Universidad Tecnológica de Tijuana

Program. Multiplatform Software Development

Group. 4°D

Student. Lara López Josue Isaac.

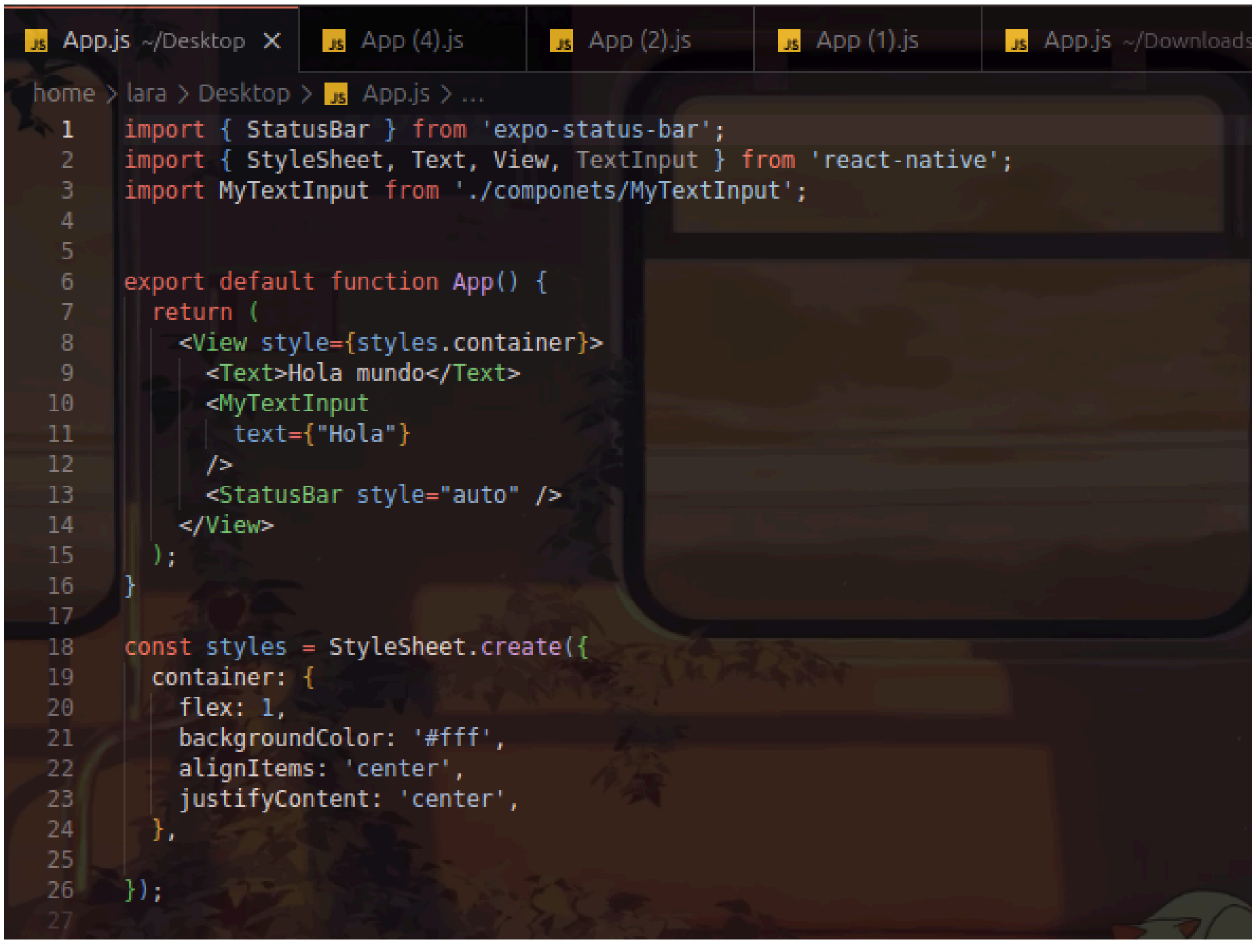
Subject. App Design.

Activity. Portafolio.

Professor. Ray Brunett Parra Galaviz
.

Tijuana, B.C., February 4, 2024.

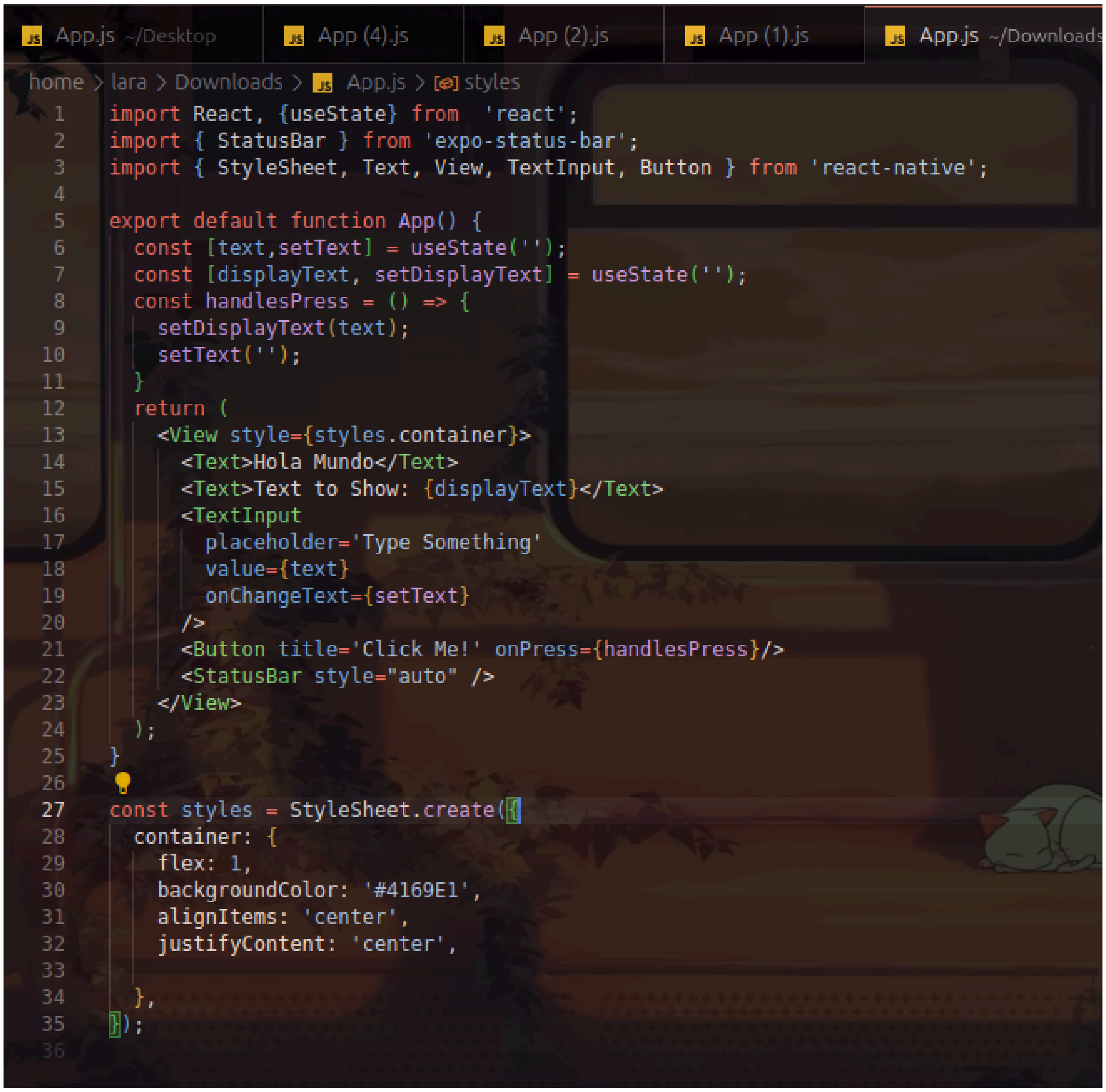
PRACTICA 1 “HOLA MUNDO”.



```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TextInput } from 'react-native';
import MyTextInput from './componets/MyTextInput';


export default function App() {
  return (
    <View style={styles.container}>
      <Text>Hola mundo</Text>
      <MyTextInput
        text={"Hola"}
      />
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },

});
```

This code in React Native defines a simple mobile app. Several components, including StatusBar, StyleSheet, Text, View and TextInput are imported from react-native, plus a custom component called MyTextInput from the components folder. The App function is the main component of the application and returns a View with styles defined in styles.container. Inside this View is rendered a Text with the message “Hello world”, the component MyTextInput to which is passed the prop text with the value “Hello” and the status bar StatusBar in automatic mode. Finally, a styles object is defined with StyleSheet.create, where the container style establishes a flexible design with flex 1, a white background with backgroundColor '#fff' and centers the elements with alignItems 'center' and justifyContent 'center'.

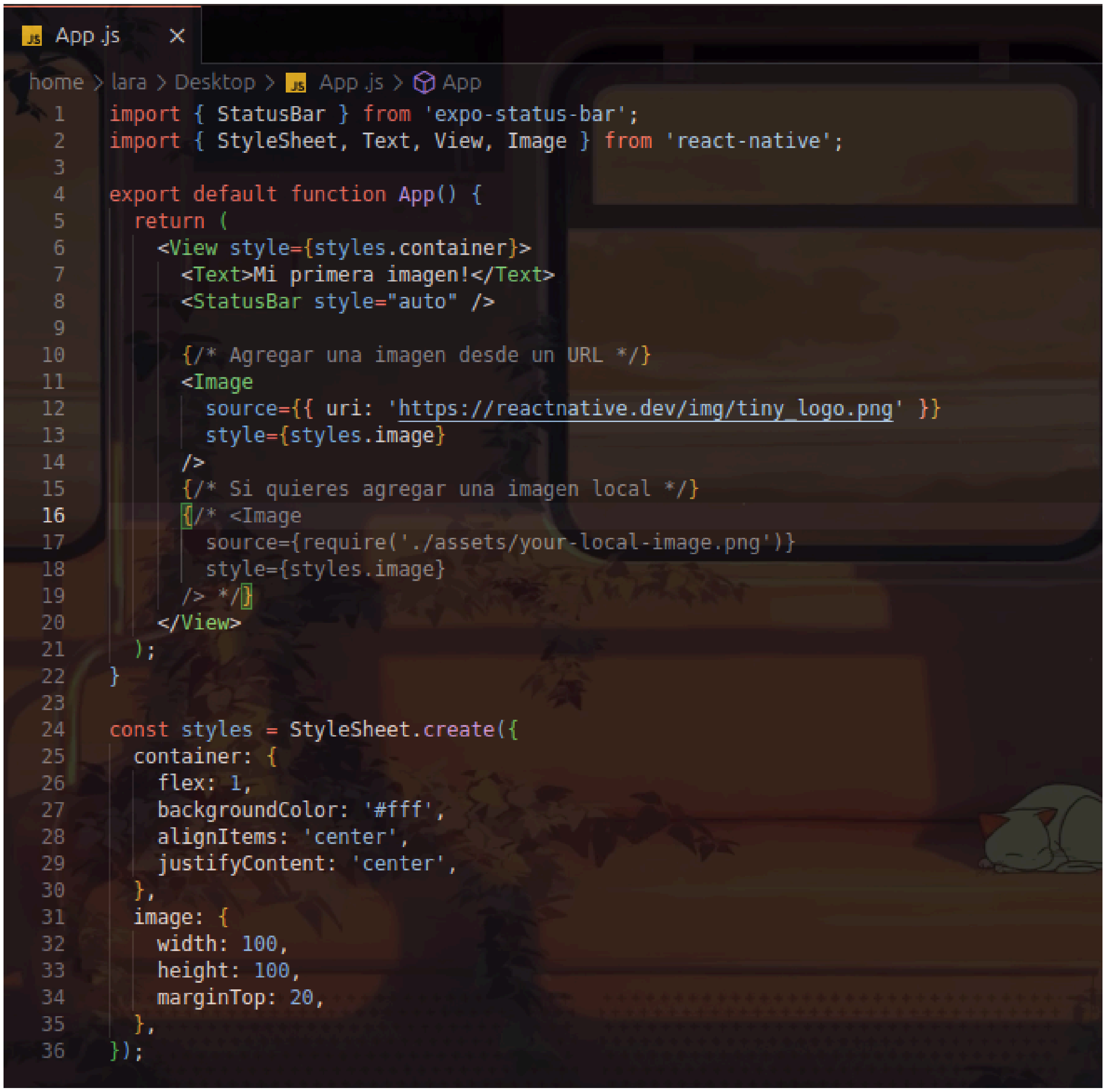
PRACTICA 2.



```
import React, {useState} from  'react';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TextInput, Button } from 'react-native';

export default function App() {
  const [text,setText] = useState('');
  const [displayText, setDisplayText] = useState('');
  const handlesPress = () => {
    setDisplayText(text);
    setText('');
  }
  return (
    <View style={styles.container}>
      <Text>Hola Mundo</Text>
      <Text>Text to Show: {displayText}</Text>
      <TextInput
        placeholder='Type Something'
        value={text}
        onChangeText={setText}
      />
      <Button title='Click Me!' onPress={handlesPress}/>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#4169E1',
    alignItems: 'center',
    justifyContent: 'center',

  },
});
```

This React Native code defines a mobile app that allows the user to enter text into an input field and display it on the screen when a button is pressed. React and the useState hook are imported to handle states within the functional component, along with several React Native components, such as StatusBar, StyleSheet, Text, View, TextInput and Button. In the App function, two states are defined: text, which stores the text entered by the user, and displayText, which stores the text to be displayed on the screen. The handlesPress function is created, which updates displayText with the value of text and then empties the input field. In the JSX return, the interface is structured with a View that contains a “Hello World” message, another Text that shows the displayText value, a TextInput where the user can write, a Button that executes handlesPress when pressed and the StatusBar status bar in automatic mode. Finally, the styles object is defined with StyleSheet.create, where the container style configures flex 1 to occupy the entire screen, sets the background color to blue (#4169E1) and centers the elements with alignItems 'center' and justifyContent 'center'.

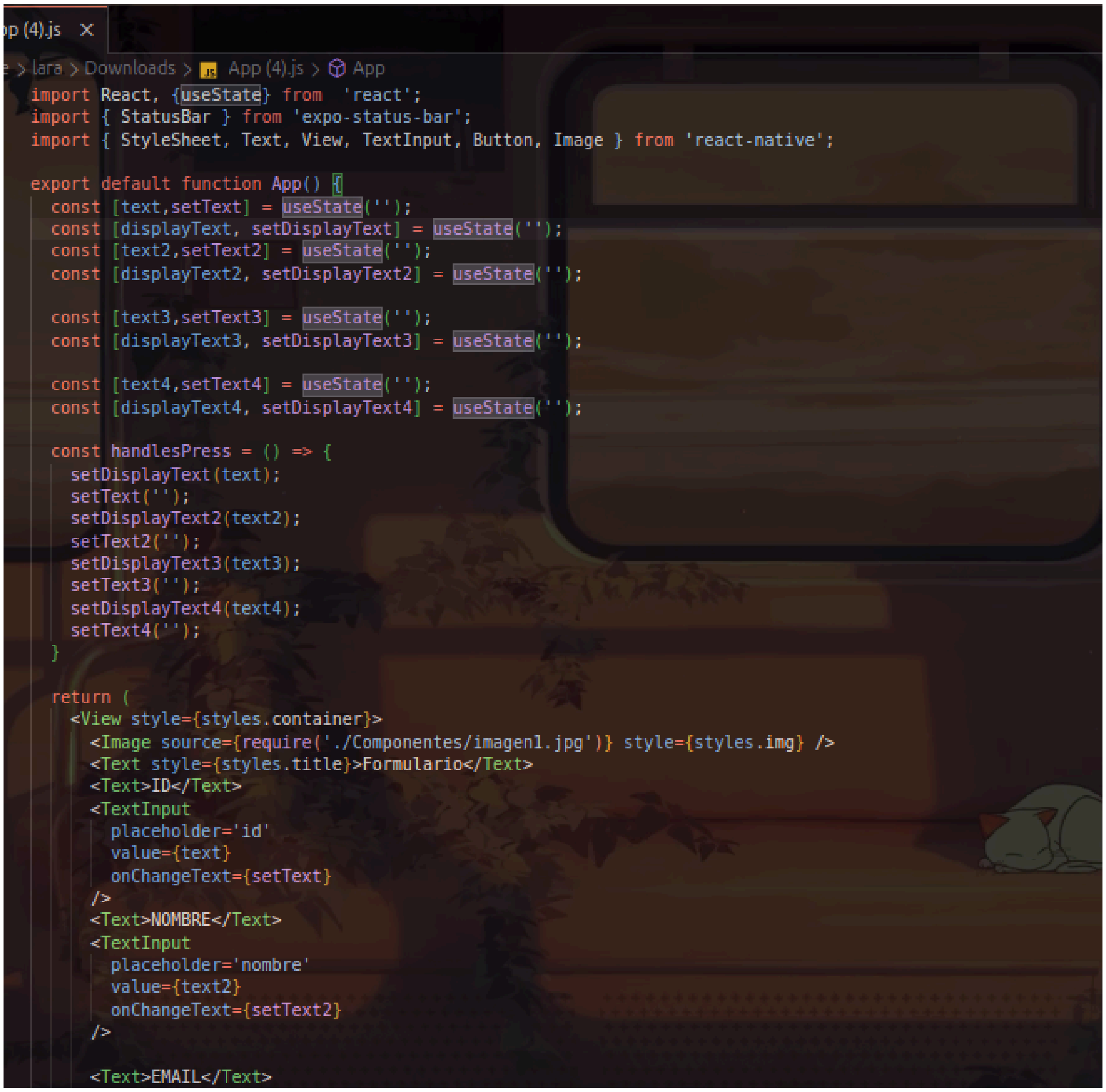
PRACTICA 3 - IMAGEN.



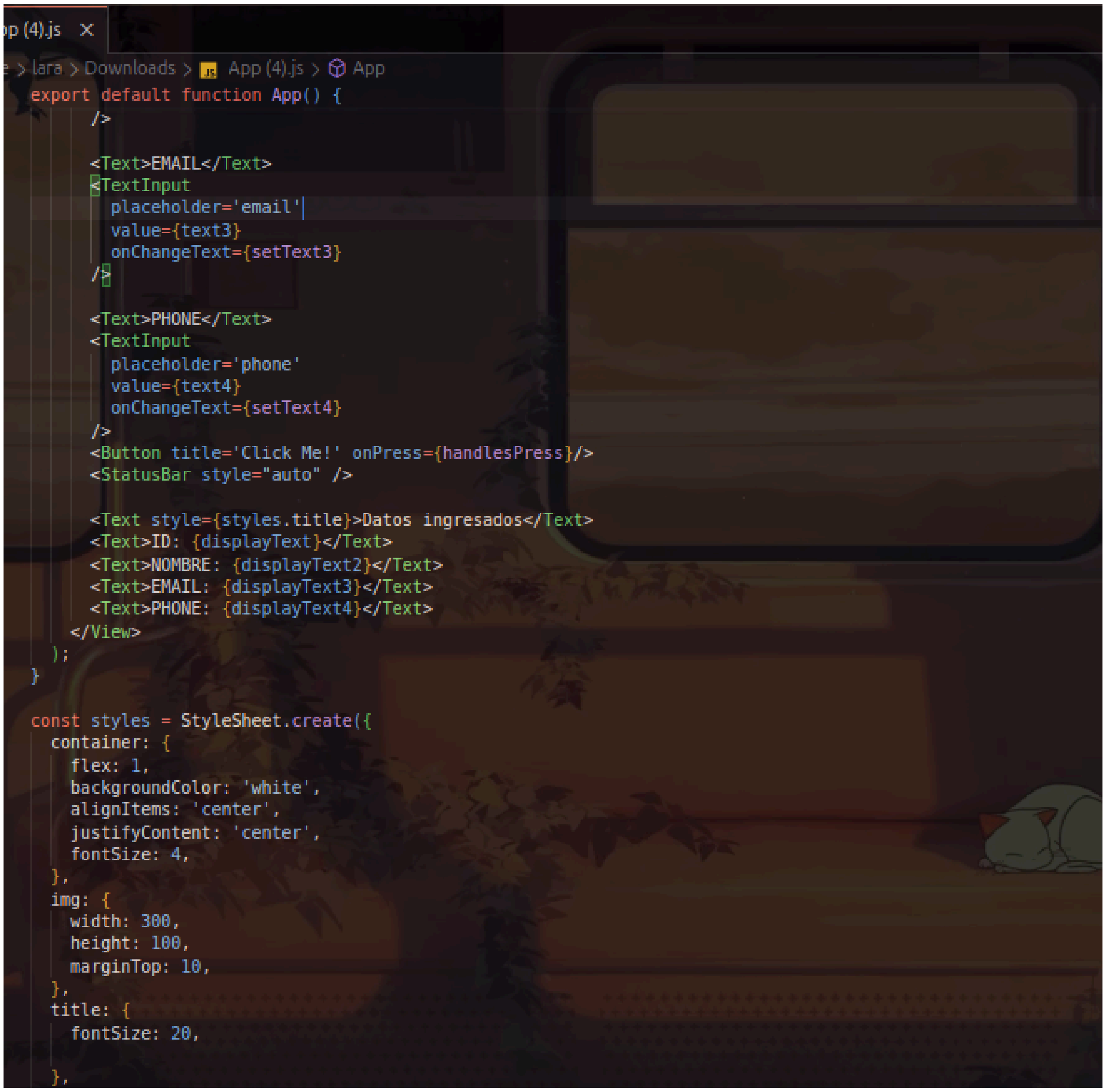
```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, Image } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text>Mi primera imagen!</Text>
      <StatusBar style="auto" />

      {/* Agregar una imagen desde un URL */}
      <Image
        source={{ uri: 'https://reactnative.dev/img/tiny_logo.png' }}
        style={styles.image}
      />
      {/* Si quieres agregar una imagen local */}
      {/* <Image
        source={require('./assets/your-local-image.png')}
        style={styles.image}
      /> */}
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  image: {
    width: 100,
    height: 100,
    marginTop: 20,
  },
});
```

This code in React Native creates a simple mobile app that displays a text and an image. The StatusBar, StyleSheet, Text, View and Image components are imported from React Native. In the App function, a View is returned with styles defined in styles.container. Inside this View, a Text is displayed with the message “My first image!” and the StatusBar status bar in automatic mode. Then, an Image component is included that loads an image from an external URL, specifically the small logo of React Native, by means of the source property with uri. We also leave commented an example to load a local image using require. Finally, a styles object is defined with StyleSheet.create, where the container style sets flex 1 to occupy the entire screen, a white background, and centers the elements with alignItems 'center' and justifyContent 'center'. The image style defines a width and height of 100 pixels and a top margin of 20 pixels to separate the image from the text.

PRÁCTICA 4 - COMPONENTES.



```
import React, {useState} from  'react';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TextInput, Button, Image } from 'react-native';

export default function App() {
  const [text,setText] = useState('');
  const [displayText, setDisplayText] = useState('');
  const [text2,setText2] = useState('');
  const [displayText2, setDisplayText2] = useState('');

  const [text3,setText3] = useState('');
  const [displayText3, setDisplayText3] = useState('');

  const [text4,setText4] = useState('');
  const [displayText4, setDisplayText4] = useState('');

  const handlesPress = () => {
    setDisplayText(text);
    setText('');
    setDisplayText2(text2);
    setText2('');
    setDisplayText3(text3);
    setText3('');
    setDisplayText4(text4);
    setText4('');
  }

  return (
    <View style={styles.container}>
      <Image source={require('./Componentes/imagen1.jpg')} style={styles.img} />
      <Text style={styles.title}>Formulario</Text>
      <Text>ID</Text>
      <TextInput
        placeholder='id'
        value={text}
        onChangeText={setText}
      />
      <Text>NOMBRE</Text>
      <TextInput
        placeholder='nombre'
        value={text2}
        onChangeText={setText2}
      />

      <Text>EMAIL</Text>
```



```
export default function App() {
      />

      <Text>EMAIL</Text>
      <TextInput
        placeholder='email'
        value={text3}
        onChangeText={setText3}
      />

      <Text>PHONE</Text>
      <TextInput
        placeholder='phone'
        value={text4}
        onChangeText={setText4}
      />
      <Button title='Click Me!' onPress={handlesPress}/>
      <StatusBar style="auto" />

      <Text style={styles.title}>Datos ingresados</Text>
      <Text>ID: {displayText}</Text>
      <Text>NOMBRE: {displayText2}</Text>
      <Text>EMAIL: {displayText3}</Text>
      <Text>PHONE: {displayText4}</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'white',
    alignItems: 'center',
    justifyContent: 'center',
    fontSize: 4,
  },
  img: {
    width: 300,
    height: 100,
    marginTop: 10,
  },
  title: {
    fontSize: 20,

  },
```

This React Native code creates an application with a form that allows you to enter and display four pieces of information: ID, Name, Email and Phone. The necessary React Native components are imported, including StatusBar, StyleSheet, Text, View, TextInput, Button and Image. In the App function, eight states are defined with useState: four to store the values entered in the text fields (text, text2, text3 and text4) and four to display the values stored after pressing the button (displayText, displayText2, displayText3 and displayText4). The handlesPress function assigns the entered values to the corresponding displayText states and then empties the input fields. In the JSX structure, inside a View with styles defined in styles.container, a local image, a title “Form” and four TextInput fields with placeholders for ID, Name, Email and Phone are displayed. A “Click Me!” button executes handlesPress when pressed. Then, another title “Data entered” and the values stored in the displayText states are shown. In the styles part, styles.container sets flex 1 to fill the entire screen, a white background and centers the elements with alignItems 'center' and justifyContent 'center'. styles.img sets the width to 300 pixels and the height to 100 pixels with a top margin of 10 pixels. styles.title defines a font size of 20 pixels.