



Universidad Tecnológica de Tijuana

Diseño de Apps

Diseño de Apps: Primer Avance Portafolio

Carrera: TSu en Tecnologías de la Información y Comunicación

Alumno: Miguel Angel Leon Ramirez

Matricula: 0323106022

Grupo: 4 D

Docente: Ray Brunett Parra Galaviz

Tijuana B.C a 04 de febrero del 2025

## Lab 01

### Importación

**React:** Se importa para crear el componente de React.

**useState:** Un hook de React que permite manejar estados en el componente.

**StyleSheet, View, Text, Button, TextInput, StatusBar:** Componentes de React Native utilizados en la UI

### Definición del componente “MyForm”

```
export default function MyForm() {
```

Es un componente funcional que se exporta como predeterminado.

### Definición del Estado

```
const [text, setText] = useState("");  
const [displayText, setDisplayText] = useState("");
```

- **‘text’:** Variable de estado que almacena el texto ingresado.
- **‘setText’:** Función para actualizar ‘text’.
- **‘displayText’:** Variable que almacena el texto mostrado en pantalla.
- **‘setDisplayTex’:** Función para actualizar ‘displayTex’

### Función ‘handlePress’

```
const handlePress = () => {  
    setDisplayText(text);  
    setText("");  
};
```

Al presionar el boton. se guarda el contenido de ‘text’

Luego, se vacia el campo de entrada de texto

## Renderizado del componente

```
return (  
  <View style={styles.container}>  
    <Text>Text to show: {displayText}</Text>  
    <TextInput  
      style={styles.input}  
      placeholder="Type Something"  
      value={text}  
      onChangeText={setText}  
    />  
    <Button title="Click Me!" onPress={handlePress} />  
    <StatusBar style="auto" />  
  </View>  
);
```

- **View**: Contenedor principal.
- **Text**: Muestra el texto almacenado en `displayText`.
- **TextInput**: Permite ingresar texto.
- **Button**: Ejecuta `handlePress` al ser presionado.
- **StatusBar`**: Maneja la barra de estado del dispositivo.

## Resultado

Text to show:

CLICK ME!

## Lab 02

**Este código crea una aplicación que muestra dos textos y dos imágenes.**

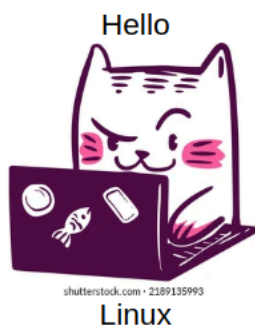
Primero, la aplicación presenta el texto "Hello", seguido de una imagen local llamada "gatito.png"

```
1 export default function App() {  
2   return (  
3     <View style={styles.container}>  
4       <Text style={styles.title}>Hello</Text>  
5       <Image style={styles.image} source={require('./assets/gatito.png')} />  
6  
7       <Text style={styles.title}>Linux</Text>  
8       <Image  
9         style={styles.image}  
10        source={{ uri: 'https://s1.elespanol.com/2016/12/07/actualidad/actualidad_176496376_129697721_1706x960.jpg' }}  
11      />  
12  
13     <StatusBar style="auto" />  
14   </View>  
15 );  
16 }
```

almacenada en la carpeta de assets del proyecto. Luego, muestra el texto "Linux" y carga una imagen, desde una URL externa.

El diseño del contenido se centra en la pantalla y mantiene un fondo blanco. En resumen, la aplicación muestra imágenes y textos en una interfaz sencilla y adaptable

## Resultado



## Lab 03

En esta aplicación se muestra una barra de navegación en la parte superior, un campo de texto donde el usuario puede escribir algo, y un botón que, al presionarlo, muestra una alerta con el texto ingresado.

```
1  import React from 'react';
2  import { StatusBar } from 'expo-status-bar';
3  import { StyleSheet, View } from 'react-native';
4  import { Button, Text, AppBar, TextInput } from 'react-native-paper';
5
6
7  export default function App() {
8    const [text, setText] = React.useState('');
9
10   return (
11     <View style={styles.container}>
12       {/* APP BAR COMPONENT PAPER */}
13       <AppBar>
14         <AppBar.Content title="React Native Paper" />
15       </AppBar>
16
17       {/* TEXT INPUT COMPONENT PAPER */}
18       <TextInput
19         label="Type Something"
20         value={text}
21         onChangeText={text => setText(text)}
22         textColor="red"
23         style={styles.input}
24       />
25
26       {/* BUTTON COMPONENT PAPER */}
27       <Button mode="contained" onPress={() => alert(`Texto: ${text}`)} style={styles.button}>
28         Show Text
29       </Button>
30
31       <Text>Hola, esta es mi tercera App</Text>
32       <StatusBar style="auto" />
33     </View>
34   );
35 }
36
37 const styles = StyleSheet.create({
38   container: {
39     flex: 1,
40     backgroundColor: '#fff',
41     alignItems: 'center',
42     justifyContent: 'center',
43     padding: 16,
44   },
45   input: {
46     width: '80%',
47     marginBottom: 16,
48   },
49   button: {
50     marginTop: 16,
51   },
52 });
```

Se usa un estado para almacenar lo que el usuario escribe en el campo de texto.

Cuando el botón es presionado, el texto ingresado por el usuario se muestra en una alerta.

Además, hay un mensaje estático que aparece debajo del botón.

El diseño de la interfaz es sencillo, centrando todos los elementos y con una barra de estado que se ajusta automáticamente.

## Resultado

### React Native Paper

Type Something

Hola!!

Show Text

Hola, esta es mi tercera App

## Mensaje de Alert

### React Native Paper

#### Alert

Texto: Hola!!

OK

Hola, esta es mi tercera App