



# TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TLAXIACO

**CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**Desarrollo Móvil**

Como usar mapas en React Native

**DOCENTE: Sánchez Salmorán Eloy**

**Alumno: Edwin Ortiz Cruz**

**SEMESTRE: SÉPTIMO**

**GRUPO: 7US**

agosto-diciembre 2023

Heroica Ciudad de Tlaxiaco, Oaxaca, a 28 de noviembre del  
2023.





## Introducción

Los mapas son una herramienta esencial para muchas aplicaciones móviles. Pueden utilizarse para navegar, encontrar lugares, o simplemente para proporcionar información sobre el entorno. React Native es un framework que permite crear aplicaciones móviles nativas para iOS y Android. También ofrece una biblioteca de mapas que permite integrar mapas de Google Maps en las aplicaciones React Native.

En React Native, el uso de mapas es común en aplicaciones que requieren la visualización de ubicaciones geográficas, como mapas interactivos. Para integrar mapas en una aplicación React Native, puedes utilizar bibliotecas como "react-native-maps" que proporcionan componentes para trabajar con mapas de manera sencilla.

## Desarrollo

### Instalar la librería react-native-maps con Google Maps

Una vez creado el proyecto el siguiente paso es añadir la librería de mapas para ello podemos utilizar el siguiente comando.

```
npm install react-native-maps --save-exact
```

```
o
```

```
yarn add react-native-maps -E
```

La diferencia entre el primer comando y el segundo es que usando el cli de Expo nos aseguramos de utilizar la última versión de la librería compatible con Expo.

Cabe mencionar que podemos utilizar la librería de **react-native-maps** tanto con **Apple Maps** como con **Google Maps**. En este tutorial nos centraremos en utilizar Google Maps como proveedor de mapas, pero los pasos para integrar Apple Maps son muy similares.

### Obtener la API Key de Google Maps

Para poder utilizar Google Maps en nuestra aplicación será necesario **habilitar el SDK de iOS y Android** en un proyecto de Google con una cuenta de facturación activa en Google Cloud Console y generar una API key para añadirla al código de nuestra aplicación.

### Añadir la API Key en Android

Si se trata de un proyecto Bare, en Android será necesario añadir el API Key en `google_maps_api.xml` en la ruta `android/app/src/main/res/values`.

```
<resources>
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">(api
key here)</string>
</resources>
```

### Añadir y personalizar un mapa en React Native



Ahora que ya tenemos la librería de mapas integrada vamos a empezar por crear una pantalla con la visualización del mapa y personalizar el estilo con las diferentes opciones que nos proporciona. Para ello vamos a **crear un componente Map.js** como el siguiente.

```
import React from 'react';
import { StyleSheet, View, Dimensions } from 'react-native';
import MapView, { PROVIDER_GOOGLE } from 'react-native-maps';
import { mapStyle } from './mapStyle';

export function MapScreen() {
  return (
    <View style={styles.container}>
      <MapView
        customMapStyle={mapStyle}
        provider={PROVIDER_GOOGLE}
        style={styles.mapStyle}
        initialRegion={{
          latitude: 41.3995345,
          longitude: 2.1909796,
          latitudeDelta: 0.003,
          longitudeDelta: 0.003,
        }}
        mapType="standard"
      ></MapView>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'black',
    alignItems: 'center',
    justifyContent: 'center',
  },
  mapStyle: {
    width: Dimensions.get('window').width,
    height: Dimensions.get('window').height,
  },
});
```

Como podemos ver el componente principal es **MapView** que cuenta con multiples props para personalizar su comportamiento. En este caso los más importantes son *provider* donde le





indicamos que queremos utilizar Google Maps, *initialRegion* que será la localización que cargará el mapa en un inicio, *mapType* donde podemos definir el tipo de mapa que se carga y por último *customMapStyle* donde indicaremos el estilo personalizado del mapa que queremos utilizar.

### Añadir Marcadores a Google Mapas en React Native

Lo siguiente que haremos será añadir marcadores a nuestro mapa. Para ello crearemos una constante `MARKERS_DATA` con la siguiente estructura.

```
import { default as Reboot } from '../assets/reboot.png';
import { default as Cravy } from '../assets/cravy.png';
import { default as Dribbble } from '../assets/dribbble.png';
import { default as Basecamp } from '../assets/basecamp.png';
import { default as Discord } from '../assets/discord.png';
import { default as OnePassword } from '../assets/onepassword.png';

export const MARKERS_DATA = [
  {
    id: '1',
    latitude: 41.3997999,
    longitude: 2.1909796,
    color: '#2F3136',
    name: 'Reboot Studio',
    direction: 'Carrer de Pujades, 100',
    img: Reboot,
  },
  {
    id: '2',
    latitude: 41.3995445,
    longitude: 2.1915268,
    color: '#A3EAD8',
    name: 'Cravy',
    direction: 'Carrer de Pujades, 101',
    img: Cravy,
  },
  {
    id: '3',
    latitude: 41.4009999,
    longitude: 2.1919999,
    color: '#E990BB',
    name: 'Dribbble',
    direction: 'Carrer de Pujades, 102',
    img: Dribbble,
  },
]
```



```
{
  id: '4',
  latitude: 41.4001999,
  longitude: 2.1900096,
  color: '#EFD080',
  name: 'Basecamp',
  direction: 'Carrer de Pujades, 103',
  img: Basecamp,
},
{
  id: '5',
  latitude: 41.40009,
  longitude: 2.1909796,
  color: '#98AFE9',
  name: 'Discord',
  direction: 'Carrer de Pujades, 104',
  img: Discord,
},
{
  id: '6',
  latitude: 41.4009999,
  longitude: 2.1909796,
  color: '#4E87EB',
  name: '1 Password',
  direction: 'Carrer de Pujades, 105',
  img: OnePassword,
},
]
```

Una vez tenemos nuestros datos preparados podemos añadirlos al mapa importando el componente Marker que nos da la librería de mapas y añadirlos dentro de la MapView. Para ello usaremos una función de Array.map juntamente con la estructura de datos MARKERS\_DATA que hemos creado.

### Gestionar la navegación por el mapa

Vamos a ver cómo podemos navegar por el mapa usando tanto la función de animateCameracomo la función de animateToRegion. Para ello deberemos crear una referencia del mapa para poder utilizarla y llamar a estas funciones, en nuestro caso hemos decidido crear un hook para gestionar esta lógica.

```
//useMap.js
```



Boulevard Tecnológico Km. 2.5, Llano Yosovee C.P. 69800. Tlaxiaco, Oaxaca. Tel. (953) 55 21322 y (953) 55 20405, e-mail: dir\_tlaxiaco@tecnm.mx; tecnm.mx | tlaxiaco.tecnm.mx



2023  
AÑO DE  
**Francisco VILLA**  
EL REVOLUCIONARIO DEL PUEBLO



```
import { useState, useRef, useCallback } from 'react';

const DEVIATION = 0.0002;

export function useMap() {
  const mapRef = useRef(null);
  const [selectedMarker, setSelectedMarker] = useState(null);

  const handleNavigateToPoint = useCallback(
    (id, lat, long) => {
      if (mapRef) {
        mapRef.current.animateCamera(
          {
            center: {
              latitude: lat - DEVIATION,
              longitude: long,
            },
            zoom: 18.5,
          },
          500
        );
        setSelectedMarker(id);
      }
    }, [mapRef, setSelectedMarker]
  );

  const handelResetInitialPosition = useCallback(() => {
    if (mapRef) {
      mapRef.current.animateToRegion(
        {
          latitude: 41.3995345,
          longitude: 2.1909796,
          latitudeDelta: 0.003,
          longitudeDelta: 0.003,
        },
        500
      );
      setSelectedMarker(null);
    }
  }, [mapRef, setSelectedMarker]);
}
```







```
return {
  mapRef,
  selectedMarker,
  handleNavigateToPoint,
  handelResetInitialPosition,
};
```

Como podemos ver en el código las funciones son bastante sencillas. A `animateCamera` le pasamos el centro con la latitud y la longitud el `Zoom` y el tiempo que tardará la animación. En el caso de `animateToRegion` es muy similar pero en cambio de utilizar el `Type Camera` utiliza el `Type Region`.

En nuestro caso también le hemos añadido un `setSelectedMarker` para poder realizar la ampliación del marcador cuando la cámara utilice este como centro.

Para utilizar el hook simplemente tenemos que añadirlo en nuestro componente de Mapa, pero antes de eso crearemos el componente por encima del mapa para poder utilizar las funciones del hook.

Concretamente vamos a crear un componente `Bottom Sheet` con el listado de localizaciones y cuando se pulse encima de una de estas la cámara se moverá a ese punto y el marcador seleccionado se ampliará. Para el componente hemos utilizado la librería '`react-native-scroll-bottom-sheet`' que utiliza `Reanimated` para gestionar las animaciones del componente

## Conclusión:

En conclusión, el uso de mapas en React Native se facilita mediante bibliotecas como "`react-native-maps`". Esta biblioteca permite integrar mapas interactivos en aplicaciones móviles de manera sencilla. Los pasos básicos incluyen la instalación de la biblioteca, la configuración de claves de API para servicios de mapas (como Google Maps), y la utilización de componentes como **MapView** y **Marker** para mostrar ubicaciones geográficas. A través de esta integración, los desarrolladores tienen la capacidad de crear aplicaciones que aprovechan las funcionalidades de mapas para diversas necesidades, como la visualización de ubicaciones, la interacción con marcadores y la personalización de la experiencia del usuario. La documentación de "`react-native-maps`" es una valiosa fuente de información para explorar características avanzadas y opciones de personalización.

## Bibliografía



Boulevard Tecnológico Km. 2.5, Llano Yosovee C.P. 69800. Tlaxiaco, Oaxaca. Tel. (953) 55 21322 y (953) 55 20405, e-mail: [dir\\_tlaxiaco@tecnm.mx](mailto:dir_tlaxiaco@tecnm.mx); [tecnm.mx](http://tecnm.mx) | [tlaxiaco.tecnm.mx](http://tlaxiaco.tecnm.mx)



2023  
AÑO DE  
**Francisco  
VILLA**  
EL REVOLUCIONARIO DEL PUEBLO



Anlramirez. (2021, 14 diciembre). Integrar mapa en React Native usando React-Native-Maps

(2020). *Medium*. <https://medium.com/@anlramirez11/integrar-mapas-en-react-native-usando-react-native-maps-2020-9ed9c81ce476>

Quispe, M. (2023a, junio 16). *Cómo utilizar mapas en React Native - parte 1*. Miguel Quispe.

<https://miguelquispe.com/blog/como-utilizar-mapas-en-react-native-expo-parte-1>

Quispe, M. (2023b, junio 23). *Cómo utilizar mapas en React Native - parte 2*. Miguel Quispe.

<https://miguelquispe.com/blog/como-utilizar-mapas-en-react-native-parte-2>

*React Native Maps: Cómo instalar y usar la librería de mapas en iOS y Android [2020]*. (s. f.).

<https://reboot.studio/blog/es/react-native-maps-2020>

