App Design

Dr. Ray Brunett Parra Galaviz

Partial I - First Portfolio Progress

Due date: 05/02/2025

TSU in Information Technologies

Meza Osuna Juan Manuel

4B

Content

PARTIAL I

PRACTICE I – INTRODUCTION TO REACT NATIVE

Activity Description

As the first activity, we started by installing the work environment for React Native, and several technology alternatives were explained, along with their pros and cons.

Expo Developer Tools

Development Environment

• Node.js → Required to run JavaScript and manage packages with npm.

• npm (Node Package Manager) → Installed with Node.js and used to manage dependencies.

• Expo CLI → Command-line tool to create and manage projects in Expo.

Project Creation and Execution

• Terminal or Console → To run commands like npx create-expo-app and expo start.

• Code Editor (recommended: Visual Studio Code) → To write and modify the app code.

Testing on Devices

• Expo Go (Android/iOS) → Mobile app that allows testing the app on a real device without compiling it.

Additional Tools

• Expo Developer Tools → Web interface that opens with expo start to manage the project.

• React Native Debugger (optional) → For debugging and improving app performance.

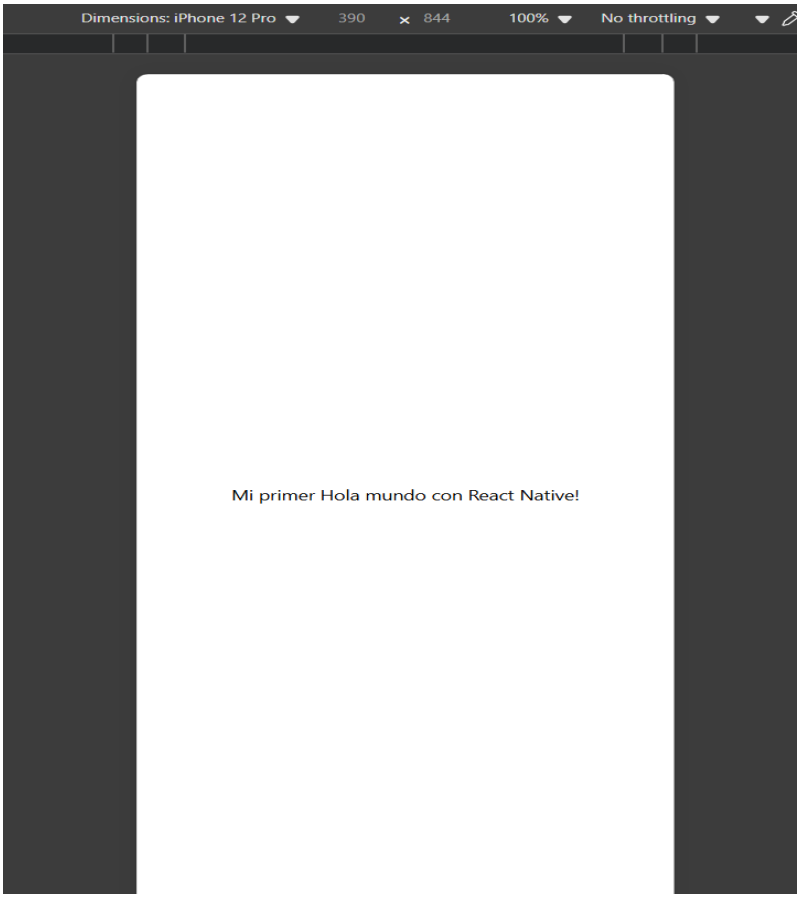

Figure 1 App View

PRACTICE II - STATIC AND DYNAMICS IMAGES

Code Description

In this practice, we developed a React Native app that displays images in two ways:

1. Static Image: We store it in the local project files.
2. Dynamic Image: We load it from an online URL.

This code helps us understand how to handle images in React Native and how to structure a simple interface using basic components.

Imports

We used the following imports:

- StatusBar from expo-status-bar to manage the device's status bar.
- useState from React to handle the text state variable.
- StyleSheet, Text, View, Image, TextInput from react-native, which allow us to build the UI.

Main Function (App)

Inside the main function:

- We declare a state variable text using useState.
- We organize the UI inside a View, displaying two images:
    1. **Static Image**, which we load from the project's local files (./assets/bayern.jpg).
    2. **Dynamic Image**, which we retrieve from an external URL.
- We add Text labels to describe each image and enhance the presentation.

```
{/*Imagen estatica*/}
<Text style={styles.texto}>Imagen estatica</Text>
<Image source={require("./assets/bayern.jpg")} style={styles.image}/>

{/* Imagen dinamica */}
<Text style={styles.texto}>Imagen dinamica</Text>
<Image
  source={{uri: "https://imageio.forbes.com/specials-images/imageserve/645c9f5b9a8c05b67fe63665/0x0.jpg?format=jpg&height=900&width=1600&fit=bounds"}}
  style={styles.image}
/>
```

Figure 2 Syntax

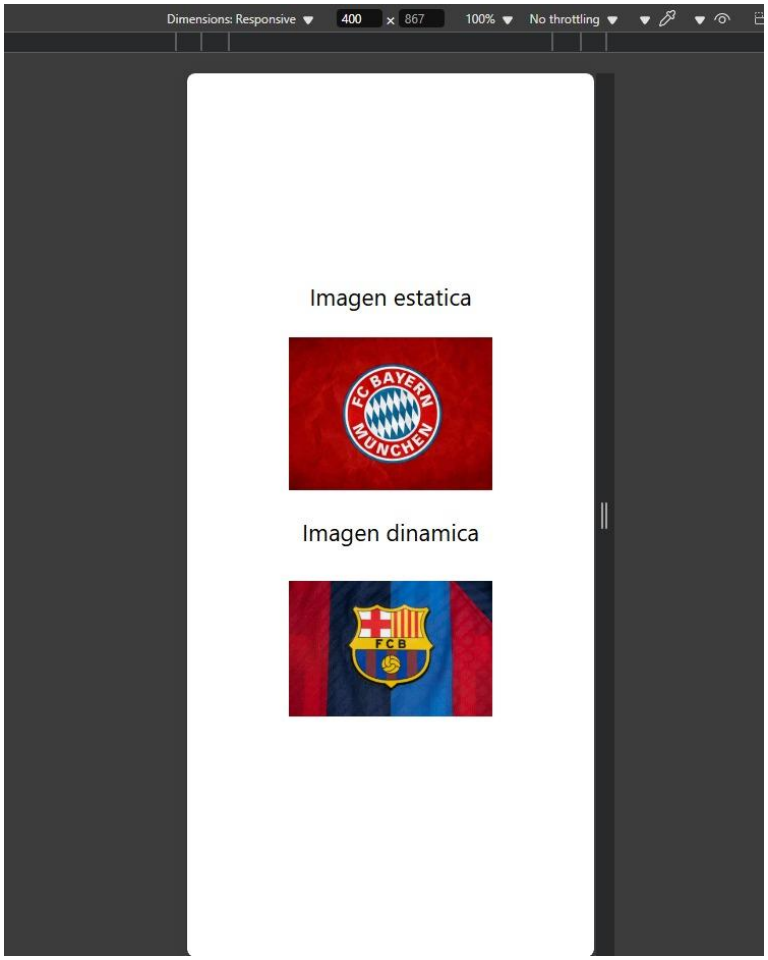

Figure 3 App View

PRACTICE III – REACT PAPER

Activity Description:
In this practice, we created a simple application with a graphical user interface using React Native Paper.

Imports:

- The necessary modules such as React, View, and Stylesheet from React Native were imported.
- Components from React Native Paper were imported, including Provider, Appbar, and TextInput.

```
PS C:\Users\manuo\Documents\paper> npm install react-native-paper

added 20 packages, and audited 903 packages in 16s

72 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Figure 4 Installation of packages for React Native Paper

Main Function of the Application:

- We used the 'text' state to store what the user types in an input field.
- It includes a text input field (TextInput) with an onChangeText that updates the state.
- It has a button (Button) that, when pressed, displays the entered text below the button.

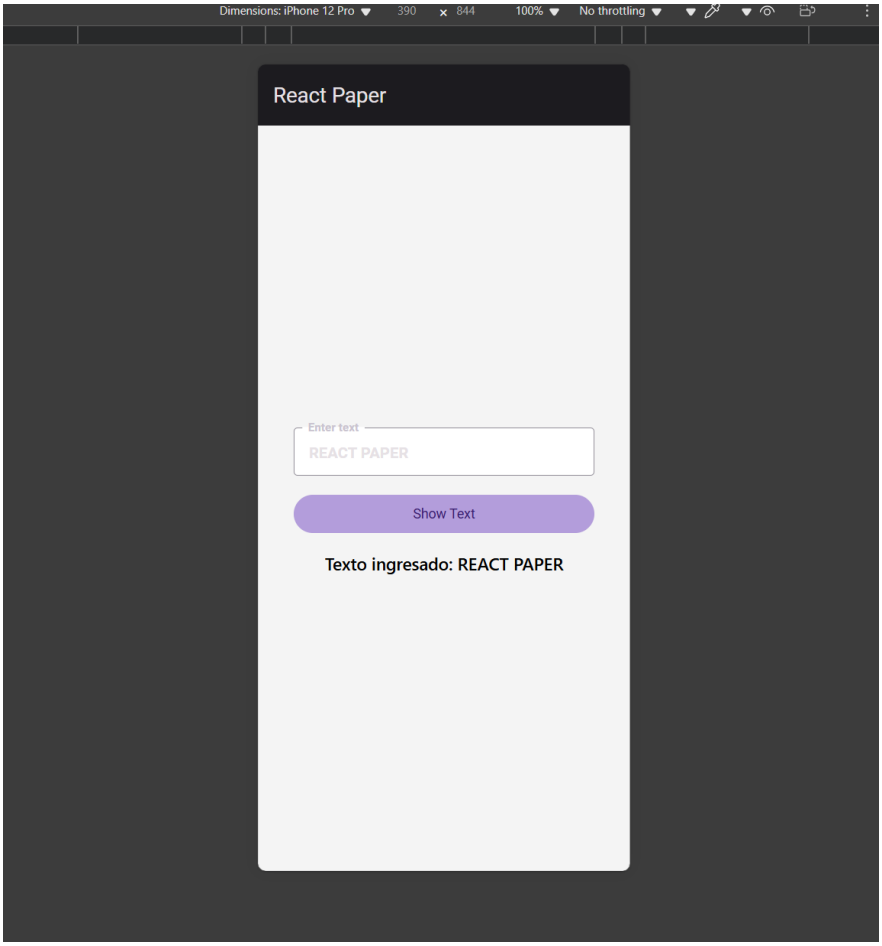

Figure 5 App View