

INFORME DE TALLER PRÁCTICO I

DATOS DE LA ASIGNATURA	
Nombre(s) del Estudiante(s): Danny Favian Ledesma Valencia	Nivel: Segundo Nivel
Carrera: Desarrollo de Software.	Docente: Ing. Viviana Flores.
Asignatura: Aplicaciones Móviles I	Periodo académico: 2023-2024.

1. TEMA DEL TALLER

Diseño de una aplicación móvil para un formulario de registro de usuarios.

2. DESCRIPCIÓN DEL TALLER

Diseñar y desarrollar una aplicación de compras multiplataforma (Android y IOS) mediante el uso del lenguaje de programación JavaScript y el framework de React Native con la finalidad de obtener una aplicación amigable, cómoda y agradable, sin importar el tipo de dispositivo a utilizar.

3. RESULTADO DE APRENDIZAJE ATADO AL TALLER PRÁCTICO EXPERIMENTAL

Los estudiantes del curso de desarrollo de software se embarcarán en la tarea de diseñar y desarrollar una aplicación que incluirá un formulario de registro de usuarios mediante el uso de React Native y sus componentes nativos.

4. FUNDAMENTACIÓN TEÓRICA DE LA PRÁCTICA

React Native es un framework de desarrollo móvil que permite construir aplicaciones móviles utilizando React y JavaScript.

La principal característica de React Native es su capacidad para crear aplicaciones móviles que se ejecutan en dispositivos iOS y Android con un solo código base.

Algunas características clave de React Native incluyen:

Componentes React:

React Native utiliza componentes React para construir la interfaz de usuario de la aplicación.

Estos componentes son similares a los que se utilizan en React para aplicaciones web, pero están adaptados para su uso en entornos móviles.

Estilo Flexbox:

La maquetación y el estilo en React Native se basan en el modelo de caja de Flexbox, que facilita la creación de interfaces de usuario flexibles y adaptables a diferentes tamaños de pantalla.

JavaScript:

La lógica de la aplicación se escribe en JavaScript, lo que permite a los desarrolladores reutilizar gran parte del código entre las versiones de iOS y Android de la aplicación.

Acceso a Funcionalidades Nativas:

React Native permite acceder a las API nativas de los dispositivos, lo que significa que los desarrolladores pueden utilizar características específicas de iOS y Android cuando sea necesario.

Hot Reloading:

Facilita el desarrollo al permitir la recarga en caliente de la aplicación para ver los cambios en tiempo real sin reiniciar la aplicación.

Bibliotecas y Módulos Nativos:

Existe una gran cantidad de bibliotecas y módulos nativos disponibles para facilitar el acceso a características específicas del dispositivo y agilizar el desarrollo.

Componentes Nativos de React Native:

React Native incluye un conjunto de componentes básicos que pueden utilizarse para construir interfaces de usuario. Algunos de los componentes nativos más comunes incluyen:

View:

Representa un contenedor que puede contener otros componentes o elementos.

Text:

Muestra texto en la interfaz de usuario.

Image:

Muestra imágenes en la interfaz de usuario.

ScrollView:

Proporciona una vista desplazable para mostrar contenido más allá de las dimensiones de la pantalla.

FlatList:

Es un componente de lista optimizado para grandes conjuntos de datos.

Touchable Components:

Incluyen TouchableOpacity, TouchableHighlight, etc., que permiten crear componentes interactivos que responden al tacto del usuario.

TextInput:

Permite la entrada de texto por parte del usuario.

ActivityIndicator:

Muestra un indicador de carga para informar al usuario sobre la actividad en curso.

Estos son solo algunos ejemplos y React Native proporciona muchos más componentes y API que pueden adaptarse a diversas necesidades de desarrollo móvil.

Ventajas.

React Native permite escribir código una vez y ejecutarlo en múltiples plataformas, como iOS y Android. Esto facilita la gestión y el mantenimiento del código base.

Componentes Reutilizables:

La arquitectura basada en componentes de React Native facilita la reutilización de componentes en diferentes partes de la aplicación o incluso en proyectos diferentes.

Acceso a Funcionalidades Nativas:

React Native permite acceder a las API nativas del dispositivo, lo que significa que los desarrolladores pueden utilizar características específicas de iOS y Android cuando sea necesario.

Desarrollo Rápido:

La recarga en caliente (Hot Reloading) de React Native permite ver los cambios en tiempo real sin tener que reiniciar la aplicación, lo que agiliza el proceso de desarrollo.

Lenguaje Familiar:

Utiliza JavaScript, un lenguaje ampliamente conocido y utilizado en el desarrollo web. Esto facilita la transición para los desarrolladores familiarizados con React o JavaScript.

Optimización de Rendimiento:

React Native utiliza un hilo de fondo para ejecutar operaciones intensivas en la CPU, lo que mejora el rendimiento y la respuesta de la interfaz de usuario.

Actualizaciones Over-the-Air (OTA):

Permite realizar actualizaciones de la aplicación de forma remota sin necesidad de pasar por las tiendas de aplicaciones, lo que facilita la implementación de correcciones de errores y mejoras.

Compatibilidad con Navegadores Web:

React Native Web permite compartir lógica de aplicación entre plataformas móviles y web, lo que puede facilitar la creación de versiones web de tus aplicaciones móviles.

Soporte de Empresas Líderes:

Grandes empresas respaldan React Native, lo que garantiza un soporte continuo y mejoras constantes en la tecnología.

5. DESCRIPCIÓN DE LAS ACTIVIDADES DESARROLLADAS DURANTE LA PRÁCTICA.

El objetivo primordial será crear una interfaz amigable y atractiva para mejorar la experiencia del usuario. Para alcanzar estos criterios, se deberá prestar especial atención a los siguientes puntos:

- **Nombre:** Un campo de texto para que el usuario ingrese su nombre.
- **Apellido:** Un campo de texto para que el usuario ingrese su apellido.
- **Correo Electrónico:** Un campo de texto para que el usuario ingrese su dirección de correo electrónico.
- **Contraseña:** Un campo de texto para que el usuario ingrese su contraseña.
- **Confirmar Contraseña:** Un segundo campo de texto para que el usuario confirme su contraseña.

- **Número de Teléfono:** Un campo de texto numérico para que el usuario ingrese su número de teléfono.
- **Fecha de Nacimiento:** Un componente de selección de fecha para que el usuario ingrese su fecha de nacimiento.
- **Género:** Un menú desplegable para que el usuario seleccione su género (Masculino, Femenino, Otro).
- **Aceptar Términos y Condiciones:** Un checkbox para que el usuario acepte los términos y condiciones de la aplicación.
- **Botón de Registro:** Un botón que envíe la información del formulario.

MATERIALES Y EQUIPOS

Corresponde a los materiales, equipos, insumos, maquinaria, instrumental, etc. disponibles en los laboratorios institucionales o escenarios de aprendizaje asignados.

MATERIAL / EQUIPO/ INSUMO		UTILIDA D
	Laptop	Diseño de la aplicación móvil.
	Internet	Observacion y consulta de React Native









6. IDENTIFICACIÓN DE RIESGOS Y EQUIPOS DE PROTECCIÓN PERSONAL

(SI APLICA) Riesgos genéricos identificados por carrera y los equipos de protección personal específicos a ser utilizados por los estudiantes durante la ejecución de los talleres práctico experimentales.

ITSQMET
INSTITUTO TECNOLÓGICO SUPERIOR
QUITO METROPOLITANO

Normas de seguridad Desarrollo de Software

<div>MANTÉN LA MESA DE TRABAJO LIMPIA Y ORDENADA</div> <div></div> <div>Es importante que se mantenga el orden también mientras se trabaja.</div>	<div>1</div>	<div>2</div> <div>LIMPIA LA MESA DE TRABAJO CON LOS ÚTILES ADECUADOS</div> <div></div> <div>Con la mano o soplando puedes hacerte daño tú o a tus compañeros y compañeras.</div>
<div>APRENDE A USAR LAS HERRAMIENTAS</div> <div></div> <div>Antes de usar una herramienta asegúrate que conoces su forma de uso y sus medidas de seguridad.</div>	<div>3</div>	<div>4</div> <div>UTILIZA LOS MEDIOS DE PROTECCIÓN ADECUADOS</div> <div></div> <div>Usa gafas de seguridad y guantes al manipular materiales y herramientas.</div>
<div>NO COMAS NI BEBAS DENTRO DEL TALLER</div> <div></div> <div>Para evitar incidentes no debe ingresar, alimentos o bebidas al laboratorio.</div>	<div>5</div>	<div>6</div> <div>NO CORRAS O JUEGUES EN EL TALLER</div> <div></div> <div>Puedes poner en peligro a tus compañeros y compañeras si no te comportas adecuadamente.</div>
<div>EVITA RIESGOS DE ENGANCHE</div> <div></div> <div>Recógete el pelo y no laves prendas o accesorios que puedan engancharse a ellas o al material que estés utilizando.</div>	<div>7</div>	<div>8</div> <div>APROVECHA EL MATERIAL</div> <div></div> <div>Intenta ahorrar material, aprovecha todo el espacio y utilízalo en su justa medida. Por ejemplo, no malgastes pegamento.</div>
<div>MANTÉN EL TALLER ORDENADO</div> <div></div> <div>Coloca las herramientas en su lugar al acabar.</div>	<div>9</div>	<div>10</div> <div>COMUNICA CUALQUIER INCIDENCIA O DUDA</div> <div></div> <div>Comunica a tu docente los defectos o averías, pregunta cuando tengas dudas e informa cuando ocurra un incidente.</div>
<div>UTILIZA CONTRASEÑAS SEGURAS</div> <div></div> <div>Que incluyan combinación de letras, números y caracteres y evite guardarlas en el navegador</div>	<div>11</div>	<div>12</div> <div>TEN CUIDADO CON LOS APARATOS ELÉCTRICOS</div> <div></div> <div>No dejes aparatos eléctricos o enchufes cerca de líquidos ni los utilices con las manos mojadas, podrías electrocutarte.</div>

7. ANÁLISIS DE RESULTADOS

Para realizar un análisis de resultados en el contexto de desarrollo de aplicaciones utilizando React Native, es importante considerar diversos aspectos. Aquí hay algunas áreas clave que podrías evaluar:

Rendimiento de la Aplicación:

Evalúa el rendimiento de la aplicación en términos de velocidad de carga, respuesta a eventos táctiles y fluidez en la interfaz de usuario.

Utiliza herramientas de perfilado para identificar posibles cuellos de botella y optimizar el rendimiento.

Experiencia del Usuario:

Recopila comentarios de los usuarios sobre la experiencia general de la aplicación.

Realiza pruebas de usabilidad para identificar áreas de mejora en la navegación y la interacción.

Desarrollo Rápido:

Analiza el tiempo que lleva desarrollar nuevas características o correcciones de errores.

Utiliza la recarga en caliente para evaluar la eficacia de las actualizaciones en tiempo real durante el desarrollo.

Compatibilidad con Dispositivos:

Verifica la compatibilidad de la aplicación con diferentes dispositivos y tamaños de pantalla.

Asegúrate de que la aplicación se comporte correctamente en diferentes versiones de sistemas operativos.

Reutilización de Componentes:

Evalúa la capacidad de reutilización de componentes en diferentes partes de la aplicación y entre proyectos.

Considera el modularidad del código y la facilidad para mantener y escalar la base de código.

Acceso a Funcionalidades Nativas:

Verifica el acceso y la integración exitosa con características nativas específicas de iOS y Android.

Asegúrate de que la aplicación pueda utilizar cámaras, sensores y otras funcionalidades del dispositivo según sea necesario.

Despliegue y Actualizaciones:

Evalúa la facilidad de implementar nuevas versiones de la aplicación en las tiendas de aplicaciones.

Considera la posibilidad de realizar actualizaciones Over-the-Air y su impacto en la experiencia del usuario.

8. CONCLUSIONES

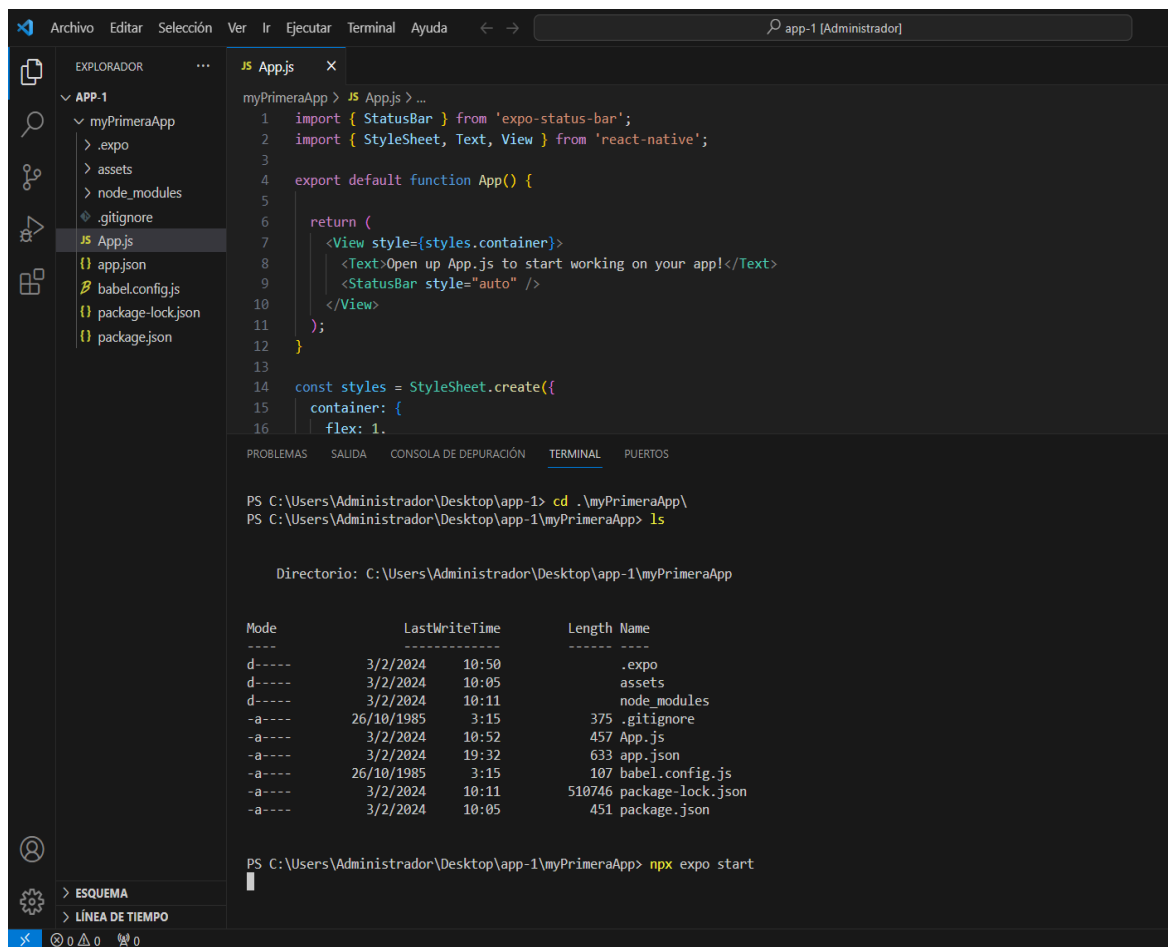
Luego de realizar un análisis exhaustivo de los resultados en el desarrollo de aplicaciones utilizando React Native, puedes llegar a diversas conclusiones

- React Native ha demostrado ser eficiente en términos de desarrollo, permitiendo la creación de aplicaciones para múltiples plataformas con un solo código base.
- La recarga en caliente y otras herramientas han acelerado el ciclo de desarrollo, facilitando la implementación y pruebas rápidas.
- La aplicación proporciona una experiencia de usuario satisfactoria con una interfaz de usuario fluida y tiempos de carga aceptables.

9. RECOMENDACIONES

Recomendaría enfocarse en la optimización continua de la aplicación. Esto implica realizar mejoras constantes en términos de rendimiento, experiencia del usuario, eficiencia del código y actualización de dependencias. La optimización continua garantiza que la aplicación siga siendo competitiva, eficiente y cumpla con las expectativas cambiantes de los usuarios a lo largo del tiempo.

10. EVIDENCIA FOTOGRÁFICA DE LA PRÁCTICA



The screenshot shows a VS Code editor with a project named 'APP-1' in the Explorer. The project structure includes:

- myPrimeraApp
 - .expo
 - assets
 - node_modules
 - .gitignore
 - App.js
 - app.json
 - babel.config.js
 - package-lock.json
 - package.json

The main editor shows the content of 'App.js':

```

1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5
6      return (
7          <View style={styles.container}>
8              <Text>Open up App.js to start working on your app!</Text>
9              <StatusBar style="auto" />
10             </View>
11         );
12     }
13
14     const styles = StyleSheet.create({
15         container: {
16             flex: 1,

```

The Terminal panel shows the following commands and output:

```

PS C:\Users\Administrador\Desktop\app-1> cd .\myPrimeraApp\
PS C:\Users\Administrador\Desktop\app-1\myPrimeraApp> ls

Directorio: C:\Users\Administrador\Desktop\app-1\myPrimeraApp

Mode                LastWriteTime         Length Name
----                -
d-----          3/2/2024   10:50             .expo
d-----          3/2/2024   10:05             assets
d-----          3/2/2024   10:11          node_modules
-a-----        26/10/1985    3:15             375 .gitignore
-a-----          3/2/2024   10:52             457 App.js
-a-----          3/2/2024   19:32             633 app.json
-a-----        26/10/1985    3:15          107 babel.config.js
-a-----          3/2/2024   10:11        510746 package-lock.json
-a-----          3/2/2024   10:05             451 package.json

PS C:\Users\Administrador\Desktop\app-1\myPrimeraApp> npx expo start

```



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

app-1 [Administrador]

EXPLORADOR

APP-1

myPrimeraApp

.expo

assets

node_modules

.gitignore

JS App.js

app.json

babel.config.js

package-lock.json

package.json

JS App.js

myPrimeraApp > JS App.js > ...

1 import { StatusBar } from 'expo-status-bar';

2 import { StyleSheet, Text, View } from 'react-native';

3

4 export default function App() {

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

(node:13036) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
Starting Metro Bundler

> Metro waiting on [exp://192.168.19.30:8081](http://192.168.19.30:8081)

> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using [Expo Go](#)

> Press **s** | switch to development build

> Press **a** | open Android

> Press **w** | open web

> Press **j** | open debugger

> Press **r** | reload app

> Press **m** | toggle menu

> Press **o** | open project code in your editor

> Press **?** | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.

ESQUEMA

LÍNEA DE TIEMPO

0 0 0 0

Lín. 22, col. 1

11. BIBLIOGRAFÍA

<https://elibro.net/en/lc/itsqmet/titulos/106397>

<https://elibro.net/en/lc/itsqmet/titulos/126402>

<https://elibro.net/es/ereader/itsqmet/126402?page=16>