

First steps with a native mobile app



Rodolfo Roballo

Começaremos agora a desenvolver nossa primeira aplicação nativa para dispositivos móveis.

Travel Agency initial transaction model

Name	Type
Country	Country
CountryId	Id
CountryName	Name
CountryFlag	Image
City	City
CityId	Id
CityName	Name

Name	Type
Attraction	Attraction
AttractionId	Id
AttractionName	Name
AttractionDescription	Description
AttractionPhoto	Image
CountryId	Id
CountryName	Name
CityId	Id
CityName	Name

- Transactions
 - Attraction
 - Associated Tables
 - Attraction_DataProvider
 - WorkWithWebAttraction
 - Country
 - Associated Tables
 - Country_DataProvider
 - WorkWithWebCountry

Aqui vemos que já criamos uma KB com alguns objetos. Como nossos exemplos serão baseados em uma realidade de uma agência de viagens, temos as transações de Países com seus dados básicos como seu identificador, nome e bandeira e um nível subordinado para modelar que um país tem muitas cidades e armazenamos o identificador e nome de cada cidade.

Temos também a transação de Atrações turísticas que a agência de viagens promove, com seu identificador, nome, descrição, foto e o país e cidade a que pertence cada atração.

Vemos que cada transação tem seu data provider para preencher com dados a tabela associada no momento de sua criação na base de dados. Vemos também que as transações têm aplicado o padrão Work With for Web, portanto esses objetos fazem parte da aplicação de backoffice da agência de viagens.

Esses objetos estão disponíveis em um xpz chamado ImportBeforeFirstRun.xpz que vocês podem baixar da página do curso, caso queiram criar a KB do zero e importar o xpz para deixá-la no estado inicial deste vídeo.

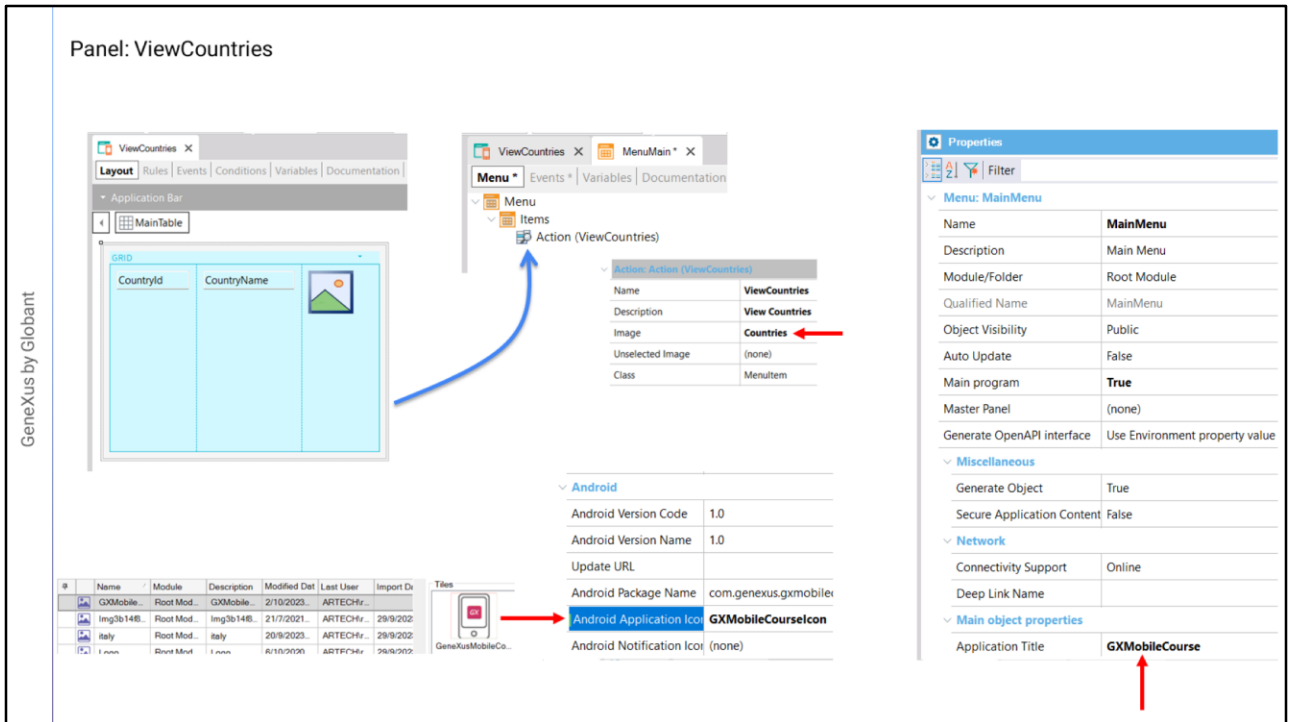
GeneXus by Globant

Travel Agency Back Office

Antes de criar nossa primeira tela mobile, vamos pressionar F5 para executar as telas do backoffice e nos familiarizarmos com os dados que iremos usar.

Executamos o objeto WWCountry para ver os dados dos países. Seleccionamos um país e vemos suas cidades.

À esquerda da tela temos disponível um menu que nos permite trabalhar com outras entidades, por exemplo as atrações turísticas, então nós o abrimos e vemos que temos várias atrações inseridas com seus dados principais.

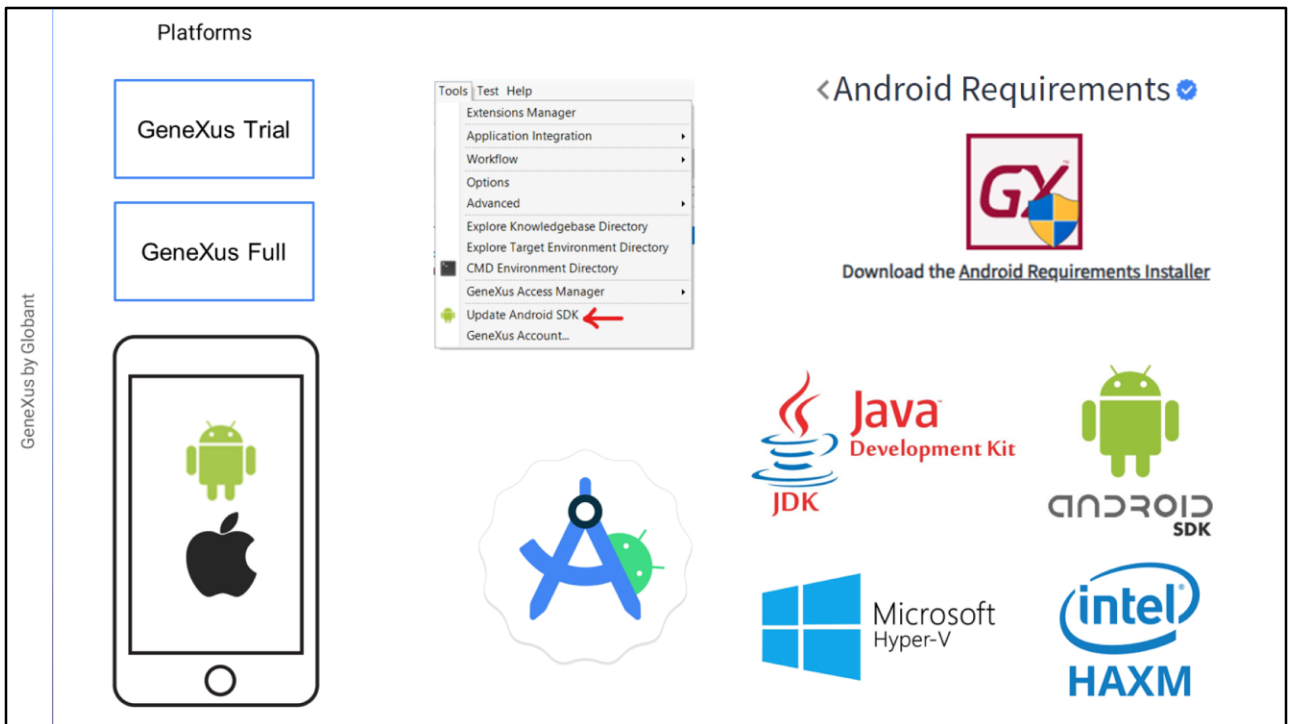


Agora vamos começar a desenvolver as telas de nossa aplicação mobile. Criamos um objeto panel chamado ViewCountries, arrastamos um controle grid para o form e selecionamos os atributos CountryId, CountryName e CountryFlag. E salvamos.

Este objeto poderia ser definido como main, mas faz sentido invocá-lo a partir de outro objeto que seja aquele que se abre ao executar a aplicação, por exemplo um menu.

Então criamos um objeto menu chamado MainMenu e arrastamos o panel ViewCountries para sua seção Items. Nas propriedades da action criada, atribuímos a propriedade Image à imagem Countries.

Se agora formos às propriedades do objeto menu, veremos que por padrão ele é um objeto main. Atribuímos o nome à propriedade Application Title e sob a seção Android escolhemos a imagem que será o ícone de nossa aplicação. E por último definimos o objeto main como objeto startup.

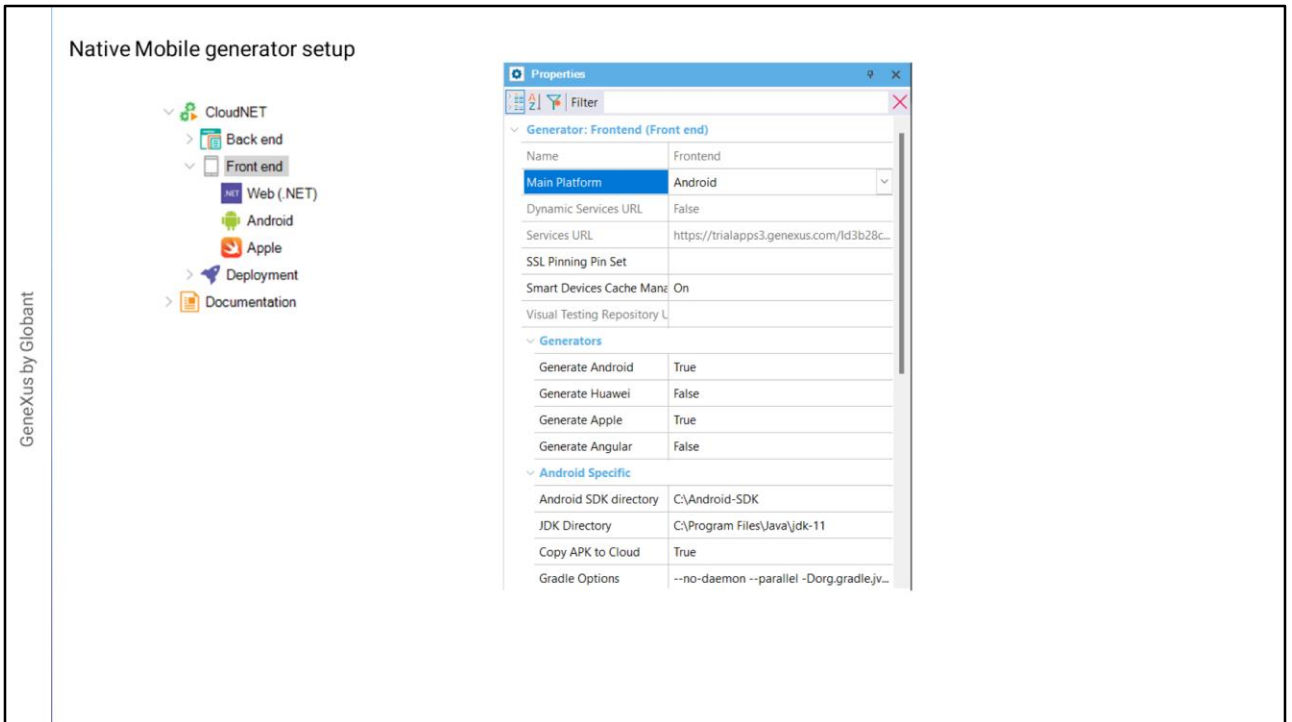


Antes de executar a aplicação pela primeira vez, devemos verificar se atendemos aos requisitos para gerar uma aplicação mobile. Neste curso iremos gerar para Android e para Apple, e começaremos prototipando com um emulador para Android, embora mais adiante veremos como prototipar em dispositivos físicos de Android e Apple.

Para realizar este curso basta instalar a versão GeneXus Trial e ao fazer isso será oferecido instalar automaticamente o software necessário para gerar aplicações em Android. Mas se você tem instalada a versão GeneXus Full ou não aceitou a instalação do ambiente para Android, no IDE de GeneXus está disponível a ferramenta Update Android SDK, que instalará todos os requisitos necessários do Android SDK.

Outra possibilidade é pesquisar na wiki por “Android Requirements”, onde a página que encontramos dispõe de um instalador automático dos requisitos de Android, e também caso seja necessário, temos informações detalhadas para fazer a instalação manual, por exemplo do Oracle JDK e o Android SDK e também do Hardware Accelerated Execution Manager (HAXM) da Intel ou a Windows Hypervisor Platform (WHPX) do Windows para que o emulador funcione mais rapidamente.

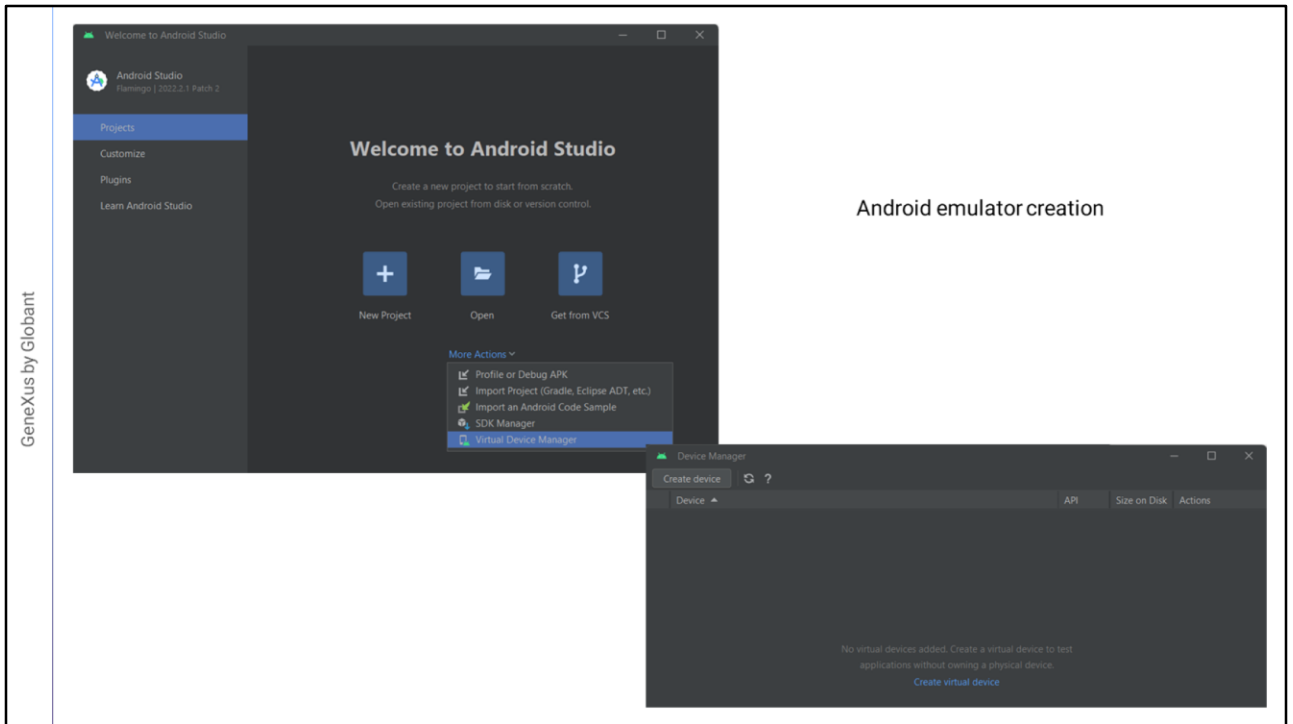
Embora GeneXus seja capaz de criar automaticamente um emulador de um dispositivo Android, é conveniente instalar também o Android Studio, que nos permitirá criar dispositivos virtuais personalizados e nos ajudará a monitorar o funcionamento de nossa aplicação gerada.



Uma vez que temos todos os softwares instalados, executaremos nossa aplicação mobile pela primeira vez, mas antes devemos verificar se temos as configurações de nosso front-end adequadamente definidas para isso.

Vamos para o KB Explorer e abrimos a seção Front end. Vemos que por padrão aparecem Android e Apple, e se formos para as propriedades do nó, vemos que isso ocorre porque as propriedades Generate Android e Generate Apple estão por padrão como True.

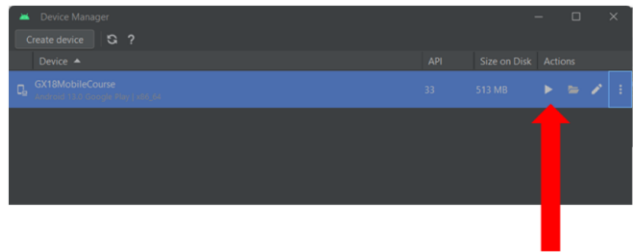
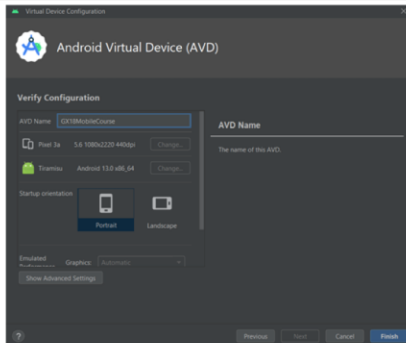
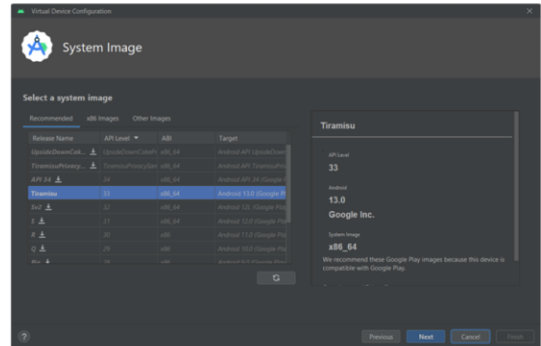
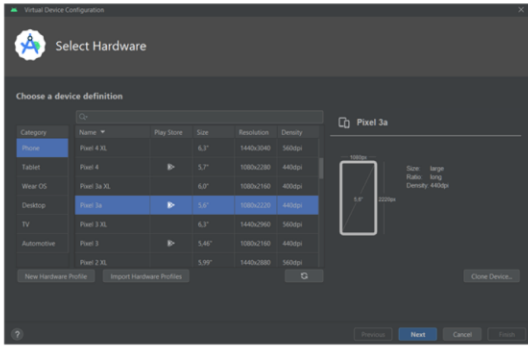
Também vemos que a propriedade Main Platform tem o valor Android e que sob a seção Android Specific a propriedade Android SDK directory foi configurada automaticamente com o diretório onde temos instalado o SDK de Android, e o mesmo para o diretório onde está o JDK.



Android emulator creation

Antes de executar, vamos abrir o Android Studio. Clicamos em More Actions e depois executamos o Virtual Device Manager.

Vamos criar um emulador que nos agrade, mas acima de tudo, que seja adequado ao que vamos fazer no curso.

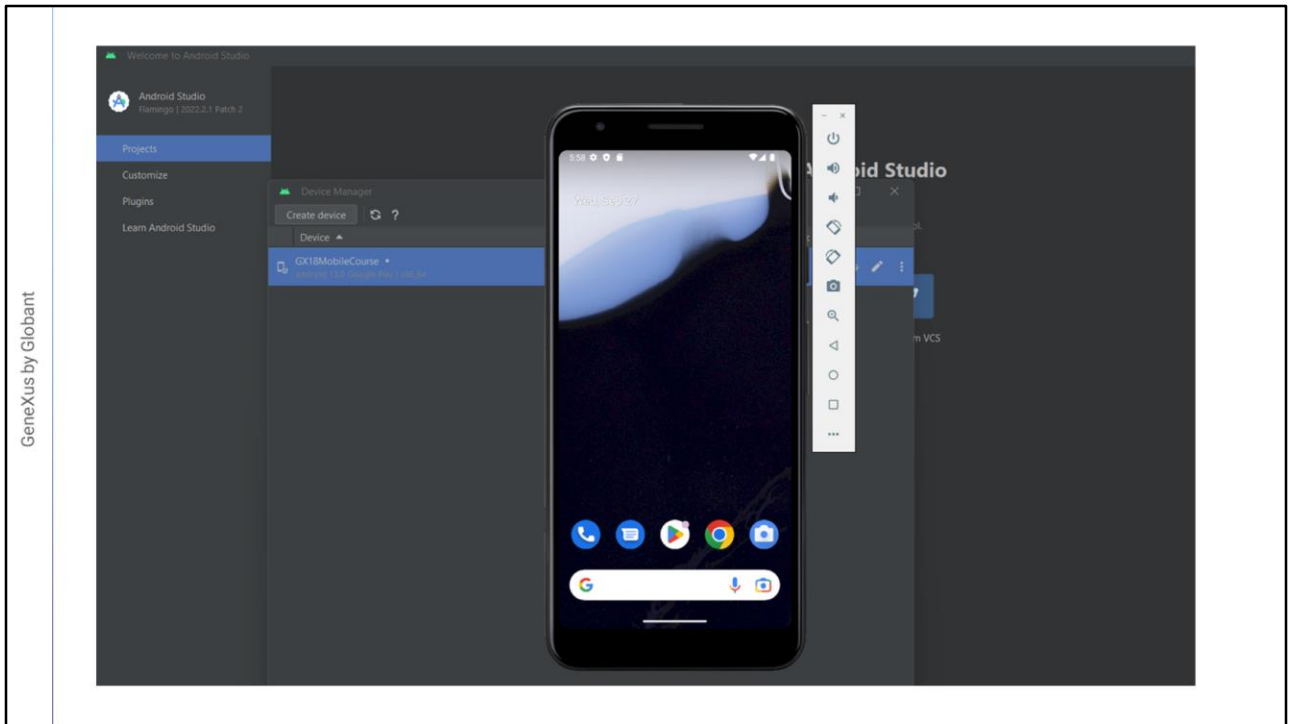


Pressionamos Criar dispositivo virtual. Escolhemos aquele chamado Pixel 3a e agora pressionamos Next.

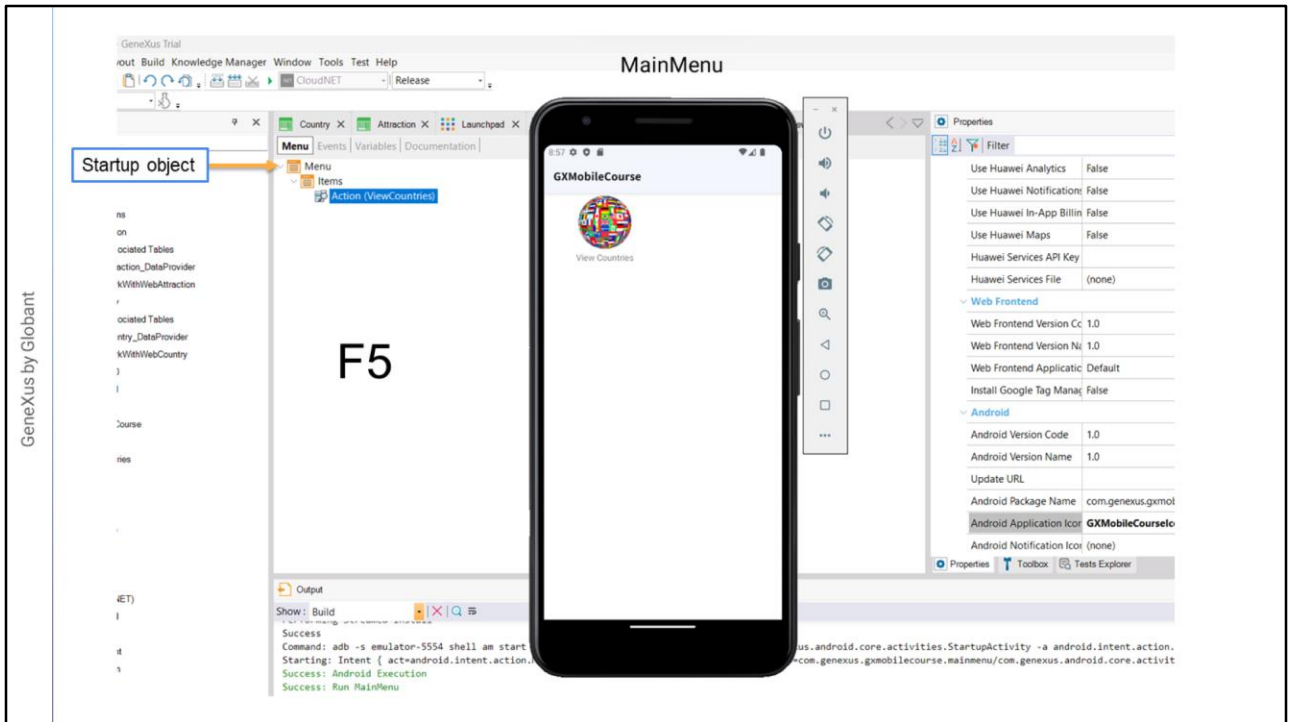
Selecionamos a linha correspondente à API 33 e pressionamos Next.

Damos o nome ao nosso Android Virtual Device, colocamos GXMobileCourse e pressionamos Finish.

Agora, para testá-lo, pressionamos o botão play...

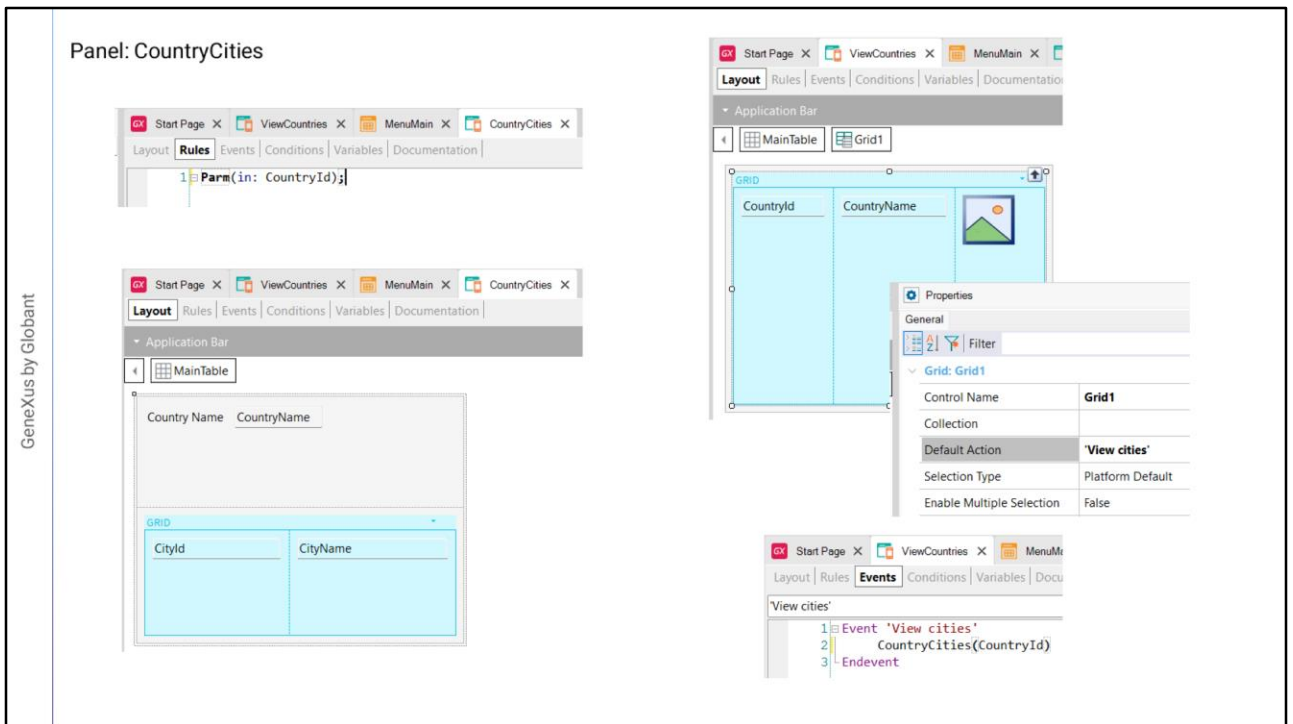


Vemos que após alguns segundos, é aberto o emulador com uma aparência idêntica a um dispositivo real.



Voltamos ao GeneXus. Como já havíamos definido o objeto MainMenu como startup object, ele será o objeto que será executado no emulador, então pressionamos F5.

Vemos na janela de Output que foi executado o objeto MainMenu e que no emulador aparece um ícone com o nome View Countries.

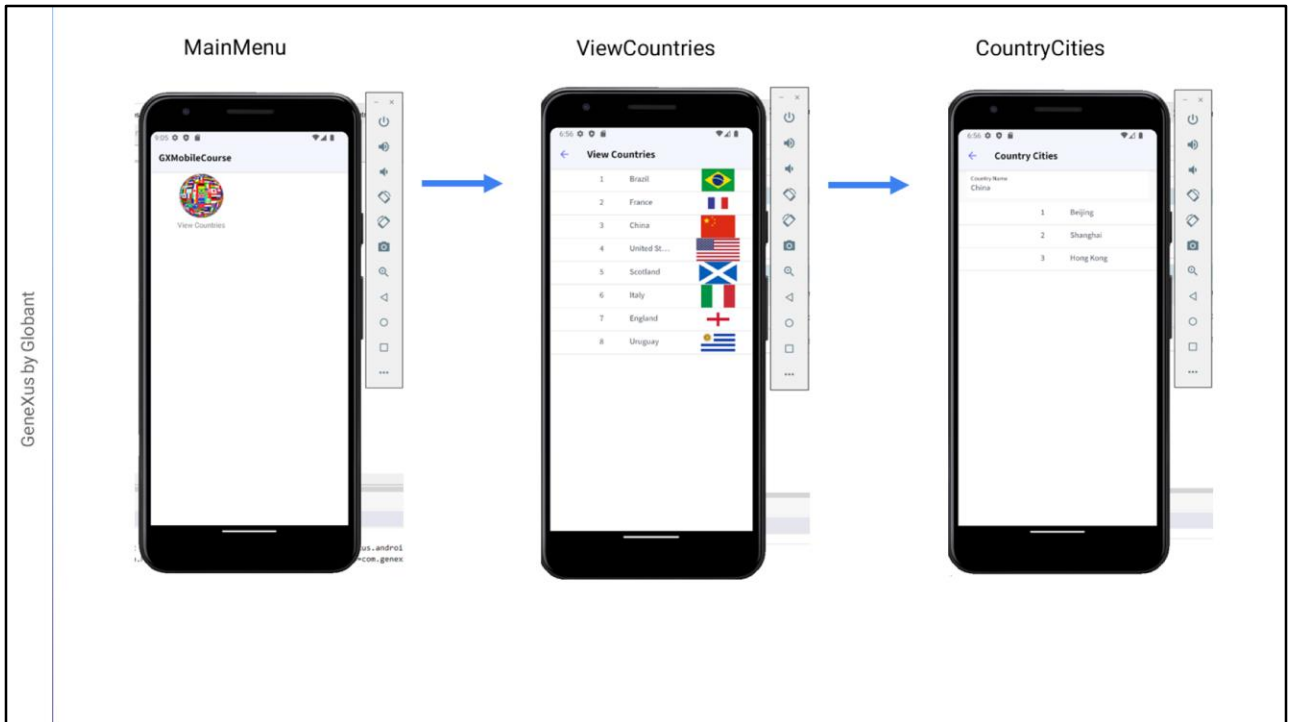


Então criamos o panel CountryCities e adicionamos nele uma regra Parm para que receba o atributo CountryId por parâmetro. Com isso garantimos que toda a informação do panel ficará automaticamente filtrada por este valor e mesmo que acessemos a tabela onde estão armazenadas todas as cidades, veremos unicamente as cidades do país que nos interessa.

Em seguida, arrastamos para o layout o atributo CountryName e um grid, que selecionamos para conter os atributos CityId e CityName. Salvamos e voltamos ao panel ViewCountries.

Selecionamos o grid e em sua propriedade Default Action escolhemos o valor <new> e escrevemos View cities. Agora abrimos a aba de eventos e no evento View Cities invocamos o panel das cidades escrevendo CountryCities e passando entre parênteses o identificador do país. Isto fará com que quando tocarmos sobre um país, seja aberto o panel com os dados de suas cidades.

Vamos testar isso executando F5.



Vemos que foi atualizada a tela do emulador e novamente vemos o ícone de View Countries. Tocamos para ver a lista de países e agora tocamos sobre China.

E vemos que nos são mostradas as cidades da China: Beijing, Shanghai e Hong Kong. Novamente, poderíamos melhorar a forma de ver esses dados.

Para voltar à tela anterior vemos que acima à esquerda nos aparece uma seta, embora o dispositivo Android nos forneça um botão específico para isso, mas nem todos os dispositivos fornecem um botão específico.

Mais adiante entraremos nesta e em outras considerações de funcionalidade definidas nos guias de cada plataforma.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com