



UNIVERSIDAD POLITÉCNICA DE QUERÉTARO

DESARROLLO DE APLICACIONES MÓVILES

GRUPO: TIID215

15/10/2025

Integrantes de equipo:

- Daniela López Pacheco
- Monserrath Alarcón Águilar
- Luis Guadalupe Barrón Duran

Componente ScrollView en React Native

1. Introducción

En esta práctica se presenta el uso del componente ScrollView de React Native, el cual permite crear vistas con contenido desplazable, tanto de forma vertical como horizontal. Este componente es esencial cuando se necesita mostrar una gran cantidad de elementos que no caben en la pantalla, manteniendo la posibilidad de hacer scroll sin complicaciones.

2. ¿Qué es ScrollView?

ScrollView es un componente de React Native que permite desplazarse (hacer scroll) dentro de una vista. A diferencia de una FlatList, ScrollView renderiza todo su contenido a la vez, lo que la hace ideal para listas cortas o diseños donde el número de elementos es limitado.

Usos comunes:

- Mostrar listas pequeñas de datos.
- Desplazar secciones de texto o imágenes grandes.
- Crear desplazamientos horizontales (por ejemplo, carruseles de tarjetas o imágenes).

3. Props más utilizadas de ScrollView

Prop	Descripción	Ejemplo / Uso
horizontal	Permite el desplazamiento horizontal.	<code><ScrollView horizontal={true} /></code>
contentContainerStyle	Aplica estilos al contenido interno.	<code>contentContainerStyle={{ padding: 20 }}</code>
showsVerticalScrollIndicator	Muestra u oculta la barra de desplazamiento vertical.	<code>showsVerticalScrollIndicator={true}</code>
showsHorizontalScrollIndicator	Muestra u oculta la barra horizontal.	<code>showsHorizontalScrollIndicator={false}</code>
nestedScrollEnabled	Permite tener scrolls anidados.	<code>nestedScrollEnabled={true}</code>
onScroll	Ejecuta una función mientras el usuario hace scroll.	<code>onScroll={(e) => console.log(e.nativeEvent.contentOffset.y)}</code>

4. Uso del State en ScrollView

Aunque ScrollView no requiere state para funcionar, podemos combinarlo con useState de React para detectar posiciones de desplazamiento, activar animaciones o mostrar u ocultar elementos según el scroll.

Ejemplo básico de uso con state:

```
const [position, setPosition] = useState(0);

<ScrollView onScroll={(e) => setPosition(e.nativeEvent.contentOffset.y)}>
  <Text>Posición actual del scroll: {position}</Text>
</ScrollView>
```

5. Código de la práctica

```
import { Text, StyleSheet, View, ScrollView } from 'react-native';

import React from 'react';

export default function ScrollViewScreen() {

  return (

    <ScrollView style={styles.container} contentContainerStyle={styles.content}
      showsVerticalScrollIndicator={true}>

      <Text style={styles.Titulo}>Práctica: ScrollView</Text>

      <Text style={styles.Titulo2}>Ejemplo de desplazamiento vertical</Text>

      <View style={styles.elementos}>

        <Text style={styles.text}>Elemento 1</Text>

      </View>

      <View style={styles.elementos}>

        <Text style={styles.text}>Elemento 2</Text>

      </View>

    </ScrollView>

  );
}
```

```

<View style={styles.elementos}>
  <Text style={styles.text}>Elemento 3</Text>
</View>

<Text style={styles.Titulo2}>Ejemplo de desplazamiento horizontal</Text>

<ScrollView horizontal nestedScrollEnabled={true}
style={styles.scrollhorizontal} showsHorizontalScrollIndicator={true}>

  <View style={styles.elementos2}>

    <Text style={styles.text}>Cuadro 1</Text>

  </View>

<View style={styles.elementos2}>

  <Text style={styles.text}>Cuadro 2</Text>

</View>

<View style={styles.elementos2}>

  <Text style={styles.text}>Cuadro 3</Text>

</View>

</ScrollView>

</ScrollView>

);
}

```

6. Explicación del código

- **ScrollView principal:** Contiene todos los elementos desplazables verticalmente. Además, en la parte de el estilo, agregamos un atributo para que los elementos tuvieran un espacio de separación para que la parte del scroll se notara bien.
- **Creación manual de elementos:** En la parte vertical se agregan cinco vistas para el scroll vertical y doce para la horizontal.

Esto permite observar cómo el contenido se puede desplazar y deslizar al exceder la altura de la pantalla.

- **ScrollView horizontal:** Dentro del scroll principal, se incluye otro ScrollView con la prop horizontal para permitir desplazamiento lateral. Se usa nestedScrollEnabled para que ambos scrolls funcionen correctamente.
- **Estilos:** Los estilos mantienen márgenes, bordes redondeados y colores diferenciados para cada tipo de scroll.

7. Resultados

Al ejecutar el código:

- Se muestra una lista de cuadros verticales que pueden desplazarse hacia abajo.
- Al llegar a la sección inferior, se observa otro grupo de cuadros desplazables de forma horizontal.
- Ambos scrolls funcionan de manera fluida e independiente.

8. Conclusión

El componente ScrollView es una herramienta fundamental en React Native para manipular contenido desplazable. Esto resulta intuitivo y permite implementar desplazamientos verticales y horizontales con facilidad. Aunque debe usarse con precaución en listas muy grandes, resulta ideal para diseños con múltiples elementos dentro de un espacio limitado.