

View, ScrollView, Text, TextInput e Image.

View = div en React

View

- Qué es: Es el componente más básico de React Native, similar a un `div` en HTML. Se usa para crear contenedores para otros componentes.
- Uso: Sirve como contenedor para otros componentes, y se puede usar para aplicar estilos, manejar layouts y gestionar eventos táctiles.
- Ejemplo:

```
jsx
import React from 'react';
import { View, Text } from 'react-native';

const MyComponent = () => {
  return (
    <View style={{ padding: 20 }}>
      <Text>Hello, View!</Text>
    </View>
  );
};

export default MyComponent;
```

ScrollView

- Qué es: Es un contenedor de desplazamiento que permite a los usuarios desplazarse por el contenido cuando este excede el tamaño de la pantalla.
- Uso: Ideal para listas de contenido, formularios largos o cualquier cosa que necesite desplazamiento vertical u horizontal.
- Ejemplo:

```
jsx
import React from 'react';
import { ScrollView, Text } from 'react-native';

const MyScrollView = () => {
  return (
    <ScrollView>
      <Text>Item 1</Text>
      <Text>Item 2</Text>
      <Text>Item 3</Text>
      <Text>Item 4</Text>
      <Text>Item 5</Text>
      <Text>Item 6</Text>
    </ScrollView>
  );
};

export default MyScrollView;
```

+ ideal para listas de contenido, formularios

Text

- Qué es: Componente usado para mostrar texto.
- Uso: Puede ser estilizado y anidado dentro de otros componentes `Text` o `View`.
- Ejemplo:

```
jsx
import React from 'react';
import { Text } from 'react-native';

const MyText = () => {
  return (
    <Text style={{ fontSize: 20, color: 'blue' }}>
      Hello, Text!
    </Text>
  );
};

export default MyText;
```

Text, id

TextInput

- Qué es: Componente usado para recibir entradas de texto del usuario.
- Uso: Útil para formularios, búsquedas, o cualquier lugar donde necesites que el usuario ingrese texto.
- Ejemplo:

```
jsx
import React, { useState } from 'react';
import { TextInput, View } from 'react-native';

const MyTextInput = () => {
  const [text, setText] = useState('');

  return (
    <View>
      <TextInput
        style={{ height: 40, borderColor: 'gray', borderWidth: 1 }}
        onChangeText={text => setText(text)}
        value={text}
      />
    </View>
  );
};

export default MyTextInput;
```

Image

- Qué es: Componente para mostrar imágenes.
- Uso: Se utiliza para cargar y mostrar imágenes desde la red, el sistema de archivos local o assets.
- Ejemplo:

```
jsx
import React from 'react';
import { Image, View } from 'react-native';

const MyImage = () => {
  return (
    <View>
      <Image
        source={{ uri: 'https://reactnative.dev/img/tiny_logo.png' }}
        style={{ width: 50, height: 50 }}
      />
    </View>
  );
};

export default MyImage;
```

Expo router



Expo Router es una herramienta que simplifica la navegación en aplicaciones desarrolladas con Expo y React Native. Proporciona una manera estructurada y fácil de definir y gestionar rutas en tu aplicación, similar a cómo funciona la navegación en aplicaciones web con frameworks como React Router.

Características principales de Expo Router

1. **Declarativo:** Permite definir rutas de manera declarativa usando archivos y directorios en lugar de configuraciones complicadas.
2. **Automático:** Genera automáticamente las rutas basándose en la estructura de los archivos dentro de un directorio específico.
3. **Integración con Expo:** Diseñado específicamente para trabajar sin problemas con aplicaciones Expo, aprovechando sus características y optimizaciones.

