

Introdução a Smart Devices

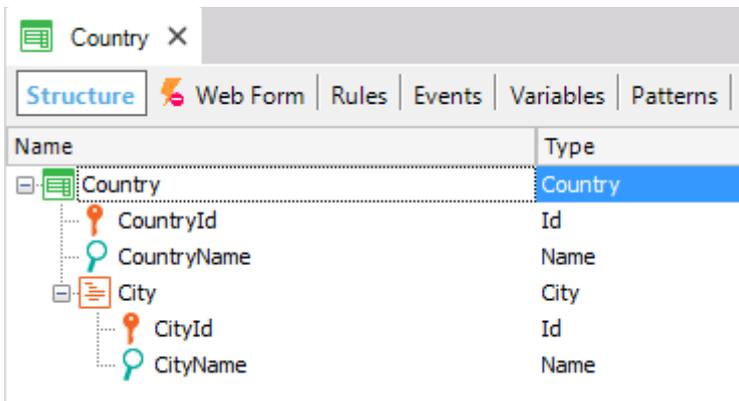
Até agora construímos todos os requisitos solicitados pela agência de viagens em relação à aplicação web a ser utilizada pelos usuários da própria agência.

Agora a agência quer oferecer também uma pequena aplicação para dispositivos inteligentes para ser utilizada pelos usuários finais.

O objetivo é que qualquer pessoa possa consultar a partir de seu Smart Device todas as cidades que a agência oferece para visitar, as excursões disponíveis, as principais atrações turísticas, etc.

Vamos então ao GeneXus, e repassemos nosso projeto de transações.

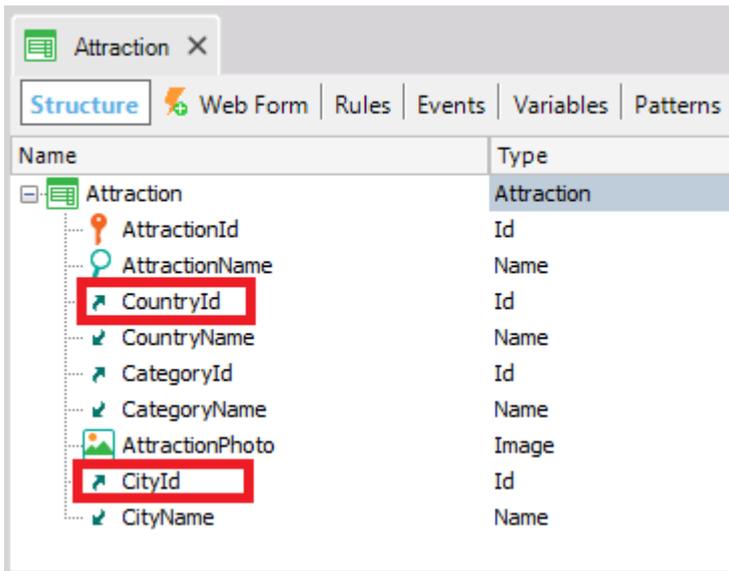
Temos a transação Country, com seu conjunto de cidades próprias:



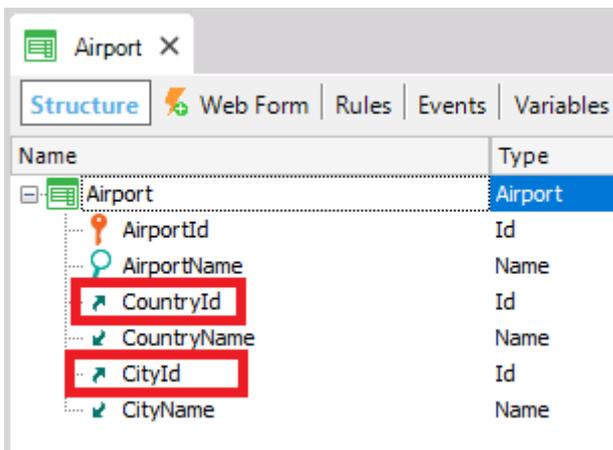
The screenshot shows the GeneXus Structure window for the 'Country' transaction. The window has tabs for 'Structure', 'Web Form', 'Rules', 'Events', 'Variables', and 'Patterns'. The 'Structure' tab is active, displaying a tree view of the transaction structure. The 'Country' transaction is expanded, showing its attributes: 'CountryId' (Id), 'CountryName' (Name), and 'City' (City). The 'City' attribute is further expanded, showing its attributes: 'CityId' (Id) and 'CityName' (Name).

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

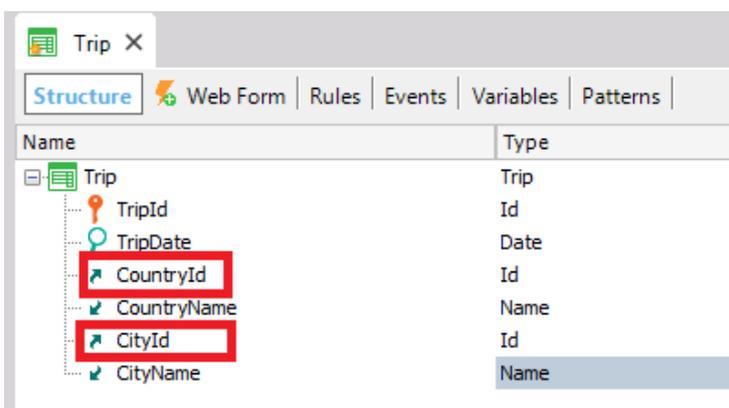
A transação Attraction, que tem os atributos CountryId e CityId como chaves estrangeiras:



A transação Airport que também tem CountryId e CityId como chaves estrangeiras:



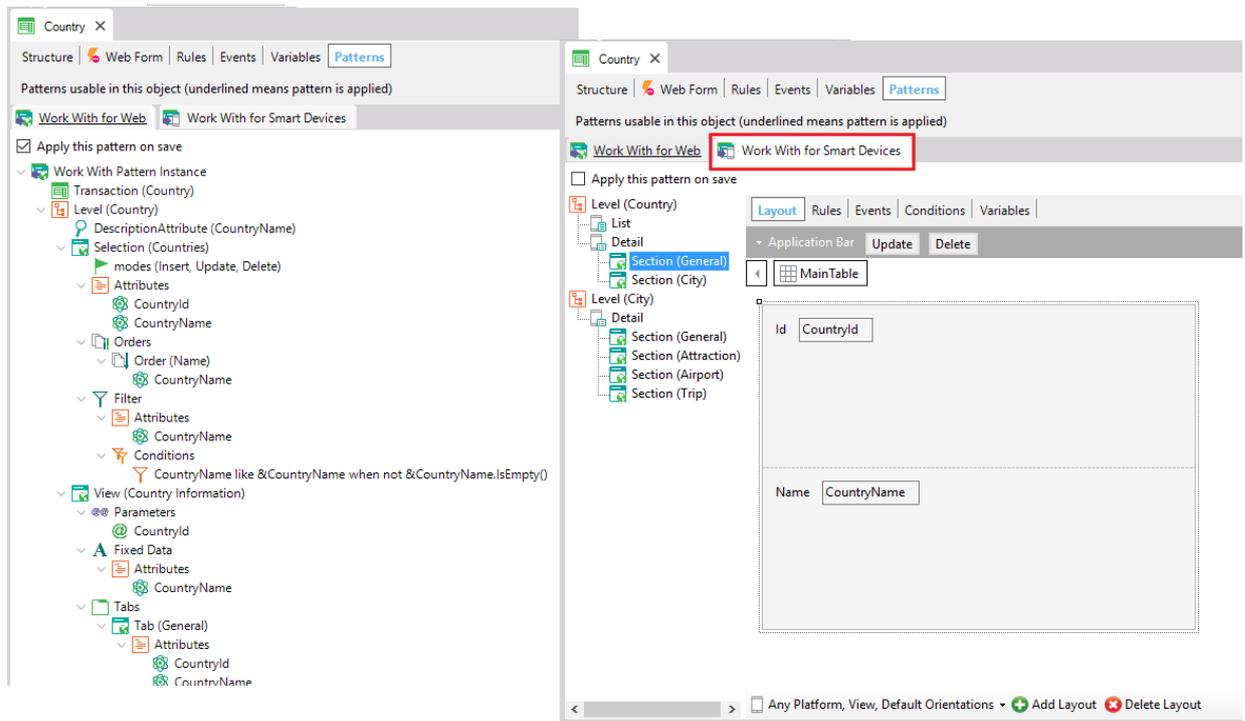
E a transação Trip que também tem CountryId e CityId como chaves estrangeiras:



Bem. Recordemos que nos vídeos anteriores aprendemos sobre o pattern Work With “for web” e o temos aplicado a algumas transações.

Vamos então para a transação Country, selecionemos a seção: Patterns.

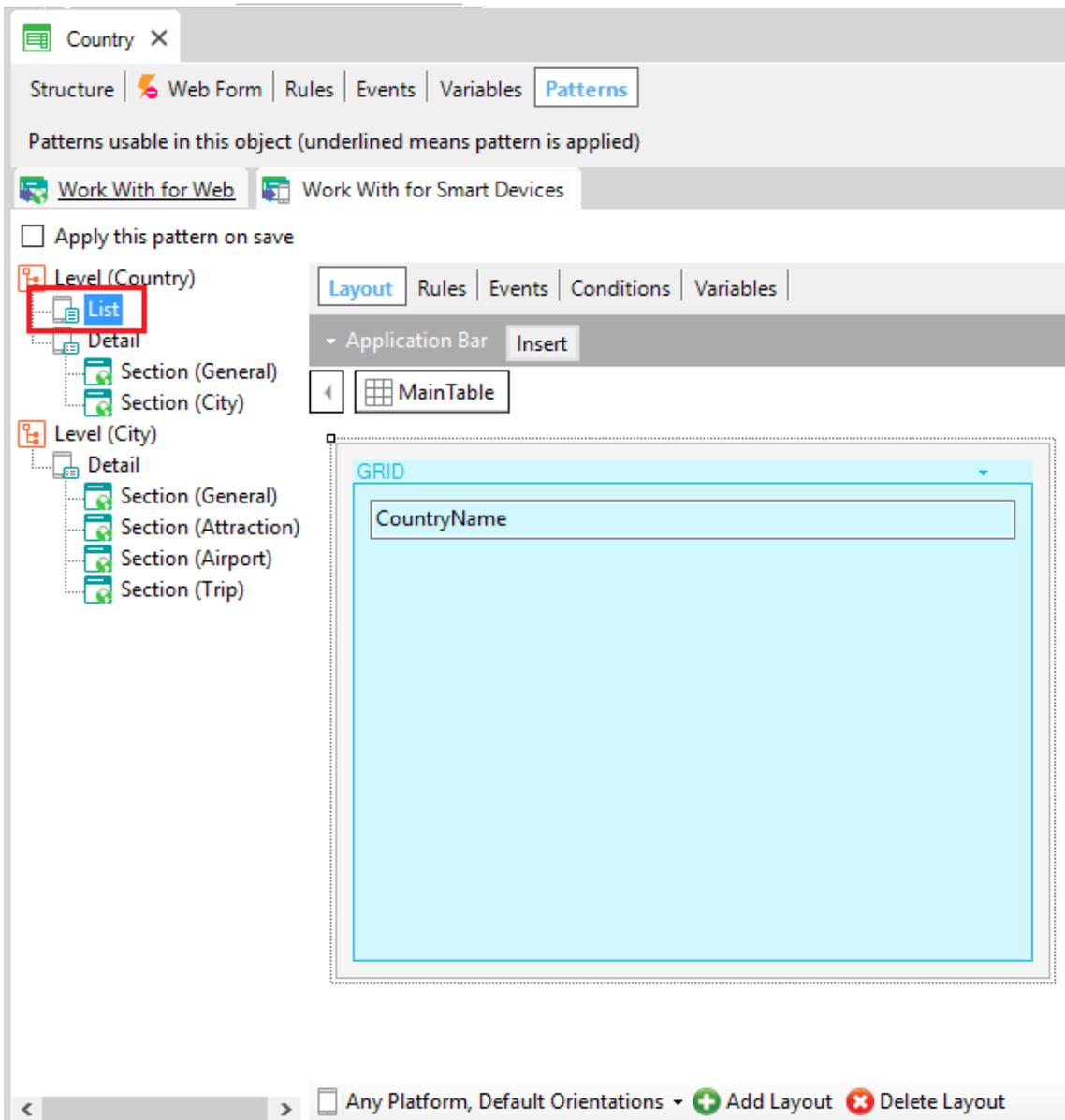
E observemos que, além da aplicação do pattern Work With para web, GeneXus também oferece a aplicação do mesmo padrão, mas para Smart Devices.



Selecionemos Work With for Smart Devices...

E vamos observar os nós da árvore da esquerda, e o que vemos à direita.

Abaixo do nó correspondente ao nível Country, aparece o nó List. Se clicarmos nele, veremos que corresponde a uma grid que tem inserido o atributo CountryName.



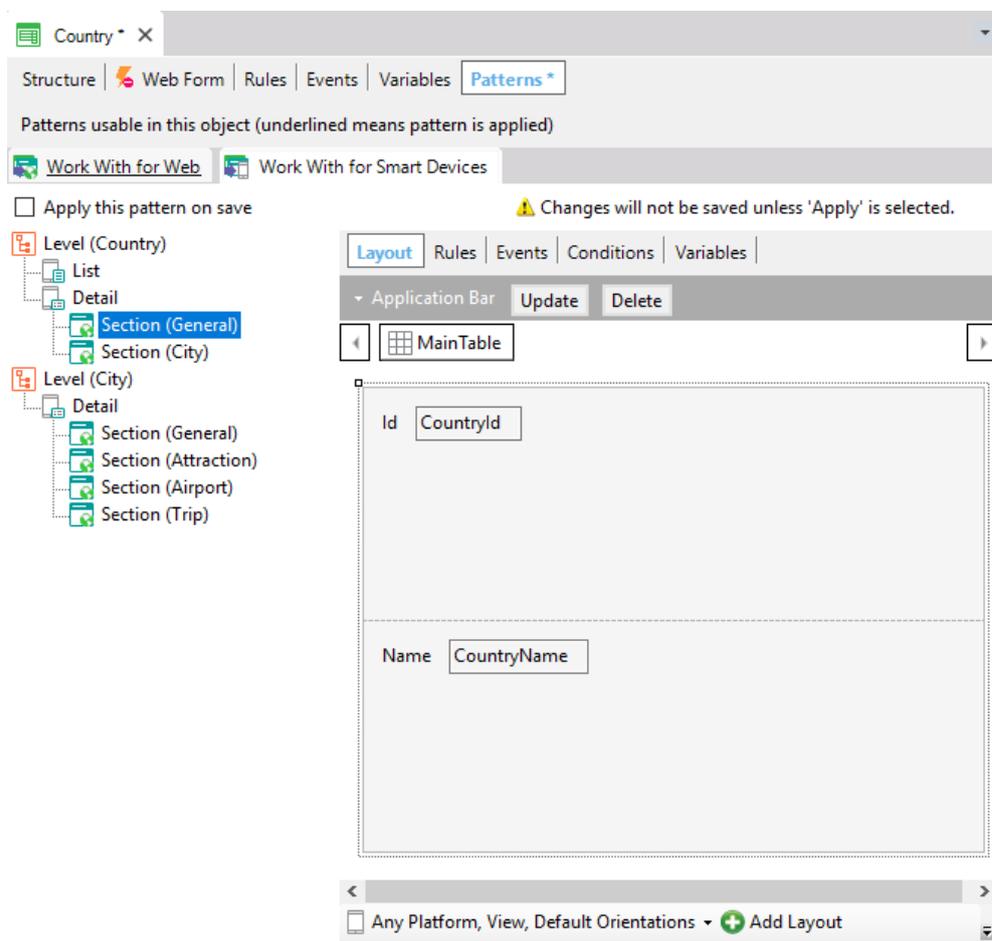
O nó denominado "List" nesse padrão corresponde ao grid que listará os dados para cada registro no painel a ser gerado. Conseqüentemente o grid no painel a ser gerado mostrará a lista de nomes dos países registrados.

Vimos que neste caso, então, já é mostrado o Layout para o nó List, em vez de ver a lista de atributos a serem incluídos no grid, abaixo do nó, como no caso do pattern Work With for Web.

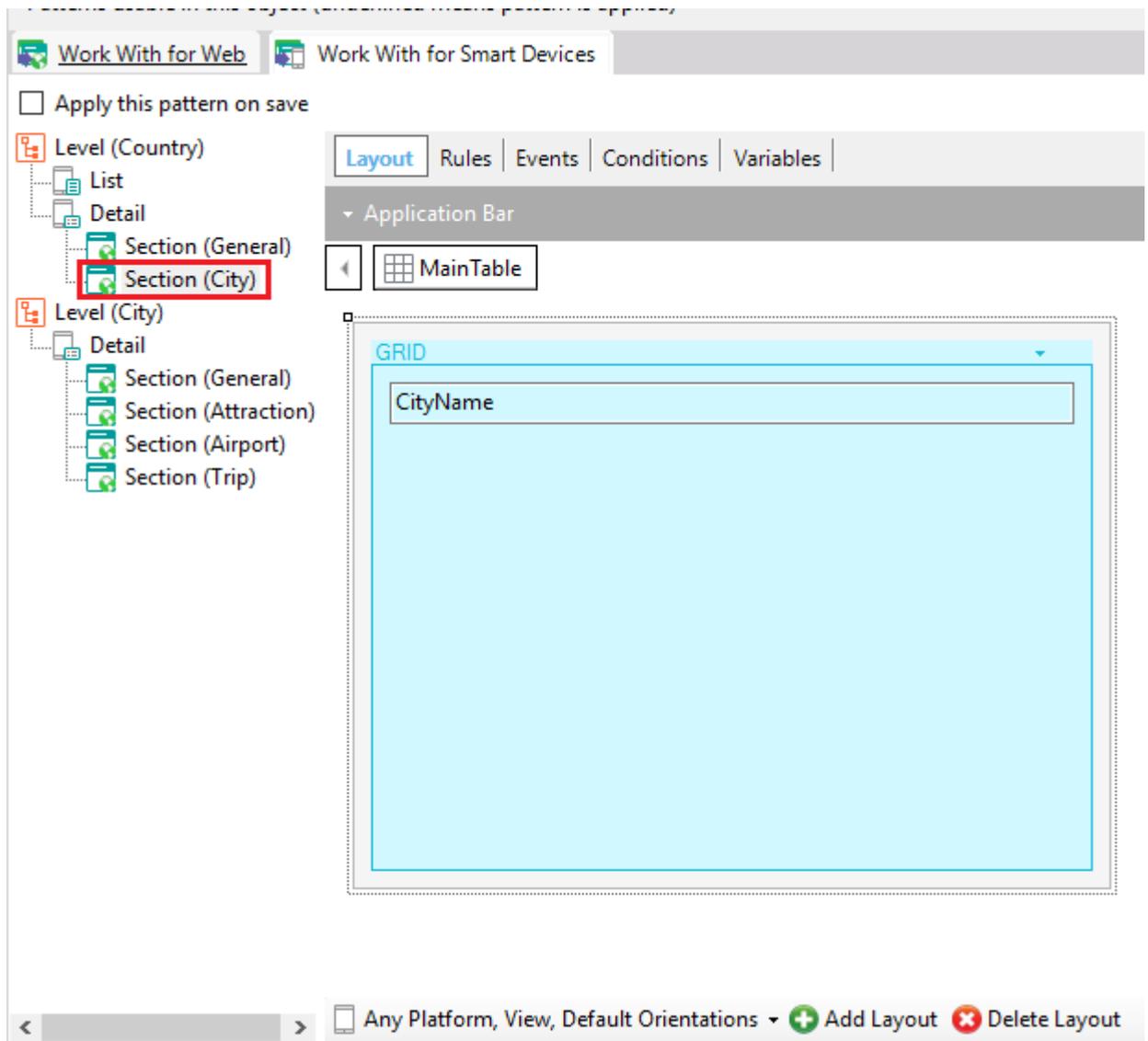
Agora vamos observar o nó Detail. Podemos associar o termo Detail ao que no Work With for web conhecíamos por "View". Isto é, para ver os detalhes de um determinado registro da lista.

O nó Detail é composto de duas seções: General e City.

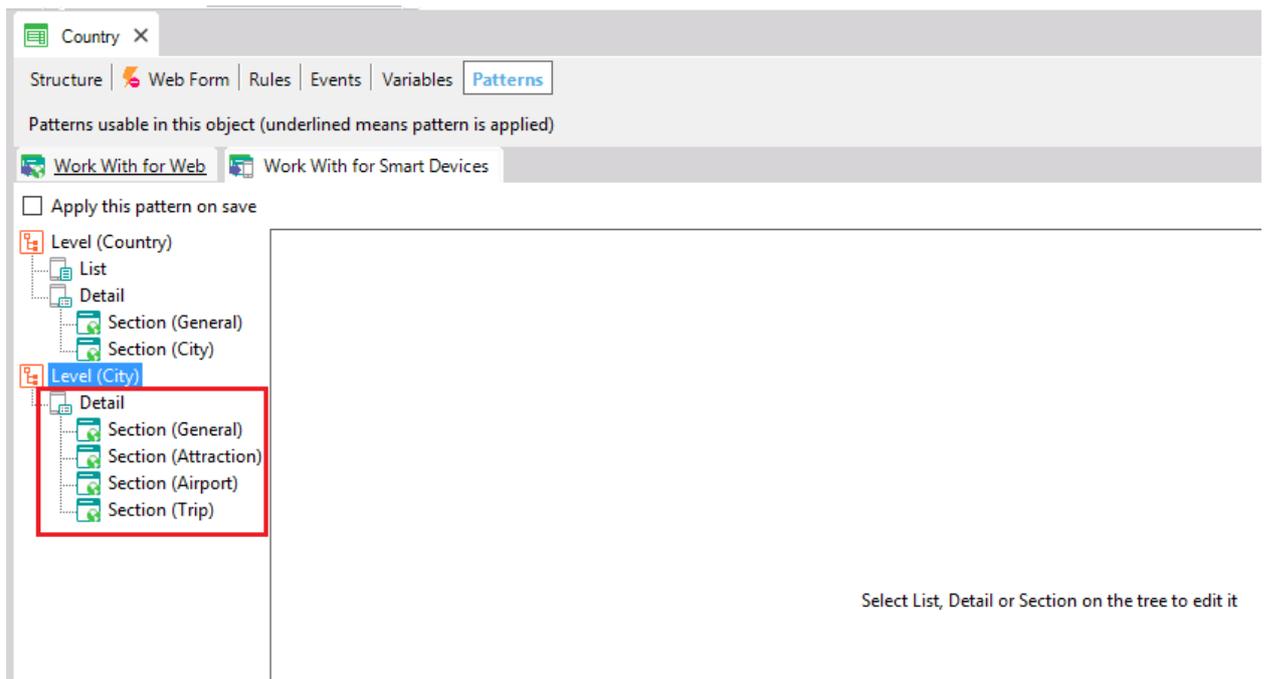
Similar à funcionalidade implementada pelo padrão Work With for Web, a seção General mostrará os dados associados ao país.



... e a seção City mostrará todas as suas cidades

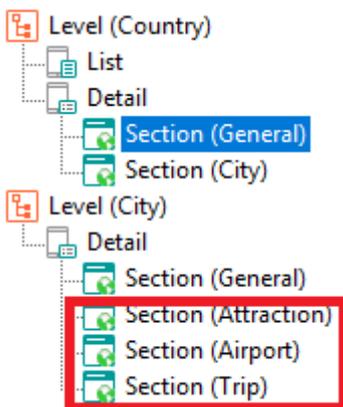


Agora observemos este nó: Level(City):

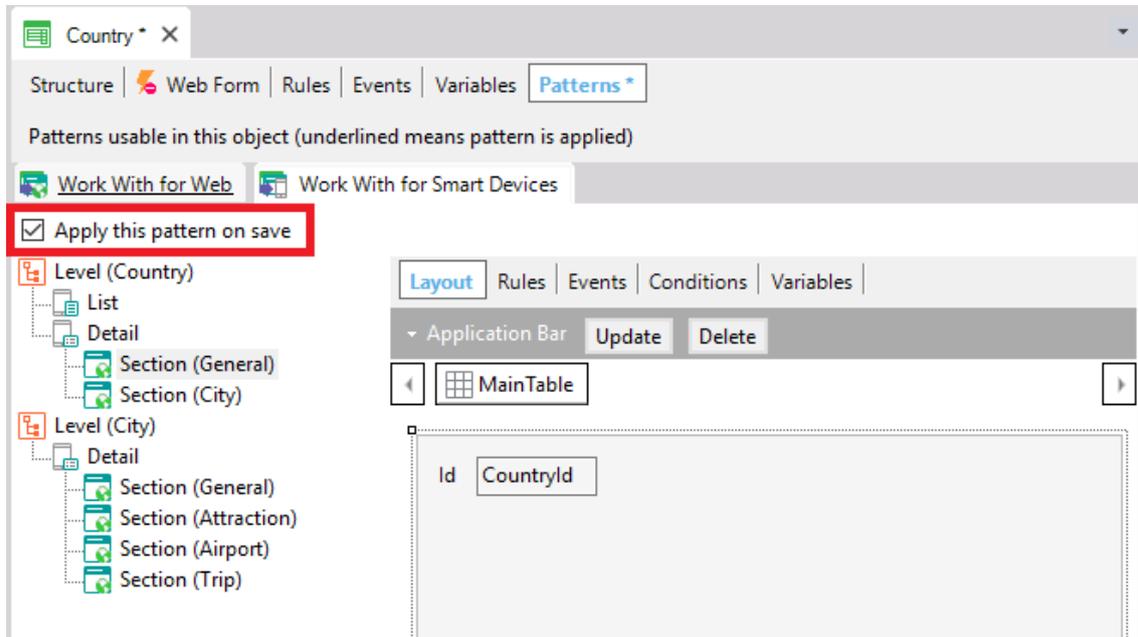


Dado que a transação Country tem dois níveis, assim como vimos recentemente o nó que corresponde ao primeiro nível, este nó corresponde ao segundo nível.

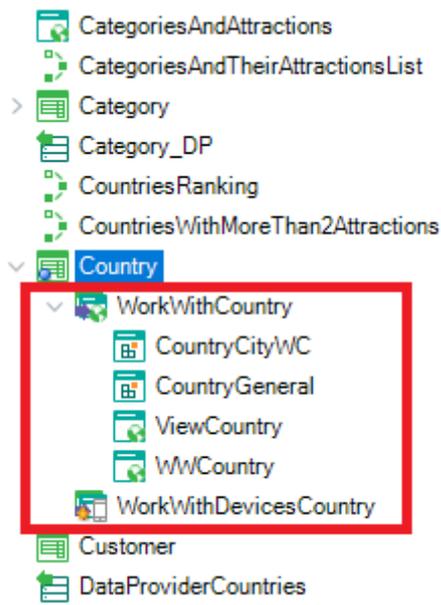
Abaixo do mesmo, vemos a seção General, tal como está aqui também... e em seguida há uma seção para cada relação 1-N que têm as cidades. Isto é, cada cidade tem N Atrações, N aeroportos e N excursões:



Vamos aplicar este padrão para ver em execução o que ele gera e oferece. Só temos que clicar na caixa Apply this pattern on save... (tal como fizemos para aplicar o pattern Work With for Web)



E observemos que abaixo da transação Country, vemos agora também o objeto WorkWithDevicesCountry:

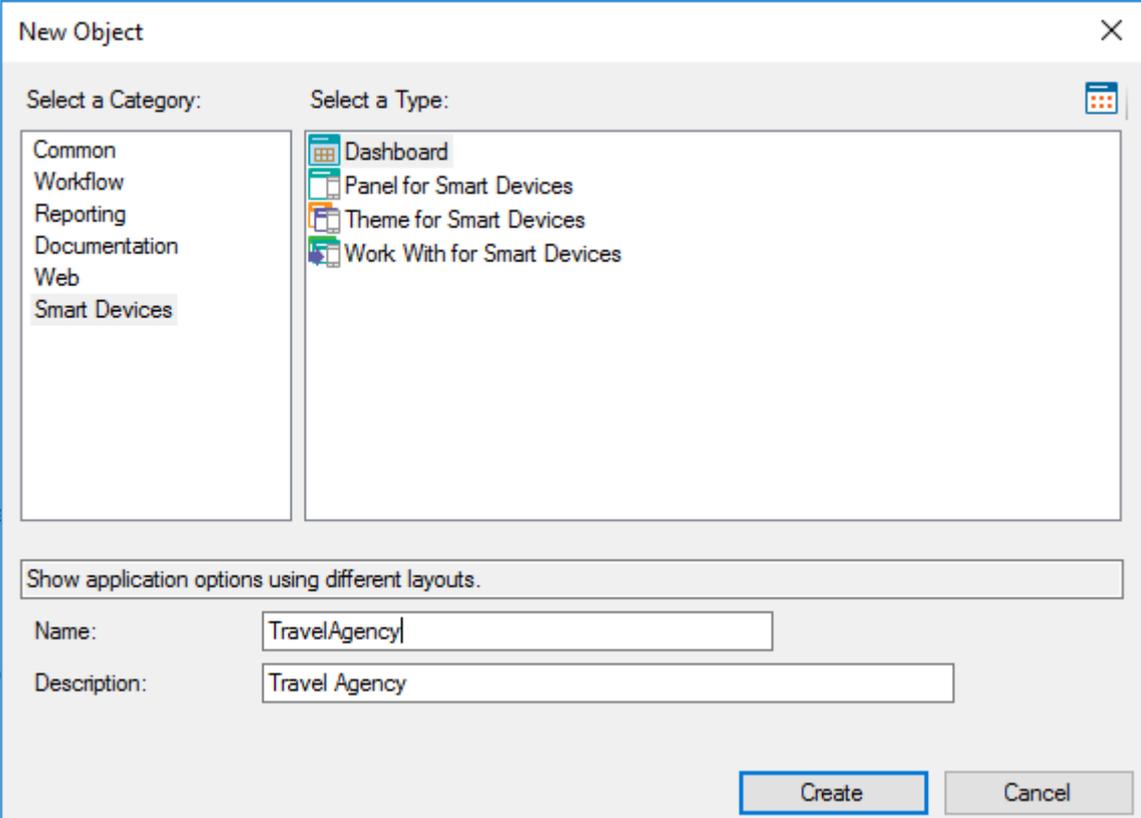


Diferente de quando aplicamos o padrão para web, aqui não vemos um conjunto de objetos gerados, mas apenas o objeto WorkWithDevicesCountry.

Agora temos que criar um menu de acesso que estará no dispositivo para que este menu acesse a funcionalidade gerada pelo padrão. Para isso, criaremos um objeto GeneXus do tipo Dashboard.

Objetos Dashboard, então, nos permitem criar menus de acesso que estejam em conformidade com nossa aplicação para Smart Devices.

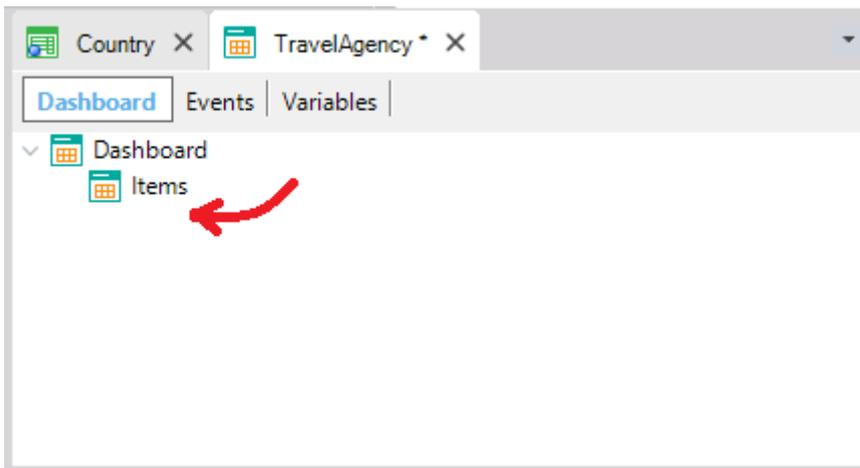
Selecionamos New object e dentro da categoria Smart Devices, selecionamos: Dashboard. Chamamos: TravelAgency...



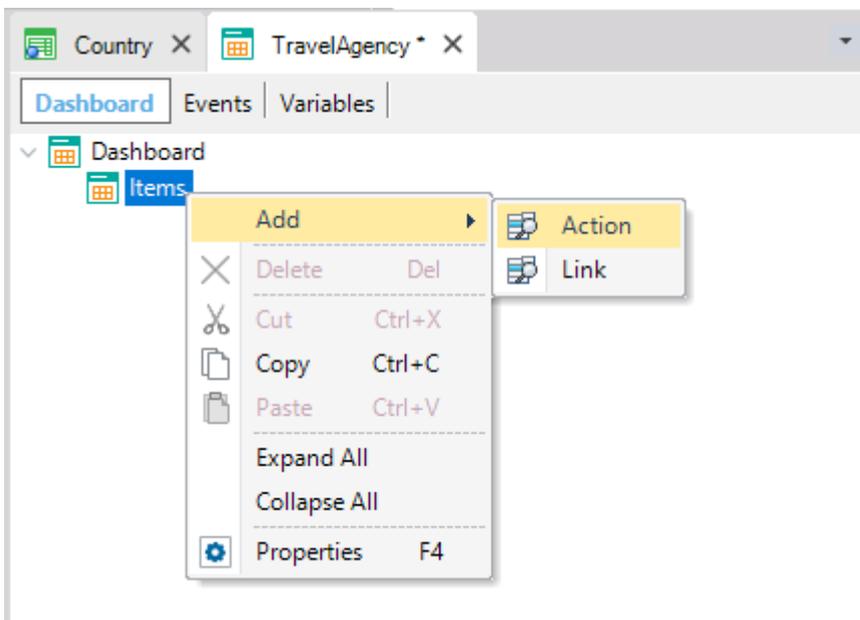
The screenshot shows a 'New Object' dialog box with the following details:

- Select a Category:** Common, Workflow, Reporting, Documentation, Web, **Smart Devices** (selected)
- Select a Type:** **Dashboard** (selected), Panel for Smart Devices, Theme for Smart Devices, Work With for Smart Devices
- Show application options using different layouts:** (checkbox)
- Name:** TravelAgency
- Description:** Travel Agency
- Buttons:** Create, Cancel

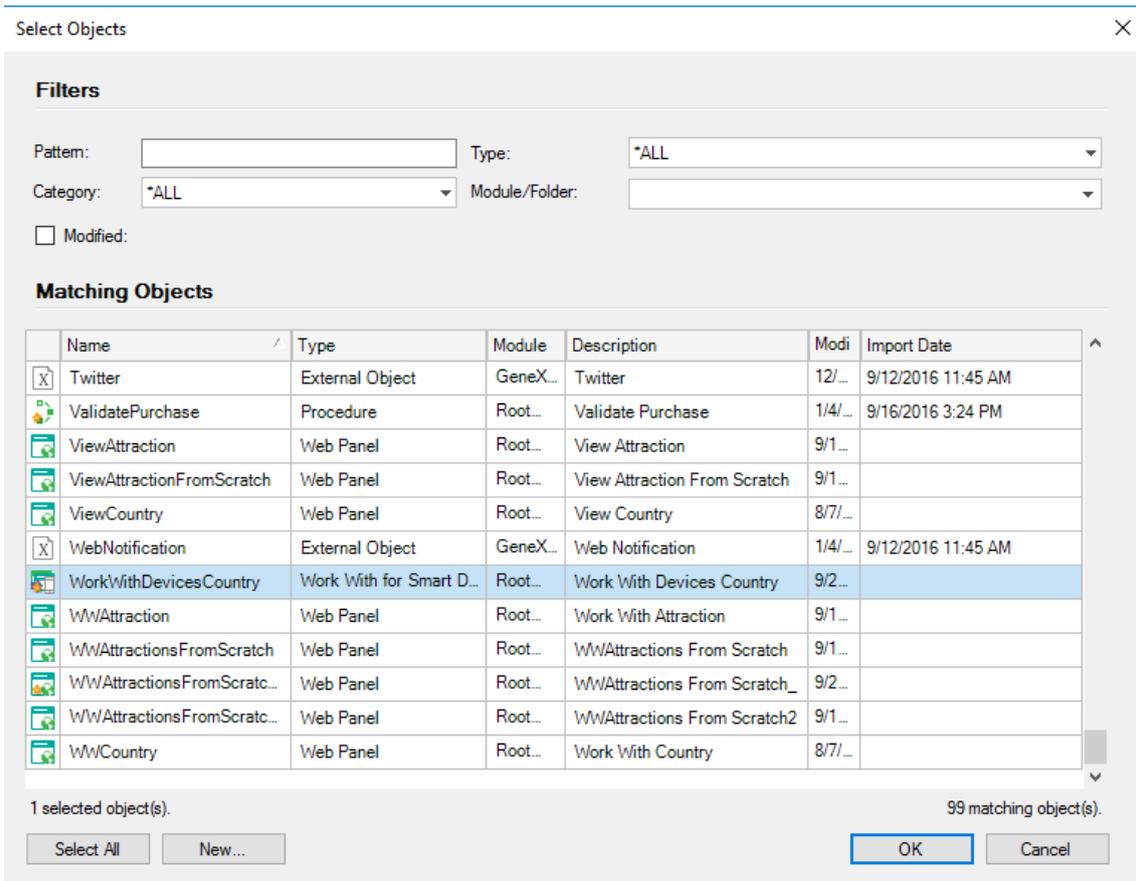
Aqui se adicionam as opções que irão compor o menu:



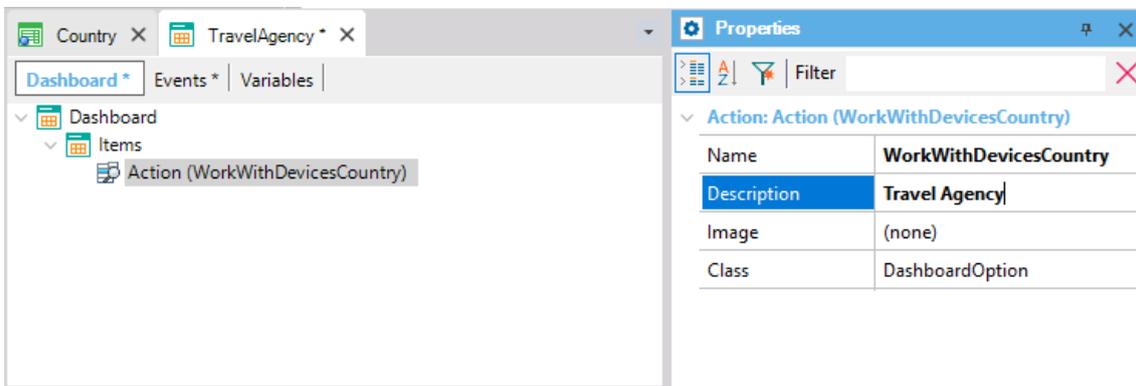
Pressionamos o botão direito do mouse no nó Item e selecionamos Add / Action:



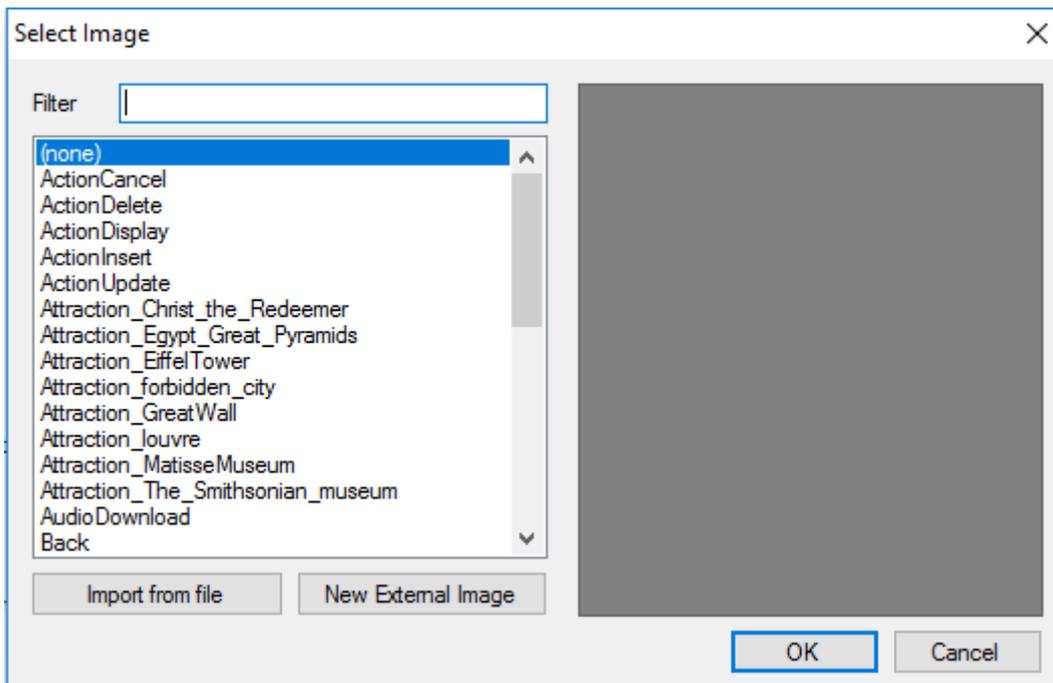
Na janela escolhemos o objeto WorkWithDevicesCountry...



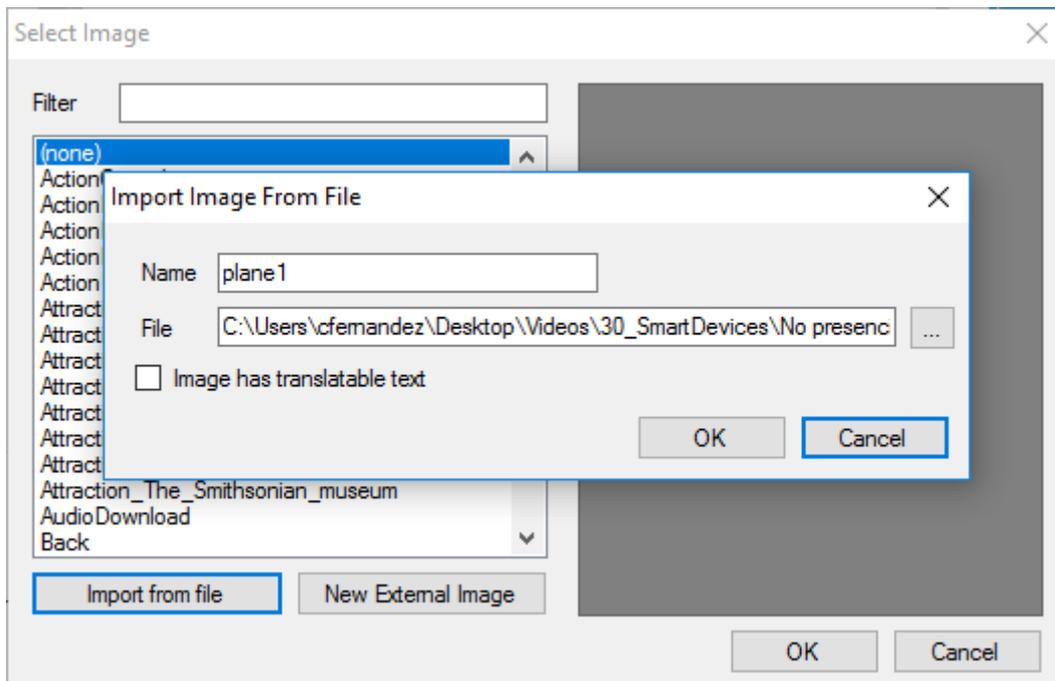
Nas propriedades do objeto Dashboard colocamos a descrição TravelAgency:

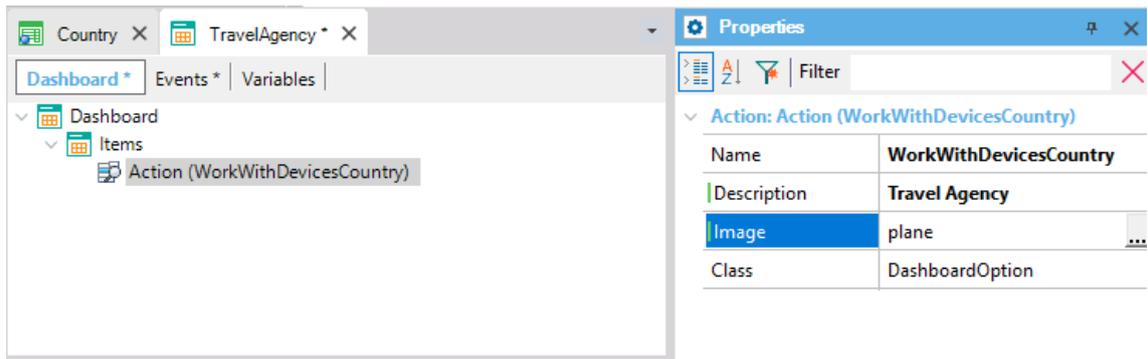


e podemos também selecionar uma imagem...



Se não gostamos de nenhuma das imagens que temos por padrão dentro de nossa base de conhecimento, podemos selecionar uma imagem a partir de um arquivo e importá-la...





Bem. Agora pelo fato de termos criado dentro de nossa base de conhecimento, objetos próprios para Smart Devices, veremos que ao pressionar F5 automaticamente se executará o emulador para o Android... podendo também acessar o nossa aplicação web a partir do menu Developer...

Pressionemos F5...

Vemos que agora o Developer menu mostra, além das opções de acesso à aplicação web, um código QR gerado automaticamente para poder fazer o download da aplicação em um dispositivo inteligente...

Browse Web Objects

 Airline	 Airport	 AttractionsPerCategory
 AttractionsWithoutParameters	 CategoriesAndAttractions	 CategoriesAndTheirAttractionsList
 Category	 CountriesWithMoreThan2Attractions	 Customer
 EnterAttractionsFilter	 Flight	 Home
 MassiveInsertRemove	 NumberOfAttractionsByCountry	 PrintRanking
 Supplier	 WWAttraction	 WWCountry

Install iOS Apps



QRCode for iOS
[Services URL for KBN](#)



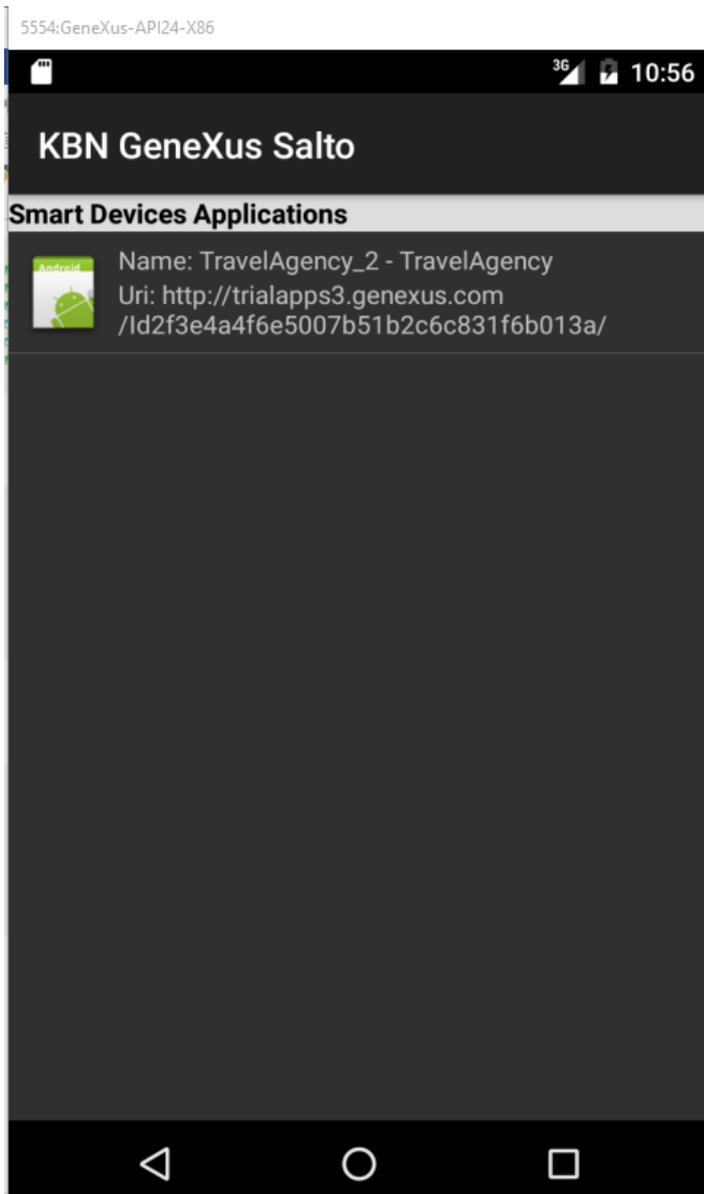
Knowledge Base Navigator
The KBN app makes native iOS online apps prototyping simple. [Download](#)

Install Android Apps

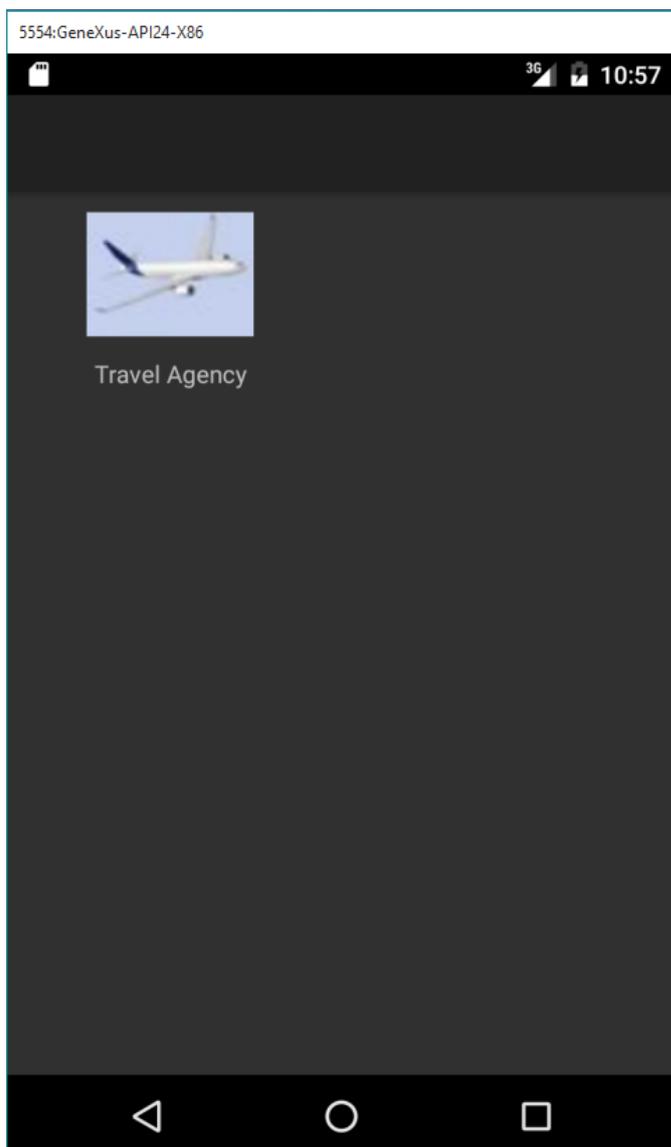


TravelAgency
[Download](#)

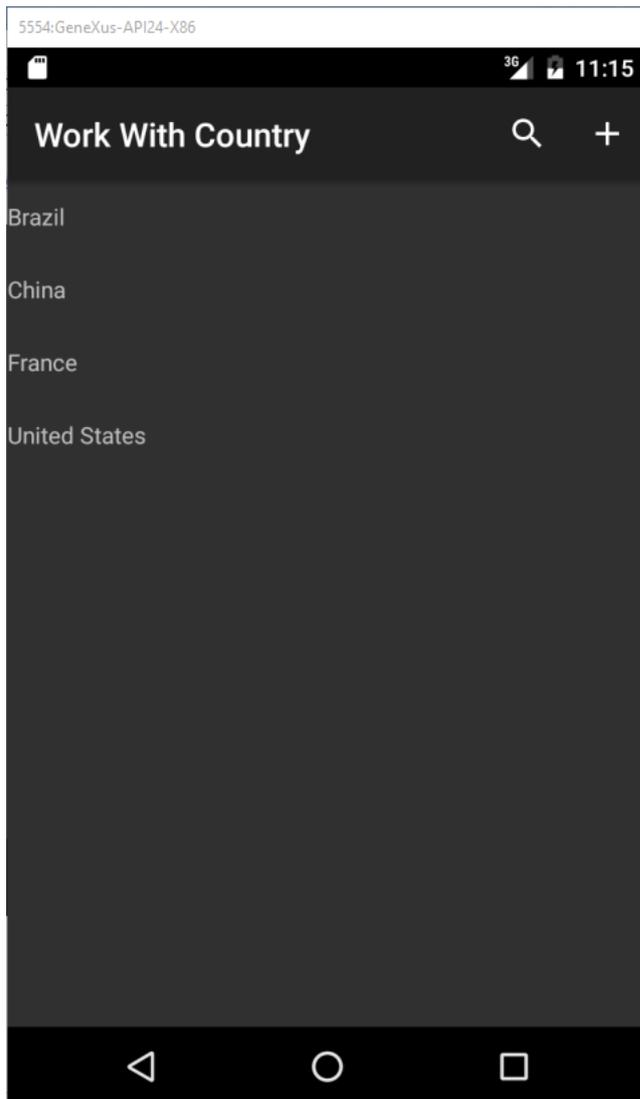
Deixemos subir agora o emulador do Android...



Clique aqui. E assim se vê o Dashboard em execução com a imagem determinada...

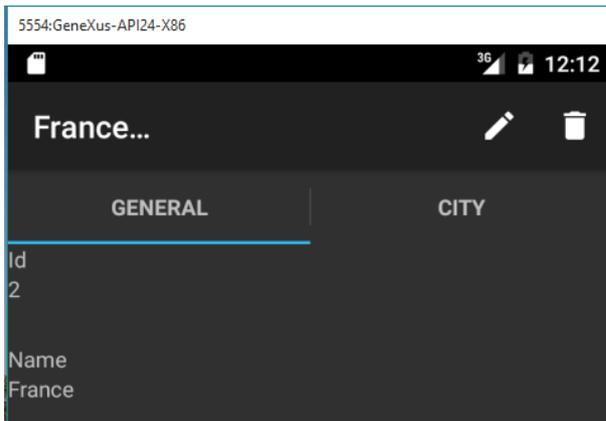


a selecionamos... e vemos todos os países registrados...

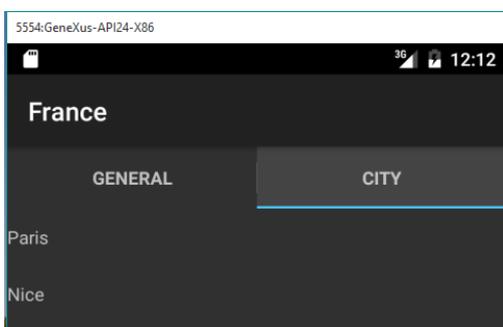


... ou o que vimos como List.

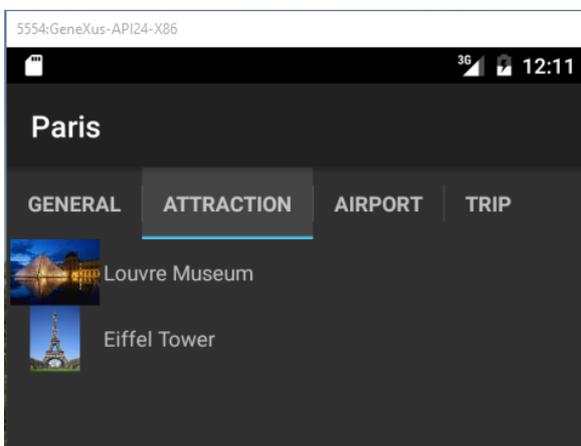
Escolhemos por exemplo France... vemos sua informação geral:



E suas cidades:



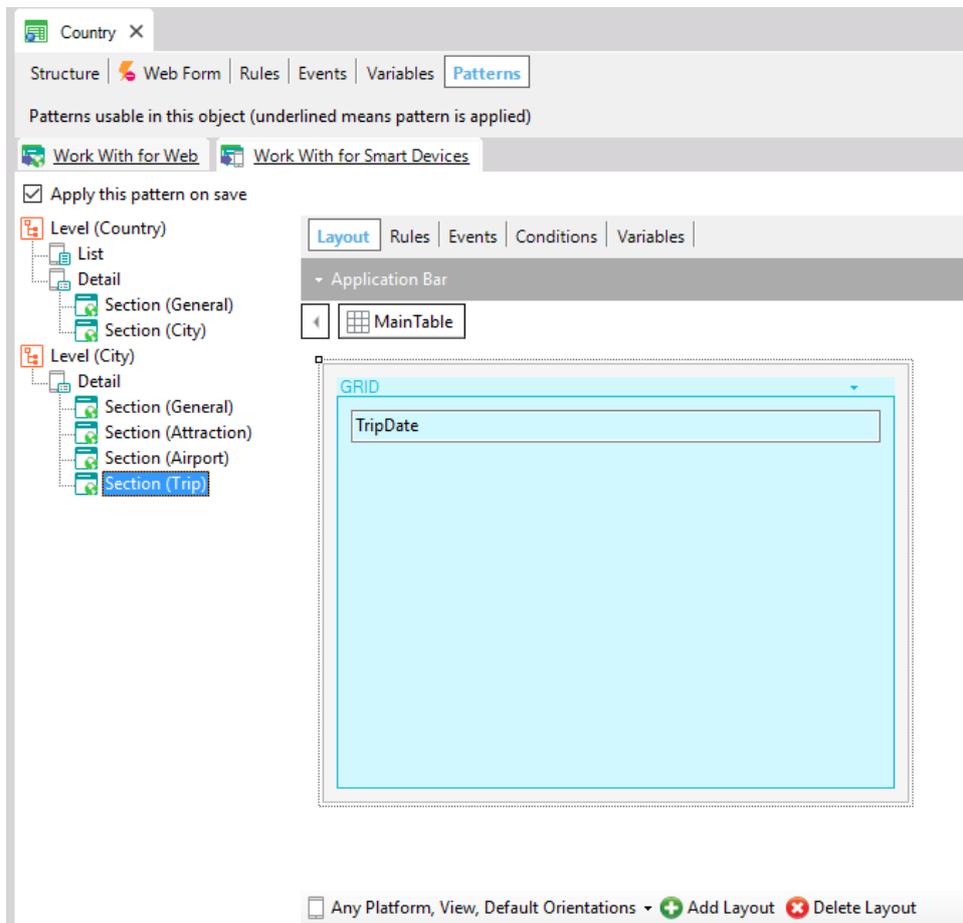
Selecionamos Paris.... E aqui vemos a seção geral... e sua informação relacionada... suas atrações, aeroportos... e excursões:



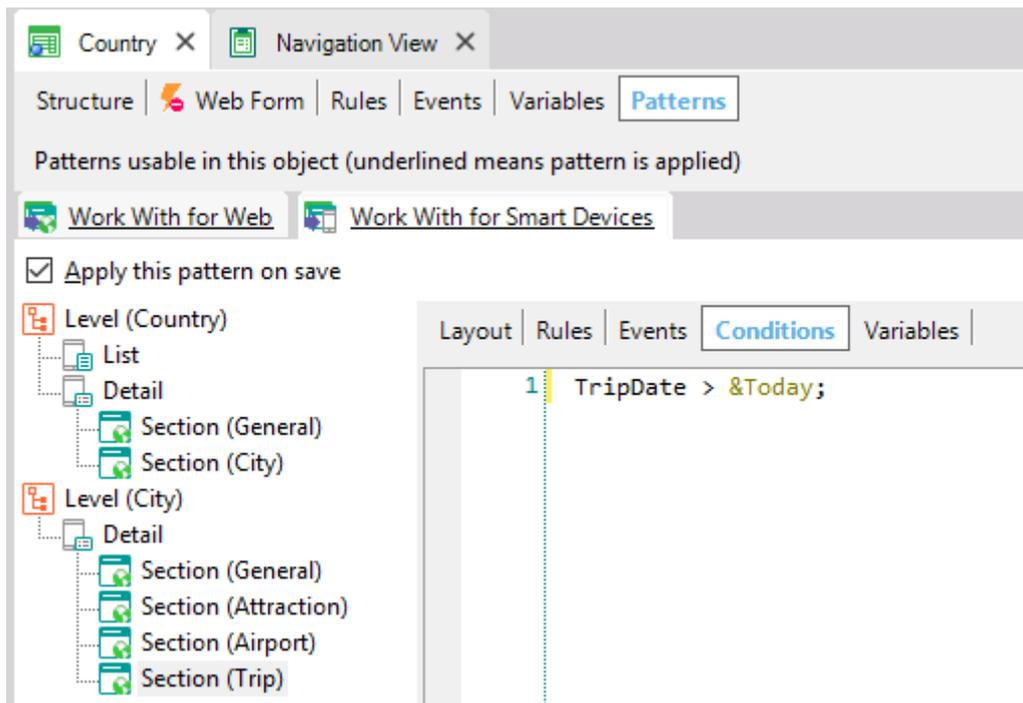
Voltemos para GeneXus para personalizar um pouco a aplicação do padrão...

Por exemplo, suponhamos que queremos ver apenas as excursões com data posterior à atual, ou seja, as excursões a serem realizadas...

Então, clicamos na seção Trip:



E nas Conditions colocamos: TripDate >&Today;



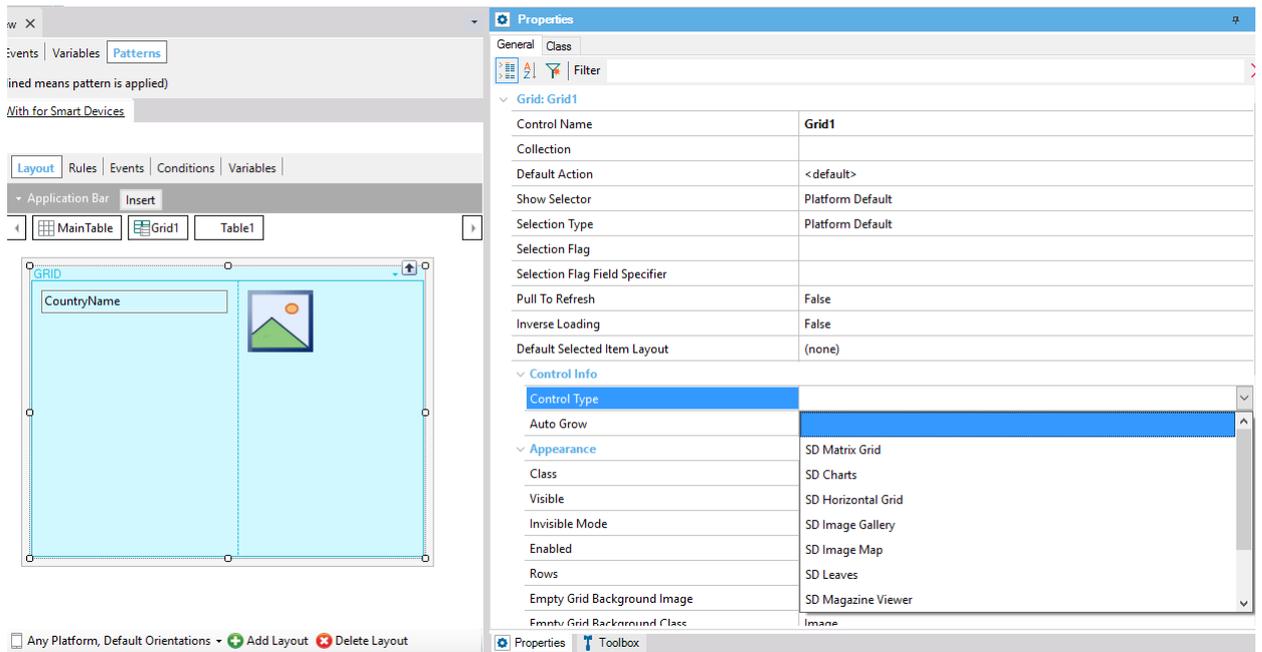
Bem. Propomos agora que a lista de países seja vista de uma forma mais atrativa, não apenas como uma lista de nomes.

Poderia ser por exemplo, que sejam vistas as bandeiras, como uma galeria de imagens, cada uma com o nome do país, e ao selecionar uma bandeira seja acessada a informação relacionada.

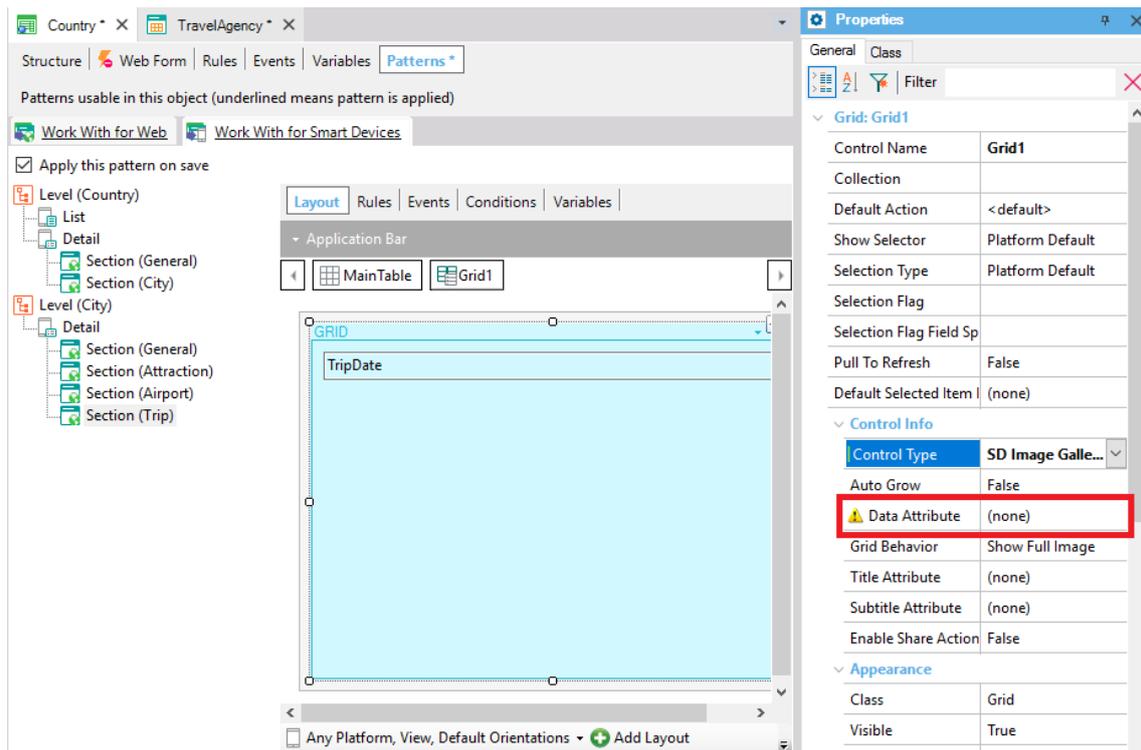
Definimos então na transação Country, um novo atributo: CountryFlag do tipo Image... e já fizemos a reorganização física e a carga de imagens, para ganhar tempo.

Voltemos agora ao padrão Work With for Smart Devices de Country, para definir agora que a lista de países seja exibida como uma galeria de bandeiras:

Na Propriedade do Control Type do grid, selecionamos SD Image Gallery:



Na propriedade Data Attribute:

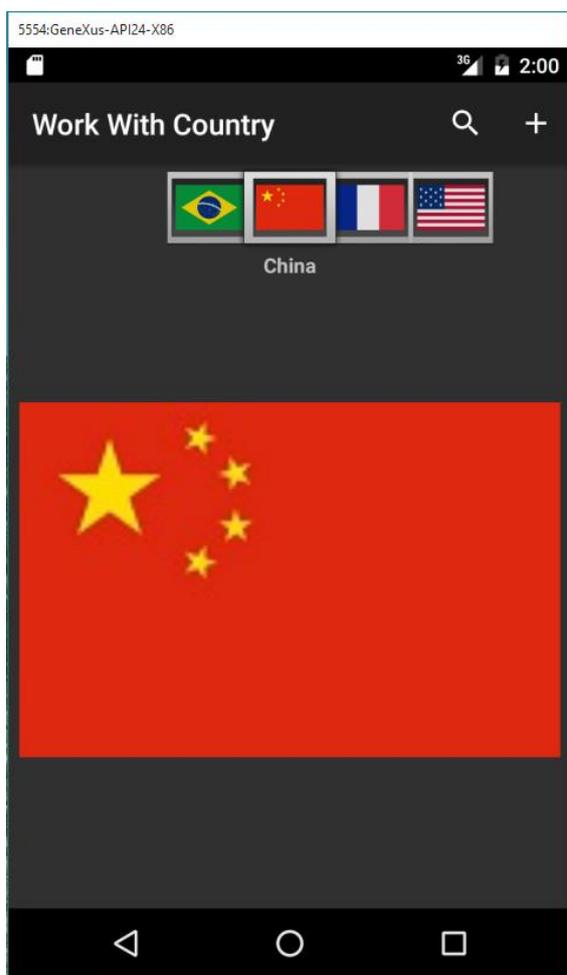


...indicamos o atributo que contém a imagem a ser exibida, neste caso é: CountryFlag e, em seguida, na propriedade: Title Attribute podemos indicar um atributo cujo valor seja o título da imagem. Seleccionamos CountryName.

Neste ponto, vejamos todas essas modificações em execução.

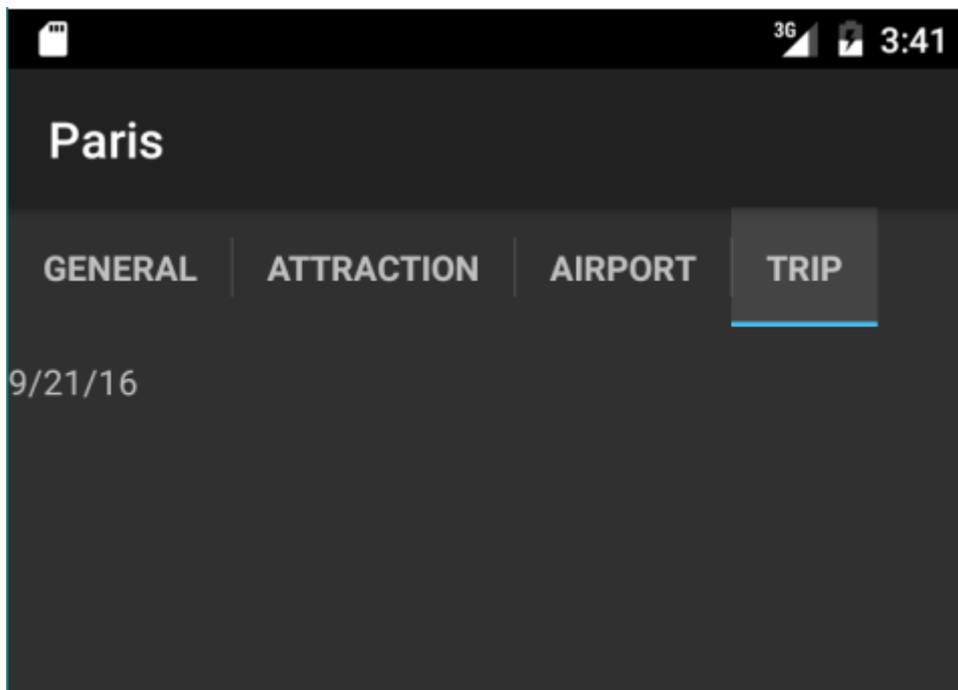
Pressionemos F5 novamente:

Abra o emulador, seleccionamos a URL da KB e vemos nossa imagem da aplicação, que está associada com o dashboard... e agora vemos a lista de países como uma galeria de bandeiras:



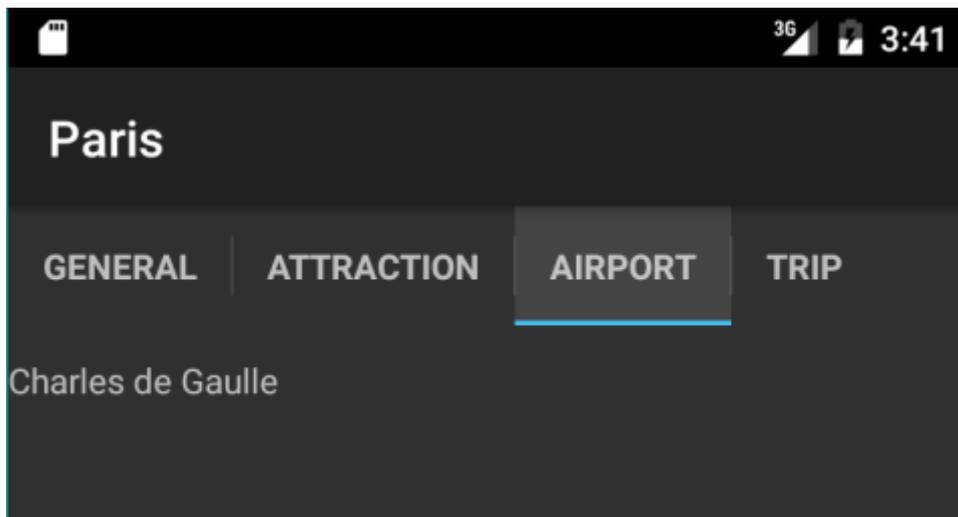
Selecionamos França, e vemos sua informação.

Escolhemos Paris... e para Paris, sua lista de excursões.



E as excursões que são exibidas têm efetivamente data posterior à atual.

Vamos agora aos aeroportos... vê-se apenas o nome, mas seria útil para os usuários da aplicação poder ver também o mapa com a localização dos aeroportos.

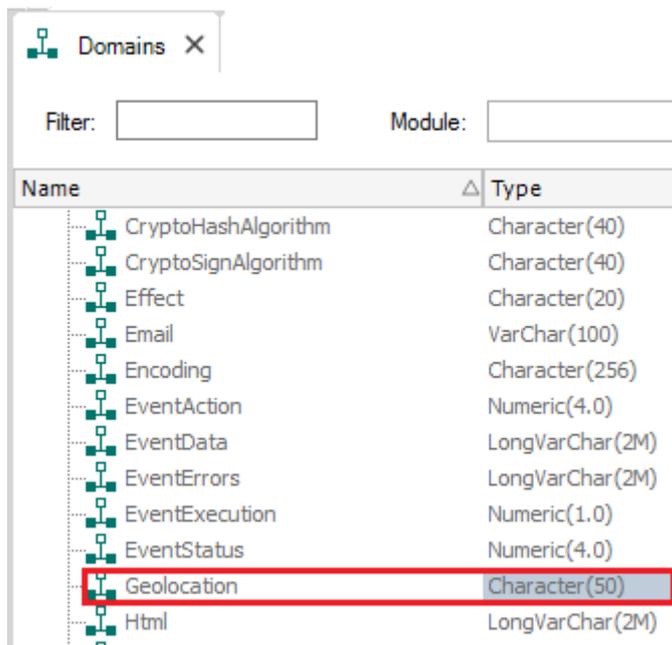


Voltemos ao GeneXus...

Vamos definir um novo atributo na transação Airport que nos permita armazenar a informação que precisamos.

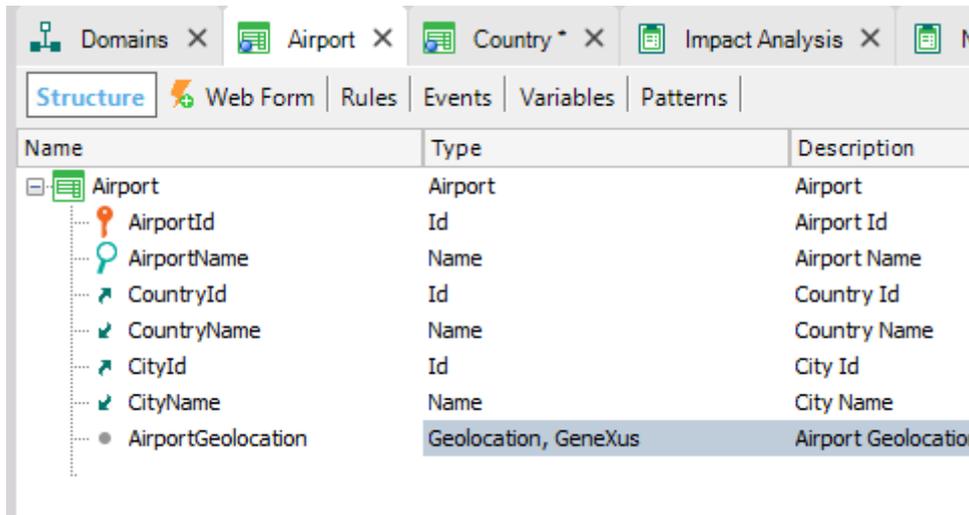
Esta informação deve ser baseada em algum tipo de dado que permita que sejam geradas as informações geográficas necessárias para que no dispositivo se visualize o mapa com a localização.

Para isso GeneXus nos oferece o domínio Geolocation:



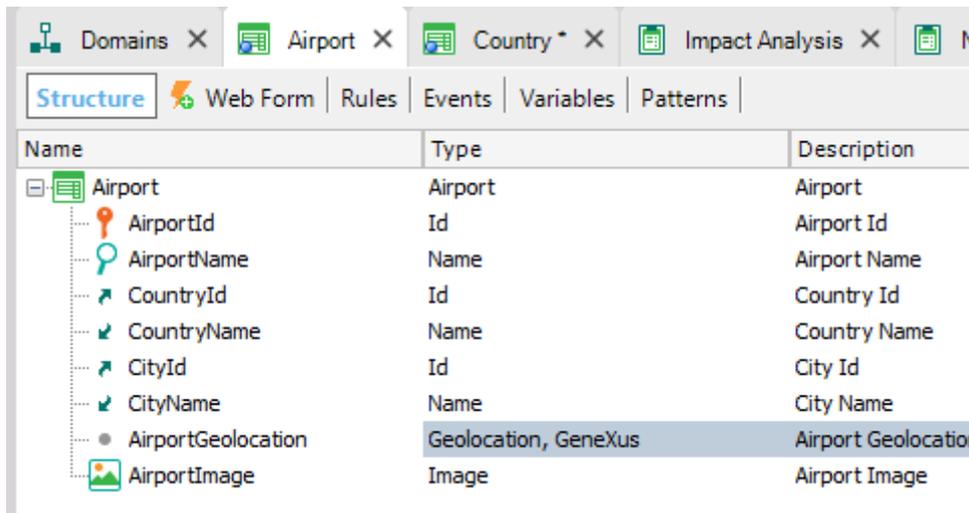
Trata-se de um tipo de dado que permite gerenciar a informação geográfica em termos de latitude e longitude.

O atributo é chamado então AirportGeolocation, e já reorganizamos e carregamos a latitude e longitude correspondentes ao local de cada aeroporto. Com estas informações em nosso banco de dados, o dispositivo mostrará o mapa com o local correspondente.



E para que a aplicação seja ainda mais atrativa, adicionaremos uma imagem de cada aeroporto.

Então, criamos também o atributo AirportImage e carregamos as fotos dos aeroportos:



Antes de executar a aplicação, verifiquemos no Work With for Devices que todas as informações estão presentes como desejamos:

Selecionemos a seção Airport...

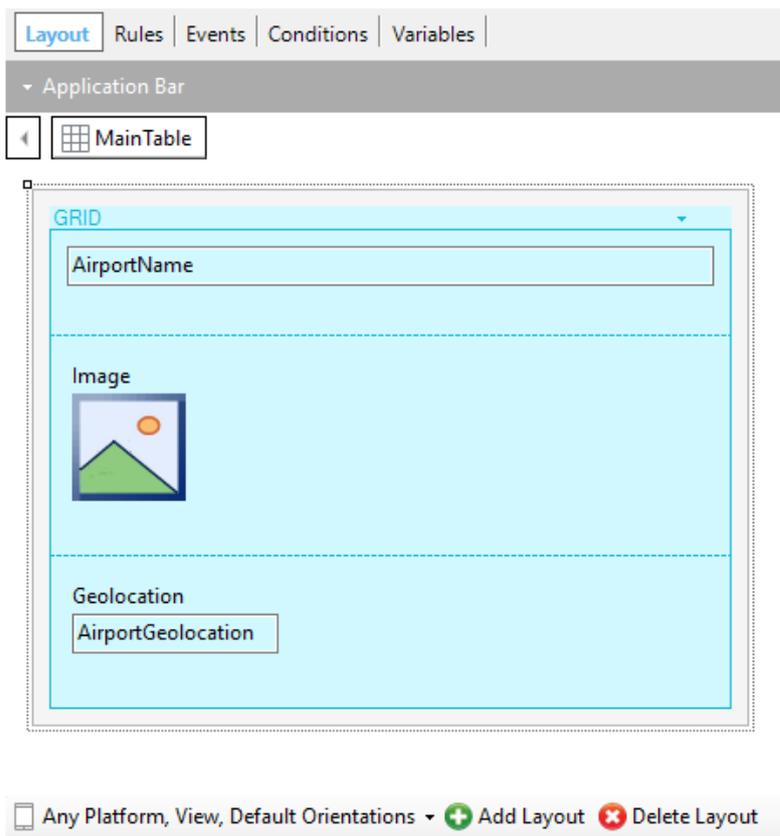
Vamos personalizar um pouco o visual...

E para aproveitar melhor o espaço na tela do dispositivo, vamos localizar os controles verticalmente em vez de horizontalmente que é como é exibido por padrão... Fazemos isso no grid.

Adicionemos os dados de geolocalização para que se possa ver o mapa com o local

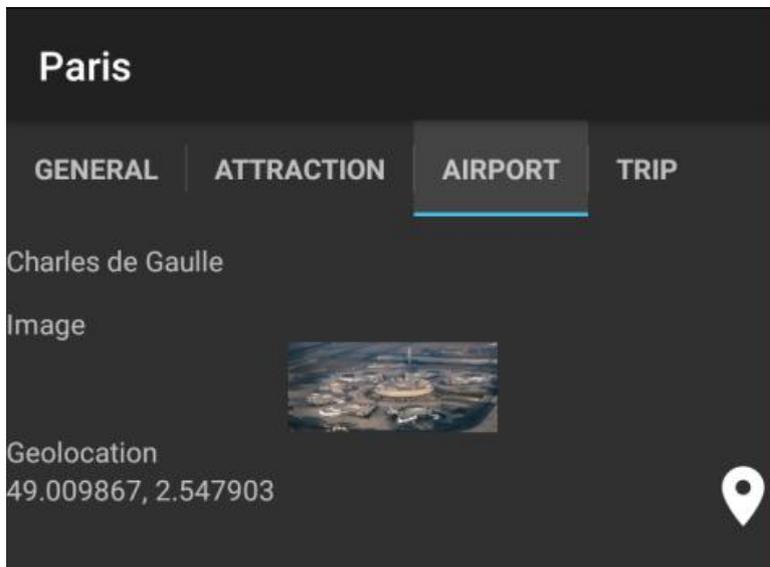
E na propriedade Label position de cada atributo, indicamos que a posição do rótulo seja Top

Inserimos o atributo AirportGeolocation e modificamos a legenda do rótulo para que diga apenas Geolocation.

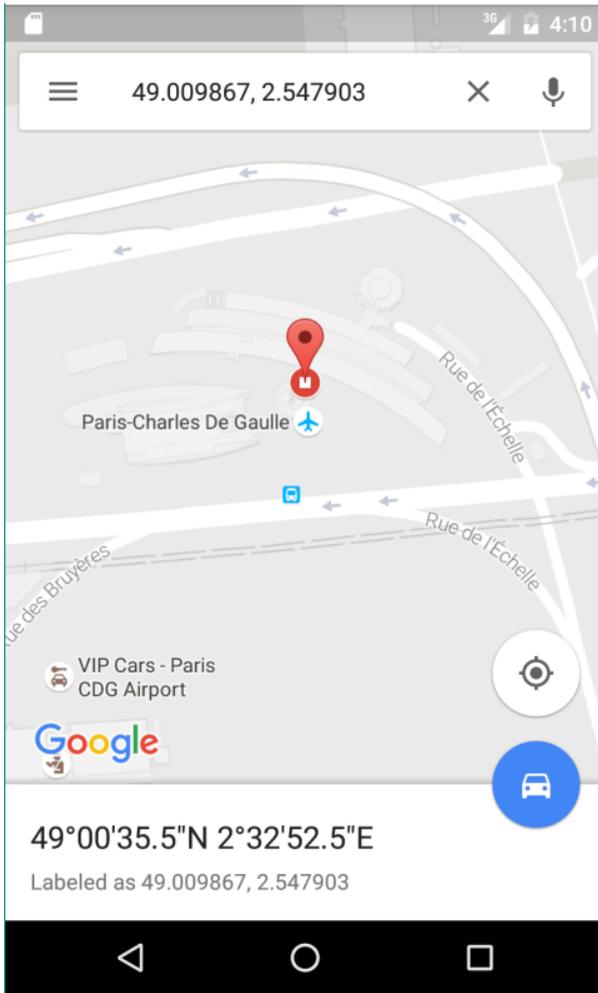


Agora sim, vejamos a aplicação novamente em execução:

Selecionamos França, Paris, Aeroporto, e vemos o seguinte:



Para ver o mapa pressionamos aqui:



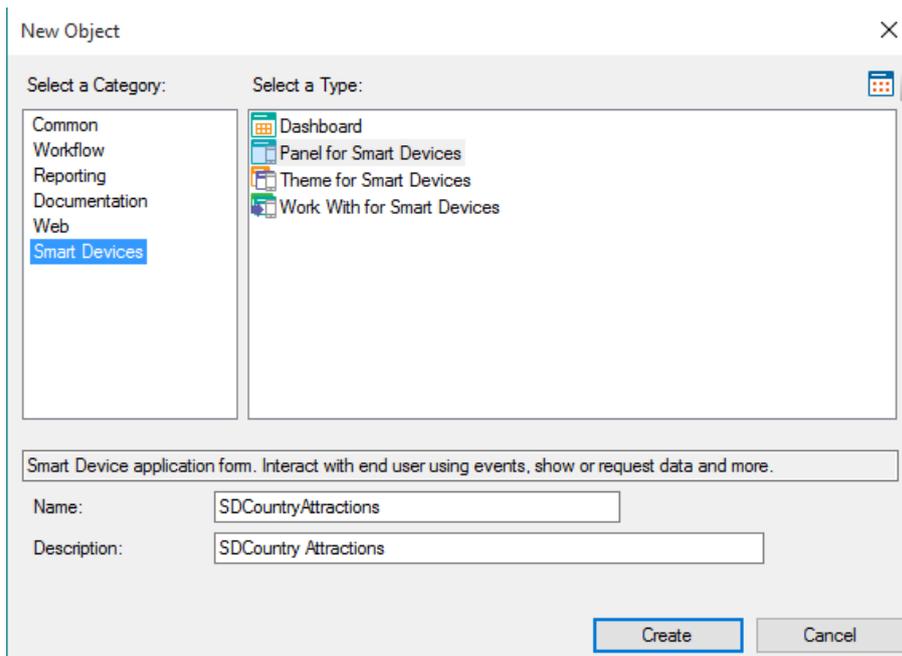
E agora, por último, suponhamos que a agência nos pede que na tela com a galeria de países, também se ofereça um botão que ao ser pressionado nos mostre algo assim:



Quer dizer, todos os países listados e para cada um deles suas atrações registradas.

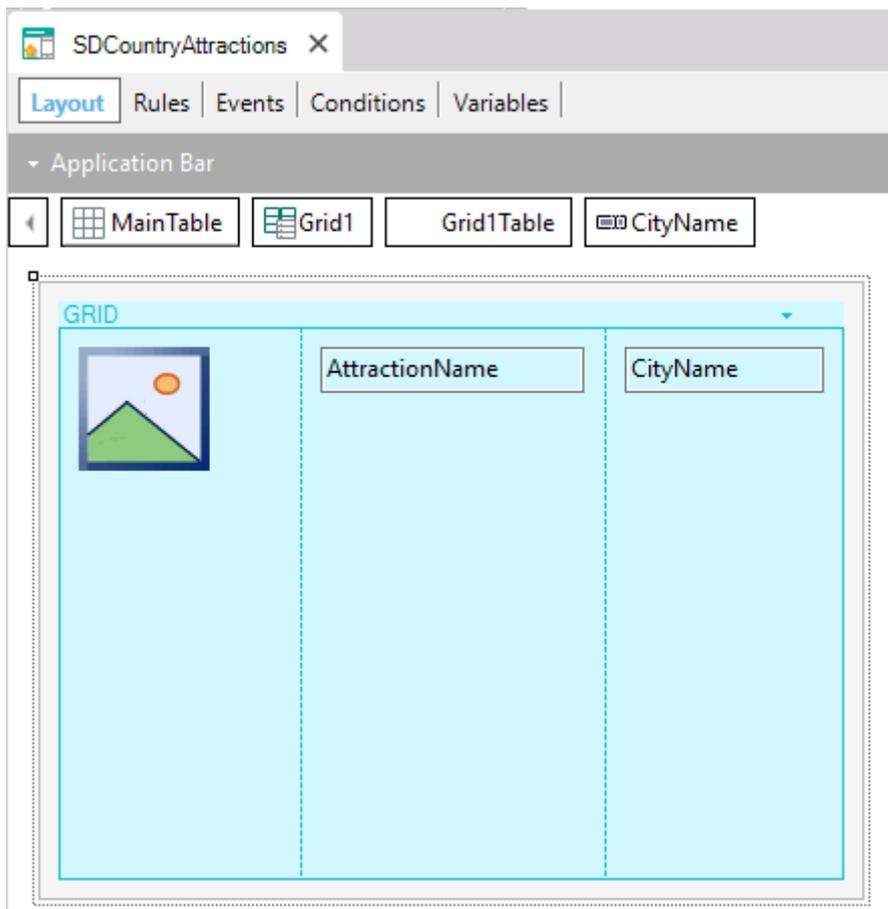
Desta forma, o usuário final poderá ter uma visão geral de quais e a quantidade de atrações cada país oferece.

Novamente em GeneXus, criamos um novo objeto, esta vez do tipo Panel for Smart Devices, que é uma espécie de web panel, mas para Smart Devices:



A partir da toolbox, arrastamos um grid e inserimos os atributos AttractionName, AttractionImage e CityName.

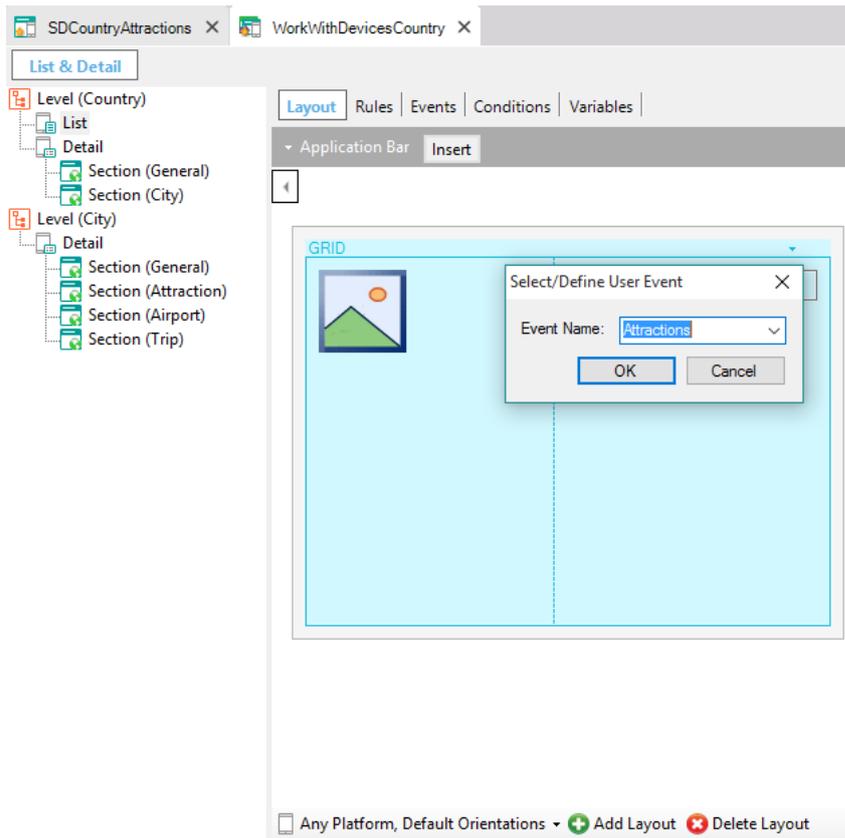
Verificamos para que não sejam exibidos os rótulos. Para isso editamos as propriedades de cada atributo e na propriedade Label Position colocamos None



Para que o usuário final possa acessar esta informação a partir da tela principal, adicionaremos o botão que mencionamos na seção List de países do objeto Work With for Smart Devices.

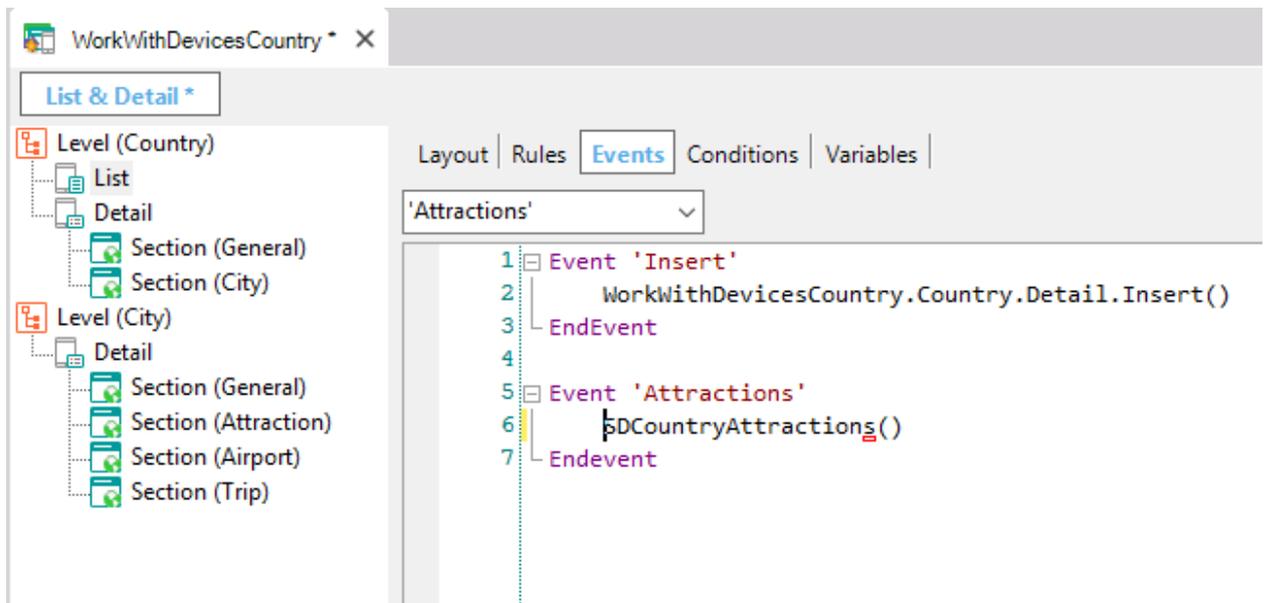
Assim, a partir da toolbox arrastamos um botão sobre a application bar.

No nome do evento do botão colocamos o nome: Attractions



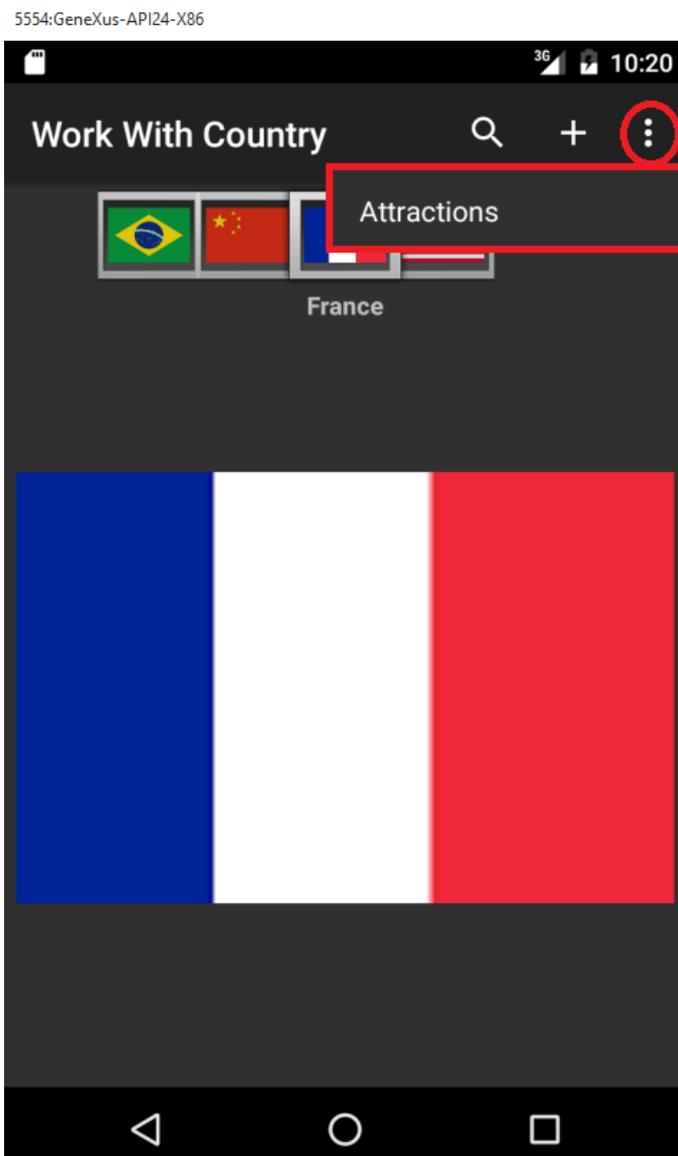
Clicamos duas vezes no botão e nos posicionamos na seção de eventos.

A partir do KB Explorer arrastamos o objeto SDCountriesAttractions e completamos a chamada com dois parênteses, como fizemos na aplicação web.

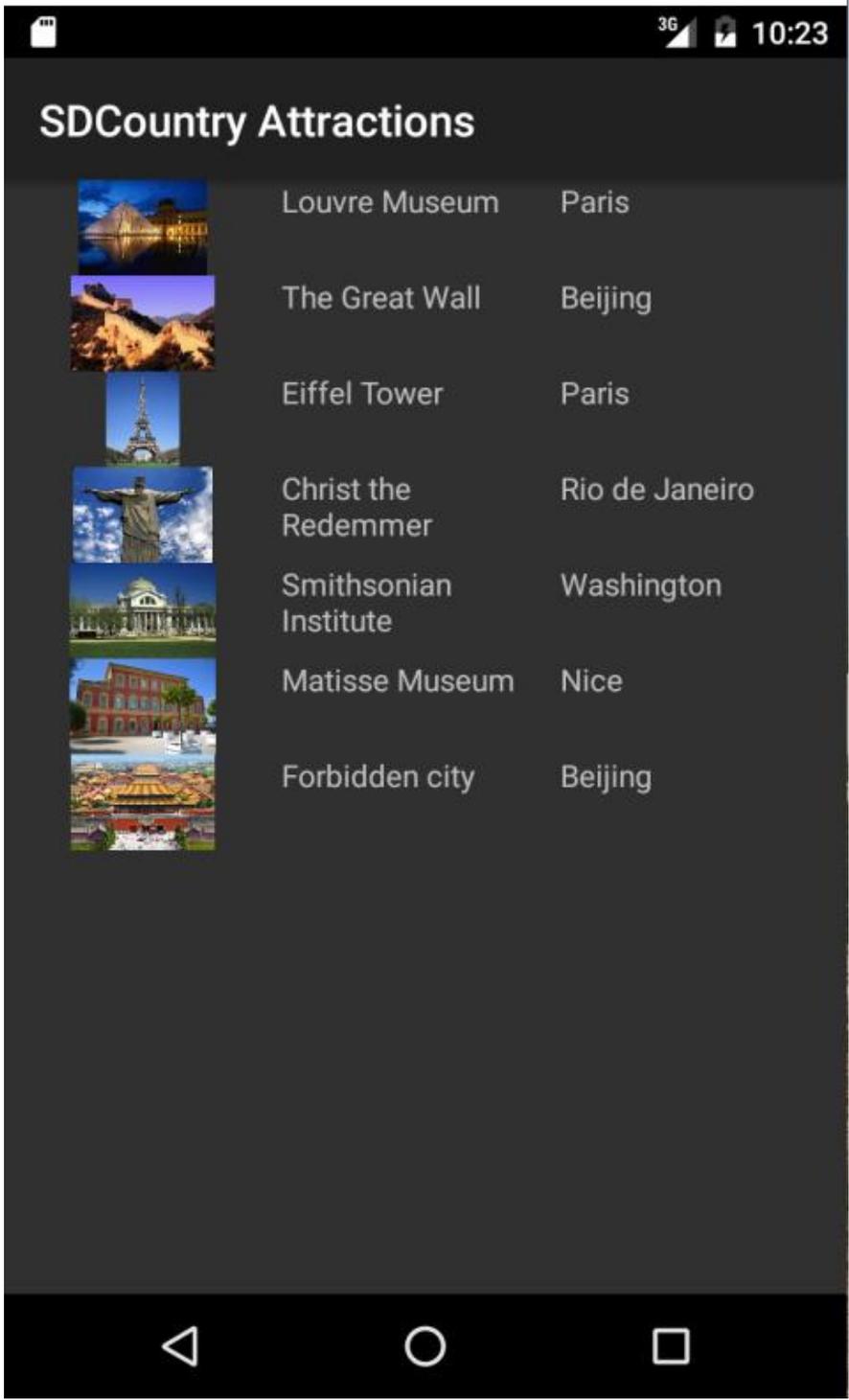


Pressionemos F5...

Na tela List, se selecionamos o botão menu do emulador, a nova opção será oferecida:



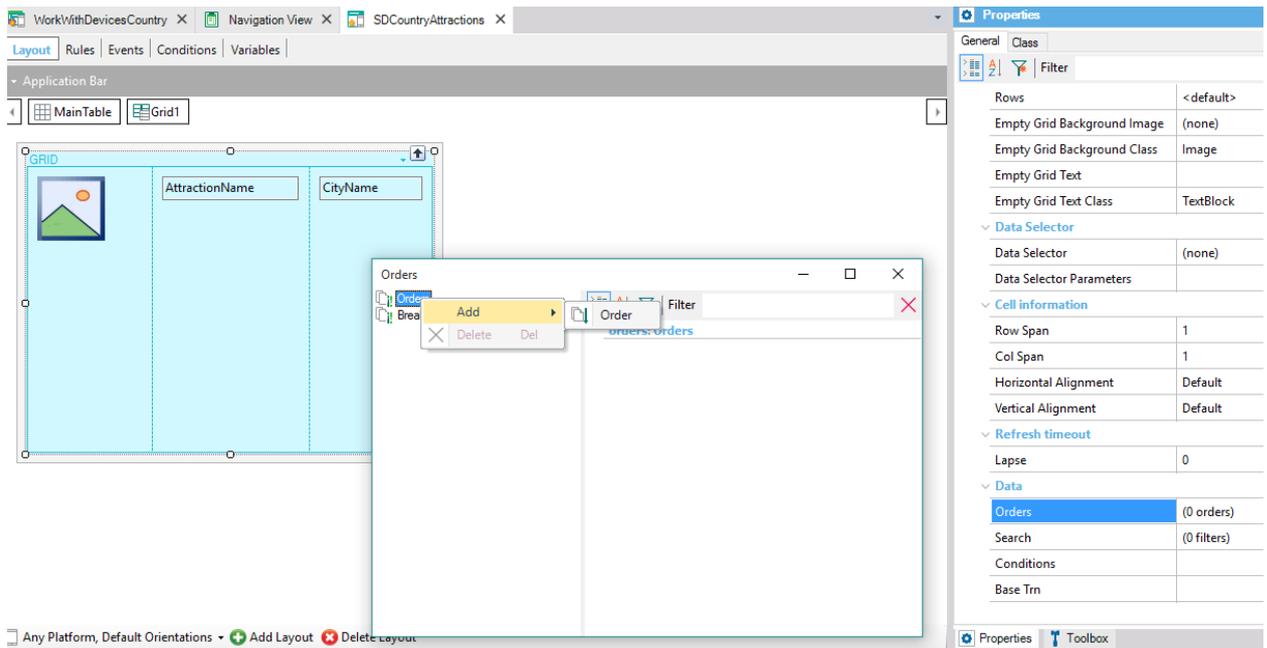
Selecionamos o botão Attractions... e vemos todas as atrações registradas:



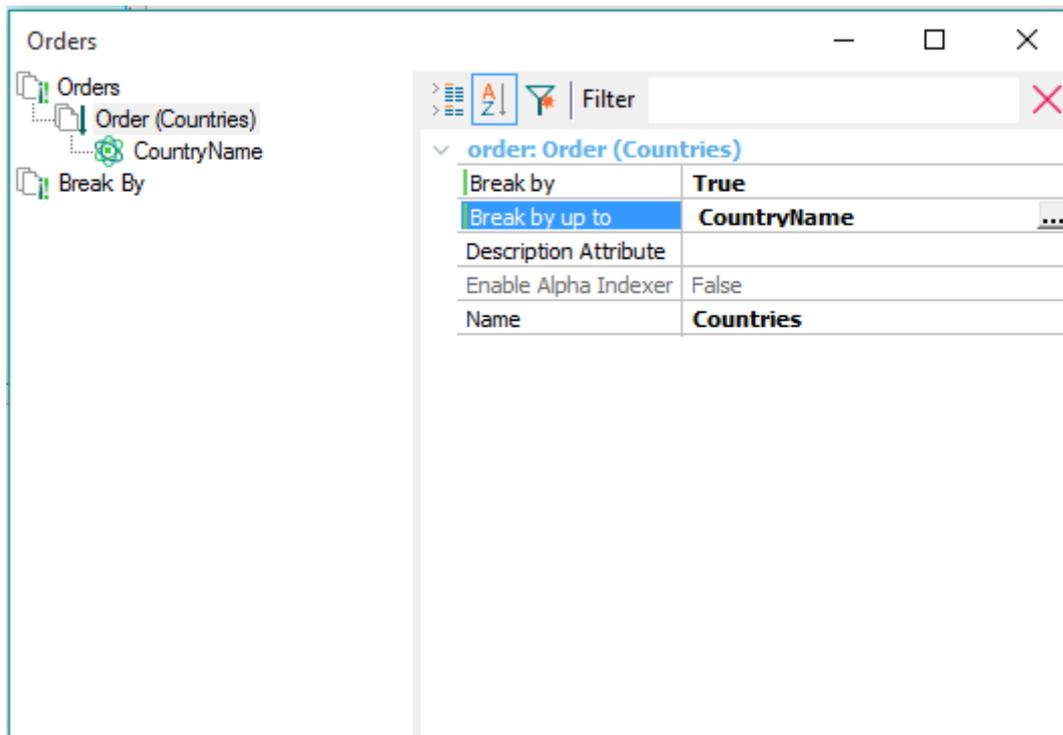
Nesta lista as cidades são repetidas, conseqüentemente os países são repetidos também embora não estamos visualizando. Esperávamos outro formato, então vamos agrupar as atrações por país.

Voltemos ao GeneXus...

Selecionamos o Grid para ver suas propriedades e observemos esta propriedade Orders. A partir daqui, vamos definir uma nova ordem de nome Countries.



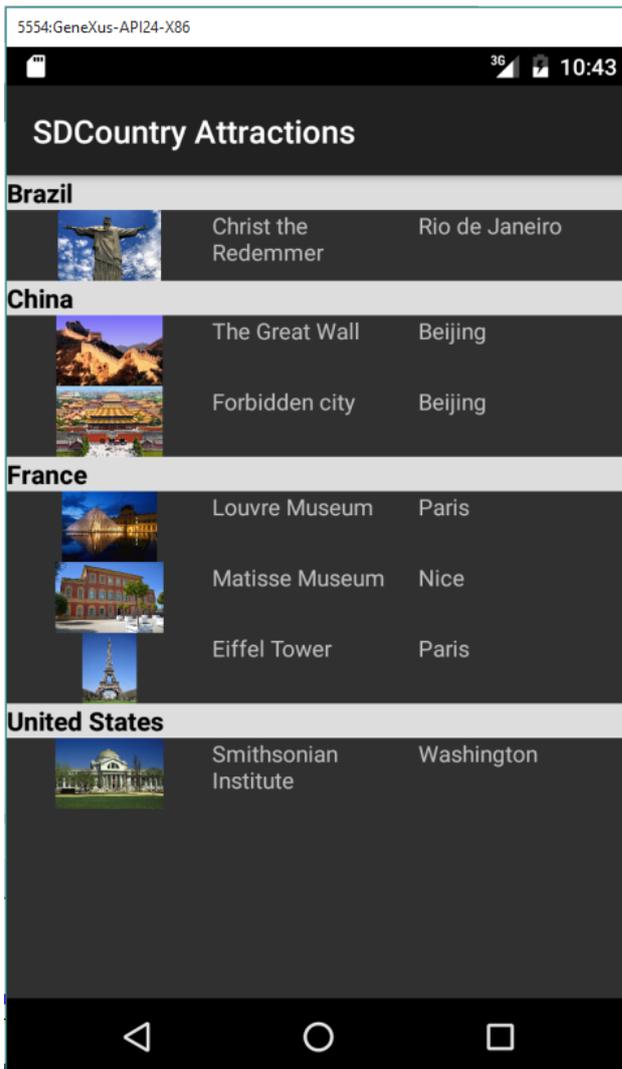
Clicamos com o botão direito do mouse em Add Attribute e selecionamos o atributo CountryName porque é o atributo selecionado para a ordem. Observemos as propriedades desta Order definida. Na propriedade Break by escolhemos True. Indicamos que o corte será feito por CountryName e, desta forma irá estabelecer o agrupamento:



e na propriedade Description Attribute também escolhemos CountryName para exibir o nome do país.

Executemos novamente a aplicação, pressionemos F5...

Executamos novamente a aplicação... pressionamos o botão Attractions e vemos as atrações agrupadas agora por país:



Desta forma, desenvolvemos um pequeno módulo para smart devices, a fim de oferecer aos usuários finais uma opção a mais de acesso à informação que a agência de viagens gerencia.