Desarrollo Nativescript

Módulo 1 - Lectura: Configuración del Marco Nativescript

Node Version Manager (nvm)

Lo primero que haremos será instalar nvm, que es un gestor de versiones de nodejs.
Es decir, usaremos el lenguaje de programación Javascript y, por ende, necesitaremos gestionar qué versión de dicho lenguaje utilizaremos.
Es recomendable hacerlo dado que, a lo largo del tiempo, las versiones cambian y esto, frecuentemente, genera conflictos con los proyectos existentes que tengamos. Para evitar conflictos de esta índole es que usaremos un gestor de versiones que nos permitirá tener más de una versión instalada al mismo tiempo y además nos permitirá seleccionar cuál queremos que sea la activa en un momento dado.
Por ejemplo, podremos dirigirnos a un proyecto X y seleccionar la versión node 8 y luego ir a otro proyecto y elegir la versión node 10.

Instalación:

Para Windows, dirígete a https://github.com/coreybutler/nvm-windows/releases y descarga la última versión (al momento de confeccionar este documento la versión es 1.1.7).

Los usuarios Linux/Mac, la instalación que realizarán será la del proyecto hermano del de Windows. Esta versión también está en github, pero en otro proyecto: https://github.com/creationix/nvm. El comando de instalación que instala la versión actual 0.34.0 será el siguiente:

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh | bash
```

Uso de comandos:

1. Listado de versiones de node en nuestro PC: “nvm list”
2. Instalar una versión de node, la actual: “nvm install node”
3. Instalar una versión específica de node: “nvm install 6.14”
4. Si tenemos más de una versión, podemos elegir cuál queremos que sea la actual a usar: “nvm use 6.14” o bien “nvm use node”
5. Si necesitamos saber dónde está instalada la versión de node en nuestro PC: “nvm which 6.14”
6. Si queremos saber qué versiones de node están disponibles para instalar:
    a. En la versión de Linux “nvm ls-remote”
    b. En la versión de Windows “nvm ls available”

Cuando instalamos nvm, tenemos disponible tanto node como npm.

¿Qué es NPM?

NPM es el gestor de paquetes de nodejs, es decir que sirve para gestionar la descarga y versionado de las librerías ejecutables en node, aunque las mismas pueden haber sido desarrolladas en nodejs o bien en Typescript.

npm consta de tres componentes distintos:

1. el sitio web
2. la interfaz de línea de comandos (CLI)
3. el registro

Utiliza el sitio web para descubrir paquetes, configurar perfiles y administrar otros aspectos de tu cuenta npm. Por ejemplo, se pueden configurar organizaciones para administrar el acceso a paquetes públicos o privados.

El CLI se ejecuta desde un terminal y es así cómo la mayoría de los desarrolladores interactúan con npm.

El registro es una gran base de datos pública de software JavaScript y la metainformación que lo rodea. Es más, ¡npm es el mayor registro de software del mundo!

npm sirve para:

1. Utilizar paquetes de código en nuestras aplicaciones, tal como son o adaptándolos.
2. Descargar herramientas independientes que se pueden usar de inmediato, por ejemplo, las herramientas de línea de comandos de Angular o Nativescript.
3. Ejecutar paquetes sin descargar, usando npx.
4. Compartir código con cualquier usuario npm, en cualquier lugar.
5. Restringir el código a desarrolladores específicos.
6. Crear organizaciones para coordinar el mantenimiento de paquetes, la codificación y los desarrolladores.
7. Gestionar múltiples versiones de código y dependencias de código.
8. Actualizar fácilmente las aplicaciones cuando se actualiza el código subyacente, gracias al versionado de paquetes.

Instalar la línea de comandos de Nativescript

Instalaremos la utilidad como un paquete npm, por ende, ejecutar:

```
npm install -g nativescript
```

Después de completar la configuración, debe haber dos comandos disponibles en la terminal: tns (abreviatura de Telerik Nativescript) y Nativescript. Los dos comandos son idénticos, por lo que probablemente querrás seguir con “tns”, ya que es más corto.

Continúa y verifica que la instalación haya sido exitosa, ejecutando tns en su terminal.
Deberías ver una larga lista de comandos a modo de listado explicativo.

Para verificar si el sistema está configurado correctamente, ejecuta el siguiente comando:

```
tns doctor
```

De la salida del comando anterior pueden desprenderse muchas configuraciones a ajustar. Principalmente nos centraremos en las de Android con el uso del emulador, dado que en todos los sistemas operativos, esto será factible de realizar, pero para el caso de iOS puedes remitirte a la documentación oficial (https://docs.nativescript.org/start/ns-setup-os-x).

Creación de una primera aplicación

Analizaremos los conceptos básicos del CLI tns, creando apps.
Usaremos el comando tns create
Abre tu terminal y ejecuta el siguiente comando para crear una nueva aplicación Nativescript:

```
tns create MyApp --template tns-template-blank
```

Aquí le está pasando dos cosas al comando crear: MyApp, que determina el nombre de la aplicación que está creando, y la opción --template, que le dice a la CLI de Nativescript que realice un andamiaje (*scaffolding*) de una aplicación, usando una plantilla predefinida llamada “plantilla-tns -blanco".

Puedes usar el comando tns create para crear andamios de aplicaciones en una variedad de puntos de partida diferentes. Para probar otras plantillas, ejecuta tns create sin la opción --template; la CLI de Nativescript te guiará a través de la selección de una plantilla mediante indicaciones interactivas.

El comando de creación tardará un minuto en completarse, ya que la CLI de Nativescript necesita descargar algunas dependencias al configurar su nueva aplicación.

Cuando el comando termine, usa el comando cd (cambiar directorio) para navegar a la carpeta de su nueva aplicación.

Puedes usar el comando tns help para ver la documentación de ayuda de la CLI de Nativescript en su navegador web.