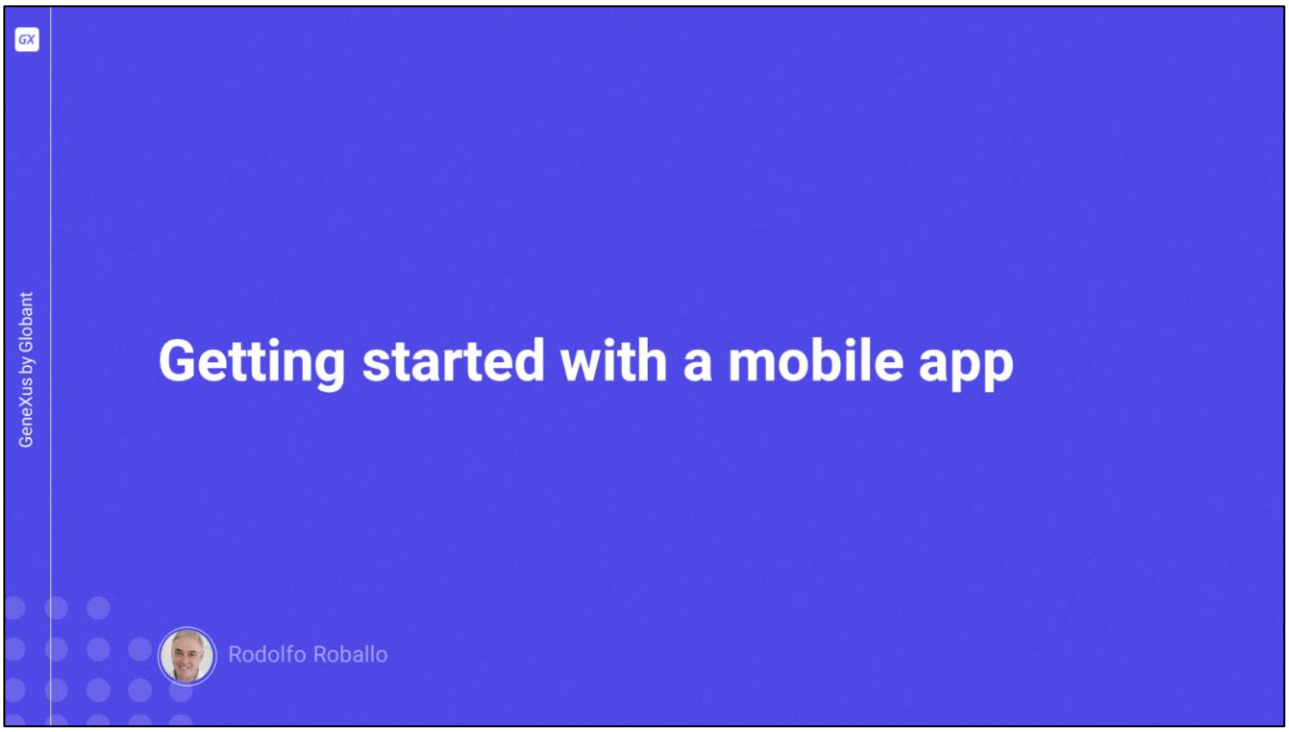
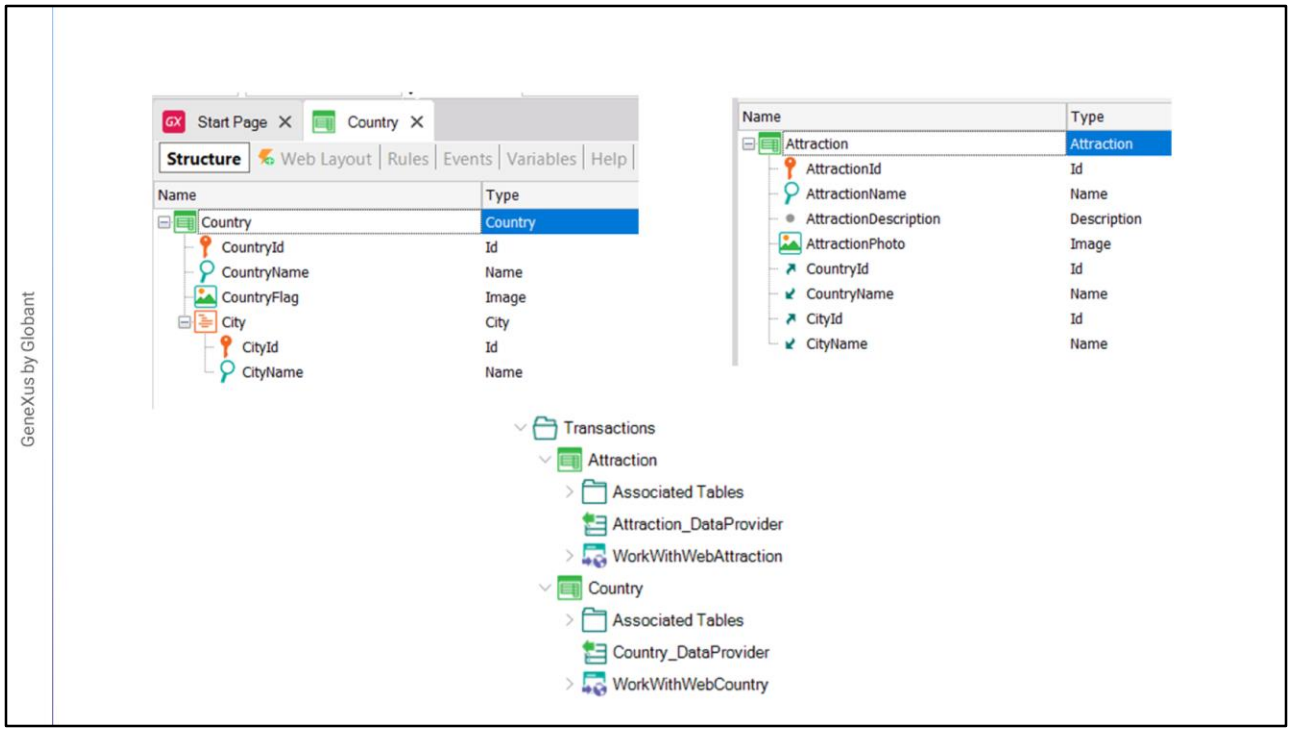


Comenzaremos ahora a desarrollar nuestra primera aplicación nativa para dispositivos móviles.

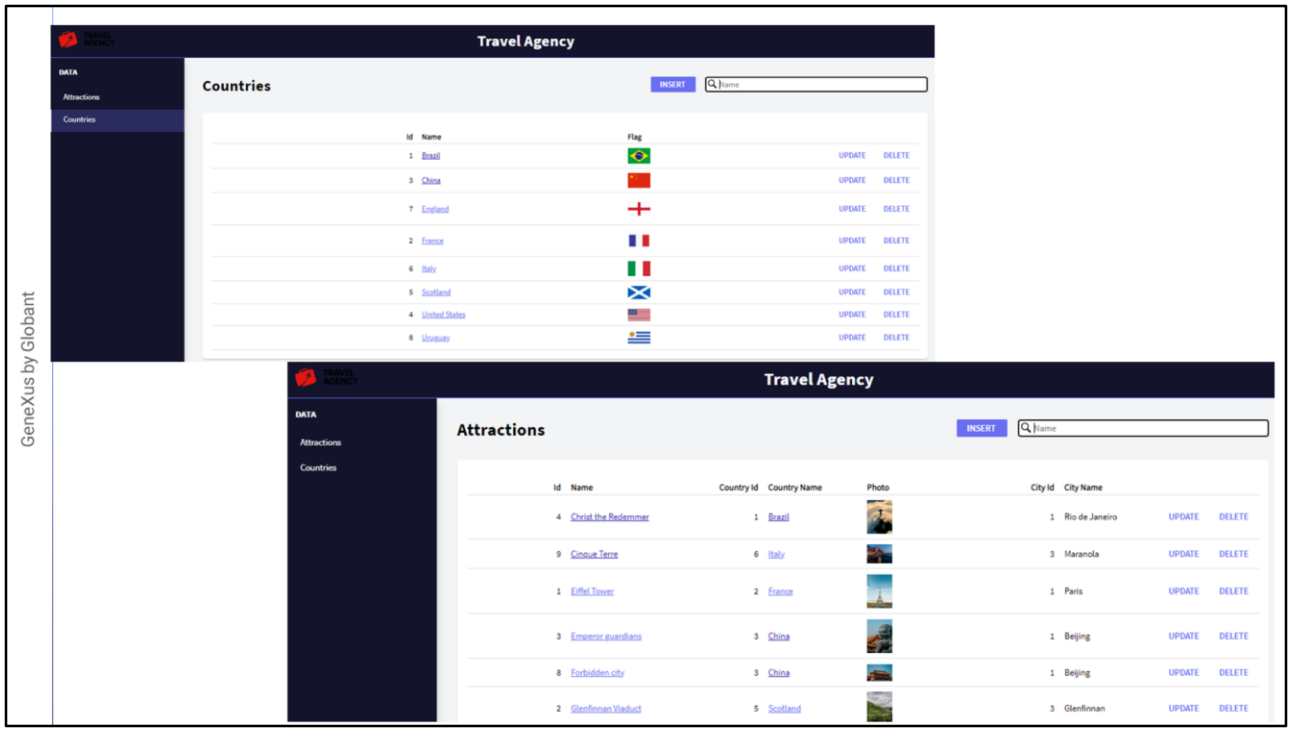


Aquí vemos que ya hemos creado una KB con algunos objetos. Como nuestros ejemplos se van a basar en una realidad de una agencia de viajes, tenemos las transacciones de Países con sus datos básicos como su identificador, nombre, y bandera y un nivel subordinado para modelar que un país tiene muchas ciudades y almacenamos el identificador y nombre de cada ciudad.

También tenemos a la transacción de Atracciones turísticas que la agencia de viaje promociona, con su identificador, nombre, descripción, foto y el país y ciudad a la que pertenece cada atracción.

Vemos que cada transacción tiene su data provider para poblar con datos la tabla asociada en el momento de su creación en la base de datos. También vemos que las transacciones tienen aplicado el patron Work With for Web, por lo que estos objetos forman una parte de la aplicación de backoffice de la agencia de viajes.

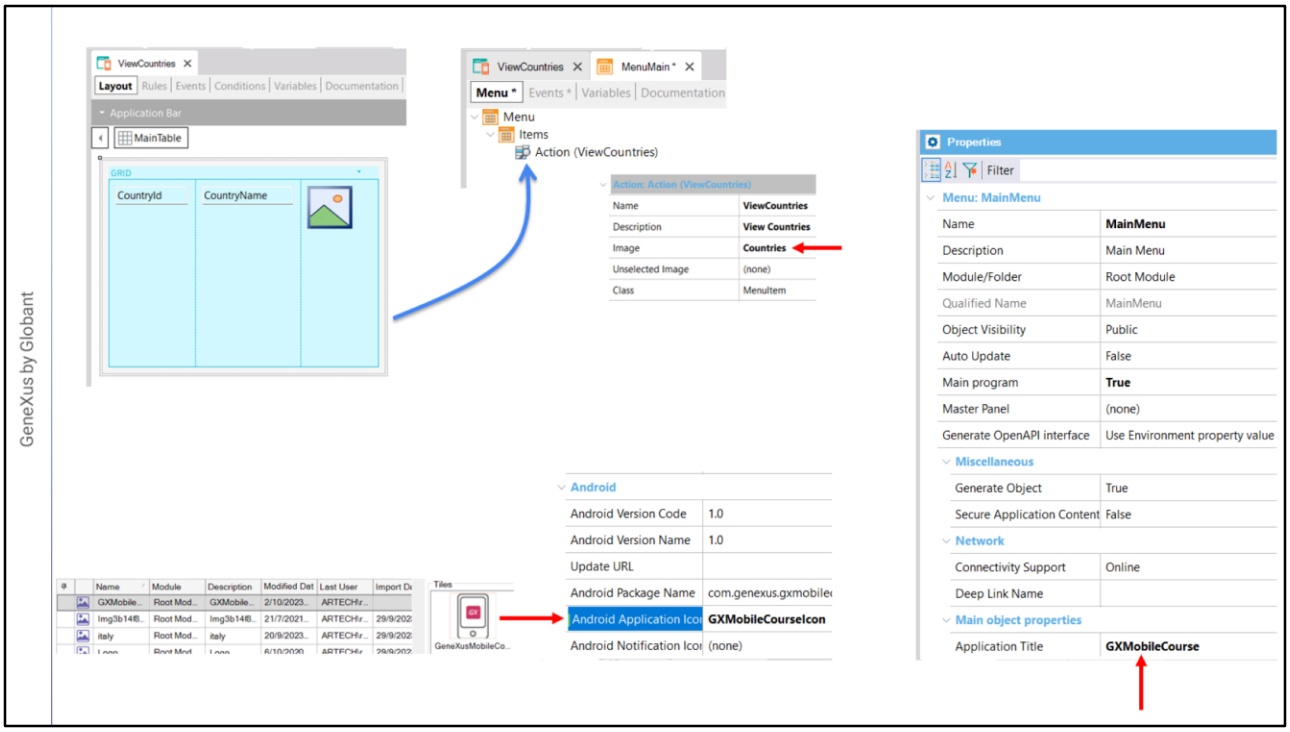
Estos objetos están disponibles en un xpz llamado ImportBeforeFirstRun.xpz que pueden descargar desde la página del curso, por si desean crear la KB desde cero e importar el xpz para dejarla en el estado inicial de este video.



Antes de crear nuestra primera pantalla mobile, vamos a dar F5 para ejecutar las pantallas del backoffice y familiarizarnos con los datos que vamos a usar.

Abrimos el launchpad y ejecutamos el objeto WWCountry para ver los datos de los países. Seleccionamos un país y vemos sus ciudades.

A la izquierda de la pantalla tenemos disponible un menu que nos permite trabajar con otras entidades, por ejemplo las atracciones turísticas, así que lo abrimos y vemos que tenemos varias atracciones ingresadas con sus datos principales.

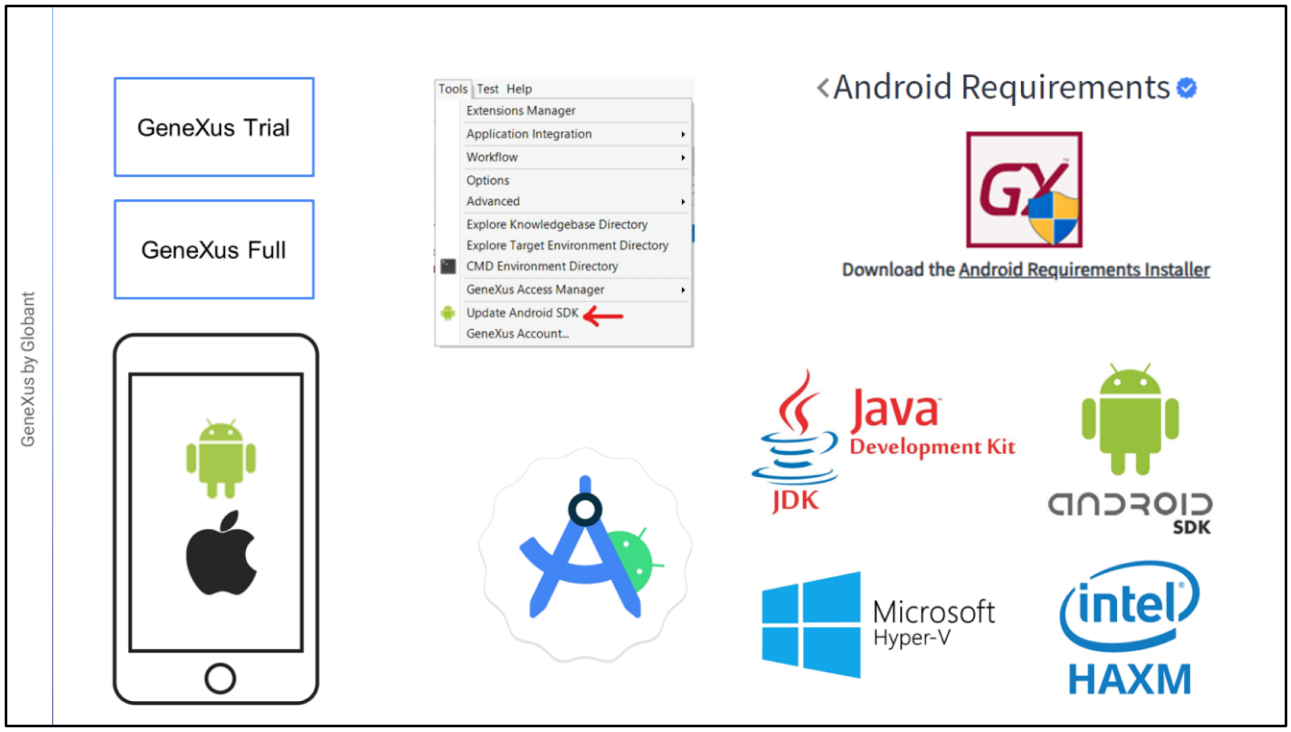


Ahora vamos a empezar a desarrollar las pantallas de nuestra aplicación mobile. Creamos un objeto panel llamado ViewCountries, arrastramos un control grid al form y seleccionamos a los atributos CountryId, CountryName y CountryFlag. Y salvamos.

Este objeto podría ser definido como main, pero tiene sentido que lo invoquemos desde otro objeto que sea el que se abra al ejecutar la aplicación, por ejemplo un menú.

Así que creamos un objeto menu de nombre MainMenu y arrastramos al panel ViewCountries a su sección Items. En las propiedades del action creado, asignamos la propiedad Image a la imagen Countries.

Si ahora vamos a las propiedades del objeto menu, vemos que por defecto es un objeto main. Asignamos el nombre a la propiedad Application Title y bajo la sección Android elegimos la imagen que será el icono de nuestra aplicación. Y por ultimo seteamos al objeto menu como objeto startup.

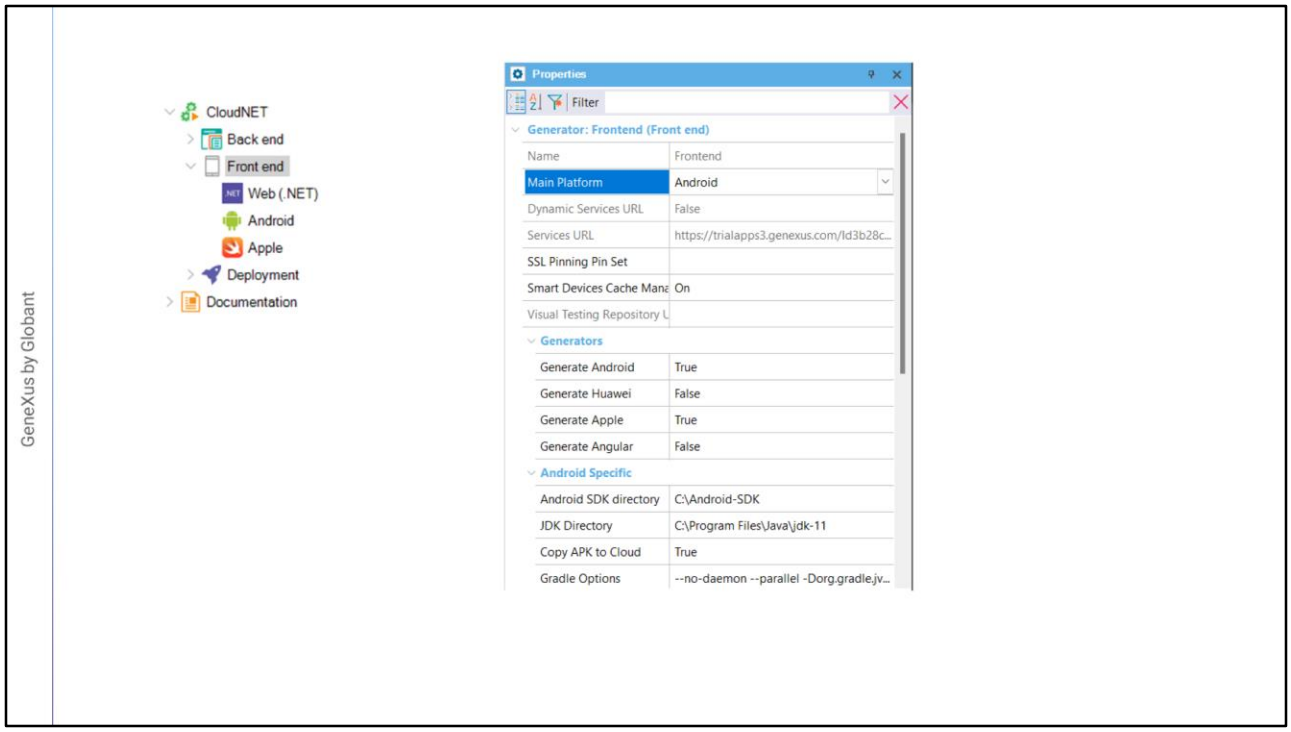


Antes de ejecutar la aplicación por primera vez, debemos verificar que cumplimos con los requerimientos para generar una aplicación mobile. En este curso vamos a generar para Android y para Apple, y comenzaremos prototipando con un emulador para Android aunque más adelante veremos cómo prototipar en dispositivos físicos de Android y Apple.

Para realizar este curso alcanza con que nos instalemos la versión GeneXus Trial y al hacerlo se ofrecerá instalar automáticamente el software necesario para generar aplicaciones en Android. Pero si tienes instalada la versión GeneXus Full o no aceptaste la instalación del entorno para Android, en el IDE de GeneXus está disponible la herramienta Update Android SDK que instalará todos los requerimientos necesarios del Android SDK.

Otra posibilidad es buscar en el wiki por "Android Requirements", donde la página que encontramos dispone de un instalador automático de los requerimientos de Android, y también si llegara a ser necesario, contamos con información detallada para hacer la instalación manual, por ejemplo del Oracle JDK y el Android SDK y también del Hardware Accelerated Execution Manager (HAXM) de Intel o el Windows Hypervisor Plataform (WHPX) de Windows para que el emulador funcione más rápidamente.

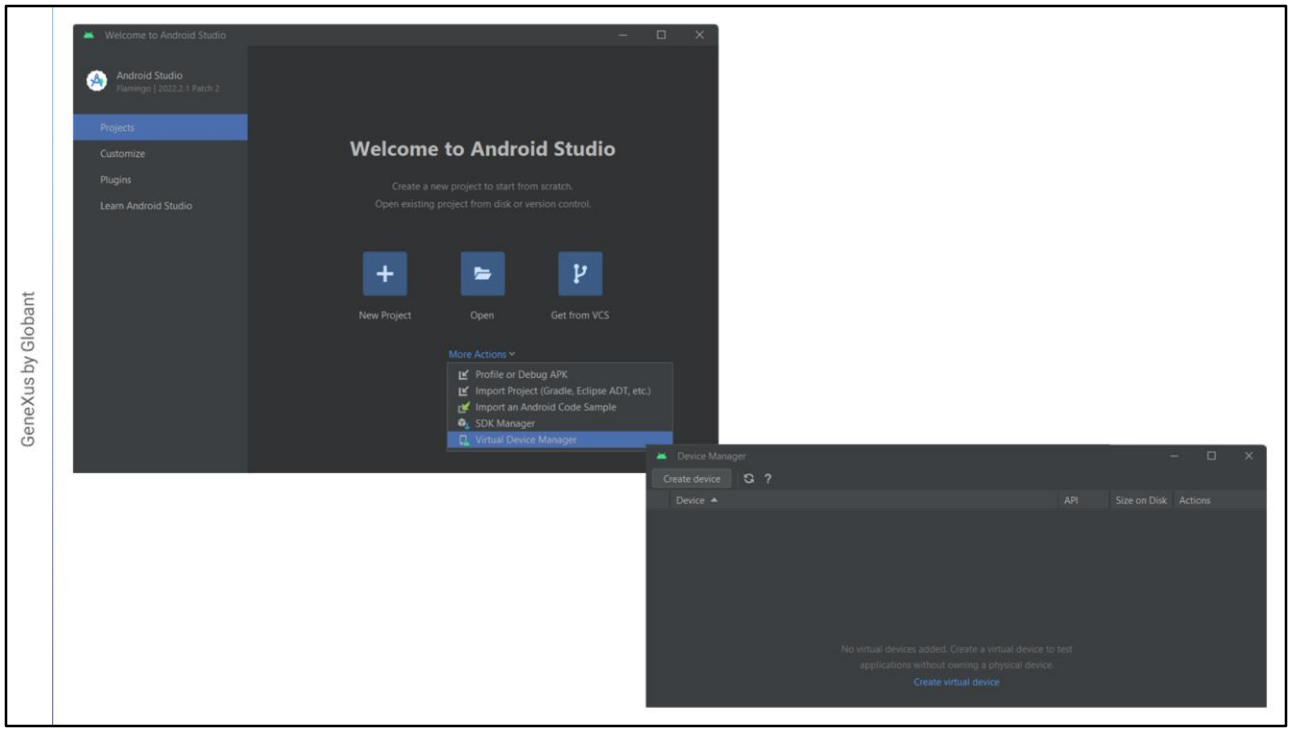
Si bien GeneXus es capaz de crear automáticamente un emulador de un dispositivo Android, es conveniente instalarse también el Android Studio, que permitirá crear dispositivos virtuales personalizados y nos ayudará a monitorear el funcionamiento de nuestra aplicación generada.



Una vez que tenemos todos el software instalado, vamos a ejecutar nuestra aplicación mobile por primera vez, pero antes debemos verificar que tengamos los seteos de nuestro front end adecuadamente configurados para hacerlo.

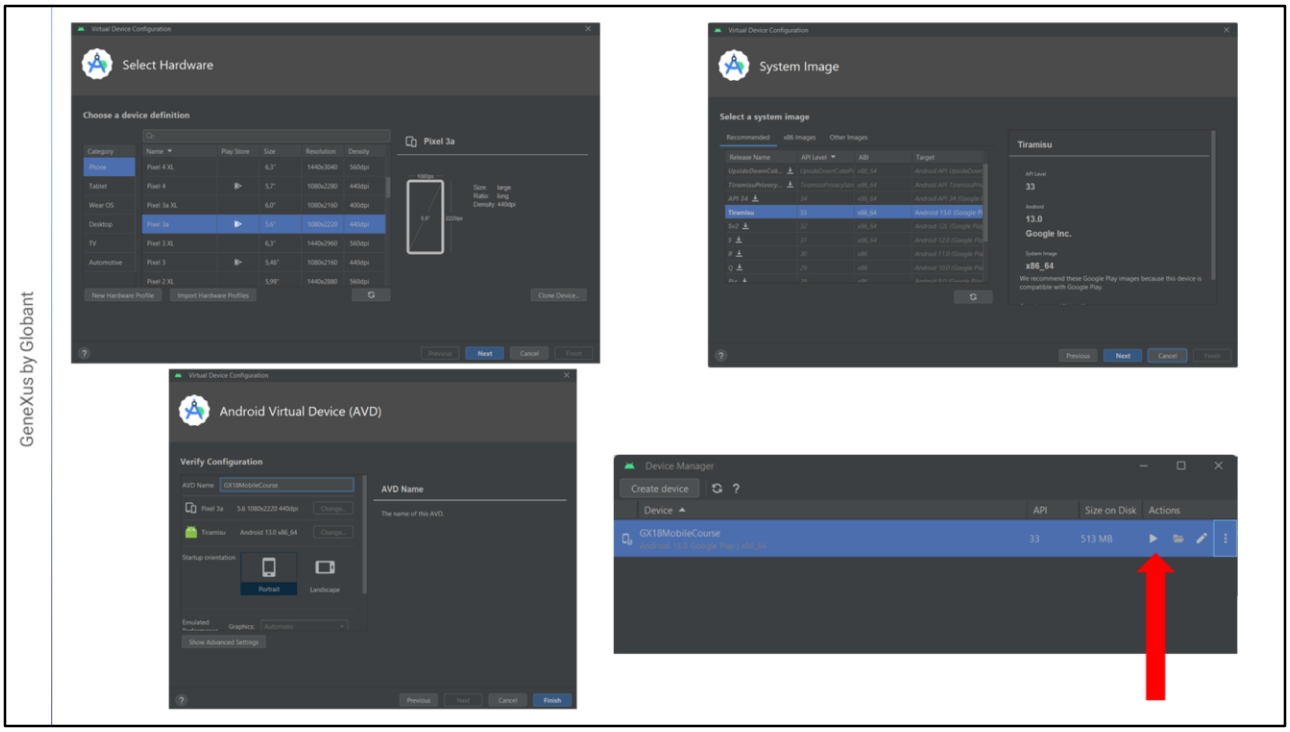
Vamos al KB explorer y abrimos la sección Front end. Vemos que por defecto aparecen Android y Apple, y si vamos a las propiedades del nodo vemos que esto es porque las propiedades Generate Android y Generate Apple están por defecto en True.

También vemos que la propiedad Main Platform tiene el valor Android y que bajo la sección Android Specific la propiedad Android SDK directory se seteó automáticamente con el directorio donde tenemos instalado el SDK de Android, y lo mismo para el directorio donde está el JDK.



Antes de ejecutar, vamos a abrir el Android Studio. Hacemos clic en More Actions y luego ejecutamos el Virtual Device Manager.

Vamos a crear un emulador que nos guste, pero sobre todo, que sea adecuado para lo que vamos a hacer en el curso.

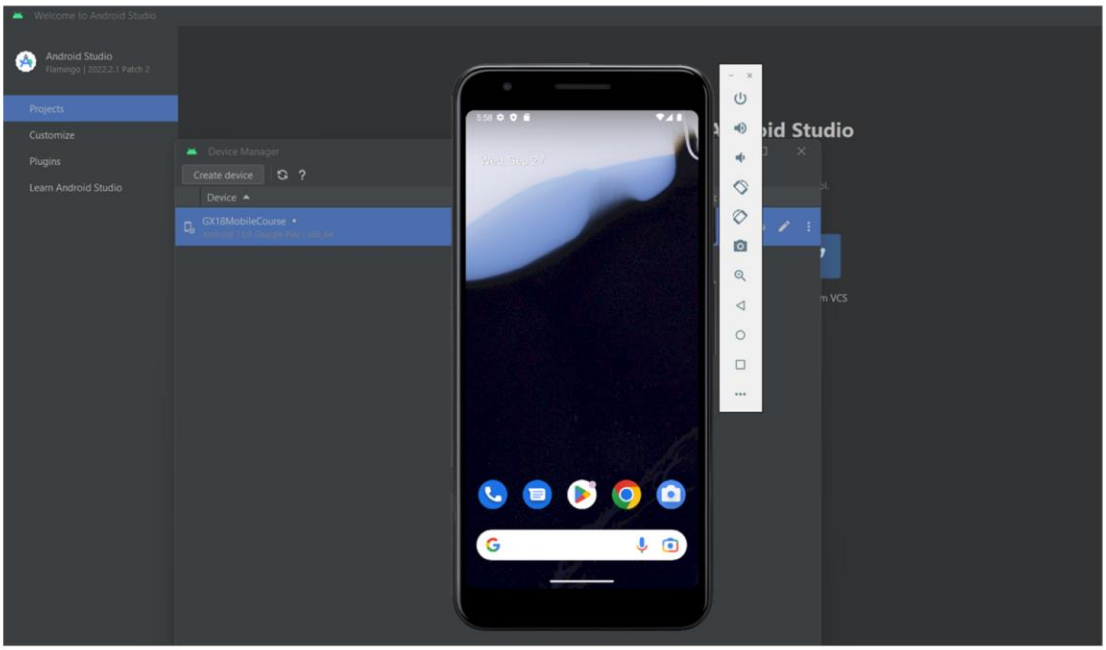


Presionamos Crear dispositivo virtual y ahora damos Next. Elegimos el que se llama Pixel 3a y presionamos Next.

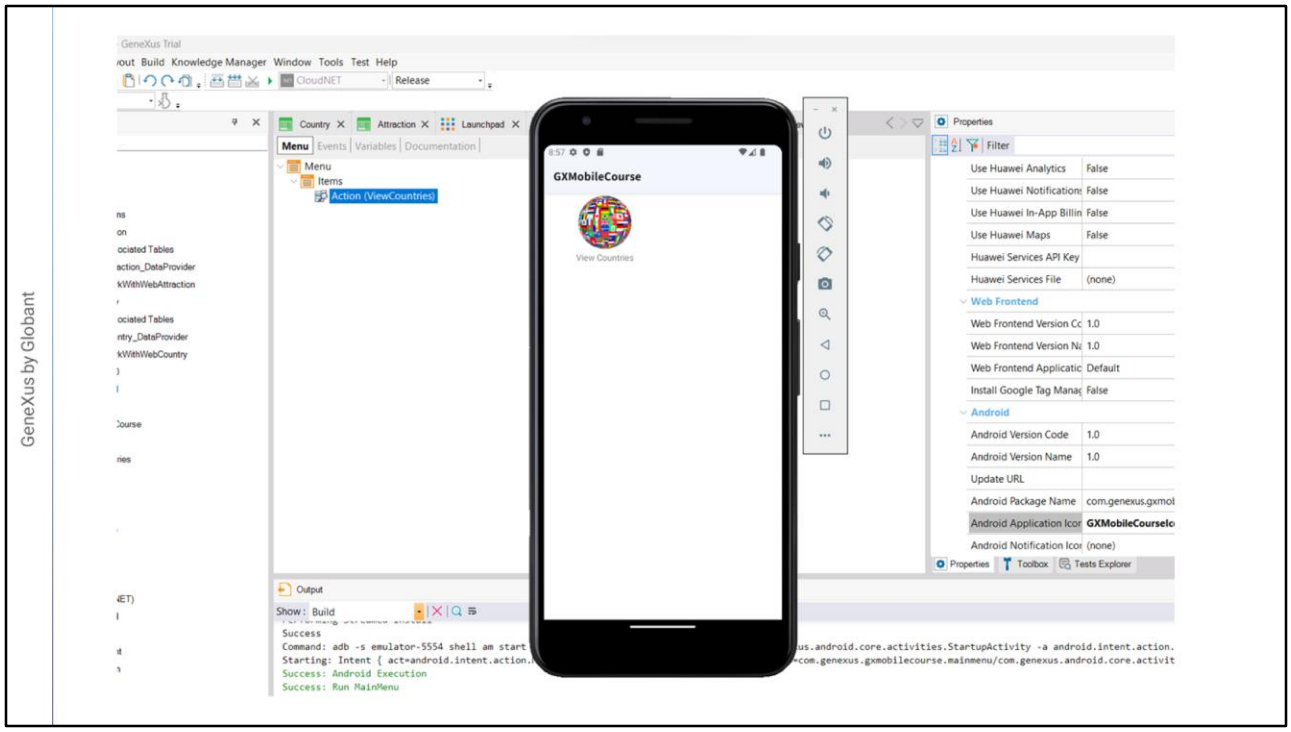
Seleccionamos el renglón correspondiente a la API 33 y presionamos Next.

Le damos el nombre a nuestro Android Virtual Device, le ponemos GXMobileCourse y presionamos Finish.

Ahora para probarlo presionamos el botón de play…

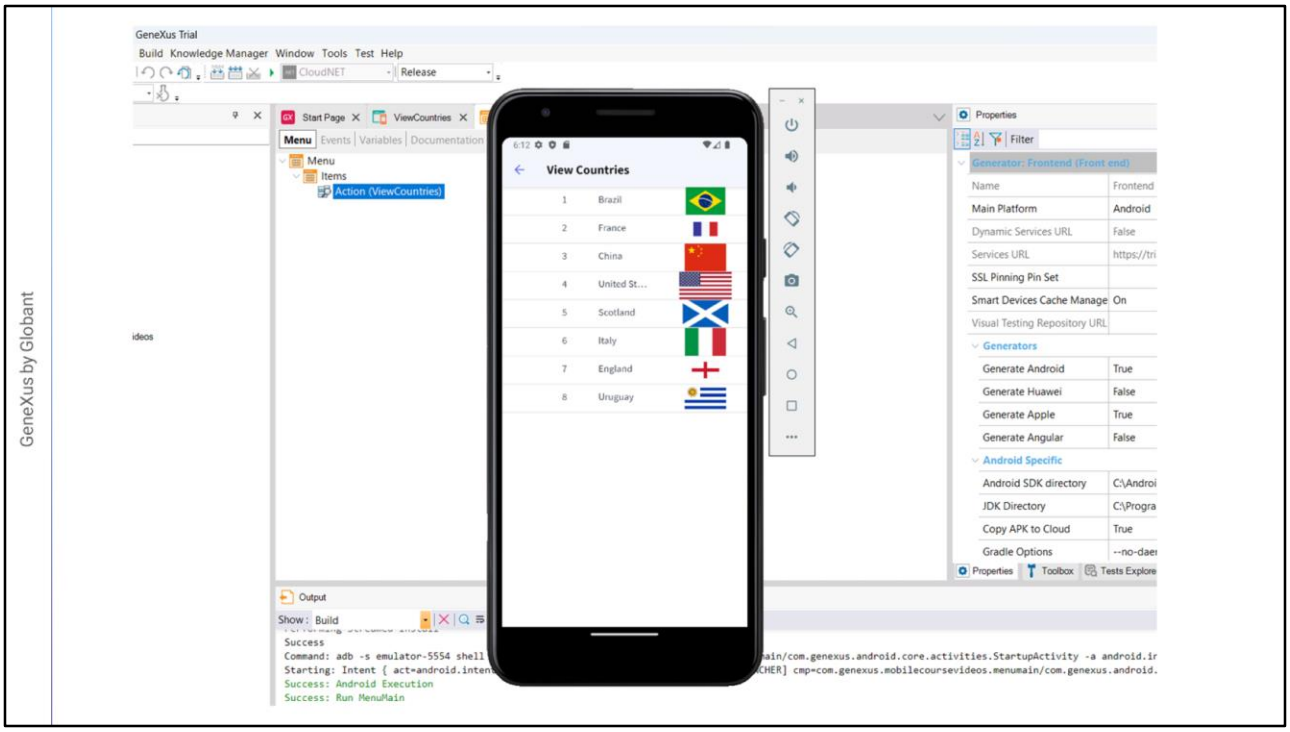


Vemos que después de unos segundos, se abre el emulador con una apariencia idéntica a un dispositivo real.



Volvemos a GeneXus. Como ya habíamos seteado al objeto MainMenu como startup object, será el objeto que se ejecutará en el emulador, así que presionamos F5.

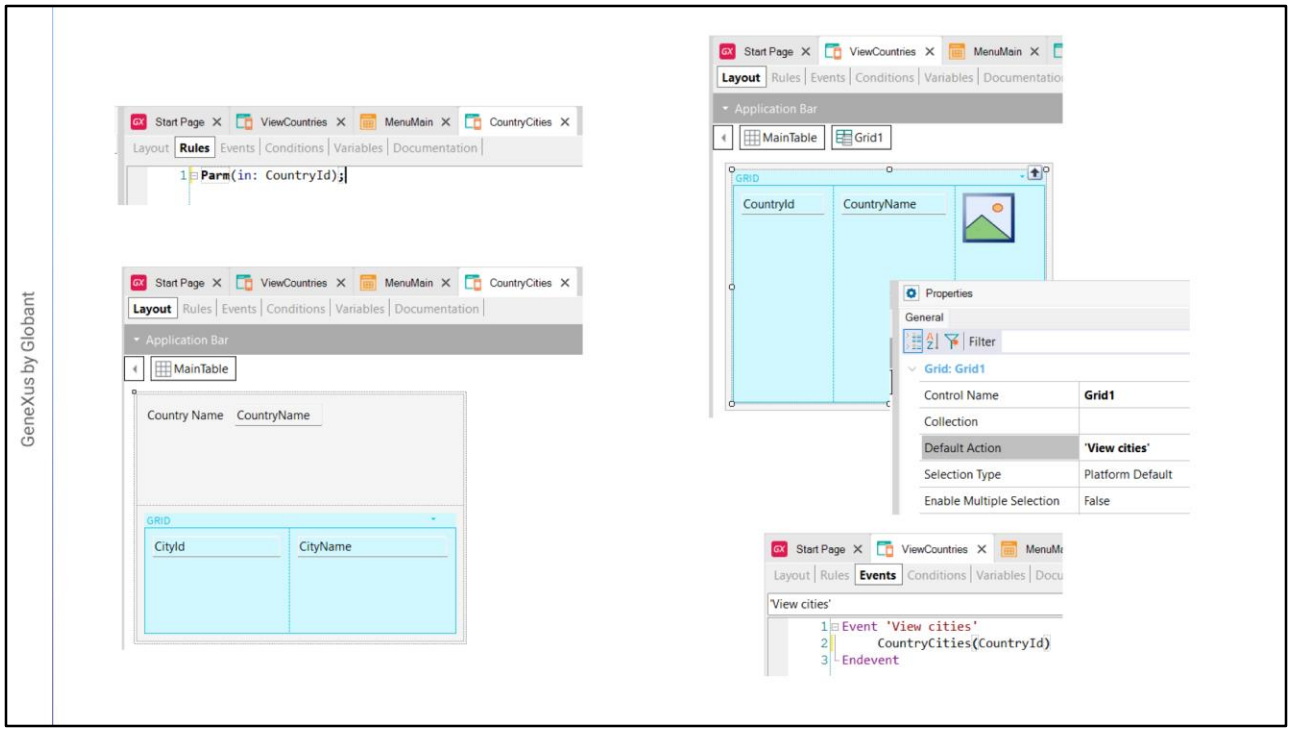
Vemos en la ventana de Output que se ejecutó el objeto MainMenu y que en el emulador aparece un icono con el nombre View Countries.



Damos tap sobre el icono y nos aparecen los datos de los países almacenados en la base de datos.

Tenemos nuestra primera aplicación nativa mobile funcionando! Después nos preocuparemos de que el diseño sea más adecuado.

Para poder ver la información de las ciudades de un país, debemos crear un panel que se ejecute cuando se haga tap sobre una fila del grid.

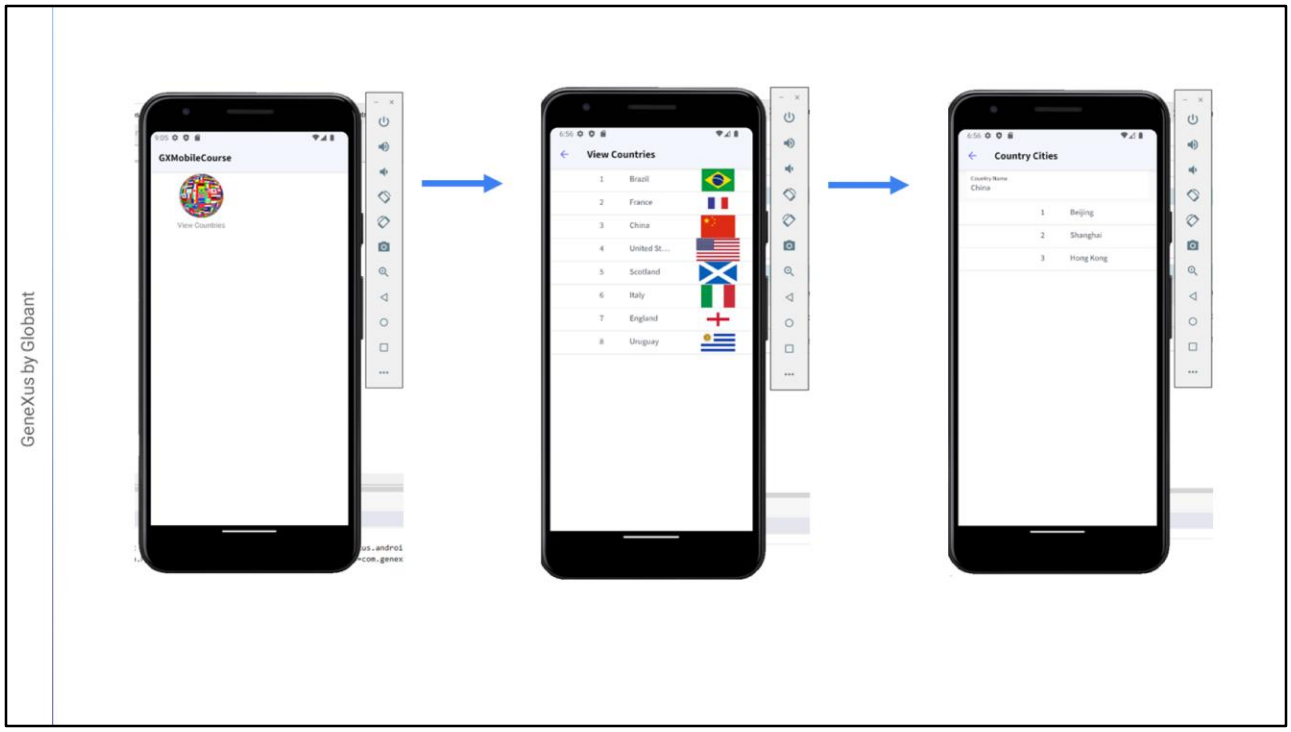


Así que creamos el panel CountryCities y le agregamos una regla Parm para que reciba al atributo CountryId por parámetro. Con esto nos aseguramos que toda la información del panel quedará automáticamente filtrada por este valor y aunque accedamos a la tabla donde están almacenadas todas las ciudades, veremos únicamente las ciudades del país que nos interesa.

Luego arrastramos al layout al atributo CountryName y a una grilla, a la que seleccionamos que contenga a los atributos CityId y CityName. Salvamos y volvemos al panel ViewCountries.

Seleccionamos el grid y en su propiedad Default Action elegimos el valor <new> y escribimos View cities. Ahora abrimos la solapa de eventos y en el evento View Cities invocamos al panel de las ciudades escribiendo CountryCities y pasando entre paréntesis el identificador del país. Esto hará que cuando hagamos tap sobre un país, se abra el panel con los datos de sus ciudades.

Vamos a probar esto haciendo F5.



Vemos que se actualizó la pantalla del emulador y otra vez vemos el icono de View Countries. Hacemos tap para ver la lista de países y ahora damos tap sobre China.

Y vemos que se nos muestra a las ciudades de China: Beijing, Shanghai y Hong Kong. De nuevo, podríamos mejorar la forma de ver estos datos.

Para volver a la pantalla anterior vemos que arriba a la izquierda nos aparece una flecha, si bien el dispositivo Android nos provee un botón específico para eso, pero no todos los dispositivos proveen un botón particular.

Más adelante entraremos en ésta y otras consideraciones de funcionalidad definidas en las guías de diseño de cada plataforma.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com