Universidad Politécnica de Madrid

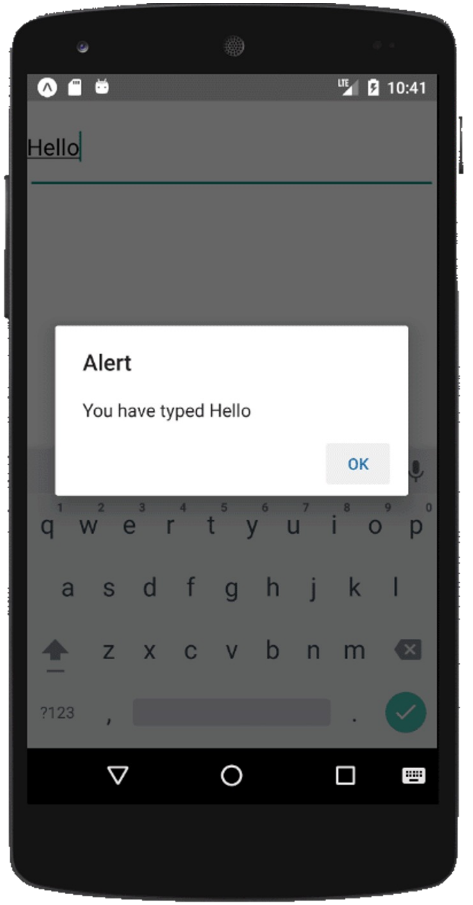
React Native

Componentes básicos

Álvaro Alonso González
Enrique Barra Arias

TextInput

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

export default function App () {
  const _onTextInputChange = (text) => {
    if(text === "Hello"){
      alert("You have typed Hello")
    }
  }
  return (
      <View>
       <TextInput style={{height: 80, fontSize: 20}}
         placeholder= "Type Hello"
         onChangeText={_onTextInputChange} />
      </View>
    )

}
```


10:41

Hello

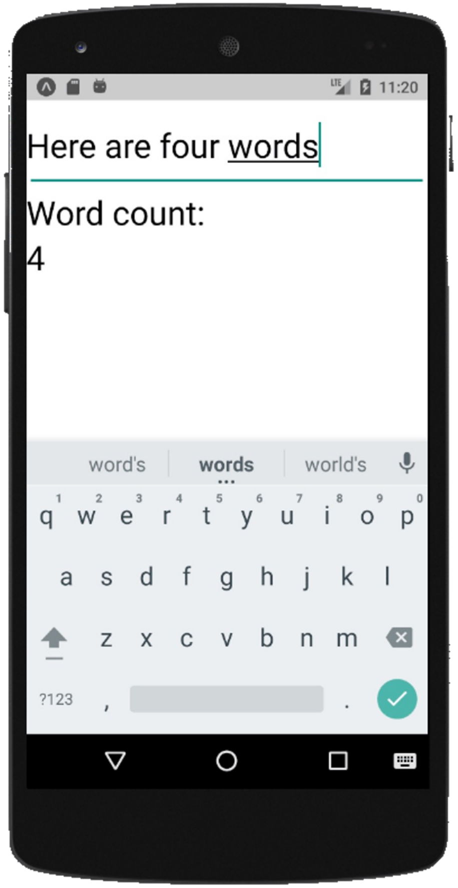
Alert

You have typed Hello

OK

Text y TextInput

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

export default function App () {
   const [value, setValue] = useState(“”);

  const _onTextInputChange = (text) => {
    setValue(text);
  }
  return (
      <View>
        <TextInput style={{height: 80, fontSize: 30}} placeholder="Type here"
           onChangeText={_onTextInputChange} />
        <Text style={{fontSize: 30}}>Chars count:</Text>
        <Text style={{fontSize: 30}}>
          {value. length}
        </Text>
      </View>
    )

}
```



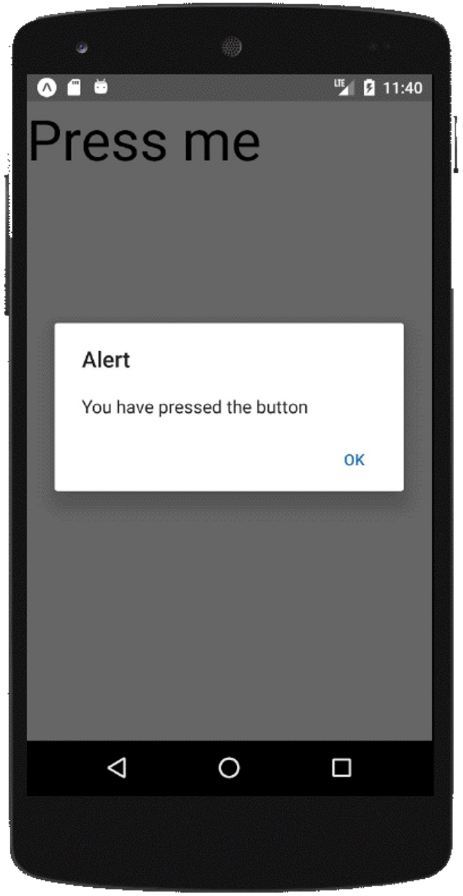
Here are four words

Word count:

4

TouchableHighlight

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

export default function App () {
  const _onPressButton= () => {
    alert("You have pressed the button")
  }
  return (
     <View>
      <TouchableHighlight onPress={_onPressButton}>
        <Text style={{fontSize: 50}}>Press me</Text>
      </TouchableHighlight>
     </View>
    )

}
```



Press me

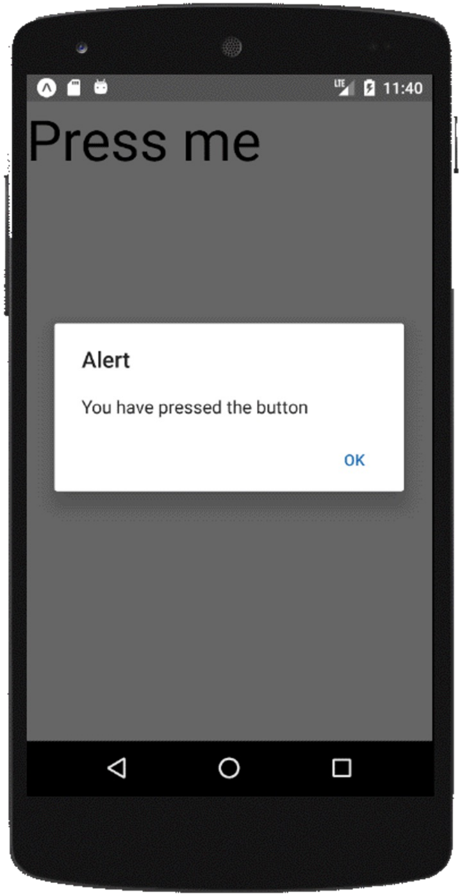
Alert

You have pressed the button

Componente MyButton

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

export default function MyButton (props) {
  return (
    <TouchableHighlight onPress={props.onClick}>
      <Text style={{fontSize: 50}}>{props.text}</Text>
    </TouchableHighlight>
   )

}
```

```
import React from 'react’;
import MyButton from “./MyButton’;

export default function App () {
  return (
    <MyButton
      text="Press me"
      onClick={() => alert("You have pressed the button")}>
    </MyButton>
   )

}
```



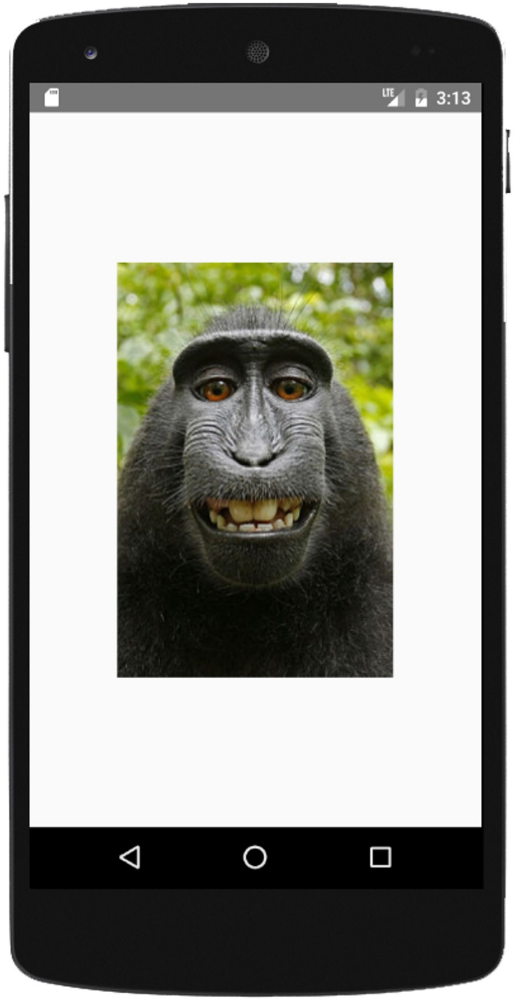
Press me

Alert

You have pressed the button

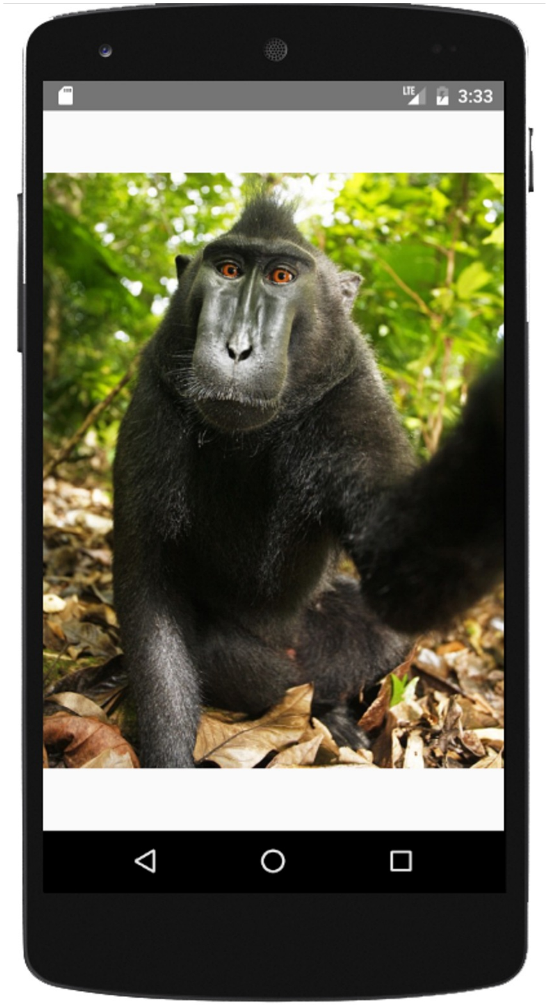
Image

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

export default function App () {
  return (
      <View style={{
        flex:1,
        alignItems: 'center',
        justifyContent: 'center'
      }}>
        <Image source={require('./img/macaca_nigra.jpg')} />
      </View>
    )

}
```



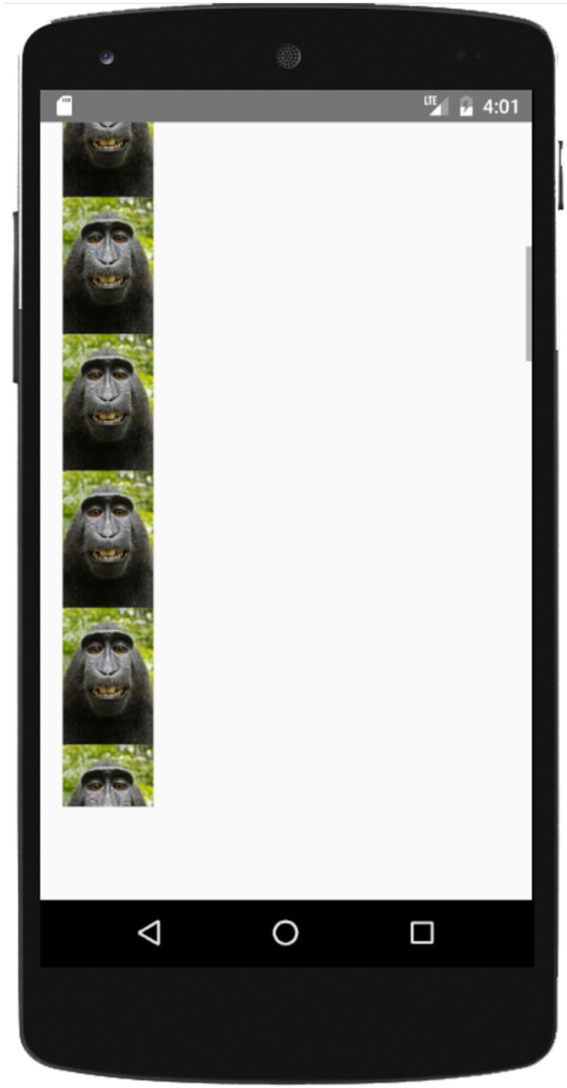
Image

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

export default function App () {
  return (
    <View style={{
      flex:1
    }}>
      <Image
        resizeMode='contain'
        style={{flex:1}}
        source={{uri: "https://vishub.org/pictures/8490.jpg"}}
      />
    </View>
  )

}
```



ScrollView

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

export default function App () {
  var images = [];
  for(let i=0; i<30; i++){
    images.push(
      <Image resizeMode='contain'
        style={{width:100, height: 100}}
        source={require('./img/macaca_nigra.jpg')} key={i} />
    )
  }
  return (
    <View style={{height: 500}}>
     <ScrollView>{images}</ScrollView>
    </View>
  )

}
```



ListViews

- React Native proporciona varios componentes para presentar listas de datos (https://reactnative.dev/docs/using-a-listview)
  - **FlatList** (https://reactnative.dev/docs/flatlist)
    - Permite mostrar de forma eficiente datos similarmente estructurados en listas con scroll
  - **SectionList** https://reactnative.dev/docs/sectionlist)
    - Permite mostrar conjuntos de datos en listas con diferentes secciones
    - Para crear listas sin secciones mejor usar FlatList

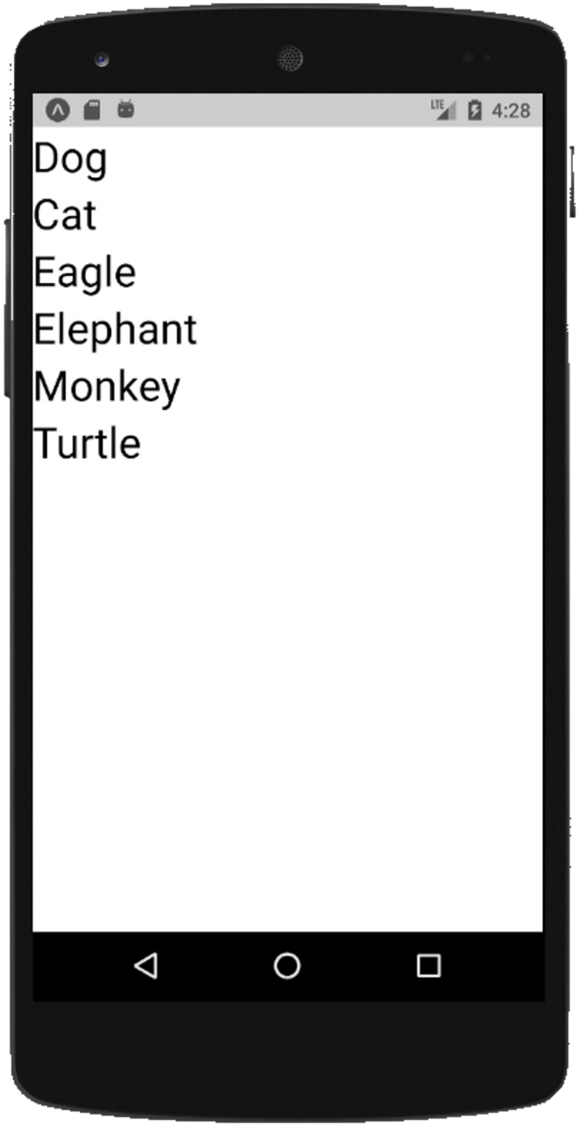
Tras crear una lista con FlatList o SectionList se pueden modificar los datos de la misma

FlatList

- **FlatList** permite mostrar de forma eficiente datos similarmente estructurados en listas verticales u horizontales con scroll
- Los datos se pueden modificar una vez creada la lista
- Componente pensado para grandes listas de datos: *FlatList* solo renderiza aquellos elementos que se muestran por pantalla mientras que *ScrollView* renderiza todos los elementos
- FlatList requiere dos propiedades:
  - **data**: fuente de información de la lista
  - **renderItem**: función que recibe como parámetro un elemento de la lista y devuelve un componente renderizable

FlatList

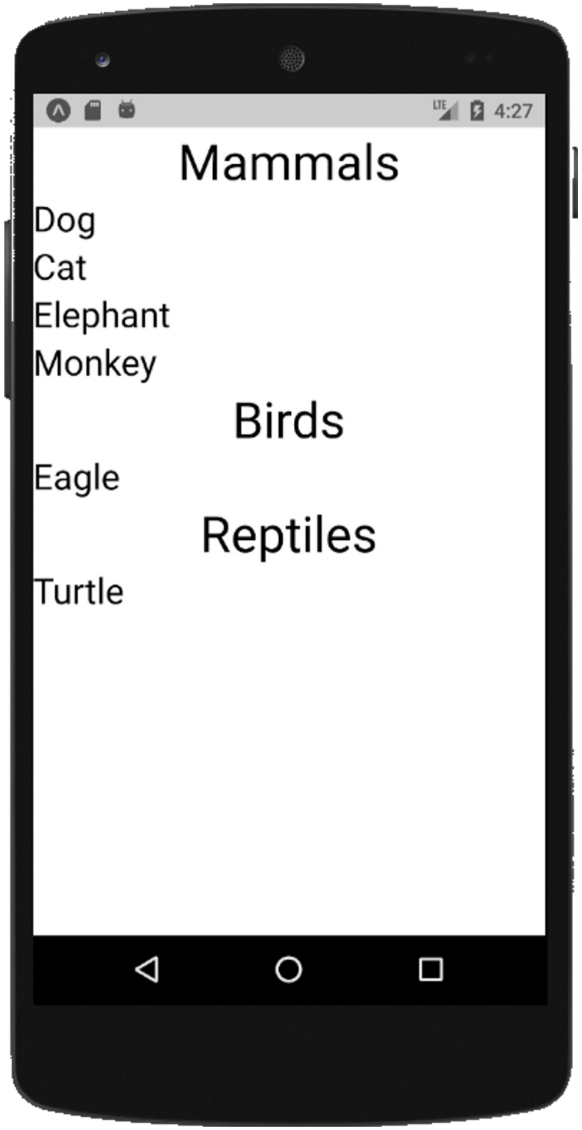
```
import React from 'react';
import { Text, FlatList } from 'react-native';

export default function App () {

  var animals = [{key:'Dog'},{key:'Cat'},{key:'Eagle'},
        {key:'Elephant'},{key:'Monkey'},{key:'Turtle'}];
   return (
    <FlatList
      data={animals}
      renderItem={({item}) =>
      <Text style={{fontSize: 30}}>{item.key}</Text>}
    />
   )

}
```



```
Dog
Cat
Eagle
Elephant
Monkey
Turtle
```

SectionList

```
import React from 'react';
import { Text, SectionList } from 'react-native';
export default function App () {

  var mammals = {key:'Dog'},{key:'Cat'},{key:'Elephant'},
                {key:'Monkey'}];
  var birds = [{key:'Eagle'}];
  var reptiles = [{key:'Turtle'}];
  return (
   <SectionList
    sections={[
     {data: mammals, title: 'Mammals'},
     {data: birds, title: 'Birds'},
     {data: reptiles, title: 'Reptiles'}
    ]}
    renderItem={({item}) =>
     <Text style={{fontSize:25}}>{item.key}</Text>}
    renderSectionHeader={({section}) =>
      <Text style={{textAlign:'center', fontSize:35}}>
     {section.title}</Text>}
   />
  )
}
```



```
Mammals
Dog
Cat
Elephant
Monkey
Birds
Eagle
Reptiles
Turtle
```

MOOC

Universidad Politécnica de Madrid

React Native

Componentes básicos

Álvaro Alonso González
Enrique Barra Arias