Universidad Tecnológica de Tijuana

Career: IT Multiplatform Software Development

Project: Industrial Inventory Control System

Subject: Application Design

Work: Software Requirements Specification

Student:
Moreno Ramirez Josue Elihu

Group: 4B

Teacher: Ray Brunett Parra Galaviz

Practice 1

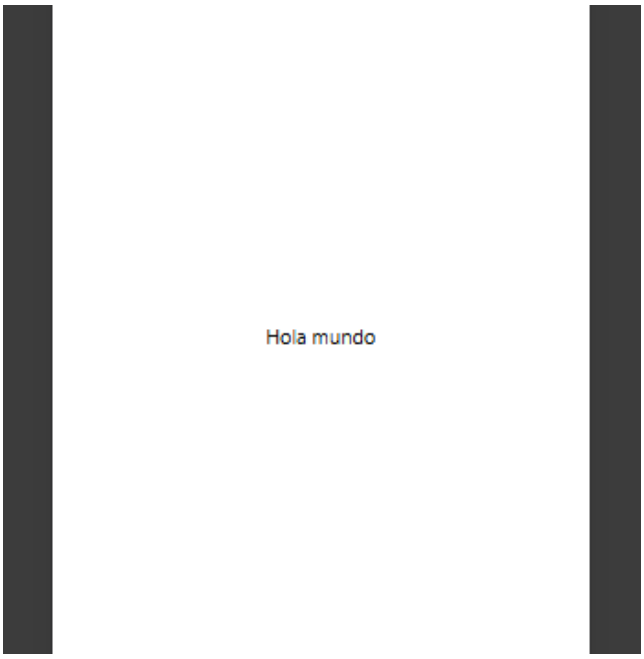
Practice one consisted of creating a blank project. The challenge was to install the necessary dependencies to use React. Once everything was installed, we could access our app.js file, where we modified a message to be displayed in the application. Using the command npx expo start, we could access our application from the browser.

```
› Metro waiting on exp://192.168.1.208:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Web is waiting on http://localhost:8081

› Using Expo Go
› Press s │ switch to development build

› Press a │ open Android
› Press w │ open web

› Press j │ open debugger
› Press r │ reload app
› Press m │ toggle menu
› shift+m │ more tools
› Press o │ open project code in your editor
```

The application code was simple: a Text component displaying the words "Hello World."

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

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

This was the application's compilation:



Practice 2

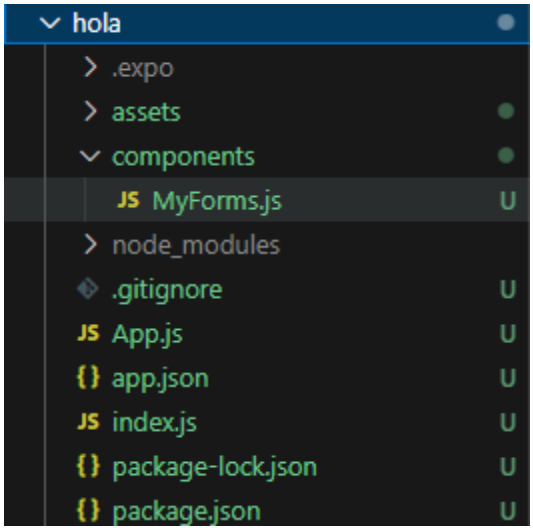
Practice two involved using components in different parts of our application. The components were divided, with the main file containing the following code, which called another file with the form.

```
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet, View } from 'react-native';
import MyForms from './components/MyForms';


export default function App() {
  return (
    <View style={styles.container}>
      <MyForms />
      <StatusBar style="auto" />
    </View>
  );
}
```

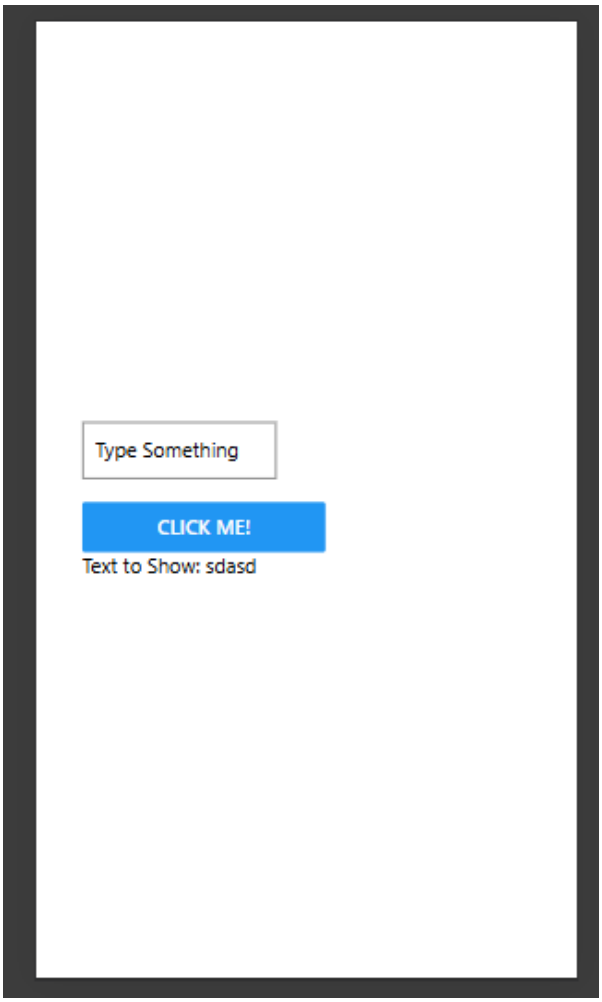
```
import React, { useState } from 'react';
import { Button, StyleSheet, Text, TextInput, View } from "react-native";

export default function MyForms() {
      const [text, setText] = useState('');
      const [displayText, setDisplayText] = useState('');

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

    return (
    <View style={styles.container}>
    <TextInput
        style={styles.input}
        placeholder="Type Something"
        value={text}
        onChangeText={setText}
    />

    <Button title="Click Me!" onPress={handlePress} />
    <Text style>Text to Show: {displayText}</Text>
    </View>
);
}
```

The application worked with a TextInput, which stored text. A button was included with a function that saved the stored text in a variable and displayed it.

These were the files for the practice, organized within the project folder.

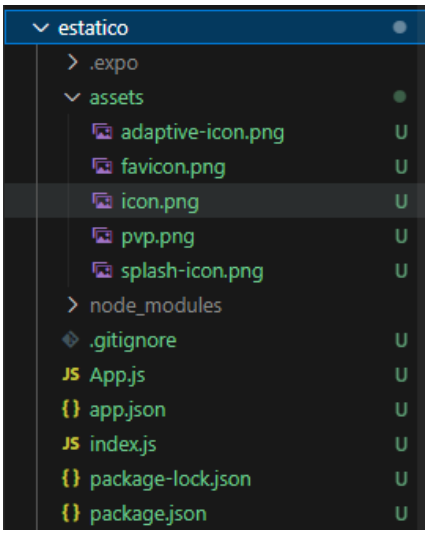


And this was the application's compilation:



Practice 3

This practice consisted of implementing static elements into the application, such as images. Two images were added: one via URL and another downloaded and stored in the project's assets folder.

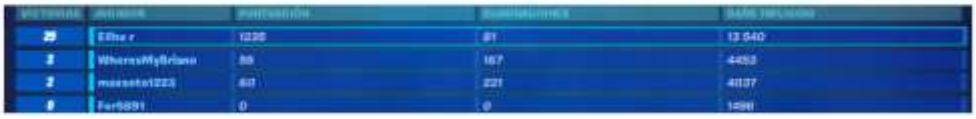



```
import { StatusBar } from 'expo-status-bar';
import { Image, StyleSheet, Text, View } from 'react-native';

// remplaza por el link de tu imagen
  const third_image = "https://static.wikia.nocookie.net/hollowknight/images/1/13/Silksong_cover.jpg";

  //sube tu imagen a la carpeta assets y remplaza el nombre de la imagen

export default function App() {
  return (
    <View style={styles.container}>
                                                    (alias) class Text
                                                    import Text
    <Text style={styles.title}>Source: Third Image</Text>
      <Image style={styles.image} source={{ uri: third_image }} />

      <Text style={styles.title}>Source: Local Image</Text>

      <Image style={styles.image_pvp} source={require("./assets/pvp.png")} />


      <StatusBar style="auto" />
    </View>
  );
}
```

This was the application's compilation:

Source: Third Image

Source: Local Image



Practice 4

This practice involved integrating a framework called **React Native Paper**, which provided various UI components. The application featured a TextInput where a value was stored. A button captured the value from the TextInput and displayed it in an alert so the user could see the entered value.

```
export default function App() {

    const [text, setText] = React.useState('');

  return (
    <View style={styles.container}>

      {/* Appbar component paper */}
      <Appbar>
        <Appbar.Content title="React Native Paper" />
      </Appbar>

      {/* TextInput component paper */}
      <TextInput style={styles.input}
      label='Type something'
      value={text}
      onChangeText={text => setText(text)}
      textColor='blue'>
      </TextInput>

      {/* Button component paper */}
      <Button mode='contained' onPress={() => alert(`Texto Ingresado: ${text}`)}>
        Click me
      </Button>

      <StatusBar style="auto" />
    </View>
  );
}
```

This was the application's compilation: