

## ACTIVIDADES PRÁCTICAS:

Título      Funciones de orden superior en React Native: map

---

Objetivos

---

*Aprender a utilizar map aplicado a componentes en React Native.*

Temporalización

---

*Dedicación estimada: 1,5 horas.*

Actividades:

---

En este bloque de actividades prácticas aprenderemos a aplicar la función de orden superior ‘map’ a componentes. Trabajaremos a partir del siguiente código:

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

export default function App() {
  return (
    <View style={styles.container}>
      <View style={{ flexDirection: 'row' }}>
        <View width={100} height={100} backgroundColor={'yellow'} />
      </View>
      <View style={{ flexDirection: 'row' }}>
        <View
          width={0}
          height={0}
          backgroundColor={'transparent'}
          borderStyle={'solid'}
          borderLeftWidth={50}
          borderRightWidth={50}
          borderBottomWidth={100}
          borderLeftColor="transparent"
          borderRightColor="transparent"
          borderBottomColor="blue"
        />
      </View>
      <View style={{ flexDirection: 'row' }}>
        <View
          width={100}
          height={100}
          borderRadius={100 / 2}
        />
      </View>
    </View>
  );
}
```

```

        backgroundColor={'green'}
      />
    </View>
    <View style={{ flexDirection: 'row' }}>
      <View
        width={100}
        height={100}
        borderTopLeftRadius={150}
        borderTopRightRadius={0}
        borderBottomLeftRadius={0}
        borderBottomRightRadius={0}
        backgroundColor={'red'}
      />
      <View
        width={100}
        height={100}
        borderTopLeftRadius={0}
        borderTopRightRadius={150}
        borderBottomLeftRadius={0}
        borderBottomRightRadius={0}
        backgroundColor={'black'}
      />
    </View>
    <View style={{ flexDirection: 'row' }}>
      <View
        width={100}
        height={100}
        borderTopLeftRadius={0}
        borderTopRightRadius={0}
        borderBottomLeftRadius={150}
        borderBottomRightRadius={0}
        backgroundColor={'black'}
      />
      <View
        width={100}
        height={100}
        borderTopLeftRadius={0}
        borderTopRightRadius={0}
        borderBottomLeftRadius={0}
        borderBottomRightRadius={150}
        backgroundColor={'red'}
      />
    </View>
  </View>
);
}

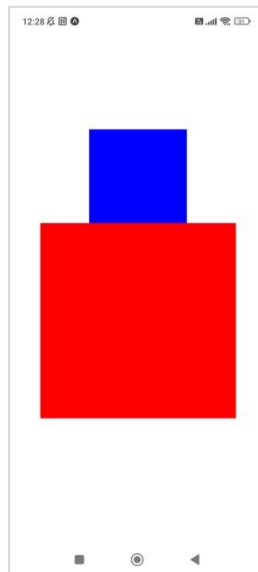
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});

```

El código proporcionado contiene varias figuras geométricas –cuadrado, rectángulo, círculo y un círculo dividido en cuatro-. Debes utilizar en cada ejercicio únicamente la parte de código necesaria para poder implementar las soluciones.

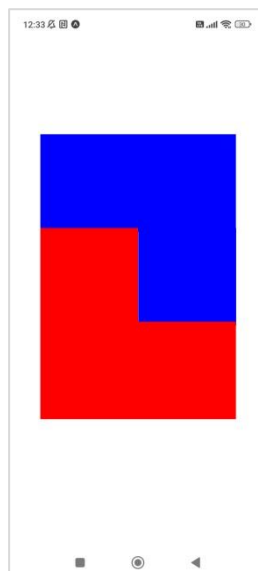
### Ejercicio 1.

A partir del código proporcionado, utiliza `map` y un `cuadrado` para renderizar una figura similar a la siguiente:



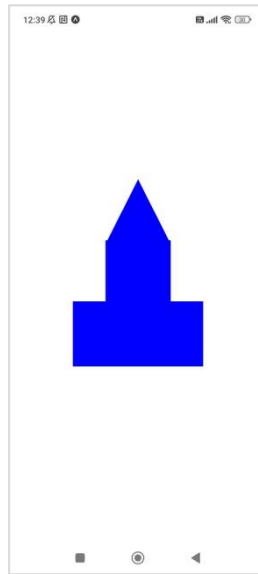
### Ejercicio 2.

A partir del código proporcionado, utiliza `map` y un `cuadrado` para renderizar una figura similar a la siguiente:



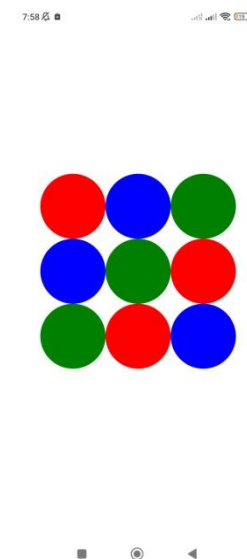
### Ejercicio 3.

A partir del código proporcionado, utiliza map, un cuadrado y un triángulo para renderizar una figura similar a la siguiente:



### Ejercicio 4.

A partir del código proporcionado, utiliza map y un círculo para renderizar una figura similar a la siguiente:



### Ejercicio 5.

A partir del código proporcionado, utiliza map y un círculo dividido entre cuatro para renderizar una figura similar a la siguiente:

