



## **PRACTICA 8 – Buttons & Switch**

**ESTUDIANTE: PÉREZ MELLADO JENNYFER**

**GRUPO: 211**

**MAESTRO: IVAN ISAY GUERRA LOPEZ**

**MATERIA: DESARROLLO DE APPS MÓVILES**

**FECHA: 16 – 10 – 2025**

## Practica 8:

Librerías y estados que se usaran para la elaboración de la práctica.

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

export default function BotonesScreen() {
  const [prendido, setPrendido] = useState(false);
  return (
    <View>
      <Text>Proximamente.....</Text>
    </View>
  )
}
```

Hacemos uso de un operador ternario, el cual funciona como un if-else en base al estado con un valor inicial de falso

```
return (
  <View>
    <Text>Estado: {prendido ? 'Prendido' : 'Apagado'}</Text>
  </View>
)
```

Estilos que tendrá nuestro apartado

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '■ #FFFFFF',
  },
  texto: {
    fontSize: 20,
    marginBottom: 30,
    fontWeight: 'bold',
  },
});
```

Se agregan los botones que van a modificar la variable a true en el botón encendido, y lo va a cambiar a false en el botón de apagado

```
export default function BotonesScreen() {
  const [prendido, setPrendido] = useState(false);
  return (
    <View style={styles.container}>
      <Text style={styles.texto}>Estado: {prendido ? 'Prendido' : 'Apagado'}</Text>
      <TouchableOpacity
        style={style.botonEncender}
        onPress={() => setPrendido(true)}>
        <text style={styles.textoBoton}>Prender</text>
      </TouchableOpacity>

      <TouchableOpacity
        style={style.botonEncender}
        onPress={() => setPrendido(false)}>
        <text style={styles.textoBoton}>Apagar</text>
      </TouchableOpacity>
    </View>
  )
}
```

Agregamos los estilos

```
botonEncender: {
  backgroundColor: '■ #4CAF50',
  paddingVertical: 10,
  paddingHorizontal: 25,
  borderRadius: 8,
  marginBottom: 20,
},
botonApagar: {
  backgroundColor: '■ #F44336',
  paddingVertical: 10,
  paddingHorizontal: 25,
  borderRadius: 8,
  marginBottom: 20,
},
textoBoton: {
  color: '■ #fff',
  fontSize: 16,
  fontWeight: '600',
},
```

Con todo puesto, en la pantalla se vera de esta forma

## Estado: Apagado

Prender

Apagar

Agregamos el import switch para la siguiente parte, y en la zona de estilos, agregamos todos los estilos pertinentes para la parte del switch

```
swithContainer: {  
  flexDirection: 'row',  
  alignItems: 'center',  
},  
switchLabel: {  
  fontSize: 16,  
  marginRight: 10,  
},  
}
```

Agregamos la parte del switch para controlar también el prendido y apagado

```
<View style={styles.swithContainer}>  
  <text style={styles.switchLabel}>Control de Switch: </text>  
  <Switch value = {prendido} onChange={setPrendido}></Switch>  
</View>  
</View>
```

En pantalla se ve de esta forma

## Estado: Prendido

Prender

Apagar

Control de Switch: ☒

Agregamos dos constantes las cuales van a modificar el color de fondo y de texto en base a la variable de prender y apagar, quedando el código de esta forma

```
export default function BotonesScreen() {  
  const [prendido, setPrendido] = useState(false);  
  
  const backgroundColor = prendido ? '#fab8e4ff' : '#121212';  
  const textColor = prendido ? '#000' : '#fab8e4ff';  
  
  return (  
    <View style={[styles.container, {backgroundColor}]}>  
      <Text style={[styles.texto, {color: textColor}]}>Estado: {prendido ? 'Prendido' : 'Apagado'}</Text>  
      <TouchableOpacity  
        style={styles.botonEncender}  
        onPress={() => setPrendido(true)}>  
        <text style={styles.textoBoton}>Prender</text>  
      </TouchableOpacity>  
  
      <TouchableOpacity  
        style={styles.botonApagar}  
        onPress={() => setPrendido(false)}>  
        <text style={styles.textoBoton}>Apagar</text>  
      </TouchableOpacity>  
  
      <View style={styles.swithContainer}>  
        <Text style={[styles.switchLabel, {color: textColor}]}>Control de Switch: </Text>  
        <Switch value = {prendido} onChange={setPrendido}></Switch>  
      </View>  
    </View>  
  )  
}
```

De esta forma, queda la pantalla de encendido:



Y de esta forma la pantalla de apagado:

