

Hola Mundo con NativeScript

Rodriguez Perez, Juan Manuel



Contenidos

Contenidos.....	1
1. Introducción	2
2. Elementos a instalar	2
2.1 Preparación de entorno en Windows	3
2.1.1 Instalación Chocolatey.....	3
2.1.2 Instalación de Node.....	3
2.1.3 Instalación de JDK8 de Java	3
2.1.4 Instalación de SDK de Android.....	3
2.1.5 Instalación complementos del SDK de Android.....	4
2.1.6 Instalación AVDs (Dispositivos virtuales Android).....	4
2.1.7 Emulador Windows.....	4
3. Creando nuestra aplicación	5

1. Introducción

Nativescript es un framework opensource para desarrollar aplicaciones nativas en IOS y Android en formato cross-platform.

Su código se apoya en Angular, es recomendable para su uso apoyarse en typescript y angular aunque se puede desarrollar con javascript.

Más información en su web: <https://www.nativescript.org/>

Guía de Inicio con JavaScript: <https://docs.nativescript.org/tutorial/chapter-0>

Guía de Inicio con TypeScript y Angular: <https://docs.nativescript.org/angular/tutorial/ng-chapter-0>

2. Elementos a instalar

- a. El primer paso de todo es instalar en nuestro equipo Node, para ello podéis acceder a su zona de descargas (<https://nodejs.org/es/download/>).

Una vez instalado node podemos buscar su consola de comandos para seguir instalando diferentes componentes usando NPM (<https://www.npmjs.com/>)

- b. Entornos necesarios para poder instalar NativeScript:

En este punto deberemos tener varias cosas claras:

- NativeScript es para desarrollar en Android e iOS, por lo que necesitaremos tener estos entornos funcionando correctamente.
- Para Android deberemos tener JDK de java instalado (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
- También deberemos tener el SDK de Android instalado (<https://developer.android.com/sdk/download.html>)
- Para iOS se debe tener macOS

2.1 Preparación de entorno en Windows

En mi caso solo le he preparado Android ya que he creado el proyecto en Windows.

En diferentes foros te recomiendan instalar el JDK y AndroidStudio, a mí este punto es posiblemente el que me ha dado más guerra; al final he conseguido preparar el entorno siguiendo los siguientes pasos:

2.1.1 Instalación Chocolatey

Para ello desde la terminal de comandos escribimos:

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command "iex ((new-object net.webclient).DownloadString('https://chocolatey.org/install.ps1'))" && SET PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin
```

Una vez instalado debemos reiniciar el terminal

2.1.2 Instalación de Node

Aunque ya lo tengamos instalado nos aseguramos usando Chocolatey instalar la última versión estable usando el comando:

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

2.1.3 Instalación de JDK8 de Java

En la terminal escribimos el comando:

```
choco install jdk8 -y
```

2.1.4 Instalación de SDK de Android

En la terminal escribimos el comando:

```
choco install android-sdk -y
```

Después reiniciar el terminal que estemos usando

2.1.5 Instalación complementos del SDK de Android

En la terminal escribimos el comando:

```
%ANDROID_HOME%\tools\bin\sdkmanager" "tools" "platform-tools"
"platforms;android-25" "build-tools;25.0.2"
"extras;android;m2repository" "extras;google;m2repository"
```

2.1.6 Instalación AVDs (Dispositivos virtuales Android)

Desde la carpeta creada por el anterior proceso (Android/Android-sdk) nos dirigimos a la carpeta "tools" y ejecutamos "Android.nat", nos abre el sdkmanager; una vez abierto instalamos las APIs que consideremos oportunas.

Para lanzar el AVD Manager.exe lo que he hecho es copiarlos de la carpeta C:\Android\android-sdk\tools\lib to C:\Android\android-sdk y ejecutar para crear el device.

2.1.7 Emulador Windows

Para Windows 8 o superior se puede usar alternativamente su emulador de Android, (no lo he probado), para más info:

<https://www.visualstudio.com/vs/msft-android-emulator/>

3. Instalación NativeScript, usaremos el comando "**npm install -g nativescript**" en la línea de comandos de NPM.
4. Para comprobar que todo haya ido bien se puede usar el comando:
tns doctor

3. Creando nuestra aplicación

He usado VSCode para el que tenemos la extensión oficial de NativeScript (telerik), para instalarla (code --install-extension Telerik.nativescript).

Crear proyecto desde el prompt y situados en la raíz donde queramos crear el nuevo proyecto escribimos:

tns create HelloWorld --ng (opción con template de angular)

Otras opciones:

tns create MyApp: Crea la aplicación con el tema básico.

tns create MyApp --tsc: Crea la aplicación usando el template de TypeScript

tns create MyApp --template nativescript-template-tutorial: Crea la aplicación con el template del tutorial oficial.

Una vez creado abrimos la carpeta desde nuestro IDE, desde este momento las instrucciones serán referentes al uso del VSCode que nos incorpora la terminal en el propio IDE.

- a. Preparar las platform:

tns prepare android (tns prepare ios).

tns platform add <ios or Android> (añadir platform)

- b. Ejecutar la aplicación, por ejemplo en Android escribimos en la terminal:

tns run Android o (tns run Android --emulator) o iOS: **tns run ios**

En el caso de Android si no se ejecuta y nos dice no encuentra el SDK suele ser problemas en el path.

- c. Para saber los dispositivos disponibles:

tns device <platform> --available-devices

d. Algunos errores:

Para ciertos problemas no reconoce el device, en mi caso era porque no estaba bien instalado "Intel HAXM":

<https://software.intel.com/en-us/articles/intel-hardware-accelerated-execution-manager-intel-haxm>

Para ver los dispositivos disponibles podemos usar el comando:

tns device android --watch

Al hacer `tns run Android --emulator` he obtenido errores casi al final de la preparación de la aplicación para la instalación en el emulador.

Lo que he hecho es eliminar las platforms y ejecutar `tns run Android`. Y ha ido todo OK