UNIVERSIDAD POLITÉCNICA DE QUERÉTARO

DESARROLLO DE APLICACIONES MOVILES

Grupo: TIID-215

Practica 5 - Reporte

PROFESOR: Iván Isay Guerra López

ALUMNO: Luis Guadalupe Barrón Durán

Componentes

Elementos del Framework que podemos crear y reutilizar para construir las interfaces de nuestras aplicaciones móviles

- Principales
- Nativos
- Comunidad: NativeBase, Shoutem, RN Material UI

Core Components

- <View>
- <Text>
- <Image>
- <ScrollView>
- <TextImput>

Props

Propiedades. Son valores que se pasan a los componentes para configurarlos. Son como los parámetros de una función.

<Button title=”Incrementar” onPress= />

Hooks

Son funciones especiales que react proporciona para que puedas ejecutar o ”engancharte” al sistema de React desde componentes funcionales.

Permiten usar estados, ciclos de vida y otras funcionalidades de React en componentes funcionales, sin tener que escribir clases.

Todos los estados tienen variable y función.

Hooks

1. **useState**:

   Es un hook de React que asignara una variable de estado, con la cual crearemos y manejaremos un estado dentro de un componente funcional

   Se compone de 2 elementos

   1. Variable: valor inicial
   2. Función: encargada del cambio de estado

Practica:

1. Para esta practica debemos de importar el componente de botón dentro de la línea 4, justo después de view. Después en el Main agregamos el componente debajo de <Text>

```
1  //Zonas
2  //1. imports: Zona de importaciones
3  import { StatusBar } from 'expo-status-bar';
4  import { StyleSheet, Text, View , Button} from 'react-native';
5  //2. Main:Zona de componentes
6  export default function App() {
7    return (
8
9      <View style={styles.container}>
10       <Text>Contador: </Text>
11       <Button></Button>
12       <StatusBar style="auto" />
13     </View>
14
```

2. Importamos el Hook de useState como se muestra en la línea 5 y en el método creamos la desestructuración como se muestra en la línea 9, asignando el valor de 0 a useState .

```
2  //1. imports: Zona de importaciones
3  import { StatusBar } from 'expo-status-bar';
4  import { StyleSheet, Text, View , Button} from 'react-native';
5  import React, {useState} from 'react';
6  //2. Main:Zona de componentes
7  export default function App() {
8
9    const [contador, setContador]=useState(0);//desestructuracion
```

3. Debajo del return en la parte donde agregamos el botón agregaremos la función de onPress y con la variable flecha mandamos llamar el método de setContador dentro de este le asignamos el valor de la variable contador y le sumaremos el valor de 1 por cada clic que reciba el botón.

```
11 return (
12
13   <View style={styles.container}>
14     <Text>Contador: </Text>
15     <Button title='Incrementar' onPress={()=>setContador(contador+1)}></Button>
16     <StatusBar style="auto" />
```

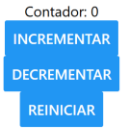
4. Por ultimo agregamos otros dos botones. Para el de decrementar o quitar es exactamente igual al de incrementar, únicamente cambiamos el signo de mas por el de menos. Por ultimo para el de reiniciar seria el mismo caso, pero multiplicaremos el valor de contador por 0 para que vuelva a el primer valor.

```
<Button title='Incrementar' onPress={()=>setContador(contador+1)}></Button>
<Button title='Decrementar' onPress={()=>setContador(contador-1)}></Button>
<Button title='Reiniciar' onPress={()=>setContador(contador*0)}></Button>
```

Ejecución en Expo Go y resultado final:

Usamos el comando npx expo start para que nos funcione el QR para emularlo en el teléfono y presionamos w para que se emule en el navegador.

Navegador:



Expo Go:

Conclusión:

En esta práctica aprendimos a como utilizar los componentes de texto y botón, además de que nos ayudo a recordar conceptos como lo es la desestructuración. Con esto empezamos a manipular y experimentar con lo que es el React.