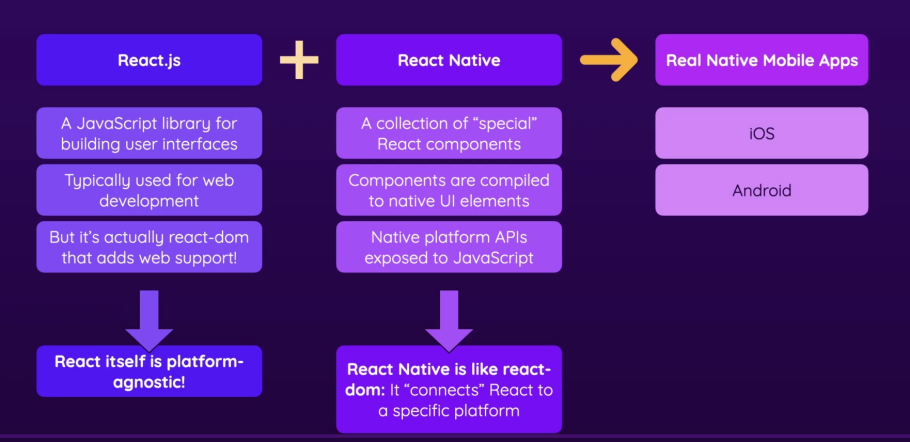
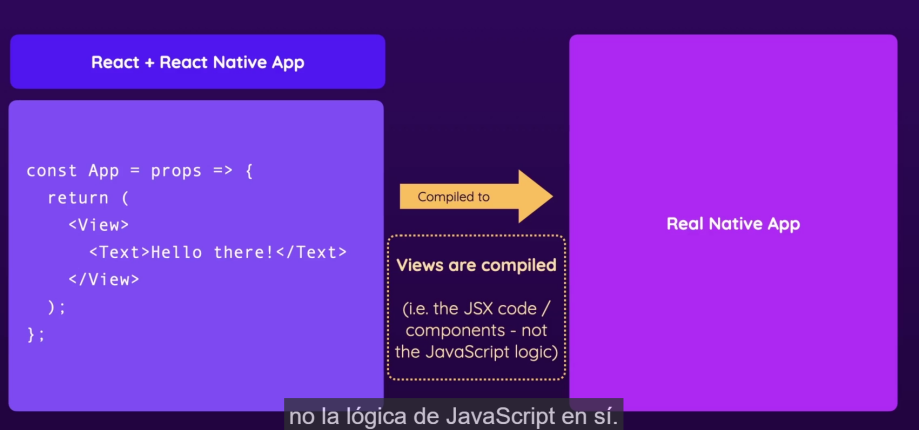
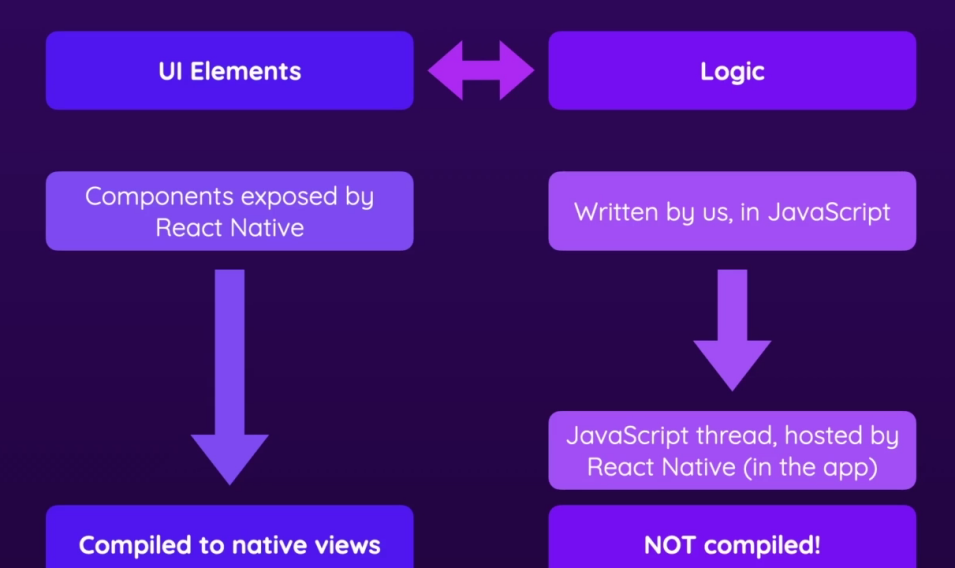
React Native:





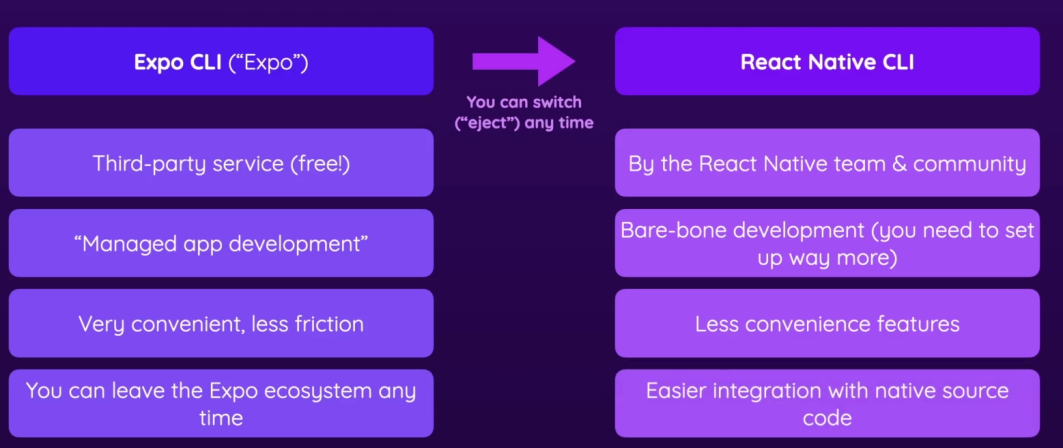
* View y Text son componentes especiales de RN que deben compilarse por RN para convertir en JS para Android o IOS, la lógica de JS no se compila.



Comparación WEB, Android, Ios y React Native (RN)

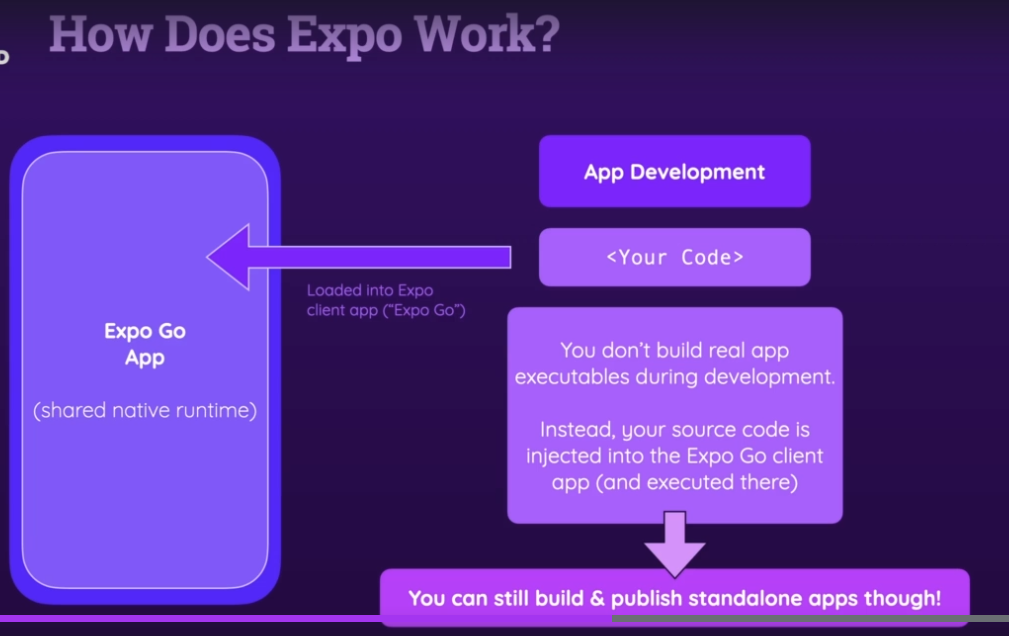


Proyectos RN

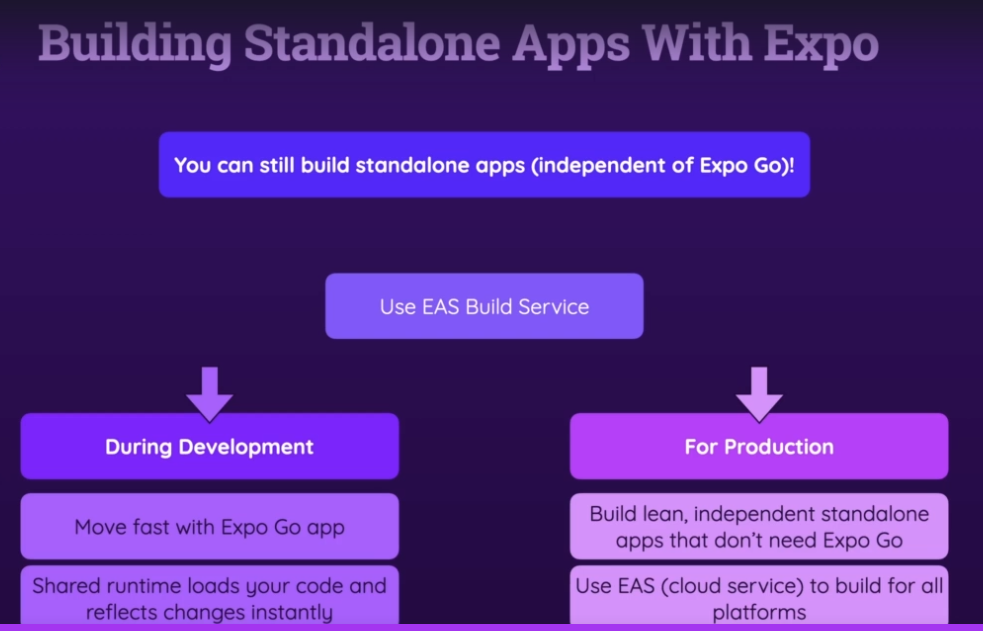


**Expo Go APP**

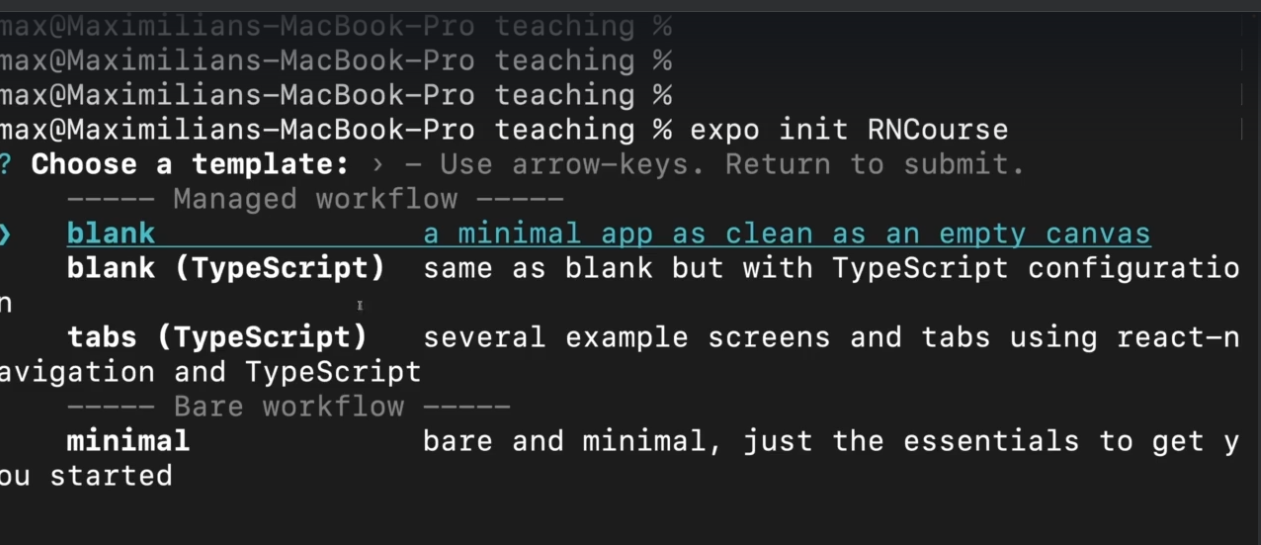
* Se crea un código fuente que se inyecta a Expo Go allí en tiempo real se muestra como una APP local y no como un ejecutable compilado. Luego de terminar el desarrollo se puede crear una APP y publicarla como una APP independiente para que los usuarios de mi APP no necesiten exportarla a través de Expo Go. Conclusión Expo Go es una ayuda durante es desarrollo



* Construir APP independientes (standAlone) se realiza con EAS Build Service. SE genera una APP independiente para Producción que no necesita Expo Go



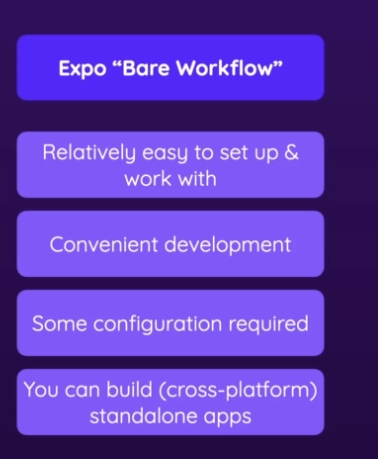
* Expo es realmente útil durante el desarrollo y ayuda a que sea más fácil
* Expo init permite crear App fácil y rápidamente. Disminuye tiempo de configuración, integra mucho código que con las características de los dispositivos para que sea mas sencillo el desarrollo y configurar o apoyo en el uso de paquetes de terceros



* Se puede utilizar un Expo que da menos configuración, pero más control durante el desarrollo
* Se puede crear y administrar proyectos con React Native CLI, que es una interfaz de línea de comandos proporcionados por el propio equipo de React Native
* Se puede utilizar Expo Go, “Managed Workflow” (Flujo de trabajo administrado)
  + Fácil de configurar y trabajar con el
  + Desarrollo rápido y sin fricciones (problemas entre los componentes)
  + Para paquetes de terceros, ayuda en su configuración en las funciones nativas del dispositivo como por ejemplo la ubicación del dispositivo, fotos, entre otros.
  + Permite crear APP independientes multiplataformas



* Si necesitamos más control, por ejemplo, escribir mi propio código nativo de Swift, Objective-C, Android o Kotlin y unirlo a nuestro RN, entonces se puede lograr utilizando algunas funciones de Expo, pero menos que en el caso anterior. Expo Bare Worflow (Flujo de trabajo básico o simple)
  + Fácil de trabajar y configurar
  + No se puede utilizar Expo Go, porque este desarrollo no corre allí. Porque estamos mezclando código nativo personalizado con el de Expo.
  + La configuración es más compleja, pero se tiene más control
  + Permite crear APP independientes multiplataformas
  + Se puede aun compilar

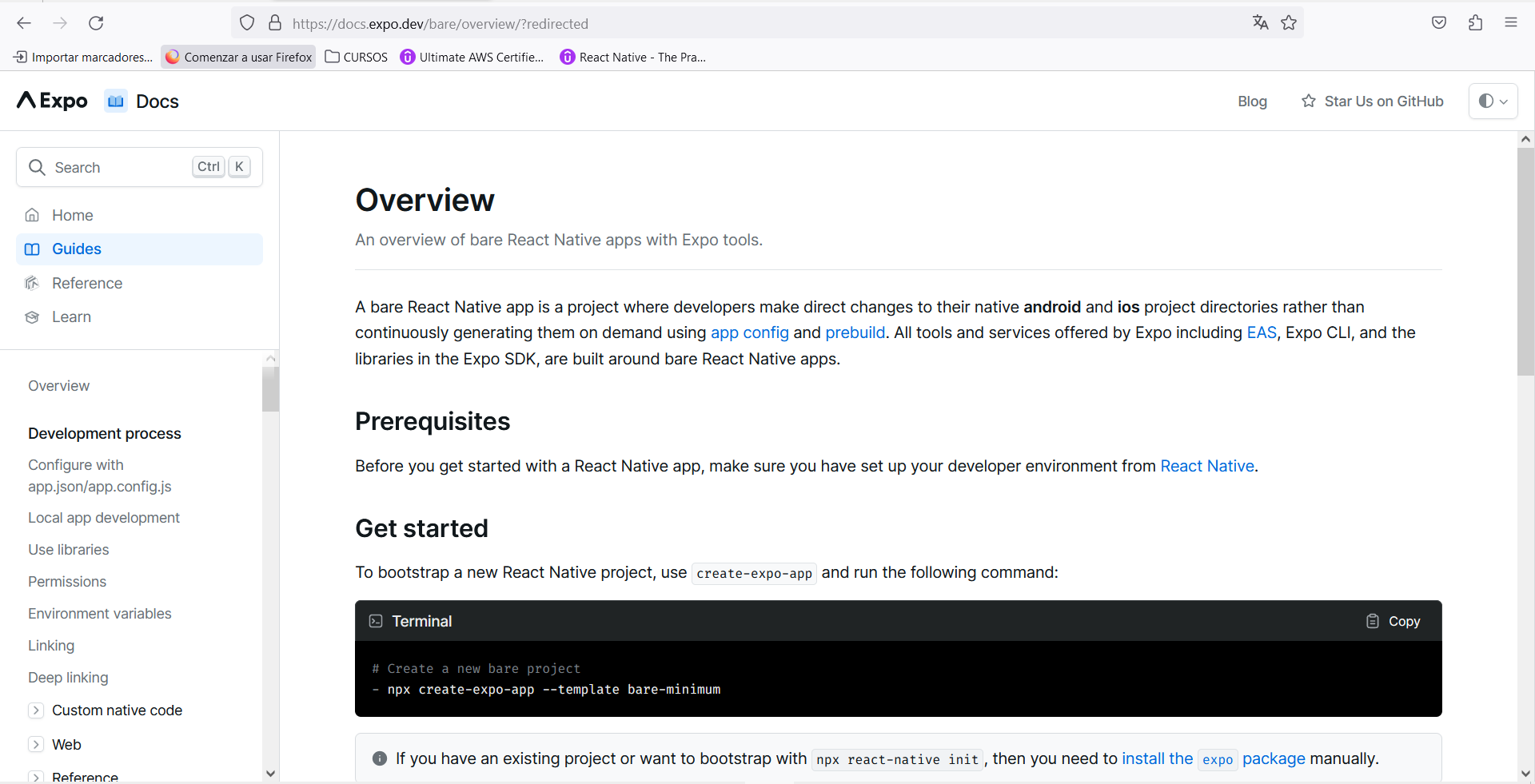


* Se puede crear APP nativas con React Native CLI.
  + Más complejas de configurar, tengo que hacer mas configuraciones por mi cuenta todo el tiempo
  + Cuando agrego paquetes de terceros o cuando necesito aprovechar las funcionalidades nativas del dispositivo a menudo se requiere mas configuración y puede ser más engorroso.
  + Se tiene más control sobre el dispositivo
  + Podría ocurrir que compañías de terceros y proveedores pueden ofrecer compilación en la nube de sus APP y podría compilar mi APP para multiplataforma pero estas funcionalidades podrían no estar integradas en mi APP, no hay manera fácil de configurar esto y, por lo tanto, se debe crear sus APP localmente. Ejemplo en maquinas de Windows no puedo crear App IOS porque es una restricción impuesta por Apple

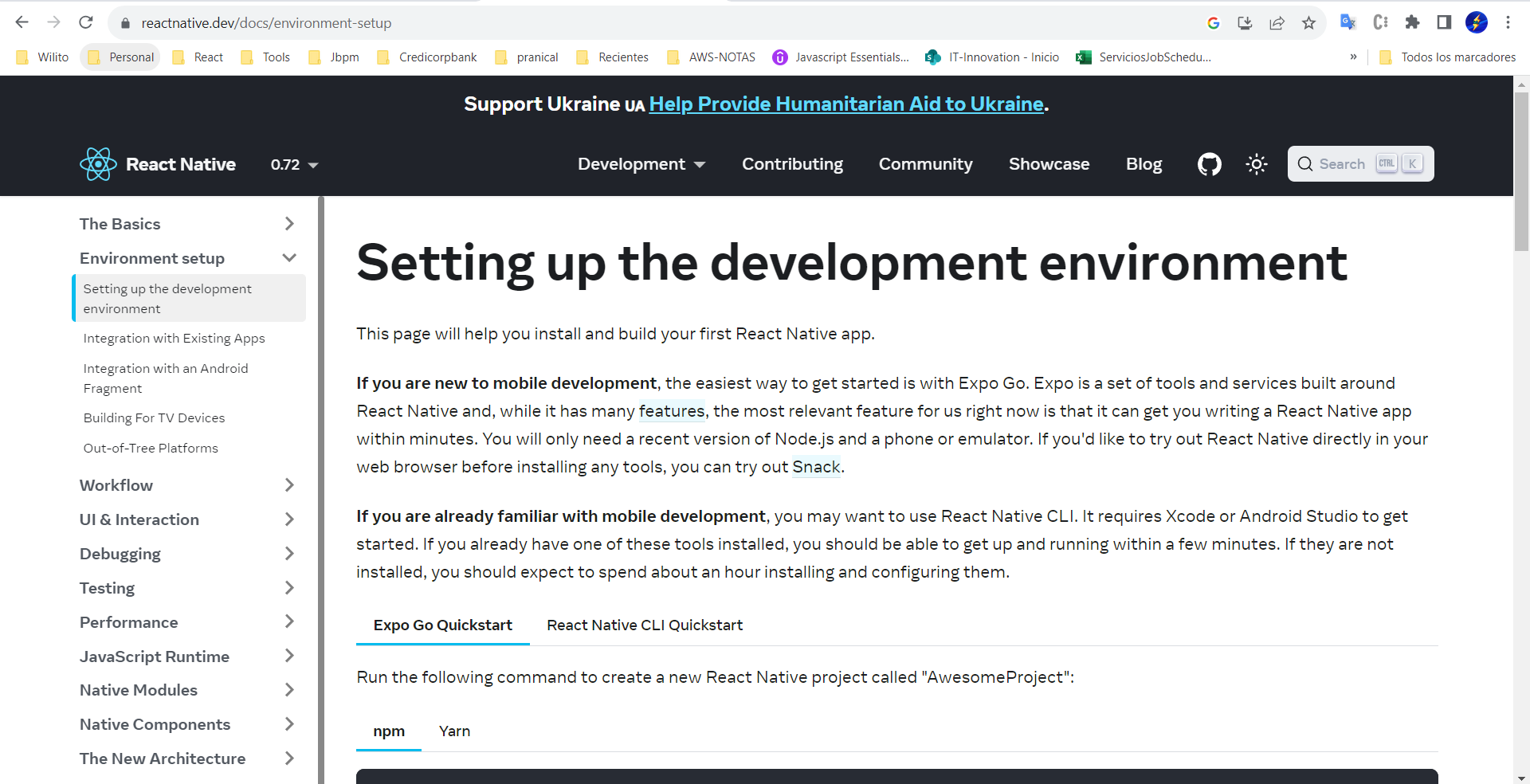


**Expo Bare Worflow (Flujo de trabajo básico o simple)**

* Los desarrollos Expo Bare Workflow son parte expo y parte desarrollo local nativo lo cual no puede ser compilado y utilizado por Expo GO. Se debe crear primero una APP independiente en su máquina antes de poder obtener una vista previa porque no puedo utilizar esta APP de envoltorio en Expo Go.

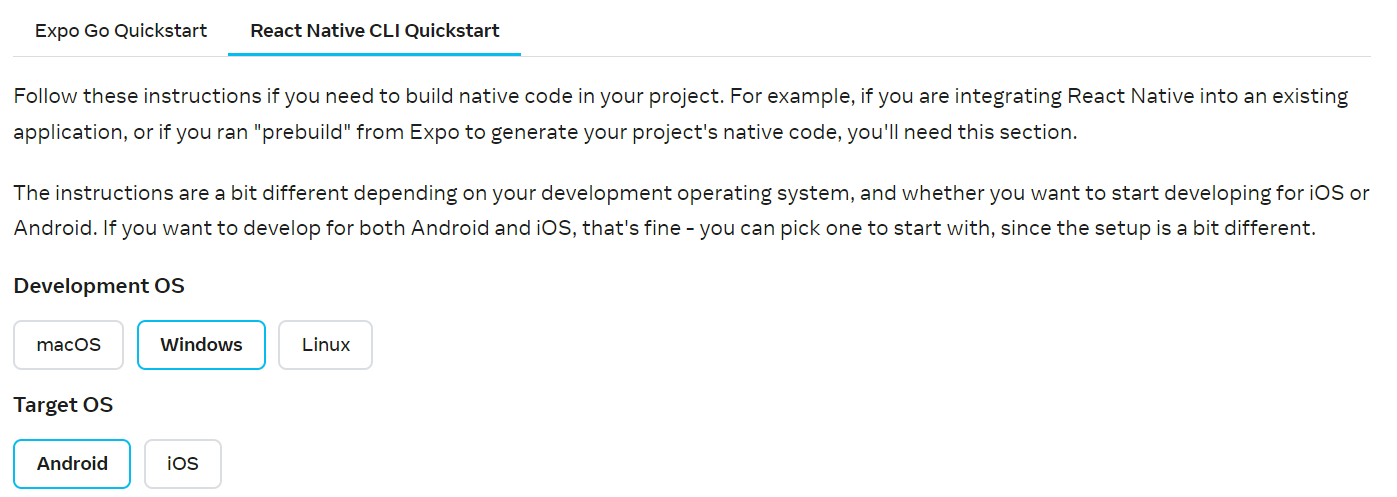


* Se debe ir a React Native en la configuración de entornos y allí configurar su entorno de trabajo para el desarrollo este también se puede utilizar con React Native CLI

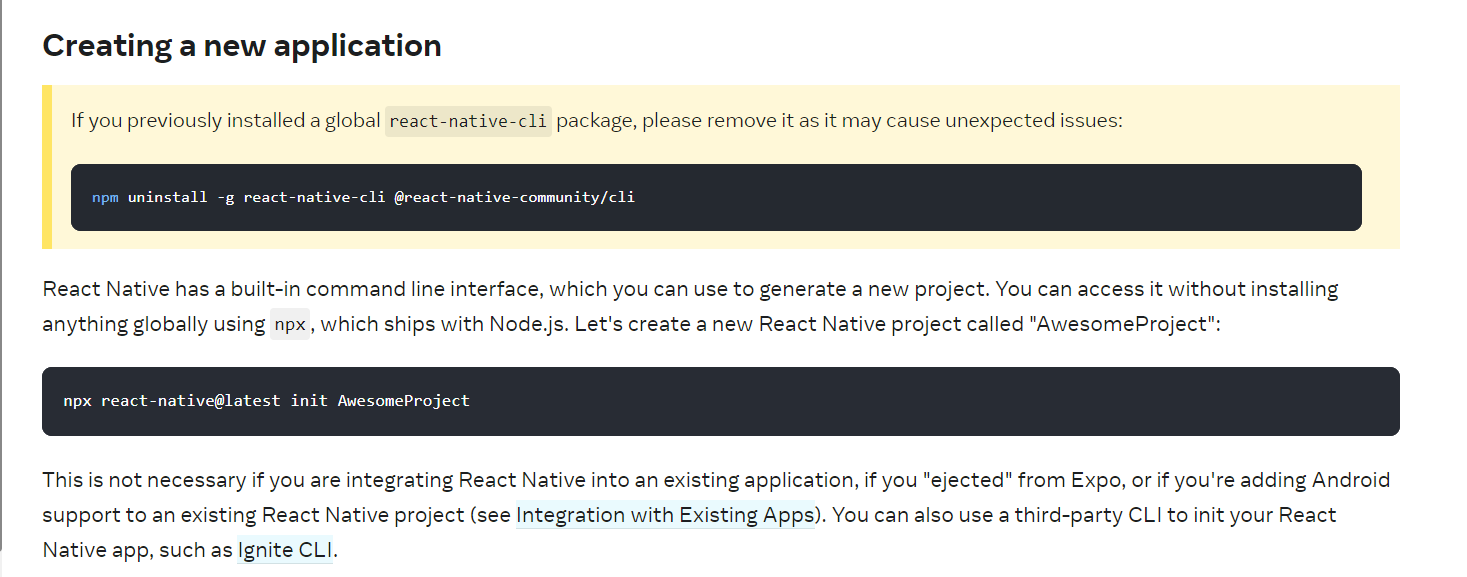


<https://reactnative.dev/docs/environment-setup>

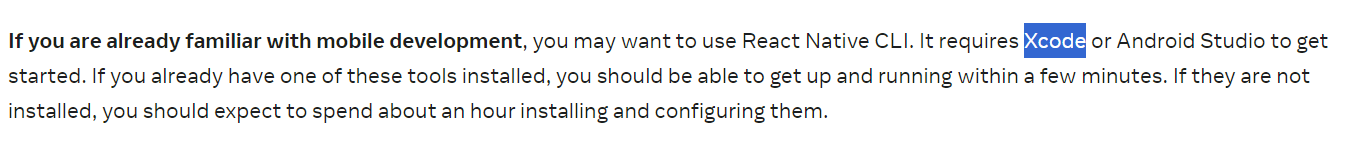
* Seguir los pasos de configuración según mi sistema operativo y plataforma

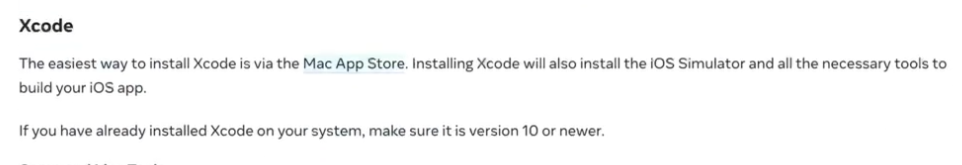


* Restricción solo puede crear APP IOS si tiene una Mac OS.
* Se debe generar los App con npx



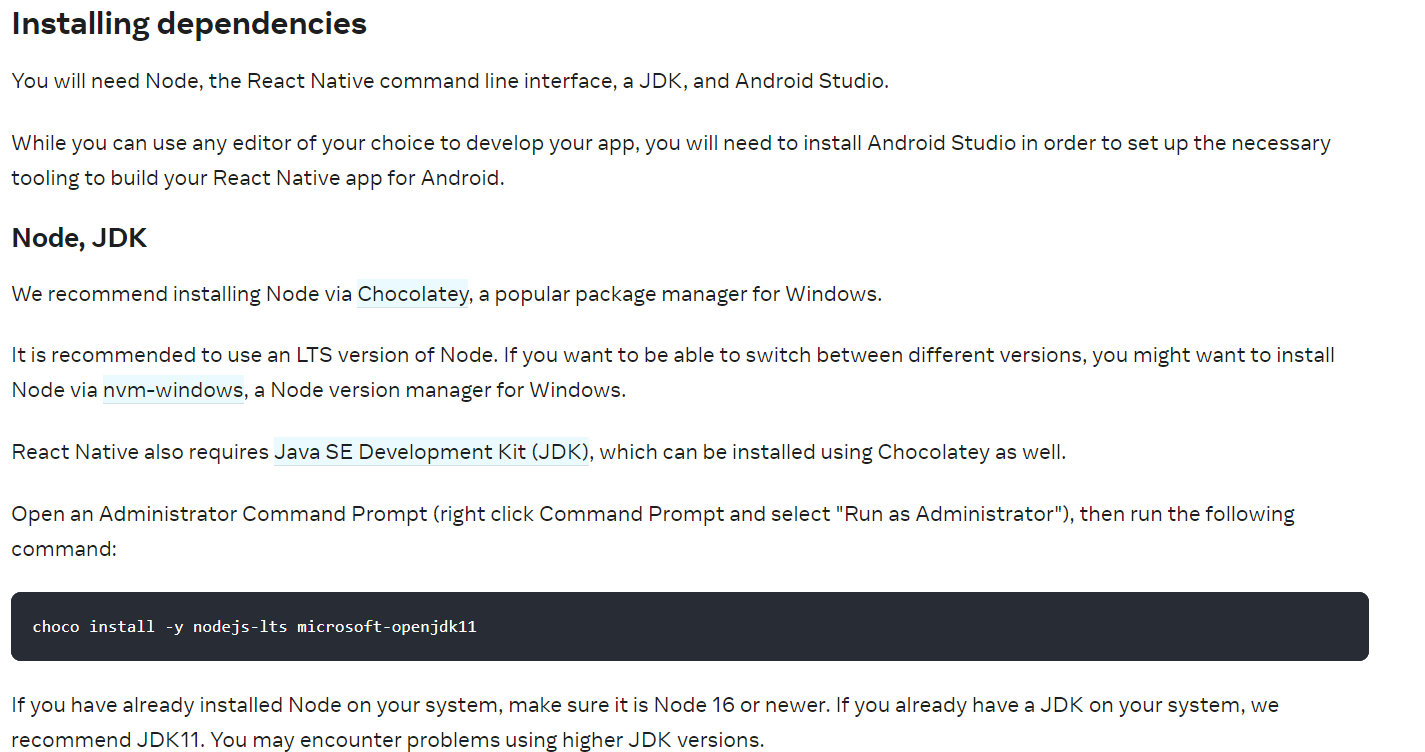
* Se debe instalar Android Studio (Android) y realizar ciertos ajustes de configuración para trabajar
* Se debe instalar Xcode (para IOS) y realizar y configurarlo de cierta manera





Esto sirve para usar los simuladores y se configuran la primera vez que vas a empezar a trabajar o programar pero luego ya quede activo en tu equipo

* Instalar otras dependencias y ya quedara configurado y listo para trabajar con Expo flujo básico

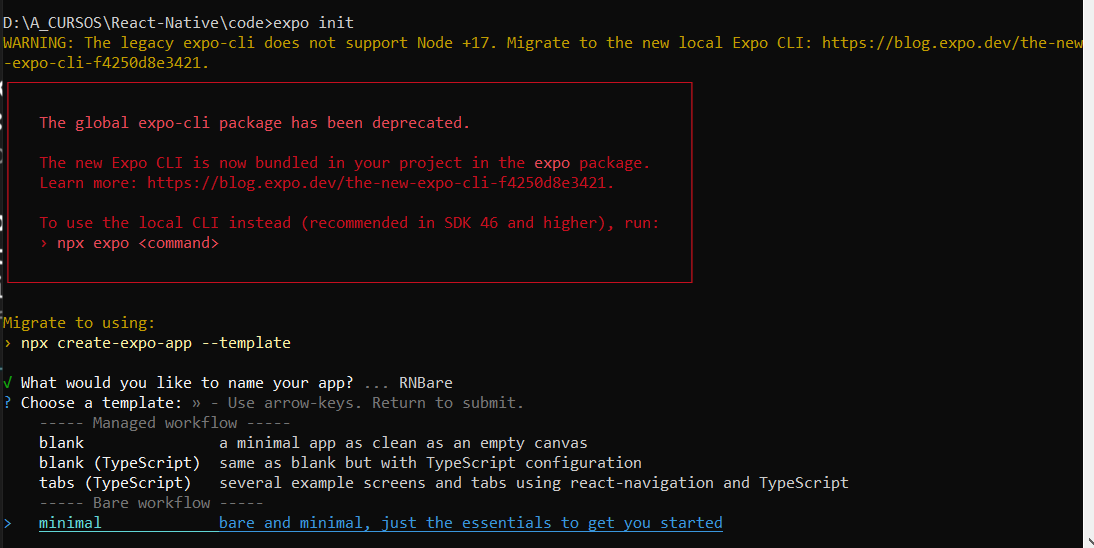


**Using Expo's Bare Workflow**

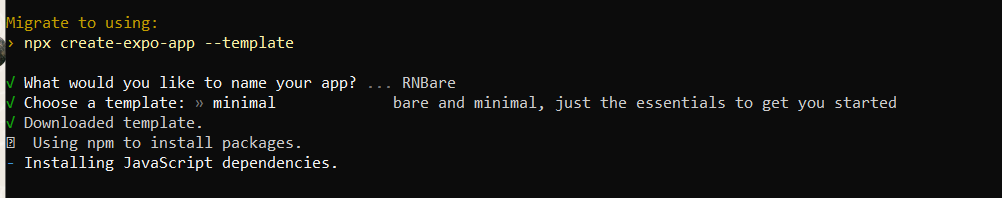
* Creamos un proyecto de React Native con Expo

$> Expo init

* Nombre del proyecto y seleccionar la opción de minimal de Template



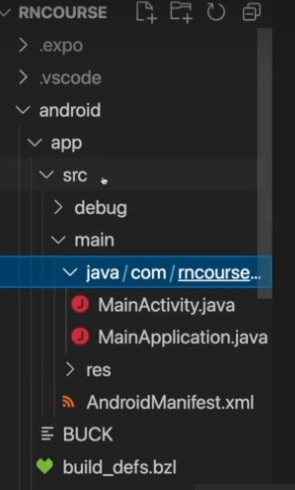
* Se realiza la instalación de dependencias mínimas requeridas por el proyecto



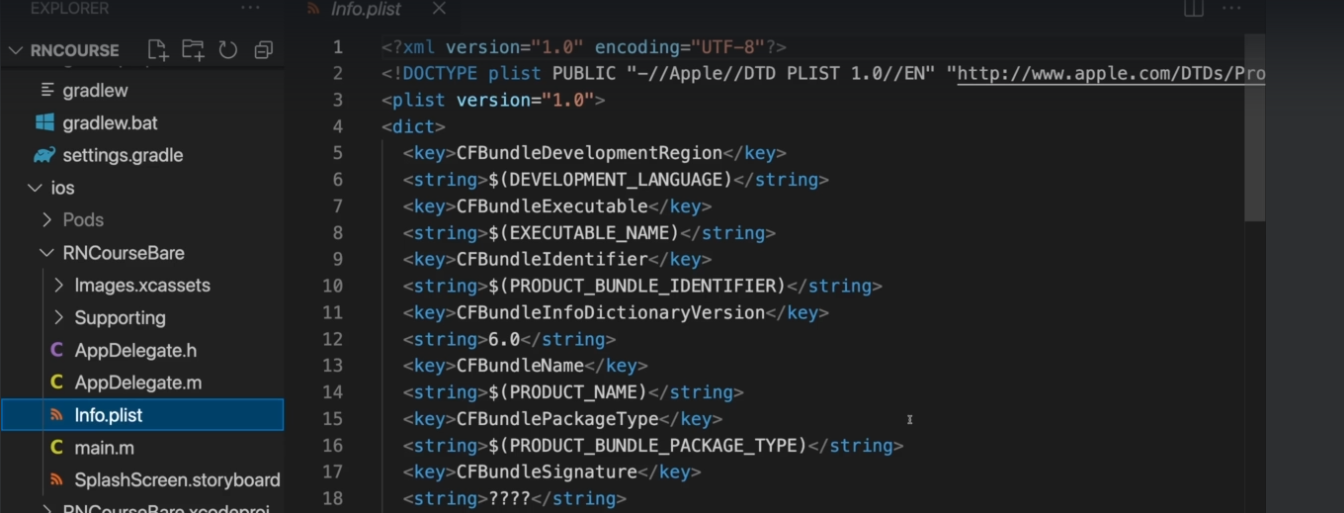
* Luego abrir el proyecto en VSCODE



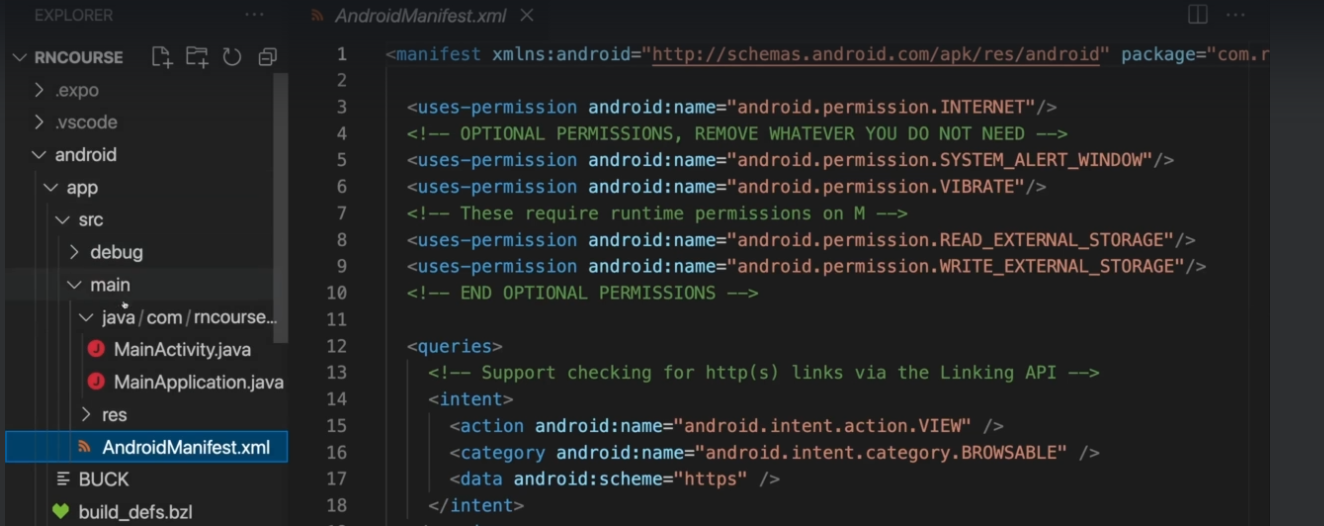
* A diferencia d ellos proyectos anteriores, se observan dos carpetas nuevas Android y IOS donde están códigos nativos y archivos de configuración de esas plataformas



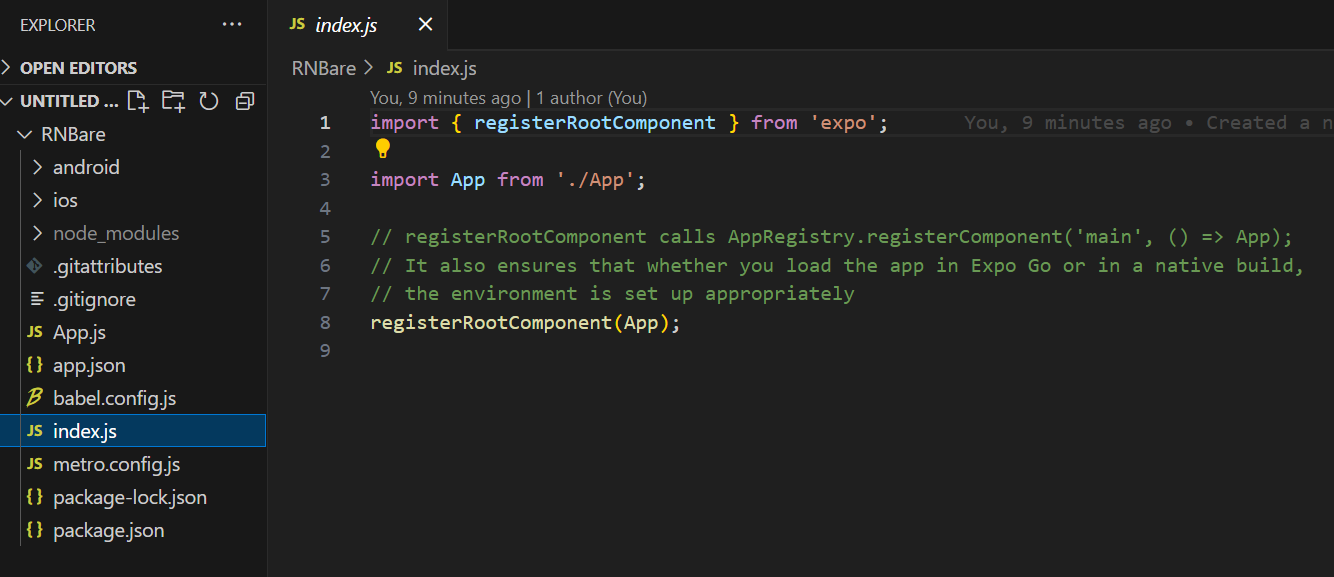
* Por ejemplo, para IOS se encuentra el archivo Info.plist con la información para configurar varias cosas de IOS



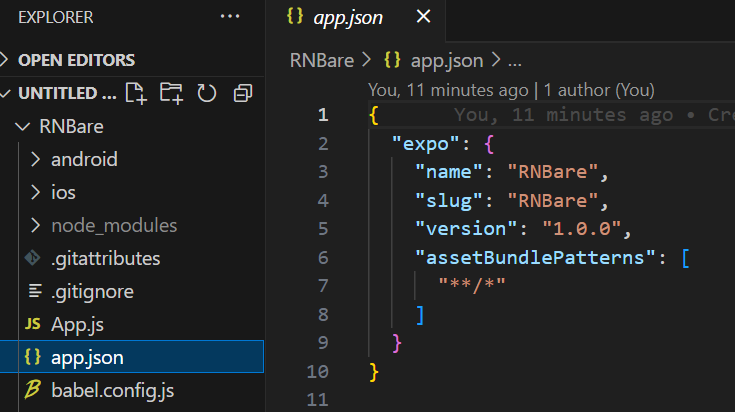
* En Android tenemos el archivo AndroidManifest.xml para configurar varias cosas, carpetas y permisos en Android



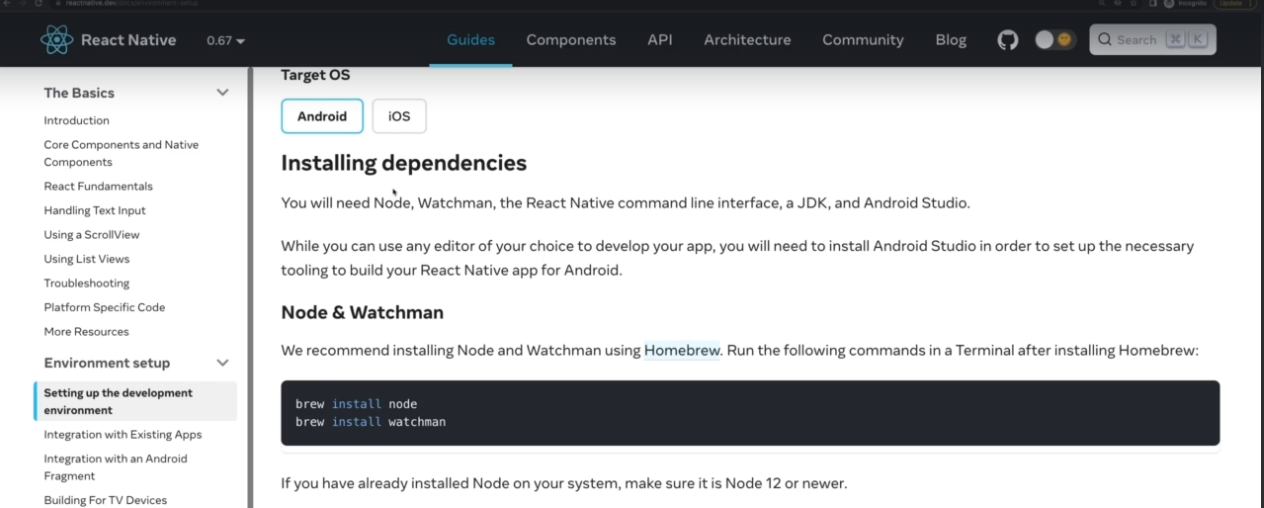
* También tenemos un index.js que es responsable de decirle a nuestro RN cual es nuestro componente raíz



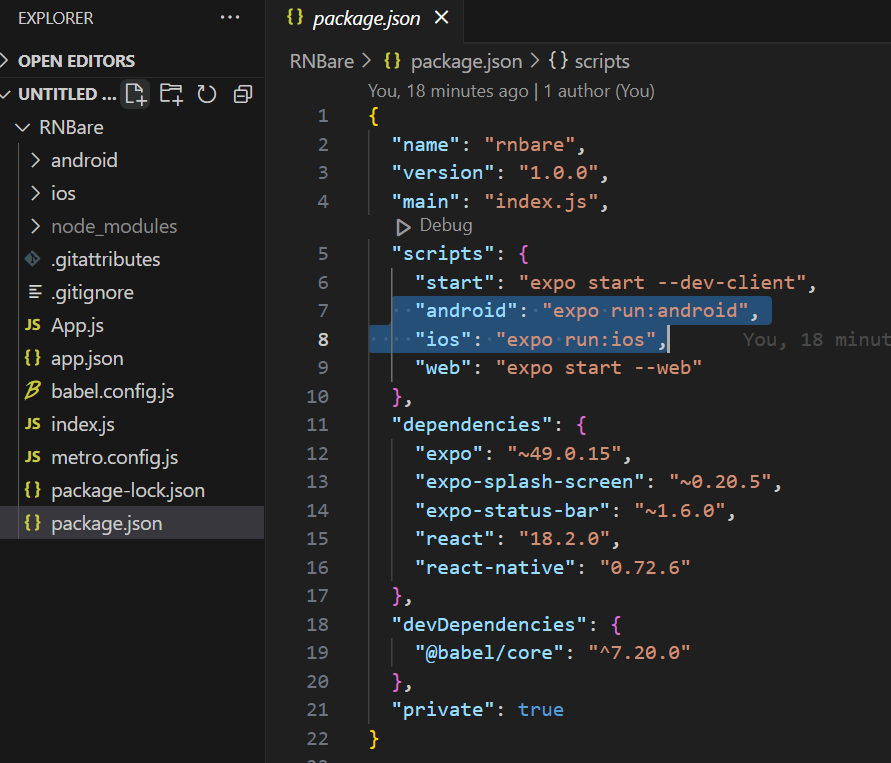
* El app.json es mucho mas reducido, menos configurado que en los proyectos Expo no básicos. En esos proyectos se configura en este archivo para ambas plataformas, ahora debemos administrar y configurar por separado para cada plataforma.



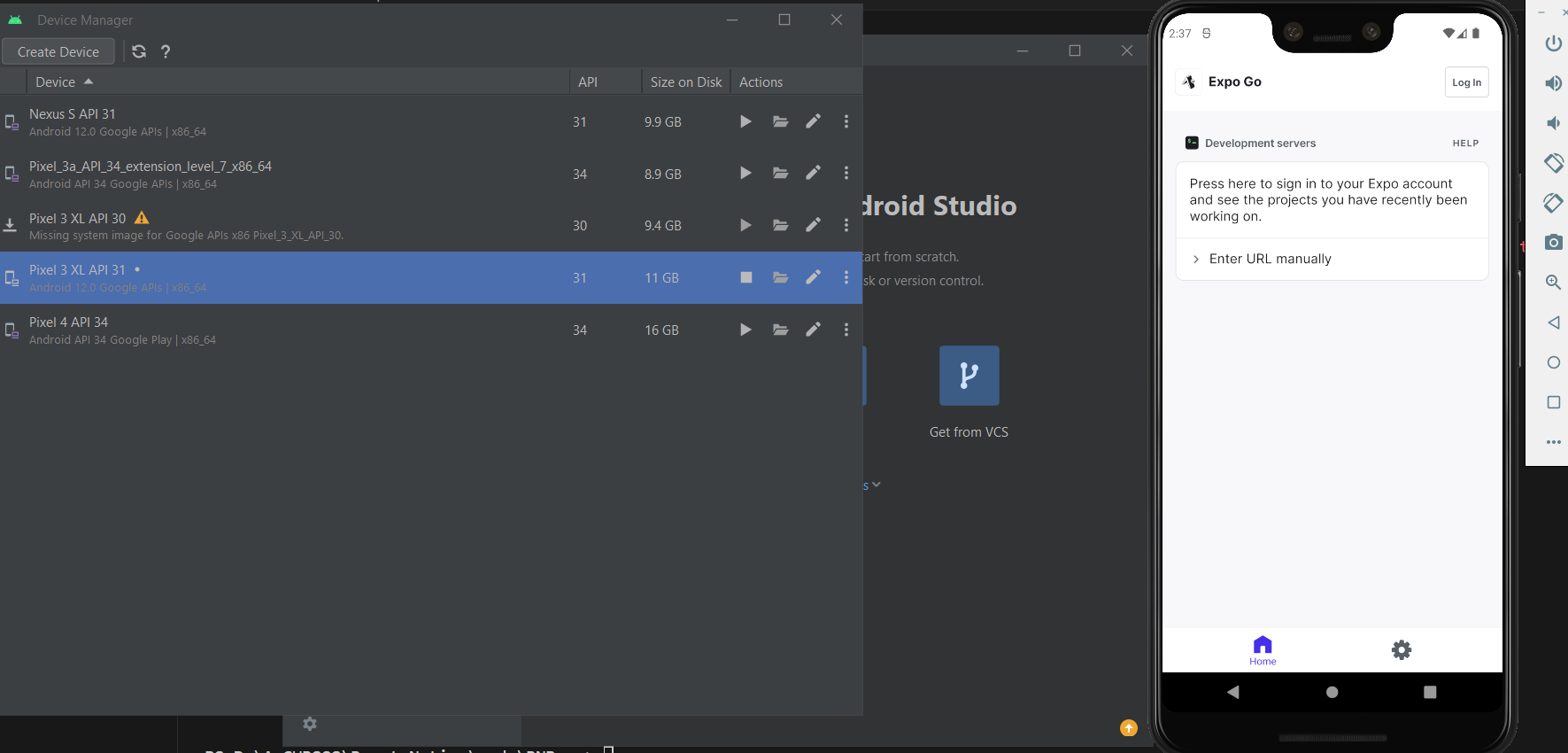
* Si levantamos el App como esta no podemos verlo en nuestro simulador (Expo Go - Android Studio, por ejemplo). Probablemente debemos crear la aplicación local y luego instalar en los simuladores (Android Studio o en mi Dispositivo real) en su dispositivo local. Por ello debemos revisar y configurar nuestros dispositivos



* Una vez configurado las plataformas, crear, levantar y probar sus aplicaciones de manera local es fácil.
* En el package.json están las tareas o script para levantar y probar la APP

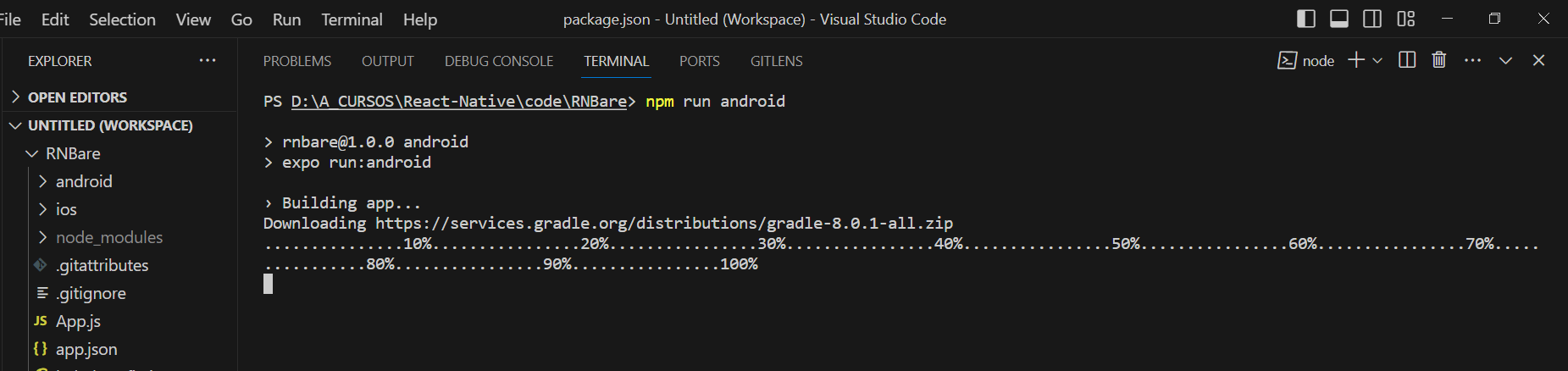


* Vamos a probar en Android.
  + Levantamos un simulador con Android Studio



* + Abrimos una terminal y llamamos la tarea o script para android

$> npm run android 🡺 expo run:android

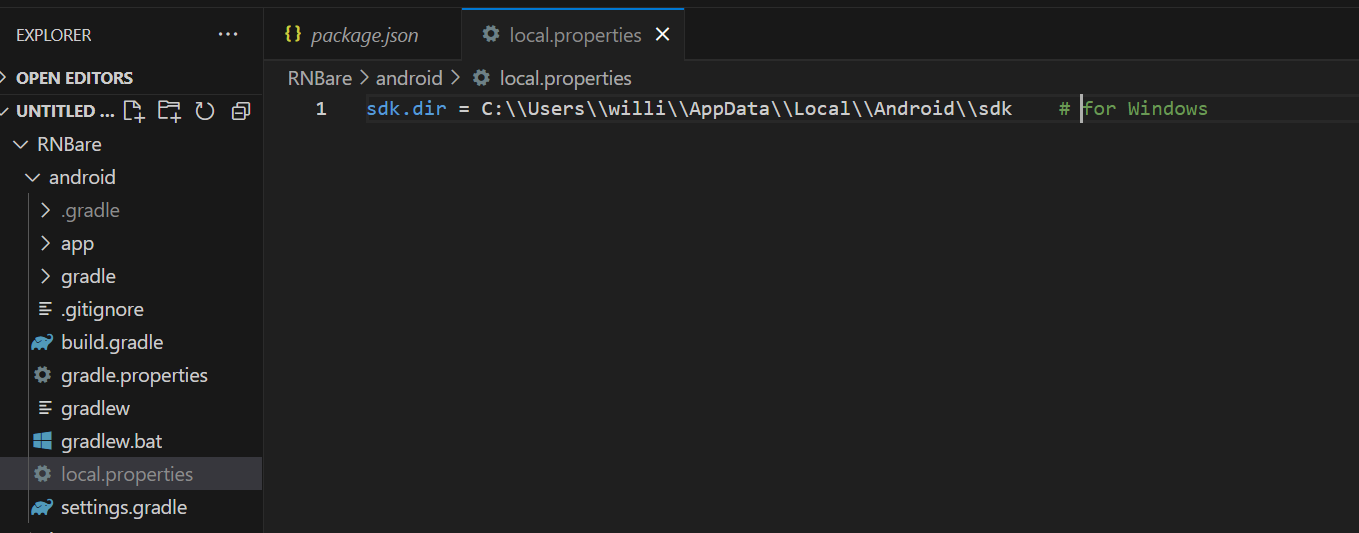


Esto creara la APP de manera local en Simulador de Android Studio

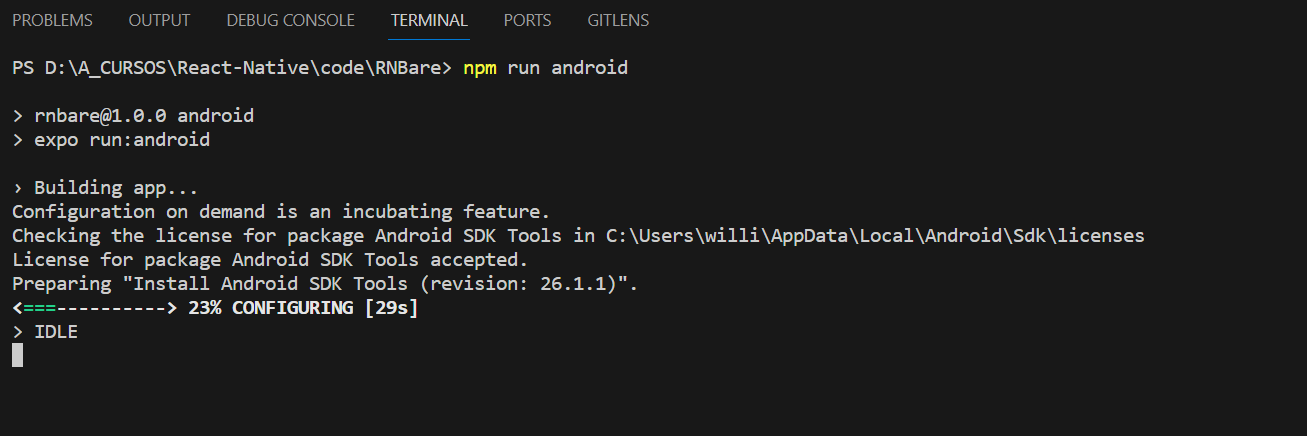
* Error al cargar la APP



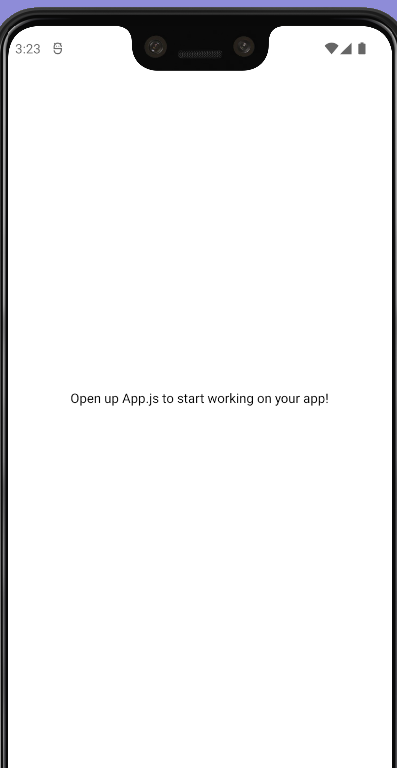
* Creo el archivo local.properties y configuro donde esta mi SDK para Android (Requerido para trabajar con Java)



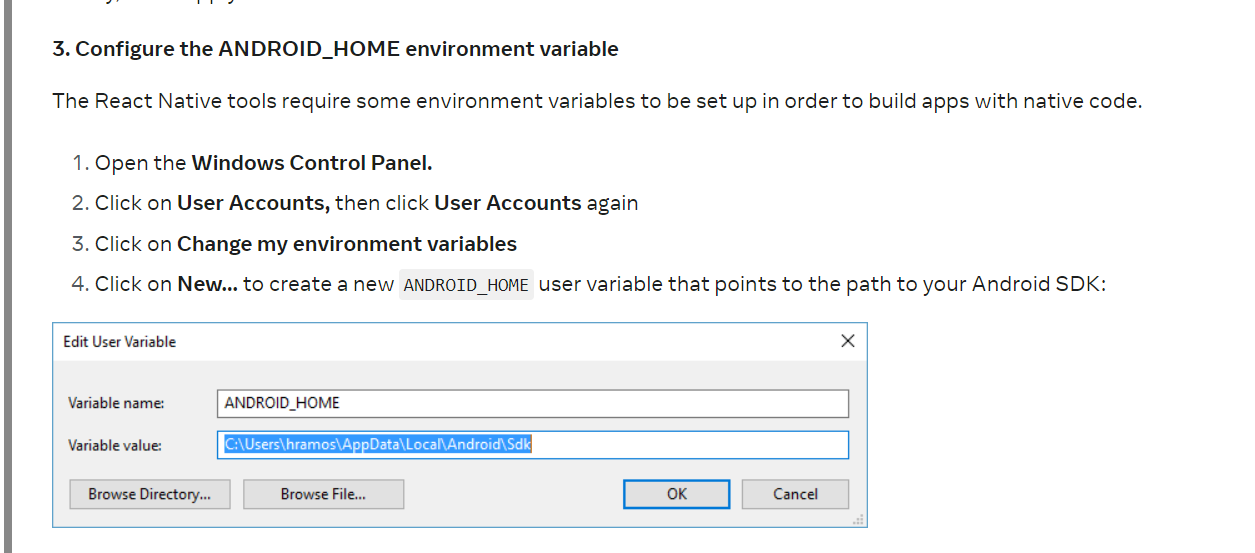
* Subo nuevamente el App



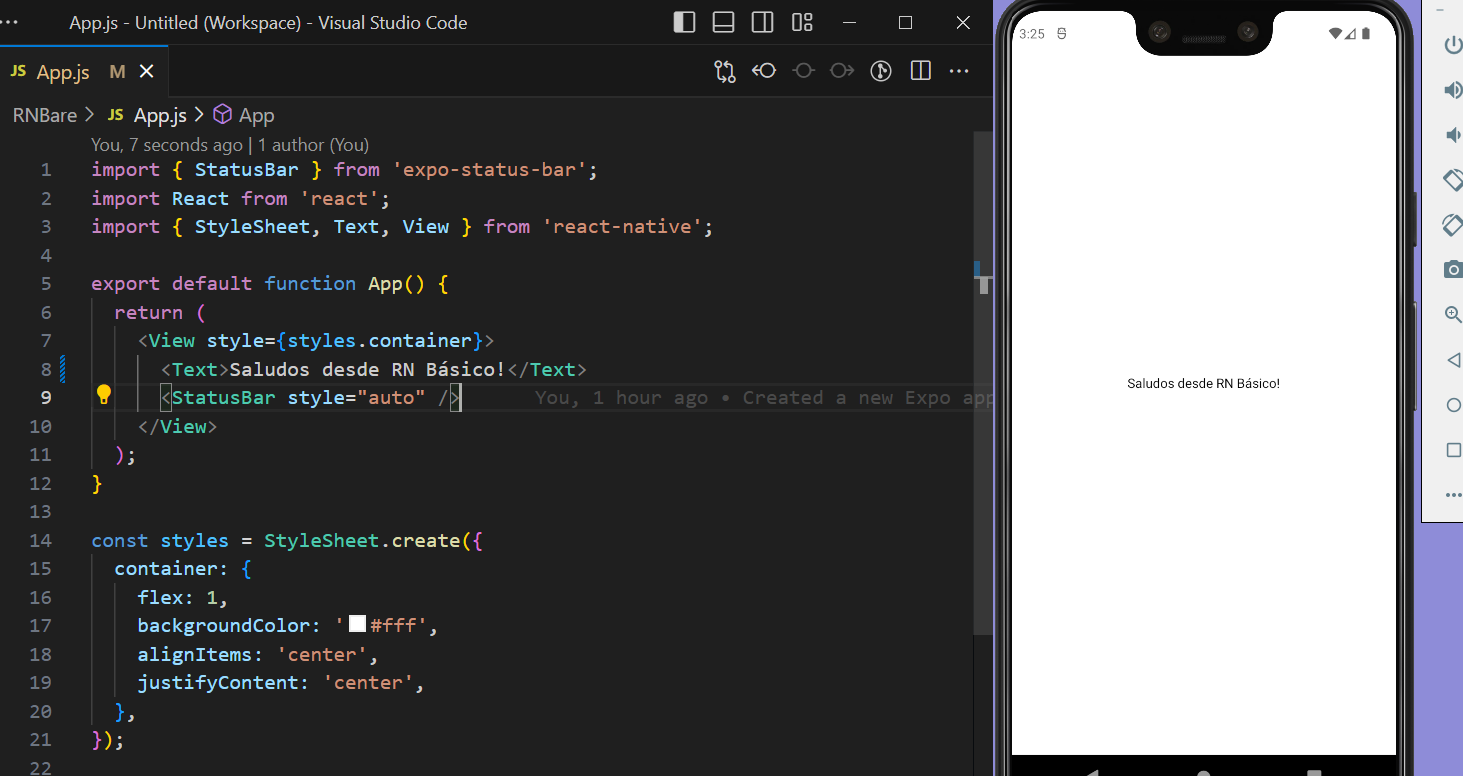




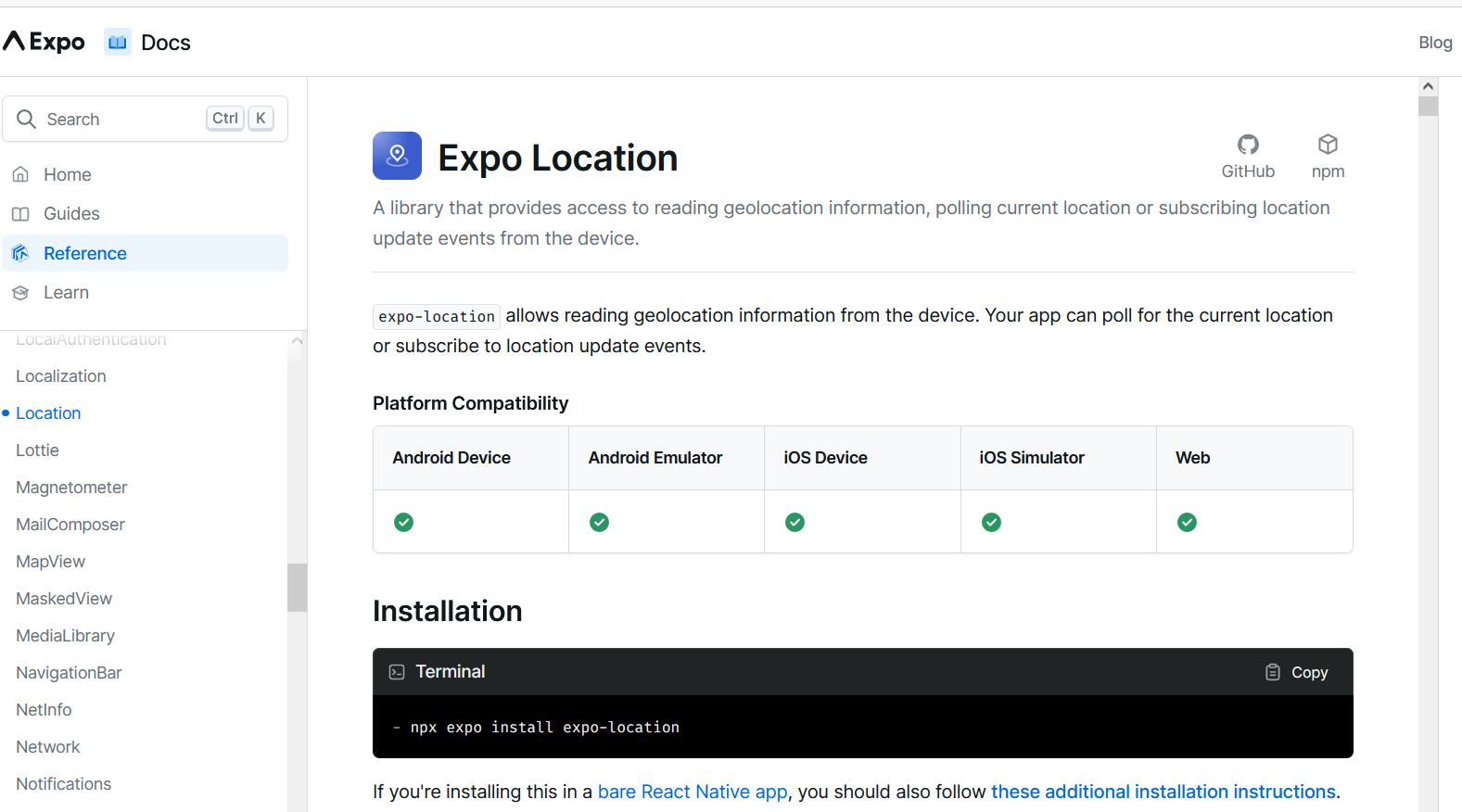
Nota para evitar este error se puede configurar una variable de ambiente para Windows

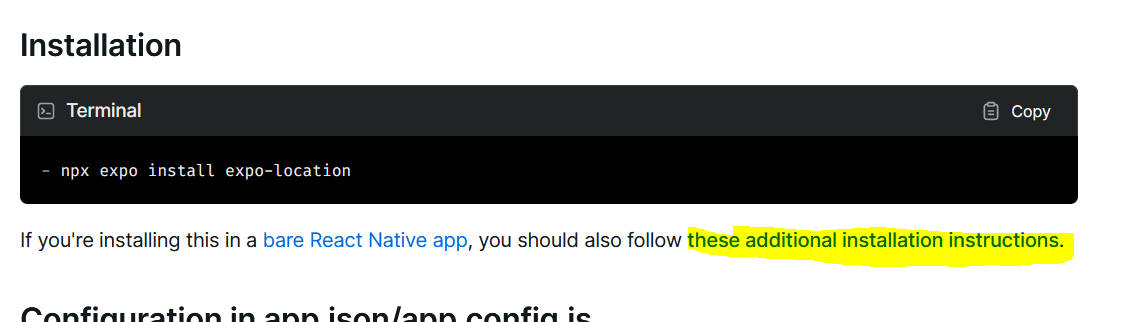


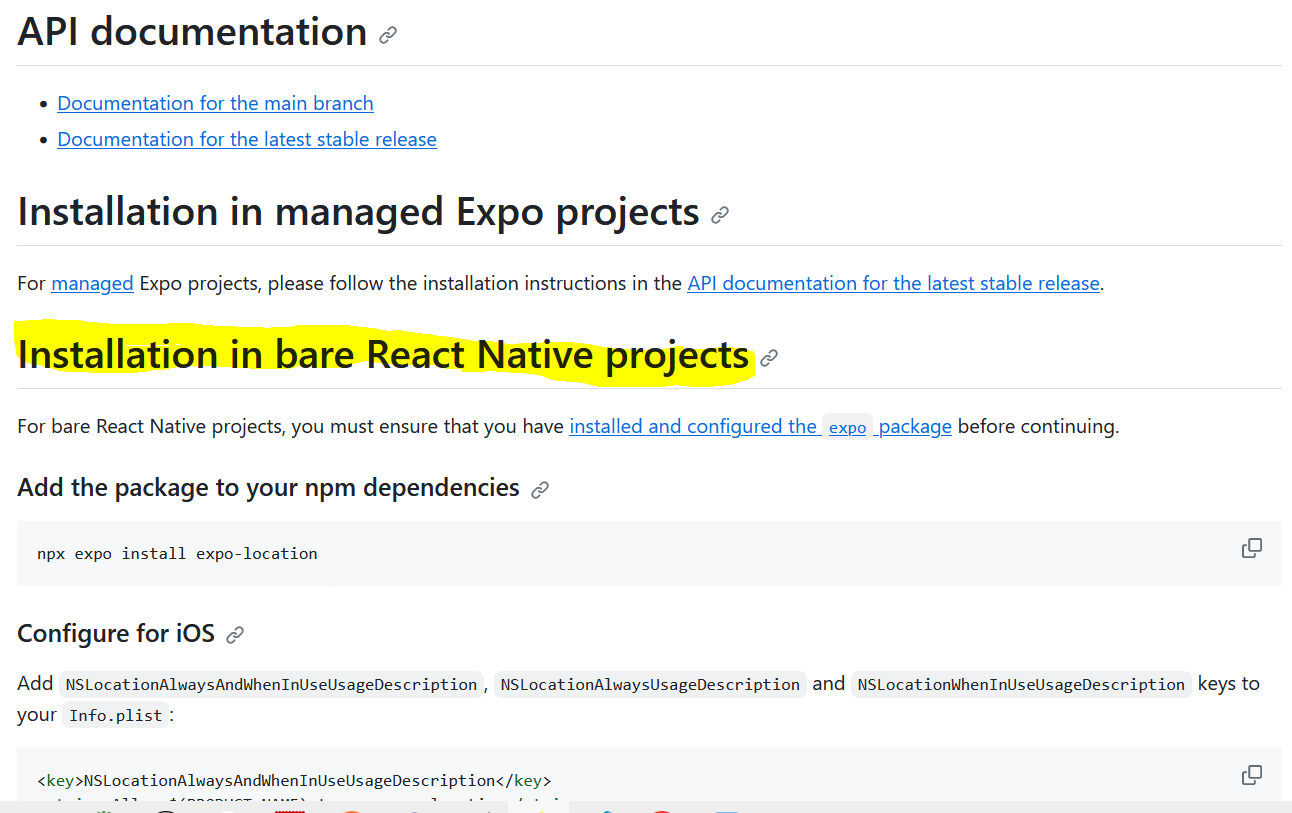
* Para probar cambiar el mensaje del APP y ver el cambio en el simulador



* Vamos a utilizar un paquete de terceros para ver cómo se configura

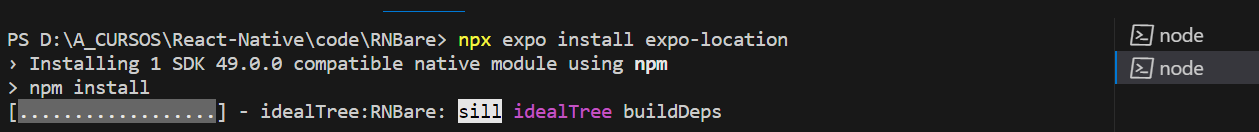






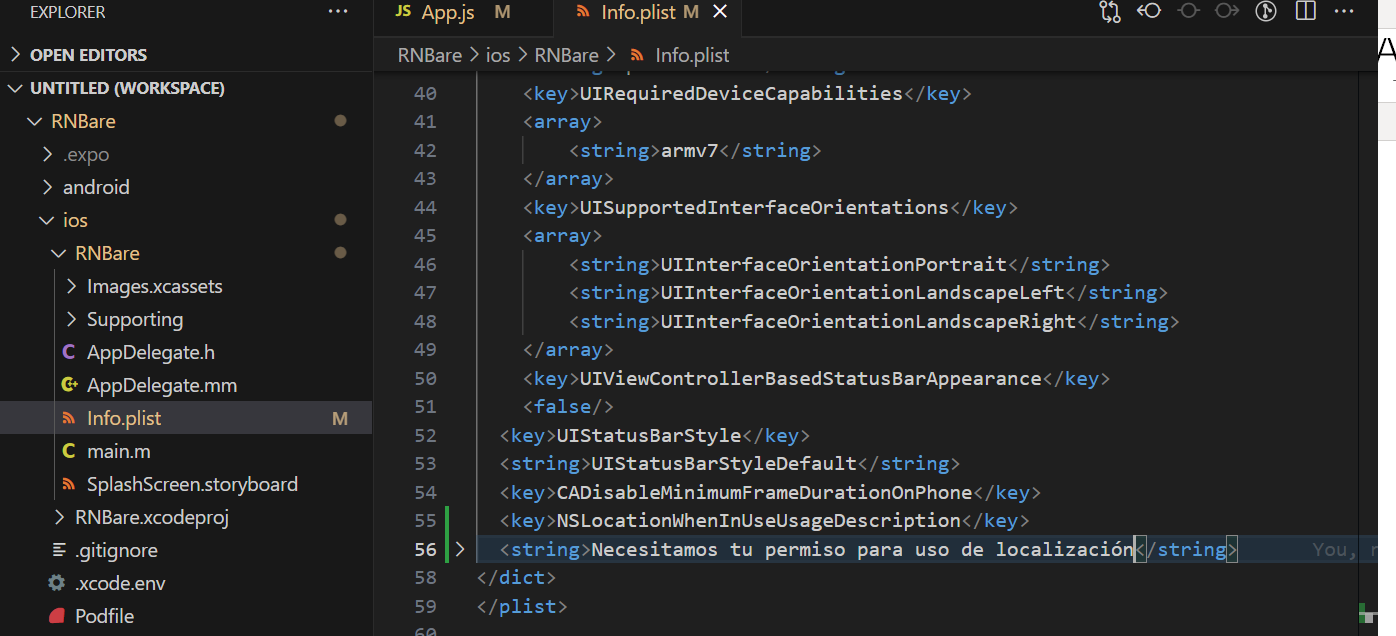
* Debemos instalar el paquete y configurar

$> npx expo install expo-location

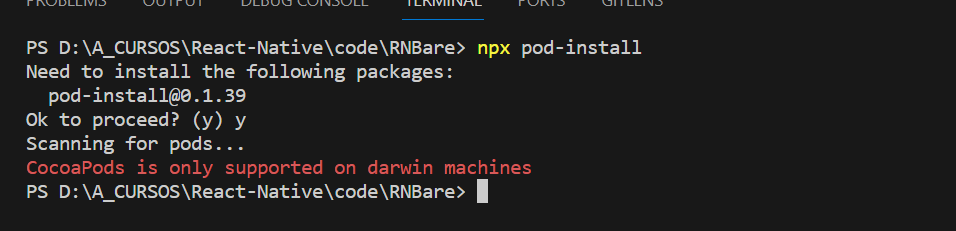


* Configurar permisos para dar permisos al dispositivo
* Para IOS

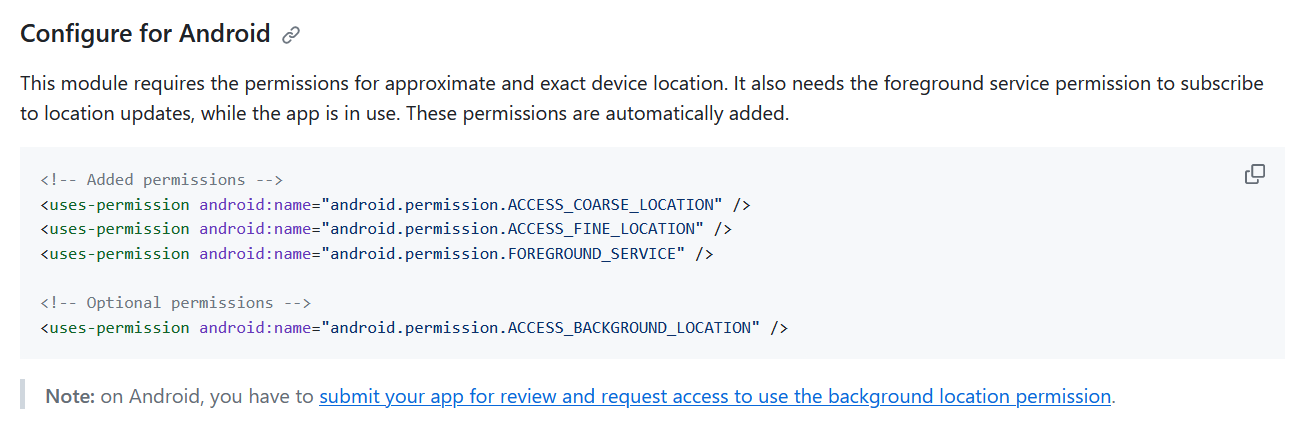




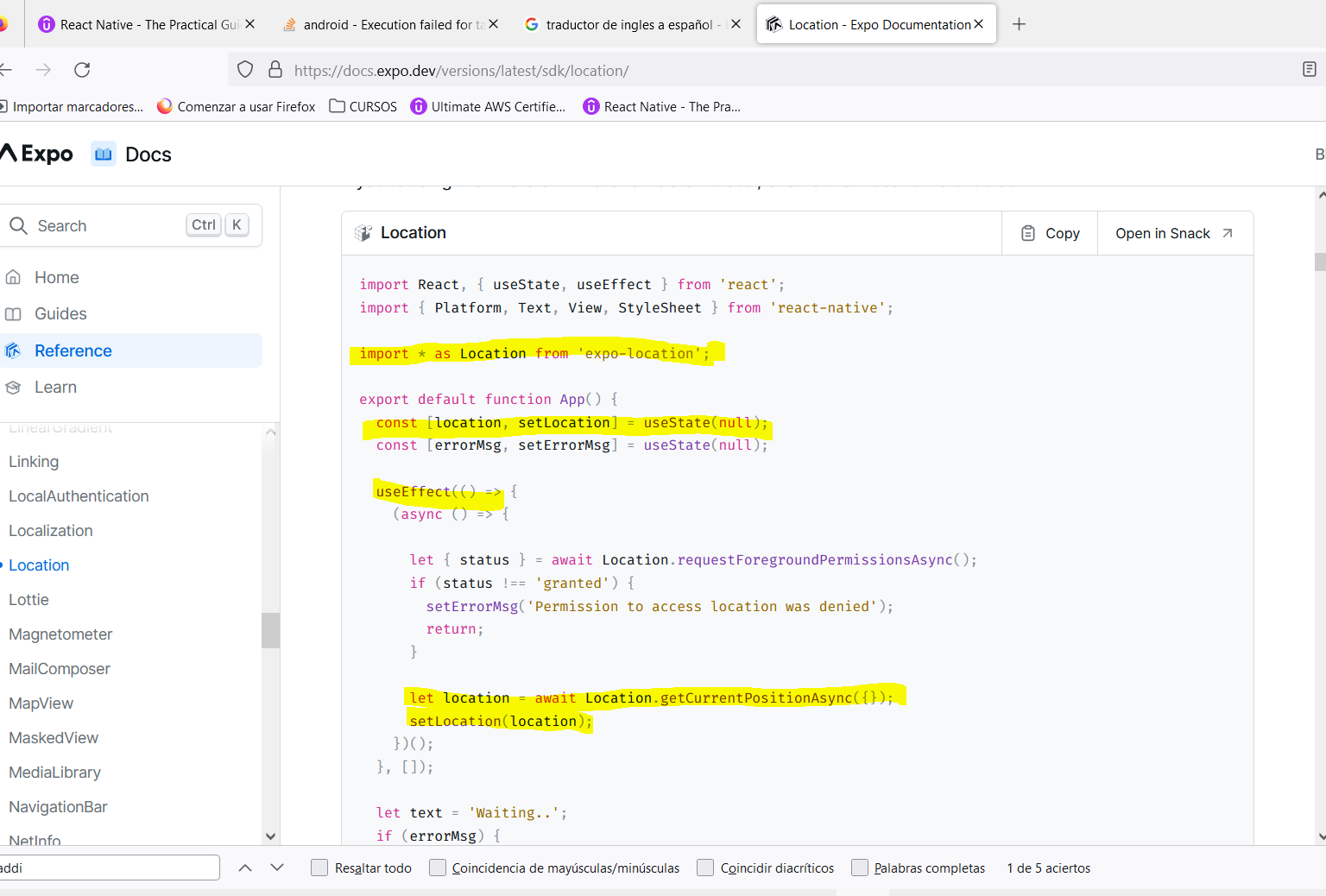
* Se debe correr el comando: $> **npx pod-install**, después de instalar el paquete



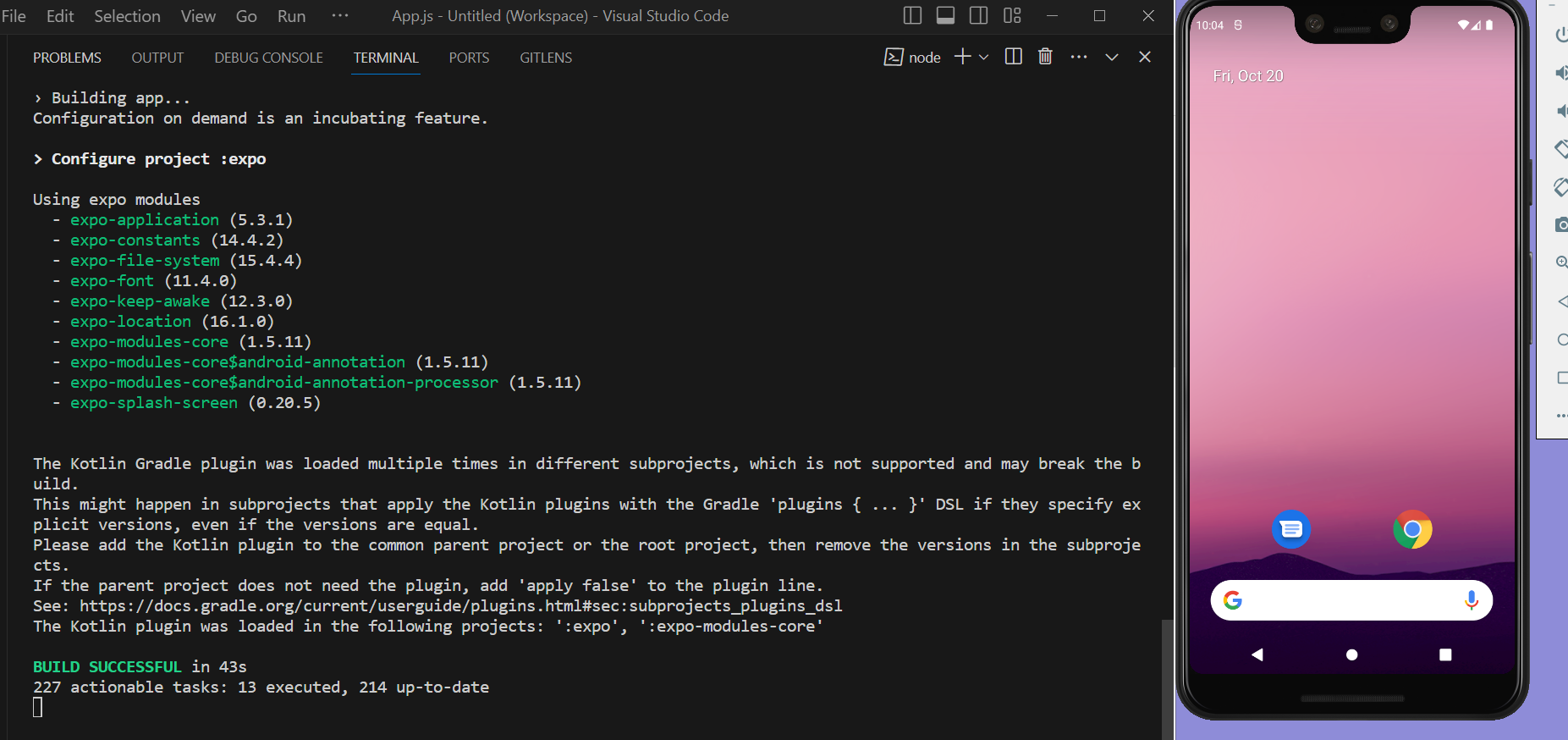
* Para Android

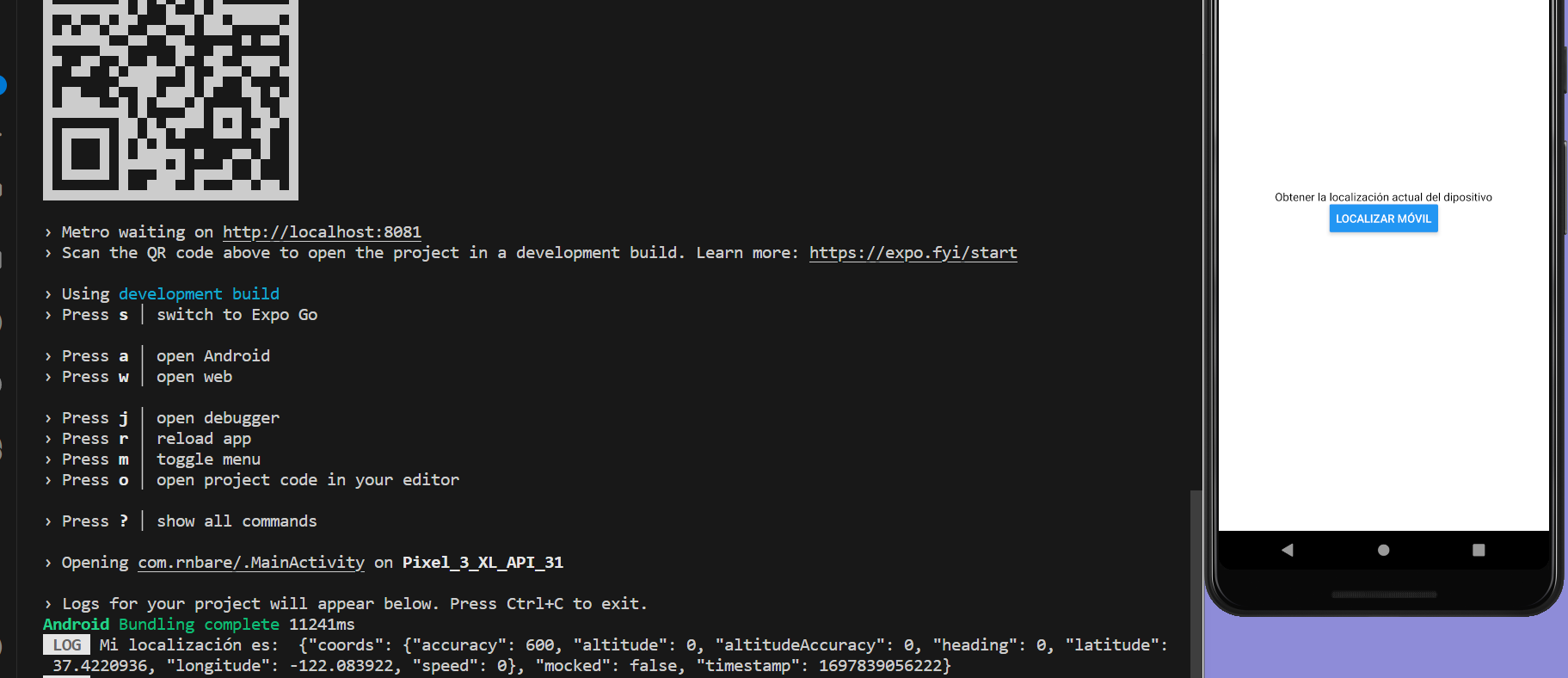
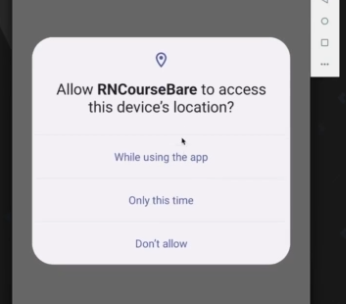


* Vamos a modificar el App.js para consultar la localización del dispositivo, solicitando permiso al usuario
* Agregar código de consulta de localización del dispositivo. Ver documentación



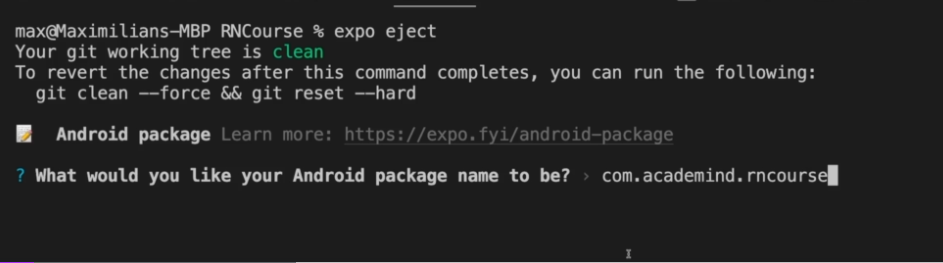
* Levantar el App y probar. Si realizo cambios de configuración de debe bajarse y subir nuevamente para que tome en cuenta los cambios en la configuración del dispositivo.
* Probar el dispositivo debe pedir los permisos y realizar la consulta. Recuerde que el dispositivo Android Studio debe estar activo antes de levantar el APP.



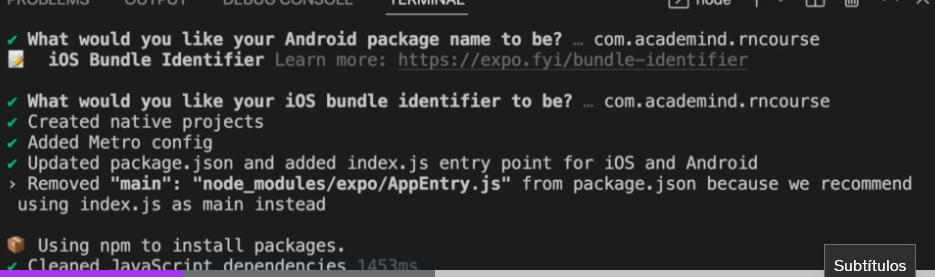
 

**Ejecting To The Bare Workflow**

* Si tengo una App con Expo Managed Workflow y la quiero llevar a una Expo Bare Workflow. No es complejo se debe expulsar la versión Expo
* Se debe utilizar el comando $> expo eject, esta consulta si realmente queremos continuar, si esta utilizando git puede guardar una confirmación antes de hacerlo para volver atrás por si lo requiere. O puede continuar y darle un nombre para su paquete de Android (Nombre único)

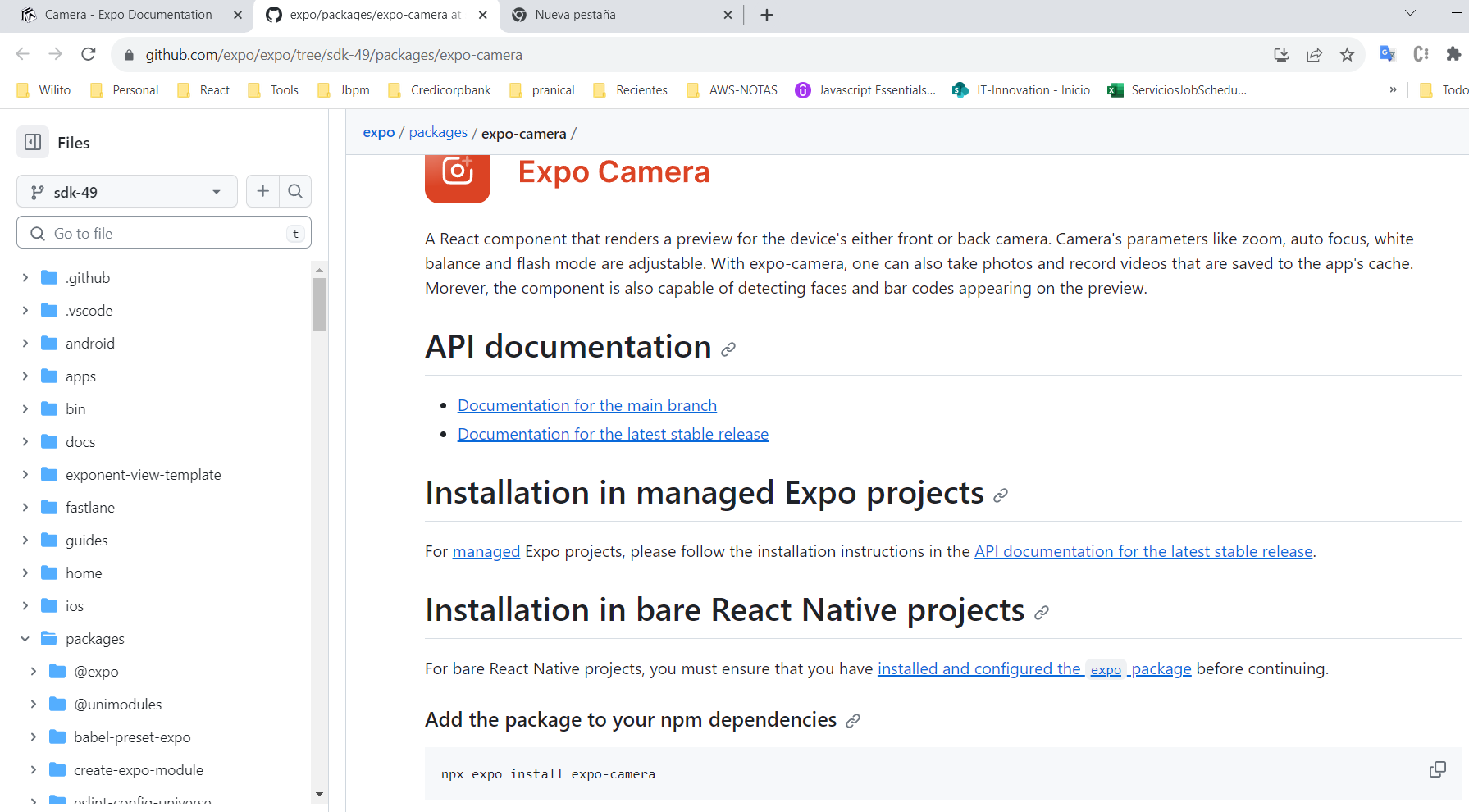


* Igual para IOS, se genera el proyecto



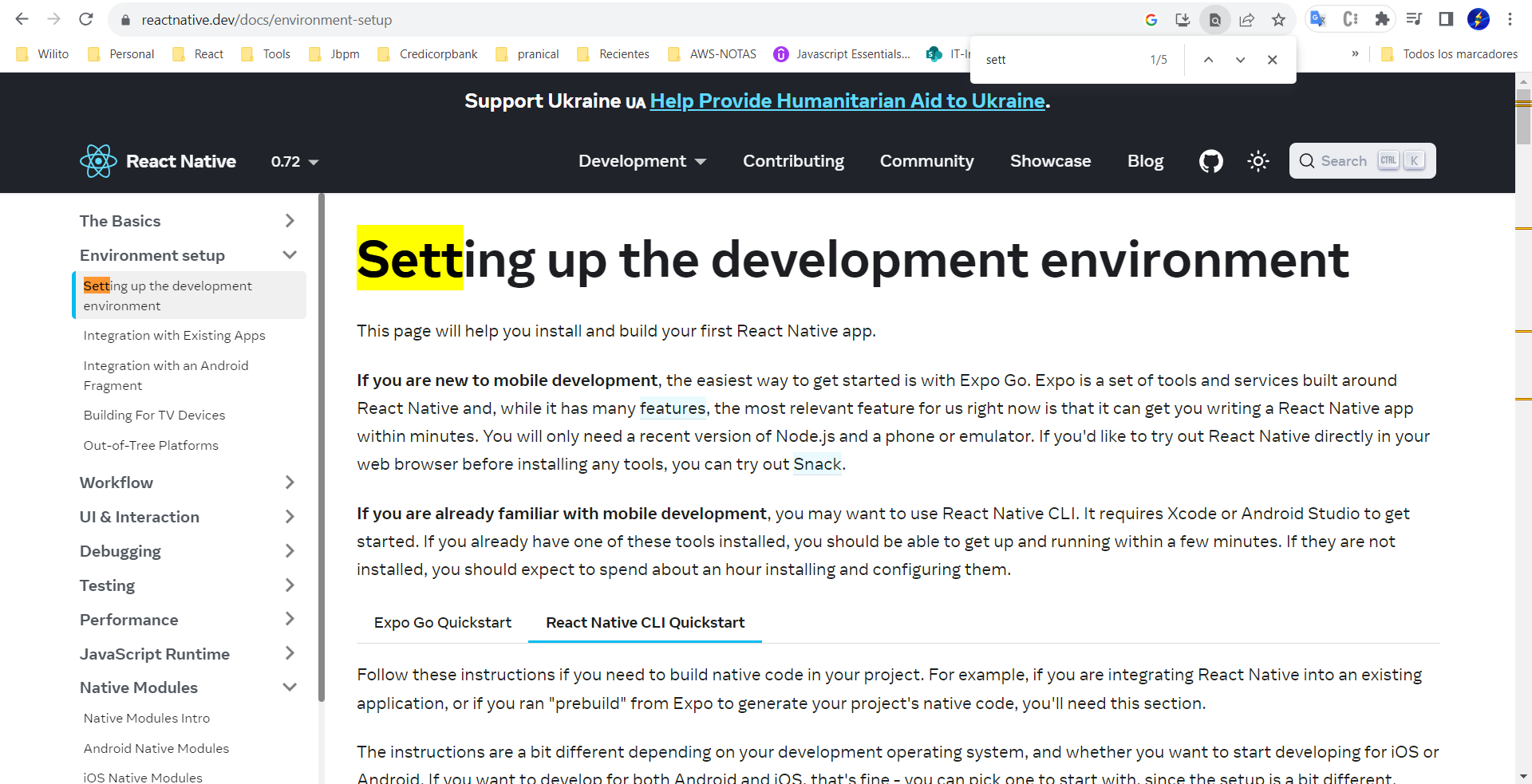
* Al terminar el proyecto queda exactamente igual como si lo hubiésemos creado con Expo básico
* Ahora podemos probarla con $> npm run android
* Ahora ya no corre en Expo Go pero si en el simulador de Android Studio.
* Para configurar componentes en Expo básico



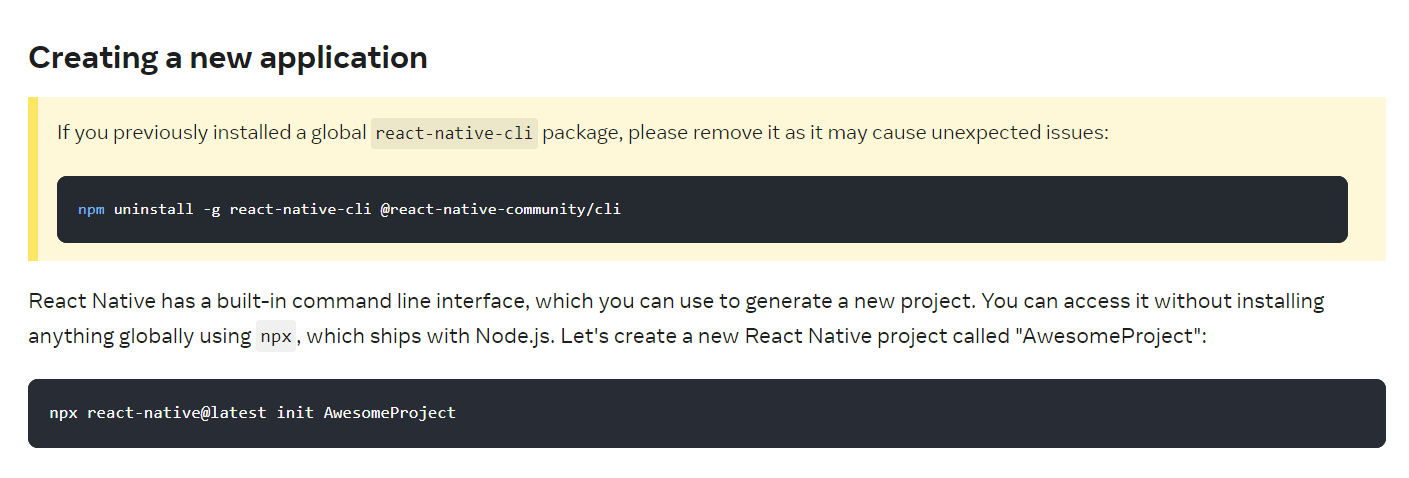


**Creating Projects with the React Native CLI (no Expo)**

* Vamos a crea una APP en React Native sin Expo

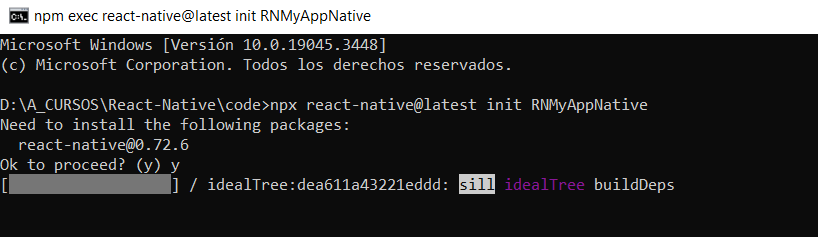


* Crear nueva App



* Comando para crear RN CLI

npx react-native@latest init RNMyAppNative



* Podría instalar globalmente react-native-cli

$> npm install -g react-native-cli

$> react-native init ProjectName