

React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma

*Ricardo Neftali Lazcano Calixto, Luis Ángel Valencia González,
Daniel Esteban Baena Díaz y Ricardo Venegas Guzmán*

Resumen

El internet, los teléfonos inteligentes, las aplicaciones móviles y la nueva forma de comunicarnos han cambiado la manera en que vivimos a nivel mundial. Se estima que nueve de cada diez personas cuentan con un teléfono móvil con capacidad de proveer aplicaciones, y que éstas se encuentren conectadas a internet. Lo anterior ha generado una alta demanda para el desarrollo de aplicaciones móviles y, por ende, empresas como Facebook están tomando ventaja de ello al crear librerías que permitan a todos los desarrolladores crear aplicaciones móviles de forma rápida y sencilla. Con la creación de la librería de React Native, Facebook ha comenzado a generar mayor interés entre la gente para que utilicen sus tecnologías.

React Native permite crear aplicaciones móviles usando sólo JavaScript. Para los desarrolladores web esto es de mucha ayuda debido a que no necesitan aprender un nuevo lenguaje de programación para incursionar en el desarrollo de aplicaciones móviles. Con React Native no se crea una aplicación web móvil, una aplicación HTML5 o una aplicación híbrida, se crea una aplicación móvil nativa que no se distingue fácilmente de entre una aplicación nativa hecha con Java para Android o con Objective C para iOS.

Actualmente React Native se utiliza para el desarrollo de diferentes aplicaciones móviles de uso diario, como redes sociales, en las que destacan Facebook, Instagram y Uber Eats entre otras.

Palabras clave: React Native, Javascript, framework, Android, multiplataforma.

REACT NATIVE SHORTENING THE DISTANCE BETWEEN DEVELOPMENT AND CROSS-PLATFORM MOBILE DESIGN

Abstract

Internet, smartphones, mobile applications and the new way of communicating have changed the way we live worldwide. It is estimated that nine out of ten people have a mobile phone capable of supporting applications connected to the Internet. This has generated a high demand for the development of mobile applications and, therefore, companies such as Facebook are taking advantage by creating libraries that allow all developers to create mobile applications quickly and easily. With the creation of the library of React Native, Facebook has begun to generate more interest among people to use its technology.

DOI: <http://doi.org/10.22201/codeic.16076079e.2019.v20n5.a5>



React Native allows you to create mobile applications using only JavaScript. For web developers, this is very helpful because they do not need to learn a completely different programming language to venture into mobile application development. With React Native, neither a mobile web application, an HTML5 application nor a hybrid application is created; instead, a native mobile application is created that is not easily distinguished from a native one made with Java for Android or with Objective C for iOS.

Currently, React Native is used for the development of different mobile applications for daily use, such as social networks, which include Facebook, Instagram and Uber Eats among others.

Keywords: React Native, Javascript, framework, Android, multiplatform.

Ricardo Neftali Lazcano Calixto

icnef2121@gmail.com

Egresado de la carrera en Tecnologías de la Información y Comunicación en 2017 por parte de la Universidad Tecnológica de Gutiérrez Zamora del estado de Veracruz. Estudiante de la Maestría en Ingeniería de software en el Instituto Tecnológico Superior de Poza Rica del estado de Veracruz, becado por el Consejo Nacional de Ciencia y Tecnología (CONACYT). Miembro activo de Saturdays.AI Guadalajara.

Interesado en el desarrollo de software e inteligencia artificial. Actualmente se encuentra desarrollando una librería *open source* para el desarrollo de aplicaciones móviles en React Native, con el propósito de mejorar la experiencia de desarrollador con firebase y react native.

Luis Ángel Valencia González

angel.valenciagz@gmail.com

Nacido en la actual Ciudad de México el 15 de julio de 1994. A los 5 años de edad se trasladó al estado de Veracruz. En el año 2013 inició la licenciatura en sistemas computacionales. En 2017 comenzó la maestría en Sistemas Computacionales con especialidad en ingeniería de software, de la que actualmente se encuentra realizando una tesis enfocada a visión artificial.

Daniel Esteban Baena Díaz

daniel.esteban.baena@hotmail.com

Nació en la ciudad de Tuxpan, Veracruz, pero radica en el Municipio de Tamiagua, Veracruz. En 2009 ingresó en el Instituto Tecnológico Superior de Poza Rica para estudiar la carrera de Ingeniería Mecatrónica optando la especialidad de Manufactura. En 2015, tras egresar, decidió realizar el posgrado de especialidad en Ingeniería Mecánica en el Instituto Tecnológico Superior de Poza Rica y en 2017 ingresó a la maestría de Sistemas Computacionales en la misma institución. De manera paralela realizó un curso de inglés en la escuela Technical English. Ha participado en varios congresos y concursos de Robótica y aplicaciones. Asimismo, ha publicado varios artículos y ganado premios en competencias locales y nacionales.

Ricardo Venegas Guzmán

ricardo.venegas@itspozarica.edu.mx

Ingeniero en Comunicaciones y Electrónica, egresado del Instituto Politécnico Nacional, con Maestría en Ciencias Computacionales, especialidad en Sistemas Digitales y Control por el Instituto Tecnológico y de Estudios Superiores de Monterrey. Tiene experiencia en investigación en el Instituto de Investigaciones Eléctricas, Cuernavaca, Morelos.

Sus áreas de investigación son los sistemas de monitoreo en tiempo real, sistemas de balanceo de rotores, análisis y graficación de señales de vibraciones para equipo rotatorio aplicadas a sistemas de mantenimiento predictivo y preventivo de las plantas termoeléctricas de Comisión Federal de Electricidad.

Actualmente se encuentra desempeñando labores de docencia, a nivel licenciatura y posgrado en el Instituto Tecnológico Superior de Poza Rica y en la Universidad Veracruzana. Imparte cátedras como Microprocesadores, Microcontroladores, Inteligencia Artificial, Redes Neuronales, Estadística, Teoría de la computación, Visión Artificial.

Introducción

En América Latina nueve de cada diez personas utilizan un teléfono inteligente según datos recabados por la segunda edición del estudio *IMS Mobile en Latam*, 81% de la población utiliza el sistema operativo Android, además de que se confirma que esta plataforma es el medio más utilizado para navegación web (Internet Media Services, 2016).

Dada esta información podemos entender fácilmente que el desarrollo de aplicaciones móviles es un área de suma importancia en el mercado y por tanto en el área de investigación también. Debido a ello ha surgido la necesidad en los desarrolladores móviles de utilizar un sólo lenguaje de programación tanto para iOS como para Android. Mientras que los desarrolladores web buscan reutilizar sus conocimientos web para incursionar en el mundo de las aplicaciones móviles, como una alternativa nace *cordova*, un empaquetador de código HTML, CSS y JavaScript, que renderiza (genera una imagen realista de un modelo 3D) el contenido dentro de un WebView¹ de Android. No obstante, esto sacrifica el rendimiento de las aplicaciones, además de que muestra un contenido poco fluido y de no tan buen aspecto.

De acuerdo con Holmes, E. y Bray, T. (2015), React Native cambia el juego con aplicaciones que son verdaderamente nativas. No usa WebView, sino que transpila² JavaScript a los idiomas nativos. Piense en ello como componentes interfaces de usuario (UI, del inglés, *User Interfaces*) nativos, controlados por un cerebro de JavaScript. El resultado es una experiencia de usuario casi indistinguible de cualquier otra aplicación nativa y una experiencia de desarrollador que aprovecha los increíbles beneficios de productividad de JavaScript y el ambiente de desarrollo de React.

¿Qué es y para qué sirve?

Antecedentes

Antes de la aparición de teléfonos inteligentes, Facebook trató de estar en los teléfonos con diferentes sistemas operativos que hoy son obsoletos, pero el funcionamiento de esta aplicación en aquellos tiempos era limitado. Con la aparición de iOS y Android, HTML5 fue la mejor solución, haciendo de Facebook una aplicación híbrida tomando su web móvil y adaptándola en estos dos sistemas operativos (Alonso, 2015). En un principio ésta no fue una mala idea pues Facebook no era tan grande ni el uso móvil tan extendido. Sin embargo, el haber adaptado su aplicación con HTML5 en iOS y Android hizo que la función fuera extremadamente lenta y la usabilidad inadecuada para generar beneficios en el uso móvil.

¹ Aplicación nativa de android que funciona como navegador web.

² *Transpilar*. Producir otro programa en otro lenguaje cuyo comportamiento es el mismo que el original.

En el año 2013 se lanza React, una librería creada por Jordan Walke, un ingeniero de software de Facebook. Esta librería hasta la fecha es utilizada por grandes empresas como Netflix, Uber, etcétera; y, por supuesto, es mantenida por Facebook, Instagram y otras comunidades de desarrolladores independientes y compañías. La importancia del desarrollo de esta librería es la capacidad de desarrollar aplicaciones móviles de forma nativa a los sistemas operativos más importantes del momento: iOS y Android.

React

React, también conocida como React.js, es una librería que funciona bajo JavaScript, por lo que es de código abierto. Así, tiene la ventaja de que no sólo se utiliza del lado del cliente sino también puede estar presente del lado del servidor y trabajar en conjunto. React está enfocado a hacer funciones. Trabaja tomando las actualizaciones del estado de página conforme el desarrollador va creando nuevo código, ya sea agregando nuevos complementos en la aplicación en desarrollo o agregando lógica en la misma. Es entonces donde entran en acción esas funciones, pues toman las actualizaciones de estado de la página y las traducen en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar dichas funciones para determinar una nueva representación virtual de la página y en seguida se traduce automáticamente ese resultado en los cambios del DOM (*Document Object Model*, una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML) necesarios para reflejar la nueva presentación de la página (Caballero, 2018).

Pareciera que todas estas funciones podrían ser algo lentas y perjudicar la agilidad del desarrollo, pero React cuenta con un algoritmo de *diffing*³ bastante eficiente, que identifica lo que ha cambiado para determinar las diferencias entre la representación virtual de la página actual y la nueva. A partir de esas diferencias, hace el conjunto mínimo de cambios necesarios en el DOM. A esto se le ha nombrado DOM virtual, que hace selectivamente subárboles de los nodos⁴ sobre la base de cambios de estado, desarrollando esto con la menor cantidad de manipulación DOM posible, con el fin de mantener los componentes actualizados, estructurando sus datos.

Actualmente React también puede ser usada en diferentes entornos de desarrollo. Existe React para desarrollo web, para aplicaciones de escritorio que en conjunto con Electron (un *framework*⁵ para JavaScript que permite el desarrollo de aplicaciones enriquecidas de escritorio mediante el uso de tecnologías web) puede desarrollar interesantes aplicaciones para PC. Para aquellos desarrolladores interesados en la realidad virtual existe React VR, el cual es un entorno de desarrollo a través de JavaScript para realizar aplicaciones funcionales con realidad virtual, siendo éste uno de los frameworks más completos e interesantes, lo que facilita diferentes cuestiones a la hora de desarrollar y explorar otros entornos de desarrollo (Boduch, 2017).

³ *Diffing*: algoritmo que se basa en resolver un problema de subsecuencia común más larga.

⁴ Una ramificación de un recorrido de árboles, refiriéndose al proceso de visitar de una manera sistemática, exactamente una vez, cada nodo en una estructura de datos de árbol.

⁵ Un *framework* es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

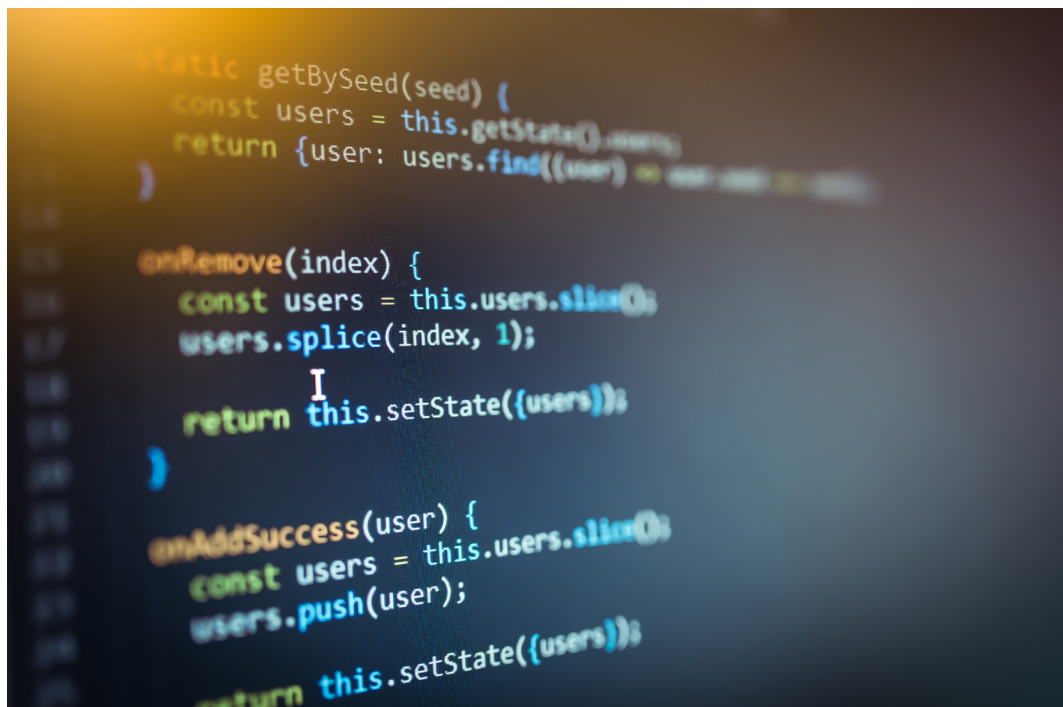
React Native

⁶ *Hackathon*. Es un término usado en las comunidades hacker para referirse a un encuentro de programadores cuyo objetivo es el desarrollo colaborativo de software y hardware.

Para React Native, todo comenzó como un proyecto de *hackathon*⁶ interno entre las paredes de las oficinas de Facebook y desde entonces se ha convertido en uno de los marcos más populares. React Native hizo algo que los desarrolladores web habían intentado hacer durante varios años antes del *hackathon*: escribir aplicaciones móviles en JavaScript (Novick, 2017).

Muchas de los conceptos introducidos por React son aplicados en React Native; por ejemplo, mejores técnicas de gestión del estado, un flujo de datos unidireccional en aplicaciones, construcción de *ui* basada en componentes y mucho más.

Actualmente es compatible con iOS y Android, y debido a su aceptación y éxito hay planes para expandirse a otras plataformas. La innovación principal de React Native es que a pesar de que las aplicaciones están escritas en JavaScript son compiladas en código nativo, por lo que su rendimiento es mucho mejor que las llamadas aplicaciones híbridas. Dichas aplicaciones son escritas en JavaScript, HTML y CSS y se ejecutan en WebView (un navegador integrado dentro de una aplicación). Además, React Native ofrece una experiencia de desarrollo similar al de la Web, como es el caso de la recarga en tiempo real de su aplicación durante el desarrollo, algo que es muy agradable.



```
static getBySeed(seed) {
  const users = this.getState().users;
  return {user: users.find((user) => user.seed === seed)};
}

onRemove(index) {
  const users = this.users.slice();
  users.splice(index, 1);
  return this.setState({users});
}

onAddSuccess(user) {
  const users = this.users.slice();
  users.push(user);
  return this.setState({users});
}
```

Figura 1. Funciones escritas en React Native con Javascript.

Modelo de Rosca

Zagallo (2016) dice que cada vez que se inicia una aplicación React Native se genera una cola de procesos empezando por el tema o hilo principal (*Main*), el cual se genera tan pronto como se inicia la aplicación. Al cargar la aplicación se inicia en automático el hilo JS para ejecutar el código JavaScript. El hilo nativo también escucha los eventos UI como “presionar”, “tocar”, etcétera. Estos eventos se pasan al hilo JS a través del puente React Native (React Render). El hilo JS ejecuta toda la lógica comercial, es decir, el código que escribimos en React Native. Una vez que se carga JavaScript, el hilo JS envía la información sobre lo que se debe representar en la pantalla (*Layout*). Esta información es utilizada por otro hilo llamado *sombra* para calcular los diseños. El hilo sombra es básicamente como un motor matemático que finalmente decide cómo calcular las posiciones de vista. Estas instrucciones se vuelven a pasar al hilo principal para representar la vista. Además de los hilos generados por React Native, también podemos generar hilos en los módulos nativos personalizados que creamos para acelerar el rendimiento de la aplicación, como se detalla en la figura 2.

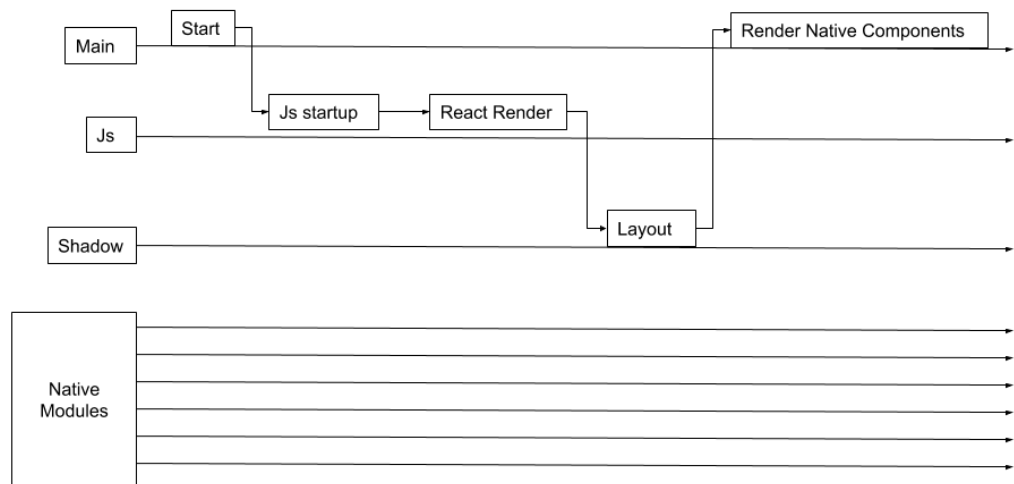


Figura 2. Cola de procesos de React Native.

Ventajas

A continuación se listan algunas de las ventajas de utilizar React Native:

- Tiene una curva de aprendizaje relativamente corta en comparación con el aprendizaje que requieren Objective-C y Java.
- Proporciona una mejor experiencia de desarrollador.
- La depuración móvil es relativamente fácil debido a que se ha logrado llevar la depuración web al mundo móvil.
- No se necesita volver a compilar una aplicación después de cada cambio de código menor, sólo con guardar los cambios realizados se puede recargar en vivo la aplicación.

⁷ API. *Application Programming Interface*, es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

⁸ *Views* nativa. Aplicación interna de una plataforma móvil que funciona como navegador web

- Utiliza Flexbox (modelo de diseño web css3) para mejorar los estilos de las aplicaciones.
- Entre 80% y 90% del código se puede reutilizar entre iOS y Android.
- Tiene acceso directo a todas la APIs⁷ y *views* nativas⁸ que ofrecen los sistemas operativos nativos (Eisenman, 2015).

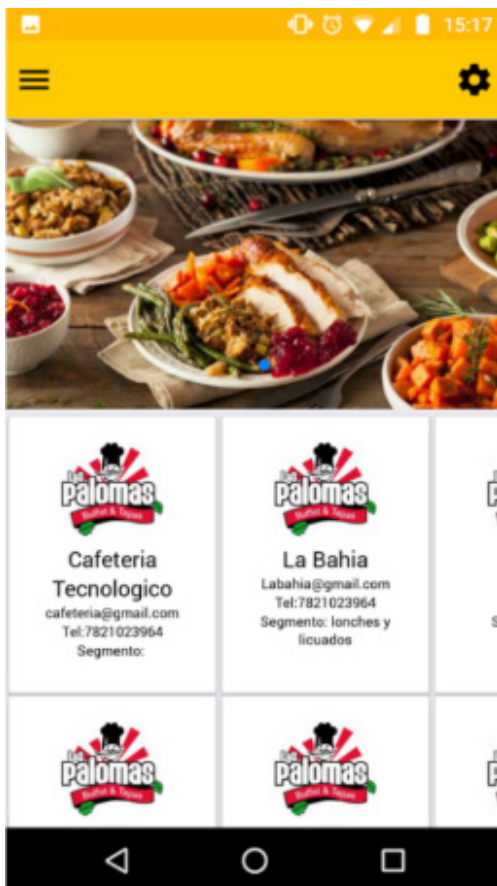
Desventajas

A continuación se listan algunas de las desventajas de React Native:

- No es posible visualizar ningún objeto 3D
- No se recomienda para el desarrollo de videojuegos.
- El entorno de trabajo es costoso de instalar en cuestión de tiempo.
- Poca información a la hora de resolver problemas.
- Si se desea desarrollar aplicaciones para iOS debe de ser en un equipo que corra el sistema operativo del mismo.

Figura 3. Interfaz realizada con React Native, catálogo de restaurantes.

Implementación de React Native



A través del framework React Native se desarrolló una aplicación móvil con el nombre de Market Eat, la cual consiste en la adquisición de alimentos de diversos restaurantes y locales de comida de la ciudad de Poza Rica de Hidalgo, Veracruz. Se crearon módulos de alta para los clientes, así como de logeo, una pantalla de inicio donde se muestran los productos, un menú lateral para poder navegar entre un catálogo de restaurantes tal como se muestra en la figura 3, y los ajustes que necesarios para cada usuario con su perfil. Además, se proporcionó un módulo de alta de productos y la visualización de cada uno; para ello también se utilizó la metodología híbrida para el desarrollo de software EssUP (*Essential Unified Process*, Proceso Unificado Esencial para el desarrollo de software) y Firebase (plataforma para el desarrollo de aplicaciones web y aplicaciones móviles) como gestor de la base de datos de todo el sistema. El uso de este framework facilitó diversas cuestiones a la hora de desarrollar la aplicación, proporcionó un ahorro de tiempo significativo en la fase de aprendizaje del mismo, permitió tiempos de compilación más cortos, además de que nos proporcionó una vista totalmente nativa con la que es muy difícil darse cuenta de que la aplicación no fue desarrollada en el ambiente de desarrollo de Android Studio.

Conclusiones

Sin duda alguna, muchas empresas seguirán desarrollando e implementando nuevas tecnologías, algoritmos y entornos de desarrollo, intentando siempre ser la mejor opción y solución para los desarrolladores a la hora de crear sus proyectos, ya sea que se trabaje de forma personal o para todas las empresas que empiezan a actualizarse debido a las grandes ventajas que adquieren al hacer uso de aplicaciones móviles.

Trabajar con JavaScript tiene la ventaja de que la curva de aprendizaje es rápida y si se tienen conocimientos previos de este lenguaje de programación el tiempo se acorta mucho más. Por otra parte, el trabajar con dos sistemas operativos muy diferentes entre sí (Android y iOS), que corren las aplicaciones de forma distinta, con diferente desempeño y con una interfaz única en cada sistema, hacía que en un principio fuera difícil y costoso el desarrollar una aplicación para ambas plataformas. Por lo tanto, pensar que se tenía que programar dos veces la misma aplicación para cada una de las plataformas definitivamente era una gran desventaja. Hoy en día eso ya forma parte del pasado, sin lugar a dudas React Native es el framework del momento, la mejor alternativa para aprender a desarrollar aplicaciones móviles ya sea para Android o iOS.

Referencias

- ❖ Alonso, M. (2015). Facebook apuesta por unificar el desarrollo móvil con React Native. *Página oficial de BEEVA*. Recuperado de: <https://www.beeva.com/beeva-view/desarrollo/facebook-apuesta-por-unificar-el-desarrollo-movil-con-react-native/>.
- ❖ Zagallo, T. (2016). *Optimising React Native: Tools and Tips* [conferencia]. Recuperado de: <https://www.youtube.com/watch?v=0MIT74erp60&t=135s>.
- ❖ Internet Media Services (2016). IMS Mobile in Latam Study. *Informe digital* Vol. 2, No. 1. Recuperado de: <https://insights.ims corporate.com/mobile2016>.
- ❖ Holmes, E. y Bray, T. (2015). *Getting Started with React Native*. Birmingham: Packt Publishing Ltd.
- ❖ Novick, V. (2017). *React Native-Building Mobile Apps with JavaScript*. Birmingham: Packt Publishing Ltd.
- ❖ Boduch, A. (2017). *React and React Native*. Livery Place, # 35 Livery Street, Birmingham. Packt Publishing Ltd.
- ❖ Caballero, J. (2018). ¿Cómo funciona React.js? *Página Oficial de DevCode*. Recuperado de: <https://devcode.la/blog/como-funciona-reactjs/>.
- ❖ Eisenman, B. (Diciembre 2015). *Learning React Native*. Sebastopol, CA: O'Reilly Media, Inc.

Agradecimientos

Agradecemos al Instituto Tecnológico Superior de Poza Rica por darnos la oportunidad de cursar la maestría y apoyarnos en cada oportunidad que nos invita a seguir creciendo. Al mismo tiempo hacemos mención de los maestros que han estado al pendiente y a disposición, en especial al M. C. Ricardo Venegas Guzmán, sin su ayuda toda esta investigación y proyectos previos no hubieran sido posibles. Y ante todo al programa CONACYT por brindarnos el apoyo y darnos la oportunidad de cumplir este gran sueño llenándonos de metas día a día y hacernos partícipes de servir a la sociedad y a futuras generaciones. Gracias infinitas.

Cómo citar este artículo

- ❖ Lazcano Calixto, Ricardo Neftali, Valencia González, Luis Ángel, Baena Díaz, Daniel Esteban y Venegas Guzmán, Ricardo (2019). React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma *Revista Digital Universitaria* (RDU). Vol. 20, núm. 5 septiembre-octubre. DOI: <http://doi.org/10.22201/codeic.16076079e.2019.v20n5.a5>.

Recepción: 22/03/2019. Aprobación: 03/07/2019
