacion
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd mi-aplicacion
  yarn start

Happy hacking!

C:\Users\W1803\REACT>cd mi-aplicacion
C:\Users\W1803\REACT\mi-aplicacion>code .
```



## Scripts

Estos scripts, serán los comandos que más se usen para la ejecución del proyecto en REACT:

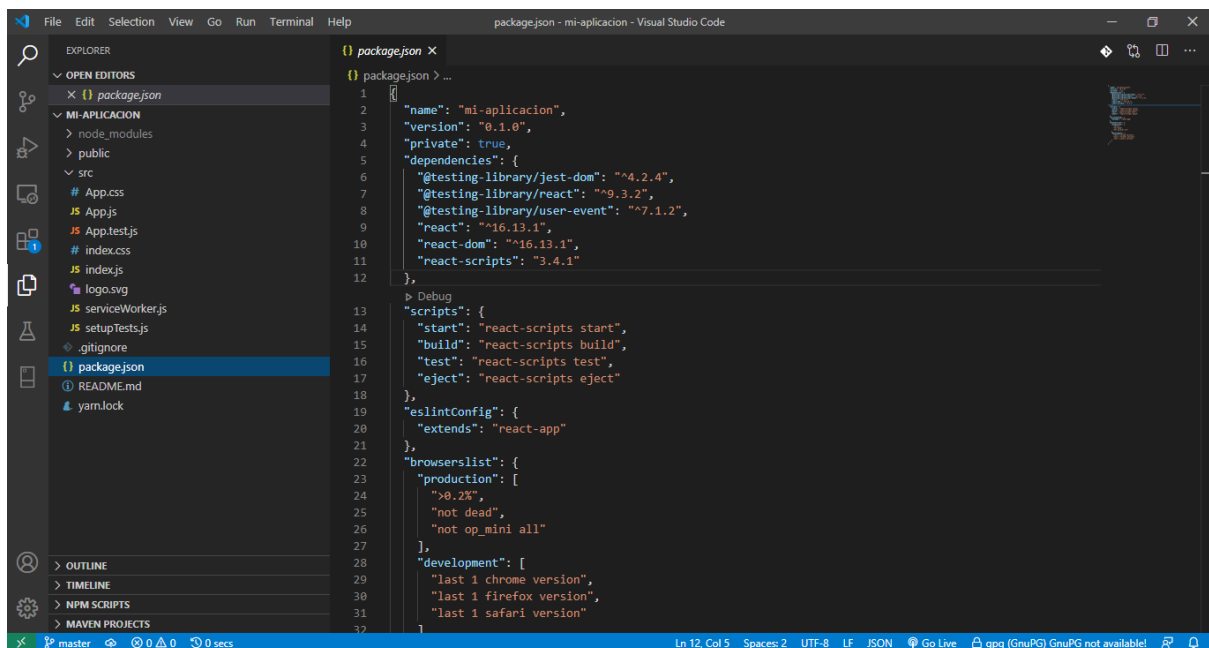
***npm start o yarn start:*** inicia el servidor de desarrollo y abre un navegador con la aplicación.

***npm test o yarn test:*** ejecuta las pruebas.

***npm run build o yarn build:*** empaqueta la aplicación para producción en la carpeta build.

***npm run eject o yarn eject:*** permite cambiar manualmente las librerías y configuración que utiliza create-react-app por defecto. Ten cuidado con este comando, una vez lo ejecutas no hay marcha atrás.

Una vez abierta la aplicación se podrá ver todas las secciones del proyecto creado.



```
1 {
2   "name": "mi-aplicacion",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^4.2.4",
7     "@testing-library/react": "^9.3.2",
8     "@testing-library/user-event": "^7.1.2",
9     "react": "^16.13.1",
10    "react-dom": "^16.13.1",
11    "react-scripts": "3.4.1"
12  },
13  "scripts": {
14    "start": "react-scripts start",
15    "build": "react-scripts build",
16    "test": "react-scripts test",
17    "eject": "react-scripts eject"
18  },
19  "eslintConfig": {
20    "extends": "react-app"
21  },
22  "browserslist": {
23    "production": [
24      ">0.2%",
25      "not dead",
26      "not op_mini all"
27    ],
28    "development": [
29      "last 1 chrome version",
30      "last 1 firefox version",
31      "last 1 safari version"
32    ]
33  }
34 }
```

Edite el archivo src/App.js, para adicionar una etiqueta p con el mensaje: Hola Mundo.



```
src > JS Appjs > App
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.js</code> and save to reload.
12        </p>
13        <p>Hola Mundo</p>
14      </div>
15    );
16  }
17
18  <div className="App-link"
19    href="https://reactjs.org"
20    target="_blank"
21    rel="noopener noreferrer">
22    Learn React
23  </div>
24  </div>
25  );
26  }
27
28  export default App;
```

Por último, ejecute la aplicación con el comando: **yarn start** o **npm start**. Para lo cual abra una terminal que se encuentra en la ruta: Terminal -> New Terminal.

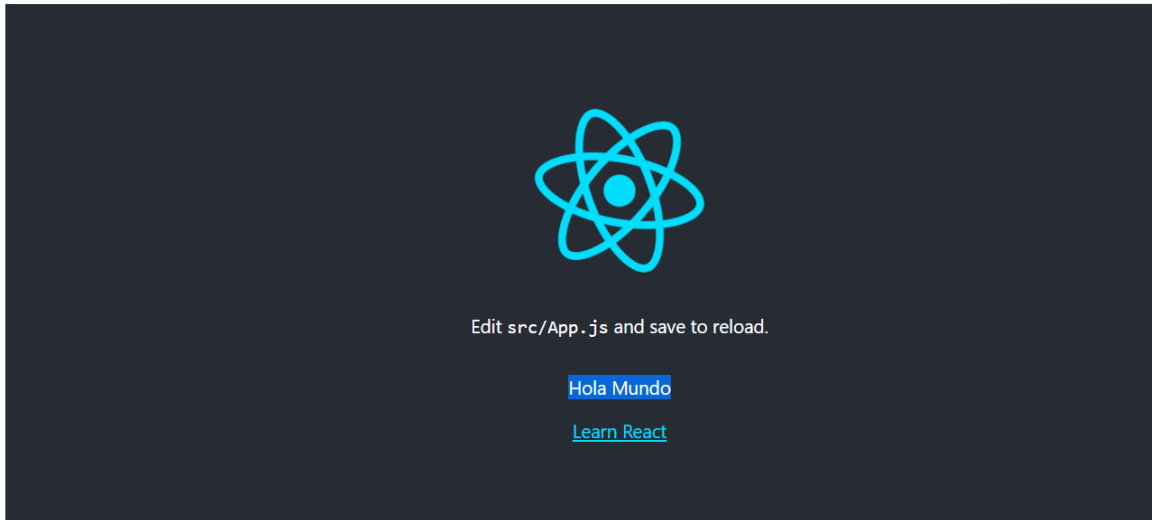
```
src > JS Appjs > App
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.js</code> and save to reload.
12        </p>
13        <p>Hola Mundo</p>
14      </div>
15    );
16  }
17
18  <div className="App-link"
19    href="https://reactjs.org"
20    target="_blank"
21    rel="noopener noreferrer">
22    Learn React
23  </div>
24  </div>
25  );
26  }
27
28  export default App;
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\W1883\REACT\mi-aplicacion> yarn start
```

De esta manera, ya puede visualizar la primera aplicación en REACT.



## Estructura de un proyecto en React

Un proyecto en React, en principio, tendrá un solo archivo HTML, y al menos un archivo JavaScript desde el que importamos la librería de React. Así que un proyecto mínimo tendrá la siguiente estructura:

```
1 mi-aplicacion
2 |   index.html
3 |   index.js
4 |   <dependencies>/
5 |   |   react.js
6 |   |   react-dom.js
```

Sin embargo, trabajaremos con una estructura bien organizada para crear proyectos de mediano tamaño con node y npm más parecida a esta:



```
1  mi-aplicacion
2  |  .gitignore
3  |  package.json
4  |  node_modules
5  |  |  react
6  |  |  react-dom
7  |  public
8  |  |  index.html
9  |  src
10 |  |  images
11 |  |  |  logo.png
12 |  |  stylesheets
13 |  |  |  index.scss
14 |  |  |  index.css
15 |  |  components
16 |  |  |  a-component.js
17 |  |  index.js
```

**package.json:** este archivo tipo manifiesto, cuenta con las configuraciones de la aplicación, así como la lista de dependencias para instalar desde npm cuando se ejecuta npm install.

**node\_modules:** la carpeta node\_modules contiene bibliotecas descargadas de npm. No debe empujarlo a github (incluso debe agregarlo a su .gitignore), porque todos los que clonan su repositorio pueden descargarlo ellos mismos (según su paquete.json).

**src:** aquí incluiremos el contenido de nuestro componente. Su estructura puede variar en base a nuestras necesidades. El código que definamos aquí es el que webpack **transpilará** (tipo especial de compilador que traduce de un lenguaje fuente a otra fuente también de un nivel de abstracción parecido) y minificará para dejar el resultado en la carpeta build.

**public/index.html:** la plantilla de la aplicación. En public van todos los archivos estáticos que necesites incluir en la plantilla.

**src/index.js:** el punto de entrada de JavaScript.

**build:** en esta carpeta se generará nuestro componente minificado y transpilado a ES6, el cual usarán las aplicaciones cuando lo instalen vía NPM.

**. gitignore** es donde definimos los archivos y carpetas que no deben subirse a git.





## PoliTips

Los PoliTips son secciones donde se profundizará algunos temas esenciales para el desarrollo de las futuras guías y se podrán ir evidenciando en algunas de las guías.

En esta primera entrega se profundizará el tema de **Eventos JavaScript**.

## Eventos JavaScript

Hasta el momento se ha visto la forma de emplear JavaScript para cambiar contenidos, estilos, etc. Pero hasta ahora nuestro script (código JavaScript) se ejecutaba al cargar la página. En esta sección emplearemos la interacción de eventos de los usuarios con un aplicativo web para modificar el contenido de esta. Vamos a ver algunos ejemplos de acciones que implican el uso de eventos:

- Mostrar una alerta cuando la usuaria hace click en un botón.
- Cambiar el tamaño de una cabecera fija cuando la usuaria llega a un punto de scroll.
- Abrir una sección oculta de un formulario cuando hago click sobre un botón.
- Cerrar una ventana modal cuando termina un temporizador de 15 segundos (aquí no hay acción de la usuaria).
- Deshabilitar algunos campos de un formulario, cuando el usuario seleccione una opción de un select.
- Enviar una petición al servidor para pedir los datos de los artículos que coinciden con la búsqueda, cuando el usuario pincha en el botón de buscar en Amazon. Y cuando los datos del servidor llegan al navegador, pintarlos en la página.

Es importante entender que nosotros no creamos eventos desde JavaScript, sino que JavaScript nos provee el mecanismo para detectar cuando se ha ejecutado algún evento dentro de un aplicativo web. Algunos ejemplos de eventos son:

- Click en un botón.
- Scroll en la página.
- Cambio en el contenido de un input.
- Expira un temporizador.
- Llegan los datos del servidor.



## Escuchando eventos desde JavaScript

Escuchar un evento desde JavaScript, es decirle al navegador que vigile un determinado elemento de HTML, y cuando algo pase sobre él, ejecute una función que se ha definido previamente. Técnicamente, registramos en el navegador una **función escuchadora** o **listener** sobre un elemento para que ejecute una **función manejadora de eventos** o **handler** cuando el evento suceda. Los principales eventos que se pueden monitorear desde JavaScript se muestran a continuación.

Tipo de evento	Nombre con prefijo on (eliminar cuando proceda)	Descripción aprenderaprogramar.com
Relacionados con el ratón	onclick	Click sobre un elemento
	ondblclick	Doble click sobre un elemento
	onmousedown	Se pulsa un botón del ratón sobre un elemento
	onmouseenter	El puntero del ratón entra en el área de un elemento
	onmouseleave	El puntero del ratón sale del área de un elemento
	onmousemove	El puntero del ratón se está moviendo sobre el área de un elemento
	onmouseover	El puntero del ratón se sitúa encima del área de un elemento
	onmouseout	El puntero del ratón sale fuera del área del elemento o fuera de uno de sus hijos
	onmouseup	Un botón del ratón se libera estando sobre un elemento



	contextmenu	Se pulsa el botón derecho del ratón (antes de que aparezca el menú contextual)
<b>Relacionados con el teclado</b>	onkeydown	El usuario tiene pulsada una tecla (para elementos de formulario y body)
	onkeypress	El usuario pulsa una tecla (momento justo en que la pulsa) (para elementos de formulario y body)
	onkeyup	El usuario libera una tecla que tenía pulsada (para elementos de formulario y body)
<b>Relacionados con formularios</b>	onfocus	Un elemento del formulario toma el foco
	onblur	Un elemento del formulario pierde el foco
	onchange	Un elemento del formulario cambia
	onselect	El usuario selecciona el texto de un elemento input o textarea
	onsubmit	Se pulsa el botón de envío del formulario (antes del envío)
	onreset	Se pulsa el botón reset del formulario
<b>Relacionados con ventanas o frames</b>	onload	Se ha completado la carga de la ventana
	onunload	El usuario ha cerrado la ventana
	onresize	El usuario ha cambiado el tamaño de la ventana

Tabla: Tabla que muestra la lista de eventos JavaScript. Recuperado de [https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=842:lista-de-eventos-javascript-on-click-dblclick-mouseover-mouseout-change-submit-keypress-cu01159e&catid=78&Itemid=206](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=842:lista-de-eventos-javascript-on-click-dblclick-mouseover-mouseout-change-submit-keypress-cu01159e&catid=78&Itemid=206).



Vamos a ver el ejemplo de mostrar una alerta pulsando un botón.

Dado este HTML:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de alerta</title>
  </head>
  <body>
    <button type="button" name="button" class="alert">Alerta</button>
    <script type="text/javascript" src="js/index.js"></script>
  </body>
</html>
```

Para empezar, tendremos recoger de HTML el elemento sobre el que queremos escuchar eventos. Para ello, usaremos nuestro ya habitual **querySelector**.

```
const button = document.querySelector('.alert');
```

A continuación, vamos a usar el método **addEventListener** de los elementos de HTML para escuchar eventos. Le pasaremos 2 parámetros: el tipo de evento a escuchar y la función que tiene que ejecutar cuando suceda el evento. Primero vamos a definir la función manejadora (**handler**), y luego registramos la función escuchadora (**listener**).

```
// elemento de HTML
const button = document.querySelector('.alert');

// handler
const showAlert = () => console.log('Alerta');

// listener sobre el elemento, con tipo de evento y handler
button.addEventListener('click', showAlert);
```

Este ejemplo lo puedes probar en el siguiente link:

<https://codepen.io/devagiloscarmesa/pen/abdqKEw?editors=1011>



## Quiero saber más ...

- <https://webpack.js.org/>
- <https://babeljs.io/>
- Para este proyecto se deja un vídeo de apoyo para la configuración y creación de un proyecto en REACT.

## Taller

1. Crear un proyecto REACT con el nombre del PPI, teniendo en cuenta de usar **Create React App**.
2. Compile el proyecto y evidencie la creación de la carpeta **build** (Toma de pantalla).
3. Modifique la aplicación agregando una tabla, donde esten los datos de los integrantes de PPI con las siguientes columnas: Nombres, Apellidos, Correo y Celular.
4. Ejecute el proyecto y muestre la evidencia con una toma de pantalla.
5. Crear un proyecto en GitHub donde se sincronice el proyecto de REACT creado en el punto 1.
6. ¿Qué es Babel y WebPack y para que se usa en REACT?
7. Crear otro proyecto HTML básico, que cuente con una caja de texto, a esta caja de texto se le deben capturar todos los eventos relacionados con el teclado y determine en que evento se puede determinar si la palabra ingresada en la caja de texto es palíndromo, si lo es mostrar un alert con el mensaje "Es palíndromo".