



“Componentes View Button”

:: UNIDAD 1: Introducción a las tecnologías móviles

:: ACTIVIDAD COMPLEMENTARIA 2

MATERIA: :: PROGRAMACIÓN DE DISPOSITIVOS MÓVILES ::

CLAVE: 1668

LICENCIATURA EN INFORMÁTICA

PLAN 2012

REALIZO: Emmanuel Alejandro Pérez Hernández

No. 423142118

Grupo: 8691

ASESOR: MARTINEZ FERNANDEZ JUAN MANUEL

viernes, 5 de septiembre de 2025

UNIDAD 1. Actividad Complementaria 2

"View" y "Button"

- **Componente View**
 - **Funciones principales:**

El componente View es el contenedor fundamental en React Native con las siguientes funciones:

1. Contenedor de layout:
 - Equivalente a `<div>` en la web
 - Soporta Flexbox para crear diseños responsivos
 - Permite organizar elementos hijos en diferentes disposiciones
2. Agrupación de elementos:
 - Agrupa múltiples componentes visuales
 - Permite aplicar estilos comunes a un conjunto de elementos
3. Soporte de estilos:
 - Acepta propiedades de estilo (StyleSheet)
 - Soporta sombras, bordes, fondos, etc.
 - Permite eventos táctiles (onTouchStart, onTouchEnd, etc.)
4. Estructura jerárquica:
 - Crea una jerarquía visual en la interfaz
 - Maneja el flujo de eventos táctiles

Ejemplo de uso:

```
<View style={styles.container}>  
  <Text>Texto dentro de View</Text>  
  <Image source={require('./image.png')} />  
</View>
```



- **Componente Button**
 - **Funciones principales:**

El componente Button es un botón interactivo con las siguientes características:

1. Interacción básica:
 - Renderiza un botón nativo de la plataforma (iOS/Android)
 - Incluye automáticamente efectos visuales al presionar
2. Propiedad de título:
 - Muestra texto mediante la prop `title`
 - No permite elementos personalizados dentro del botón
3. Manejo de eventos:
 - Gestiona pulsaciones mediante la prop `onPress`
 - No soporta eventos táctiles más complejos
4. Limitaciones:
 - Estilos limitados (solo color)
 - No permite personalización avanzada
 - Apariencia determinada por el sistema operativo

Ejemplo de uso:

```
<Button
  title="Presionar"
  onPress={() => alert('Botón presionado')}
  color="#841584"
/>
```



- **Comparación con Pressable**

Mientras que Button es un componente simple con funcionalidad limitada, Pressable (su reemplazo moderno) ofrece:

1. Mayor personalización de estilos y comportamientos
2. Feedback visual personalizable durante diferentes estados (presionado, hover, etc.)
3. Capacidad para contener cualquier tipo de elemento hijo
4. Mayor control sobre los gestos táctiles

Ejemplo de Pressable:

```
<Pressable
  onPress={() => alert('Personalizable')}
  style={({pressed}) => [
    { backgroundColor: pressed ? 'blue' : 'red' }
  ]
>
  <Text>Mi botón personalizado</Text>
  <Image source={require('./icon.png')} />
</Pressable>
```



Cambio de código

<https://snack.expo.dev/@getzemax98/actividad-boton>

```
import React from 'react';
import { Pressable, Text, View, StyleSheet, Alert } from 'react-native';

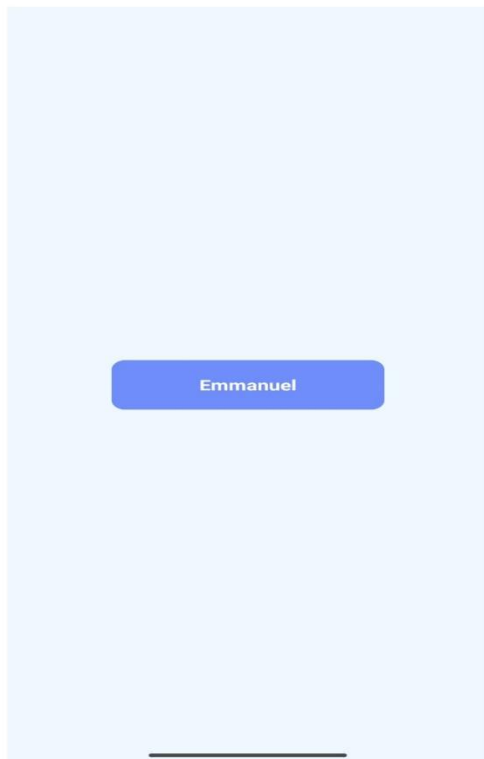
const App = () => {
  return (
    <View style={styles.container}>
      <Pressable
        style={({ pressed }) => [
          styles.button,
          { backgroundColor: pressed ? '#5a7de4' : '#6e8efb' }
        ]}
        onPress={() => Alert.alert('¡Hola!', 'Soy Emmanuel')}
      >
        <Text style={styles.text}>Emmanuel</Text>
      </Pressable>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f0f8ff',
    justifyContent: 'center',
    alignItems: 'center',
    marginTop: 100,
    marginHorizontal: 20,
  },
  button: {
    padding: 15,
    borderRadius: 10,
    minWidth: 200,
    alignItems: 'center',
  },
  text: {
    color: 'white',
    fontSize: 18,
    fontWeight: 'bold',
  },
});

export default App;
```



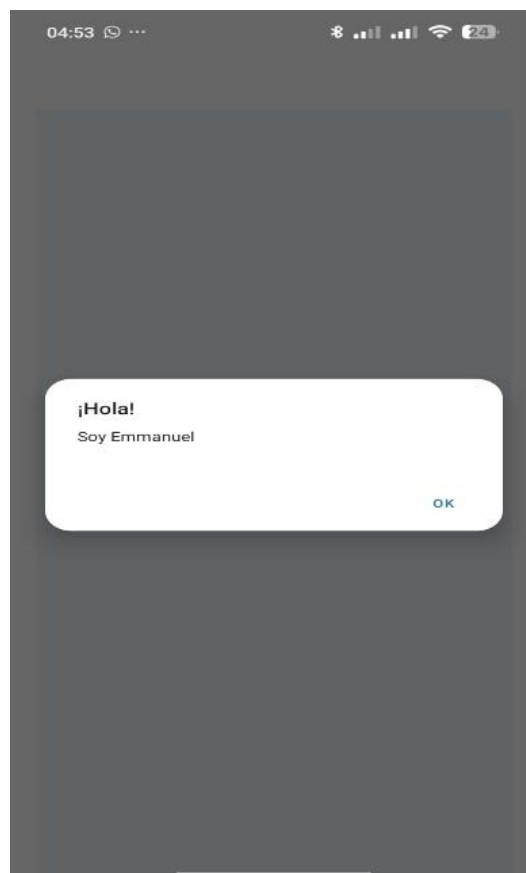
04:53 04:53 24



La aplicación funcionando en un dispositivo móvil, se notan los cambios realizados al código, como el botón con el texto "Emmanuel" (mi nombre), una interfaz limpia y diseñada, así como el cambio de fondo, márgenes y el estilo del botón.

Se muestra la respuesta al presionar el botón, con el mensaje "¡Hola! Soy Emmanuel", y al finalizar Botón "OK" para cerrar la alerta.

Confirmando que la funcionalidad onPress está trabajando correctamente.



I. REFERENCIAS BIBLIOGRÁFICAS

Button · React Native. (2025, August 20). Reactnative.dev.

<https://reactnative.dev/docs/button>

View · React Native. (2025, September 2). Reactnative.dev.

<https://reactnative.dev/docs/view#:~:text=El%20componente%20m%C3%A1s%20fundamental%20para,se%20admiten%20estilos%20en%20l%C3%ADnea>.

Silverio, A. (2024, February 27). *React Native Button Expo: How to Create Custom*

Buttons in Your App. Andreadams.com.br. [https://andreadams.com.br/react-native-button-expo-how-to-create-custom-buttons-in-your-](https://andreadams.com.br/react-native-button-expo-how-to-create-custom-buttons-in-your-app/#:~:text=Botones%20de%20estilo,%20onPress%20%20disabled%20y%20containerStyle%20.&text=La%20gesti%C3%B3n%20de%20las%20pulsaciones,ocurrir%20al%20pulsar%20un%20bot%C3%B3n.&text=propiedad%20onPress%20o%20definir%20una%20funci%C3%B3n%20separada%20para%20manejarlos)

[app/#:~:text=Botones%20de%20estilo,%20onPress%20%20disabled%20y%20containerStyle%20.&text=La%20gesti%C3%B3n%20de%20las%20pulsaciones,ocurrir%20al%20pulsar%20un%20bot%C3%B3n.&text=propiedad%20onPress%20o%20definir%20una%20funci%C3%B3n%20separada%20para%20manejarlos](https://andreadams.com.br/react-native-button-expo-how-to-create-custom-buttons-in-your-app/#:~:text=Botones%20de%20estilo,%20onPress%20%20disabled%20y%20containerStyle%20.&text=La%20gesti%C3%B3n%20de%20las%20pulsaciones,ocurrir%20al%20pulsar%20un%20bot%C3%B3n.&text=propiedad%20onPress%20o%20definir%20una%20funci%C3%B3n%20separada%20para%20manejarlos).

