

# Instalación y configuración del entorno de desarrollo

## Contenido

1. Diferencias entre Expo Cli y React Native CLI .....	1
2. Configurar entorno de desarrollo en Windows.....	2
2.1 Gestor de paquetes Chocolatey .....	2
2.3 Node JS y Open JDK .....	3
2.4 Android Studio .....	3
2.5 Crear un emulador en Windows.....	5
2.6 Crear proyecto de React Native en Windows .....	6
3. Configuración del entorno de desarrollo en Mac .....	7

## 1. Diferencias entre Expo Cli y React Native CLI

Ambas son aplicaciones en línea de comandos (CLI - Command Line Interface) para crear aplicaciones de React Native.

React Native	
Expo CLI	React Native CLI
<ul style="list-style-type: none"><li>• Expo se encarga de controlar los builds para IOS y Android.</li><li>• Provee muchas funcionalidades nativas incluidas.</li><li>• Es excelente para alguien que empieza el desarrollo móvil.</li><li>• Permite correr y probar la aplicación en IOS aunque no dispongas de una Mac.</li></ul>	<ul style="list-style-type: none"><li>• Genera un proyecto con lo mínimo necesario para correr.</li><li>• Genera los proyectos nativos para IOS y Android de forma independiente.</li><li>• Te permite escribir código nativo por plataforma (en caso de ser necesario).</li></ul>

31. Diferencia entre EXPO CLI y ReactNative CLI

React Native Cons

Expo CLI	React Native CLI
No puedes extender una funcionalidad nativa no soportada por expo ( a menos que uses el EJECT )	Necesitas los ambientes de desarrollo por separado. Android Studio ( las herramientas al menos ) y Xcode.

31. Diferencia entre EXPO CLI y ReactNative CLI

React Native Pros comunes en ambos

Expo CLI	React Native CLI
<ul style="list-style-type: none"><li>• La mayoría de los paquetes soportan ambos ambientes con instalaciones independientes.</li><li>• Cambios en caliente ( hot reload ).</li><li>• Desarrollar y ver cambios en desarrollo en dispositivos físicos.</li><li>• Utilizar el conocimiento que tienes de React en React Native.</li><li>• Desplegar en las AppStores.</li></ul>	

Vamos a trabajar con **React Native CLI** para entender en profundidad React Native y trabajar con proyectos Xcode y Android de forma nativa.

## 2. Configurar entorno de desarrollo en Windows

Seguir documentación seleccionando **React Native CLI**, SO de desarrollo (macOS, Windows, Linux) y SO destino (Android o iOS)

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

### 2.1 Gestor de paquetes Chocolatey

Instalar el gestor de paquetes para Windows [Chocolatey](#) mediante las siguientes instrucciones:

Ejecuta desde la PowerShell en modo administrador:

```
Get-ExecutionPolicy
```

Si devuelve 'Restricted':

```
Set-ExecutionPolicy AllSigned or Set-ExecutionPolicy Bypass -Scope Process
```

Ejecutar commando que instalará Chocolatey mediante el script install.ps1:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex  
(New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org  
/install.ps1'))
```

Se puede comprobar que se ha instalado correctamente:

```
choco
```

## 2.3 Node JS y Open JDK

Volviendo a la documentación de React Native y mediante el gestor de paquetes Chocolatey:

```
choco install -y nodejs-lts openjdk11
```

## 2.4 Android Studio

Descargar e instalar Android Studio: <https://developer.android.com/studio>

Abrir Android Studio:

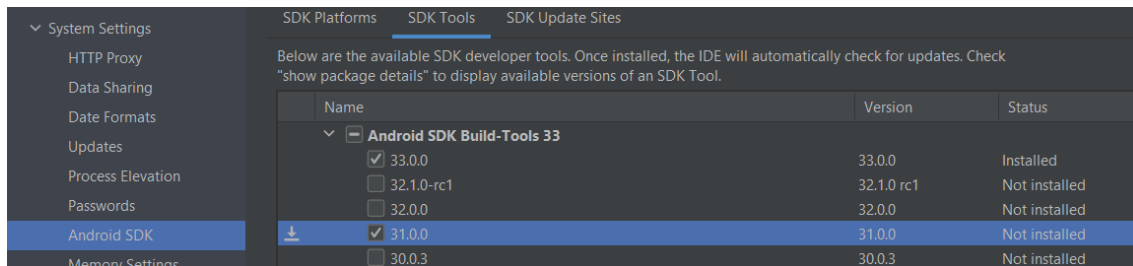
En Projects > More actions > SDK Manager

Instalar el SDK que recomiende la instalación. En este momento, el SDK Android 12 (S). Antes de instalar, hacer clic en “Show package details” para comprobar que la versión del Android SDK Platform es la requerida, en este caso la 31 y ya viene seleccionada.

Habilitar “Google APIs Intel x86 Atom\_64 System Image”:

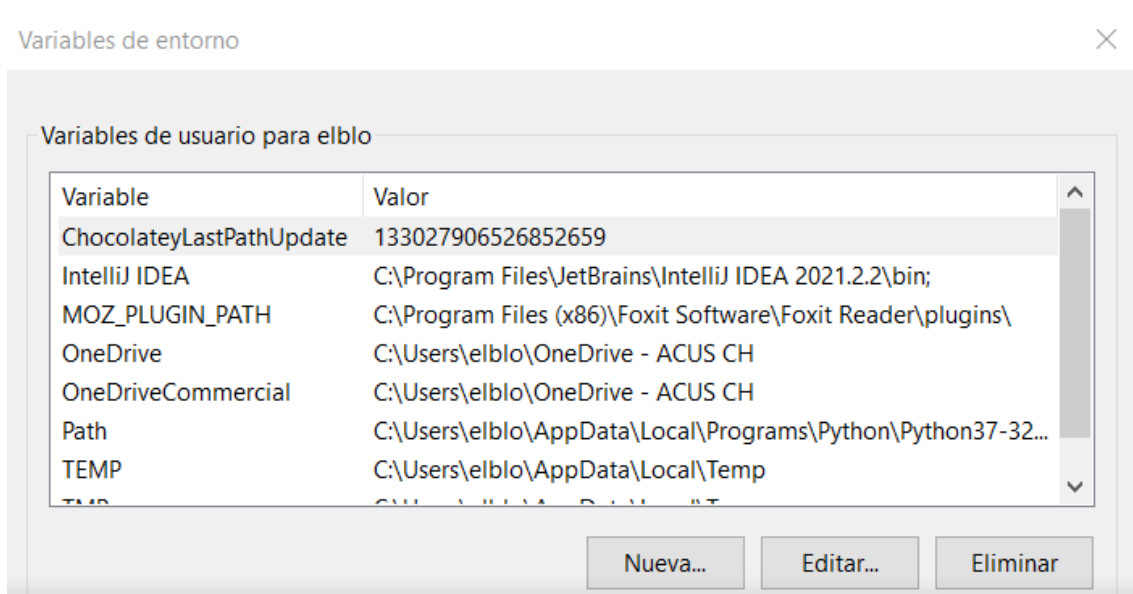
Android SDK	<input type="checkbox"/>	Google Play Intel x86 Atom_64 System Image	32	3	Not installed
Memory Settings	▼	Android 12.0 (S)			
Notifications	<input checked="" type="checkbox"/>	Android SDK Platform 31	31	1	Installed
Quick Lists	<input checked="" type="checkbox"/>	Sources for Android 31	31	1	Installed
Path Variables	<input type="checkbox"/>	Android TV ARM 64 v8a System Image	31	3	Not installed
Keymap	<input type="checkbox"/>	Android TV Intel x86 Atom System Image	31	3	Not installed
> Editor	<input type="checkbox"/>	ARM 64 v8a System Image	31	3	Not installed
> Build, Execution, Deployment	<input type="checkbox"/>	Intel x86 Atom_64 System Image	31	3	Not installed
> Languages & Frameworks	<input type="checkbox"/>	Google TV ARM 64 v8a System Image	31	3	Not installed
> Tools	<input type="checkbox"/>	Google TV Intel x86 Atom System Image	31	10	Not installed
	<input checked="" type="checkbox"/>	Google APIs Intel x86 Atom_64 System Image	31	10	Not installed

En la pestaña “SDK Tools” marcar la opción “Show Package Details” y seleccionar la que pidan, en este caso la 31.0.0:

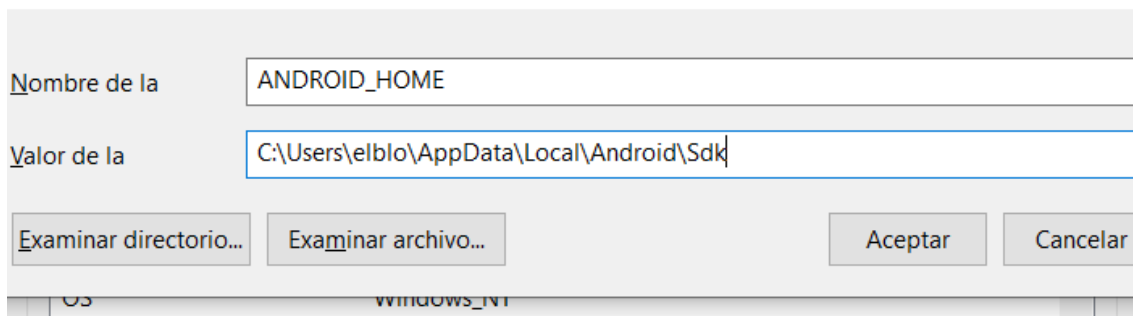


“Ok” y ya podemos salir del SDK Manager.

Ahora toca añadir la **variable de entorno ANDROID\_HOME**:



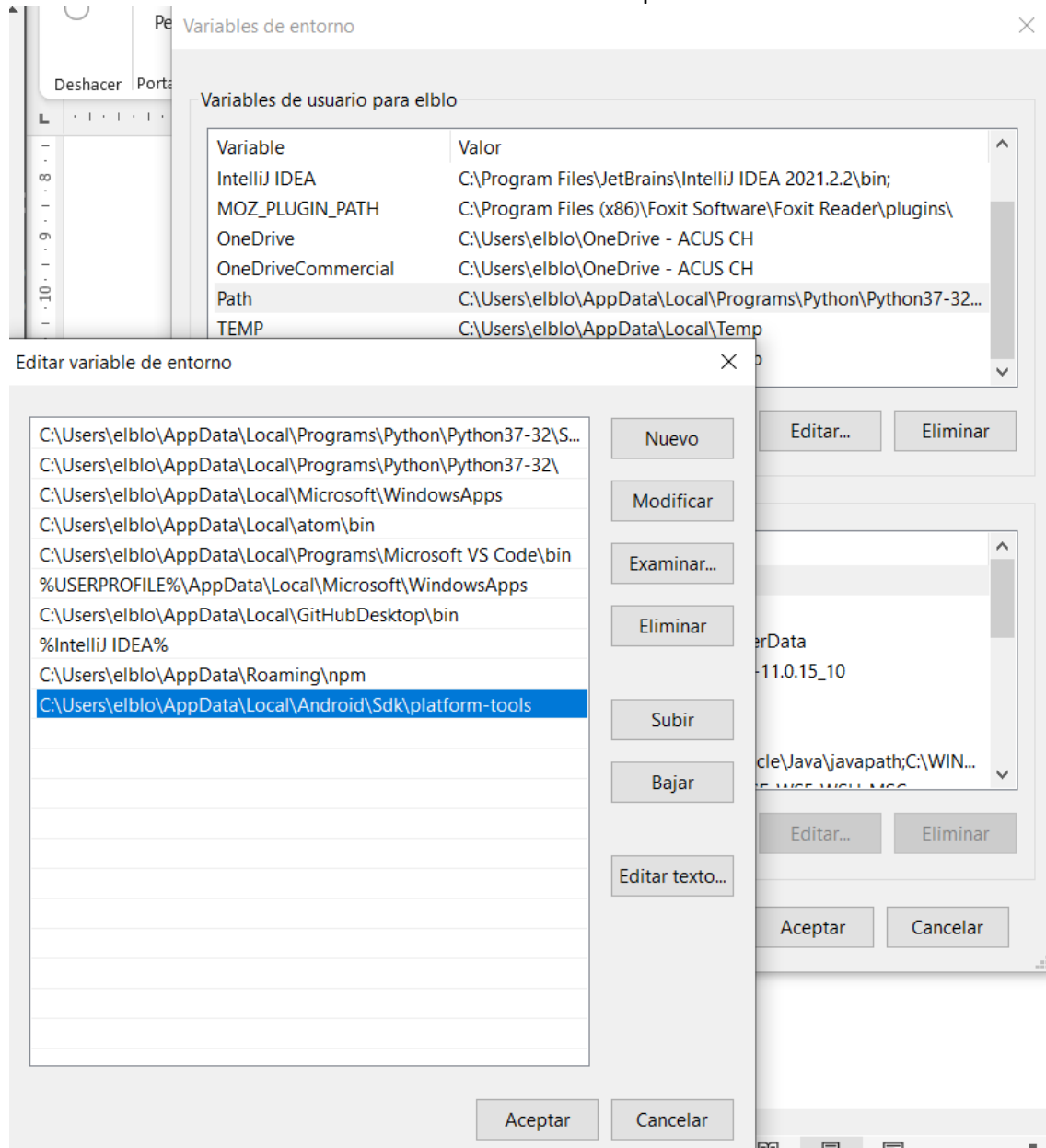
Nueva variable de usuario



Comprobar que está bien establecida (aparece en el listado devuelto) ejecutando en PowerShell:

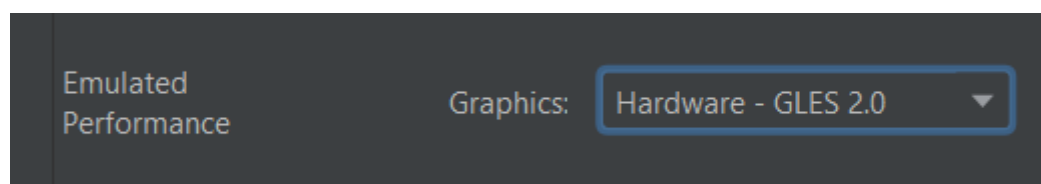
```
Get-ChildItem -Path Env:\
```

Añadir al **PATH** la ruta de antes con el subdirectorio 'platform-tools':

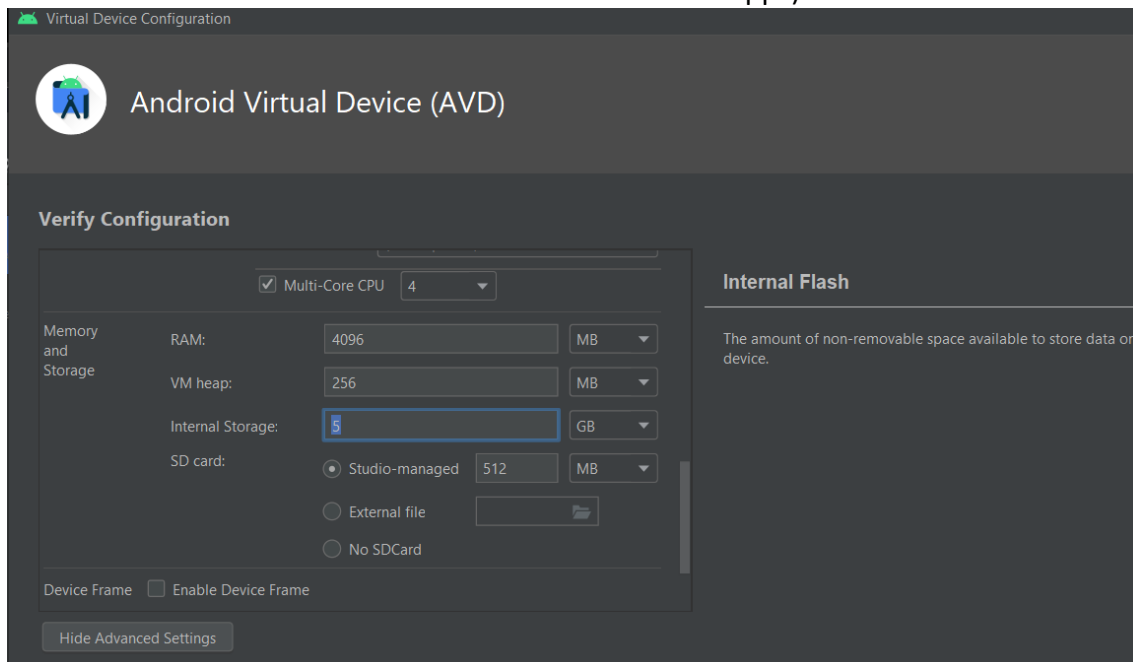


## 2.5 Crear un emulador en Windows

Desde la página inicial de Android Studio, abrir **Virtual Device Manager**, crear un dispositivo nuevo (Pixel 4 XL por ejemplo), seleccionar la versión de Android a utilizar (S, API 31 por ejemplo). Si tenemos tarjeta de vídeo independiente seleccionar la siguiente opción (y si no dejar en 'Automático'):



Y en opciones avanzadas seleccionar 4GB de RAM y 5GB de almacenamiento (para evitar fallo futuro de falta de almacenamiento al lanzar apps).



Y finalizar. Probar a lanzarlo desde el listado pulsando el botón de 'play'.

## 2.6 Crear proyecto de React Native en Windows

Si quisiéramos trabajar con JS, ejecutar:

```
npx react-native init NombreProyecto
```

Si quisiéramos trabajar con Typescript, ejecutar:

```
npx react-native init NombreProyecto --template react-native-template-typescript
```

Nota: Si aparece error de que no encuentra 'react-native-cli', desinstalar e instalar la siguiente versión ([solución](#)):

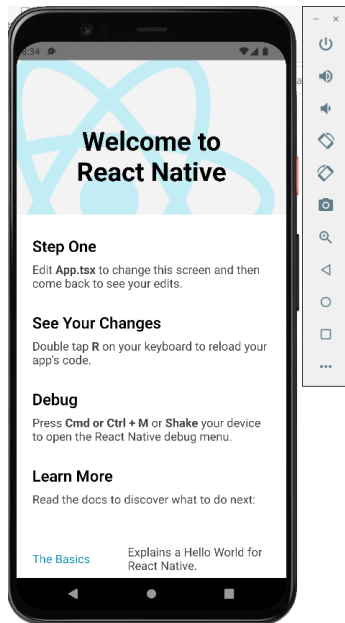
```
npm uninstall -g react-native-cli
npm i -g @react-native-community/cli
```

Y ya después volver a lanzarla con npx de arriba.

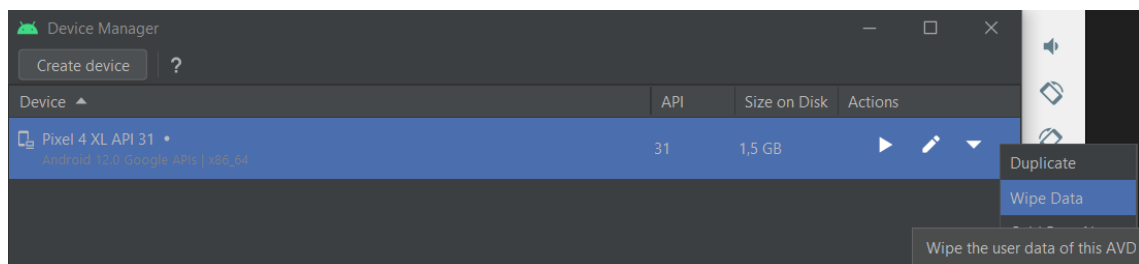
Ahora, desde la carpeta creada del proyecto, lanzar app (se abrirá el emulador, pero mejor tenerlo antes abierto para que tarde menos y evitar fallos):

```
npx react-native run-android
```

Walá!



Nota: Si se quedase colgado el emulador o no respondiese bien, se puede solucionar borrando los datos del usuario:



### 3. Configuración del entorno de desarrollo en Mac

Para mi iMac, descargué *Xcode\_12.4.xip*. Doble clic para extraer *Xcode.app* y mover este al directorio de *Aplicaciones*.

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

Seleccionar *React Native CLI Quickstart > macOS > Android*

Follow these instructions if you need to build native code in your project. For example, if you are integrating React Native into an existing application, or if you ran "prebuild" from Expo to generate your project's native code, you'll need this section.

The instructions are a bit different depending on your development operating system, and whether you want to start developing for iOS or Android. If you want to develop for both Android and iOS, that's fine - you can pick one to start with, since the setup is a bit different.

#### Development OS

☒ macOS☐ Windows☐ Linux

#### Target OS

☒ Android☐ iOS

Seguir las indicaciones que vienen después para instalar lo que sigue:

Instalar el gestor de paquetes para Mac y Linux **HomeBrew**:

Probamos en la terminal `brew --version` y si no lo tenemos instalado, copiar comando de aquí: <https://brew.sh/>

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Instalar *node* y *watchman*:

```
brew install node
brew install watchman
```

Instalar JDK

```
brew tap homebrew/cask-versions
brew install --cask zulu11
```

Instalar Android Studio: <https://developer.android.com/studio>

...

# ¿Fabio te animas a seguir?

