



Practica 5

Alumno: Valdelamar Martínez Leonardo

Profesor: Ivan Isay Guerra Lopez

Materia: Desarrollo Apps Móviles

Fecha: 7/10/2025

Notas de clase:

Componentes

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

- Principales
- Nativos
- Comunidad: NativeBase, Shoutem, React Native Material UI

Core Components

Los vamos a reconocer porque se parecen a las etiquetas de HTML pero en realidad son palabras reservadas

`<View>`

`<Text>`

`<Image>`

`<ScrollView>`

`<TextInput>`

Props

Abreviación de propiedades

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

Ej. `<Button title = "Incrementar" onPress =/>`

Hooks

Los hook son funciones especiales que React proporciona para que puedas "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.

State

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

Desarrollo:

Empezamos abriendo nuestro archivo App.js y agregamos los comentarios de las zonas. Una vez que terminamos esto añadimos una línea de export para utilizar el hook useState.

```
1 //1. Zona de los imports (donde vamos a agregar lo que necesitamos)
2 import { StatusBar } from 'expo-status-bar';
3 import { StyleSheet, Text, View } from 'react-native';
4 import React, { useState } from 'react';
5
6
7 //2. Zona de componentes
8 export default function App()
9 {
10   return (
11     <View style={styles.container}>
12       <Text>Hola Mundo React Native!</Text>
13       <StatusBar style="auto" />
14     </View>
15   );
16 }
17 //3. Zona de los estilos
18 const styles = StyleSheet.create(
19 {
20   container: {
21     flex: 1,
22     backgroundColor: '#fff',
23     alignItems: 'center',
24     justifyContent: 'center',
25   },
26 });
27
```

Después modificamos el text y ponemos la palabra “Contador:” posteriormente agregamos <Button> </Button> para poder agregar un botón.

```
//1. Zona de los imports (donde vamos a agregar lo que necesitamos)
import { StatusBar } from 'expo-status-bar';
import { Button, StyleSheet, Text, View } from 'react-native';
import React, { useState } from 'react';

//2. Zona de componentes
export default function App()
{
  return (
    <View style={styles.container}>
      <Text>Contador</Text>
      <Button> </Button>
      <StatusBar style="auto" />
    </View>
  );
}
```

Le vamos a agregar la propiedad de title

```
//2. Zona de componentes
export default function App()
{
  return (
    <View style={styles.container}>
      <Text>Contador</Text>
      <Button title = 'Agregar' />
      <StatusBar style="auto" />
    </View>
  );
}
```

Una vez que lo agreguemos lo corremos con “npm run web” y aparece así en la página web.

Contador
AGREGAR

Vamos a agregar un contador para el botón agregar.

```
//2. Zona de componentes
export default function App()
{
  const [contador, setContador] = useState(0);
  return (
    <View style={styles.container}>
      <Text>Contador</Text>
      <Button title = 'Agregar' />
      <StatusBar style="auto" />
    </View>
  );
}
```

Lo llamamos dentro del componente para que se muestre el valor de la siguiente manera.

```
//2. Zona de componentes
export default function App()
{
  const [contador, setContador] = useState(0);
  return (
    <View style={styles.container}>
      <Text>Contador: {contador}</Text>
      <Button title = 'Agregar' />
      <StatusBar style="auto" />
    </View>
  );
}
```

Para que funcione, creamos la función setContador para que lleve la cuenta.

```
//2. Zona de componentes
export default function App()
{
  const [contador, setContador] = useState(0);
  return (
    <View style={styles.container}>
      <Text>Contador: {contador}</Text>
      <Button title = 'Agregar' onPress={()=>setContador(contador+1)} />
      <StatusBar style="auto" />
    </View>
  );
}
```

Al ejecutarlo se muestra así.

Contador: 1

AGREGAR

Agregamos otros 2 botones, el de quitar y el de reiniciar.

```
//2. Zona de componentes
export default function App()
{
  const [contador, setContador] = useState(0);
  return (
    <View style={styles.container}>
      <Text>Contador: {contador}</Text>
      <Button title = 'Agregar' onPress={()=>setContador(contador+1)} />
      <StatusBar style="auto" />
      <Button title = 'Quitar' onPress={()=>setContador(contador-1)} />
      <StatusBar style="auto" />
      <Button title = 'Reiniciar' onPress={()=>setContador(0)} />
      <StatusBar style="auto" />
    </View>
  );
}
```

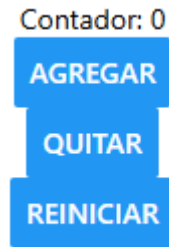
Quitar

Reiniciar

Contador: -1



Contador: 0



Conclusión:

Mediante esta práctica aprendimos la forma de importar e insertar un botón, que se pueda interactuar con él y agregarle un contador.