



React Nivel Avanzado

Fundamentos de React



El mundo actual se caracteriza por un constante avance tecnológico, lo cual implica que periódicamente se crean nuevas formas para establecer diferentes maneras de comunicarse y prepararse para la economía digital del siglo XXI. En ese sentido, la capacidad de escribir códigos es esencial en diferentes campos: la informática, la ingeniería, el diseño y otros más.

La programación está en casi cada aspecto de la vida del ser humano, por lo que saber programar, ser creativo y usar estos recursos para desarrollar nuevas aplicaciones y herramientas contribuye a una mejor calidad de vida para el ser humano.

Previo al desarrollo de cualquier aplicación, es de suma importancia conocer los principales elementos que necesitarás para llevarlo a cabo:

- Sistema operativo.
- Ambiente de desarrollo.
- Capacidades técnicas.
- Lenguaje de programación.

Lo anterior te capacitará para explotar todas las funcionalidades, a través de las cuales harás una aplicación más sencilla de programar, que utilice menos recursos e, inclusive, que te permita usar un mismo lenguaje de programación para todas las plataformas y crear una aplicación nativa.

En este tema revisarás los fundamentos de React Native, basado en JavaScript, que te ayudará a crear aplicaciones móviles nativas, dicho de otro modo, aplicaciones tanto para Android como para iOS que utilizan un mismo lenguaje de programación (JavaScript).

Explicación

En el desarrollo de software Dentro del mundo de desarrollo móvil existen diferentes frameworks que te permitirán la oportunidad de crear diferentes aplicaciones para las diferentes plataformas y sistemas operativos que hay.. Esta amplia disponibilidad de herramientas en un principio puede volverse algo confuso, sin embargo, al momento de desarrollar un proyecto, deberemos analizar las posibilidades que cada uno de los frameworks nos ofrece.

En nuestro caso usaremos el framework React Native con Expo, el cual es Uno de estos frameworks recibe el nombre de React Native.

Reun framework desarrollado por Meta (antes Facebook), a partir de la librería React, la cual se utiliza mucho popularmente para el desarrollo de sitios y aplicaciones web mediante el uso de JavaScript y JSX, permitiéndonos reutilizar componentes, esta es la idea principal de React.reutilizar compontes.

Citando la filosofía del sitio web de React (en inglés) dice: “React te permite construir interfaces de usuario a partir de piezas individuales llamadas componentes”. (React.Dev, s.f.) . La idea de esto es que cada elemento usado que se use en el sitio sea creado como un componente para finalmente construir el sitio a partir de estas piezas, como si de se tratara de un rompecabezas se tratase, de esta forma reducimos el código y también facilitamos el mantenimiento (en la medida de lo posible) para actualizar el sitio más adelante o realizar cualquier cambio que sea necesario.

React Native, por su parte al igual que su contraparte enfocada puramente en la web (ReactJS), nos permite desarrollar aplicaciones móviles utilizando código JavaScript, lo que posteriormente será compilado al lenguaje nativo de cada plataforma, ya sea Android o iOS, incluso web. Esto nos da como principal ventaja usar la misma base de código para más de una plataforma, lo que se traduce en una menor curva de aprendizaje y menos tiempos de entrega.

De acuerdo con Lopez (2022), las ventajas de usar React Native son las siguientes:

1. Solo escribes el código en JavaScript, de tal forma que React Native ejecute la *app* tanto en iOS como en Android.
2. Las funcionalidades de iOS y Android no tendrán que implementarse por separado, ya que se usa el mismo lenguaje de programación (JavaScript).
3. La aplicación se vuelve rápida porque se mezcla el potencial de React con los componentes nativos de cada sistema operativo.

Fundamentos de React Native

JSX y componentes

React Native, al igual que React JS, utiliza JSX, una versión de JavaScript que nos permite escribir código mediante etiquetas sin tener que hacer el proceso de analizar o convertir ciertas partes de la sintaxis, esto en conjunto con un IDE compatible con JSX, lo que nos facilita editar el código de forma más amigable.

Dentro del código JSX podemos llamar a un componente React y enviarle las propiedades que necesite o incluso colocar variables o métodos, sin perder la posibilidad de llamar métodos dentro del mismo componente o importar otros métodos, al igual que otro archivo JavaScript normal.

React Native utiliza los componentes nativos para poder ejecutar las aplicaciones usando un mismo lenguaje: JavaScript. De acuerdo con React Native (2022), se utiliza JSX para transformar el código a cada componente nativo de la plataforma correspondiente. React, al igual que otros frameworks y librerías modernos, usa la extensión de sintaxis JSX es una extensión de sintaxis de JavaScript usado por React.

A continuación, se muestra el típico ejemplo de Hola Mundo:

```
import React from 'react';
import { Text, View } from 'react-native';

const HelloWorldApp = () => {
  return (
    <View
      style={{
        flex: 1,
        justifyContent: "center",
        alignItems: "center"
      }}>
      <Text>Hello, world!</Text>
    </View>
  )
}
export default HelloWorldApp;
```

El código anterior nos genera la siguiente vista al ser compilado mediante React Native.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

En la última línea de código, puedes observar HelloWorldApp, lo cual está creando un nuevo componente. Cada que se compile una aplicación de React Native, se creará uno nuevo, así como todo lo que se vea en la pantalla del dispositivo al momento que se ejecuta la app.

A continuación, se ilustra la manera en que React Native toma el código JavaScript hasta llevarlo a código nativo, dependiendo de la plataforma para la que se esté generando la aplicación.

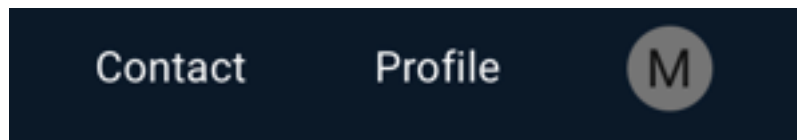
Componentes

La característica clave de React es la composición de componentes, los componentes deben poder trabajar en conjunto, pero no por eso ser poco flexibles, la idea es que cada componente pueda ser adaptable tanto al momento de desarrollarlo, como al momento de actualizar alguna de sus características, sin afectar en la medida de lo posible otros componentes que los contengan.

Una buena práctica es desarrollar componentes que se puedan reutilizar, por ejemplo, si usamos un elemento input de HTML con determinadas clases de CSS en más de una ocasión, podemos considerar convertirlo a un componente y de esta forma evitamos repetir el mismo código. Además, en caso de querer cambiar el estilo, podremos realizar el cambio únicamente en el componente, haciendo que esto se refleje en todos los demás lugares en donde el componente se utiliza.

Otra de las posibilidades que nos ofrece React al usar componentes es combinarlos para crear interfaces más elaboradas, pero a la vez fáciles de entender y mantener.

Tomemos como ejemplo un menú de navegación. Podríamos tener un componente llamado `<NavigationBar />` similar a lo que se muestra a continuación.



Fuente: Mui. (s.f.). Menu. Raterial UI. Documentación.
Recuperado de

A partir del uso de los siguientes componentes:

- `<Header />`
- `<Container />`
- `<UserPicture />`
- `<NavLink />`

Los componentes que se mencionaron anteriormente podrían estar creados de elementos HTML o de otros componentes React.



Es importante conocer todas las herramientas que se pueden utilizar, de tal forma, que sea posible identificar qué se puede hacer y qué no al momento de desarrollar una aplicación.

Conocer todos los elementos darán un mejor resultado en cualquier desarrollo que vayas a implementar.

Previo a aprender un nuevo lenguaje de programación, primero debes conocer todas las herramientas que puedes utilizar para que tu aplicación sea todo un éxito.

Referencias bibliográficas

- Android Developers. (s.f.). *Instalación Android Studio*. Recuperado de <https://developer.android.com/studio/?hl=es-419>
- Chocolately. (s.f.). *The Package Manager for Windows Modern Software Automation*. Recuperado de <https://chocolatey.org/>
- Lopez, R. (2022). *¿Qué es React Native?* Recuperado de <https://www.raullg.com/que-es-react-native/>
- React Native. (2022a). *Learn the Basics*. Recuperado de <https://reactnative.dev/docs/tutorial>
- React Native. (2022b). *Setting up the development environment*. Recuperado de <https://reactnative.dev/docs/environment-setup>
- Mui. (s.f.). *Menu*. Recuperado de <https://mui.com/material-ui/react-menu/>

Lecturas

Para conocer más acerca de los **fundamentos de React**, te sugerimos leer lo siguiente:

- React Native. (2022). *Introduction*. Recuperado de <https://reactnative.dev/docs/getting-started>
- React Native. (2022). *React Fundamentals*. Recuperado de <https://reactnative.dev/docs/intro-react>

Videos

Para conocer más acerca de los **fundamentos de React**, te sugerimos revisar lo siguiente:

- Fazt. (2020, 26 de octubre). *¿Qué es React Native?* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=iIuoJCsAxWc>
- Fernando Herrera. (2021, 5 de abril). 01/12 - *Windows: Instalaciones necesarias - React Native* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=FT4QyQADxgY>

Checkpoint

Asegúrate de:

- Comprender el uso de React Native para el desarrollo de una aplicación.
- Entender las herramientas de React Native para conocer las funciones que puede realizar.
- Reconocer cada componente de React Native para el correcto desarrollo de una aplicación.

Requerimientos técnicos

- Android Studio.
- React.

Pework

- Deberás revisar el **Tema 1. Fundamentos de React**.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.