



**Universidad Politécnica de Querétaro**

**Ingeniería en Tecnologías de la Información e Innovación Digital**

**Grupo: TIID-214**

**Desarrollo de Aplicaciones Móviles**

Iván Isay Guerra López

**Practica No.12: ScrollView & ActivityIndicator**

**Por**

Galarza Piña Ruth Verónica

30- 10 – 2025

## ScrollView

1. Lo primero que se realizara en esta práctica será poner las importaciones necesarias para que funcione correctamente nuestra función ScrollView que en este caso serán las de RefreshControl y ScrollView.

```
import { Text, StyleSheet, View, Pressable, TouchableOpacity, RefreshControl } from 'react-native';
import { useState } from "react";
import { Switch, TouchableWithoutFeedback, ScrollView} from 'react-native';

export default function ScrollScreen(){
  const [accion, setAccion] = useState('nada');
  const [isDisabled, setIsDisabled] = useState(false);
```

2. Ahora, para ahorrar un poco de trabajo reutilizaremos el código que vimos en practicas anteriores que fue la de “Botones”, en la parte de return que antes teníamos esto

```
return (
  <View style={styles.container}>
    <text>
      Activado:
    </text>
    <Switch
      onValueChange={() => setDisabled(!isDisabled)}
      value={isDisabled}
    />
```

aquí en esta parte

pondremos lo que se ve a continuación

```
return (
  <ScrollView // llamada al componente scrollview
  showsVerticalScrollIndicator={false}

  refreshControl = {
    <RefreshControl refreshing ={refreshing} onRefresh={onRefresh} />
  }

  | | stickyHeaderIndices={[0]} // sirve para fijar elementos en la parte superior
```

en esta parte

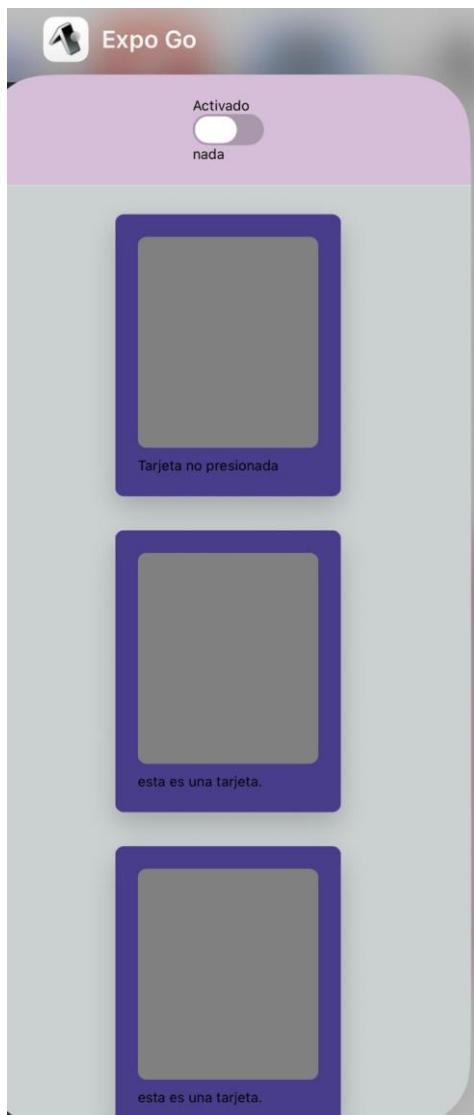
llamaremos al ScrollView y agregaremos la función de stickyHeaderIndices la cual como vemos nos mantiene fijos los elementos a comparación de en los botones.

3. Y cerramos el scroll hasta la parte de abajo en la parte que esta antes de entrar a la sección de estilos.

```
</TouchableOpacity>
<TouchableWithoutFeedback
disabled={isDisabled}
onPress={() => setAccion('Whitout Feedback')}
>
  <View style={[styles.card, styles.shadow]}>
    <View style={styles.mockImage}/>
    <Text>
      | esta es una tarjeta.
    </Text>
  </View>
</TouchableWithoutFeedback>

</View>
<ScrollView> // termina scrollview
  );
}
```

4. Y asi es como se ve, es muy parecido a la actividad pasada de botones solo que algo "mejorada" por así decirlo.



## ActivityIndicator

1. Ahora pasando al siguiente tema lo que realizaremos es un indicador que nos muestre cuando alguna cosa este cargando, para eso empezaremos importando ActivityIndicator.

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

2. Una vez hecho esto pasaremos a poner un temporizador como se ve a continuación cuando a activar para cuando se ejecute el botón.

```
export default function ActivityIndicatorScreen() {
  const [cargando, setCargando] = useState(false);

  const iniciarCarga = () => {
    setCargando(true);
    setTimeout(() => setCargando(false), 3000);
  };
}
```

3. En este apartado utilizaremos el componente de ActivityIndicator el cual nos servirá para generar la sección en la cual se verá el indicador de cargando.

```
return (
  <View style={styles.container}>
    <Text style={styles.texto}>Presiona para iniciar carga</Text>

    <ActivityIndicator
      style={styles.indicador}
      animating={cargando}
      size="large"
      color="#007BFF"
      hidesWhenStopped={true}
    />

    <View style={styles.botonContainer}>
      <Button color="red" title="Iniciar carga" onPress={iniciarCarga} />
    </View>
  </View>
);
```

4. Como podemos ver lo que se ve a continuación es la pantalla antes de que se inicie la carga de este, en el celular no se alcanza a distinguir el recuadro pero en su versión web se puede ver el recuadro de botón.

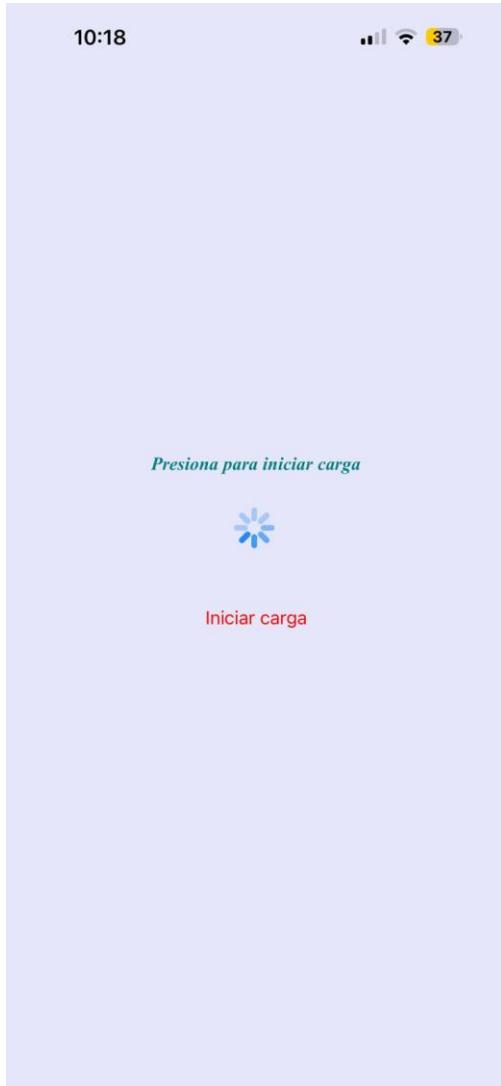
10:18

• 1 || ⌂ 37 ⌂

*Presiona para iniciar carga*

Iniciar carga

5. Una vez hecho eso lo que se vera el como se muestra la animación de carga.



Conclusión:

Se pusieron en práctica los temas vistos en clase, me había costado al principio pero fue por culpa del menú que había colocado una mayúscula por accidente lo cual me atraso pero al final logre hacer que funcionaran de una forma correcta.