

Introducción a Smart Devices

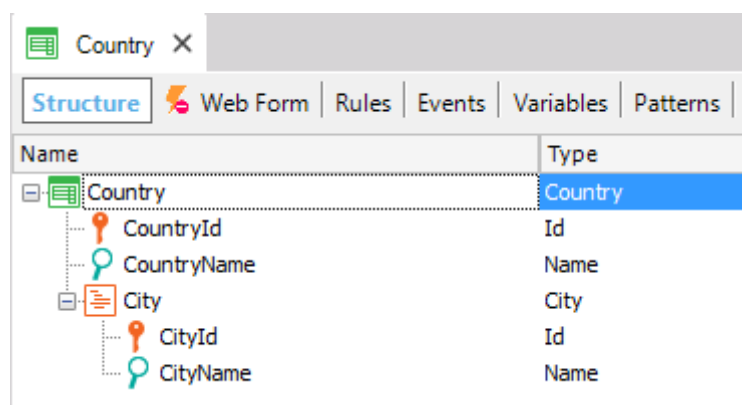
Hasta el momento hemos construido todos los requerimientos solicitados por la agencia de viajes en lo relativo a la aplicación web a ser utilizada por los usuarios de la propia agencia.

Ahora la agencia desea ofrecer también una pequeña aplicación para dispositivos inteligentes para ser utilizada por los usuarios finales.

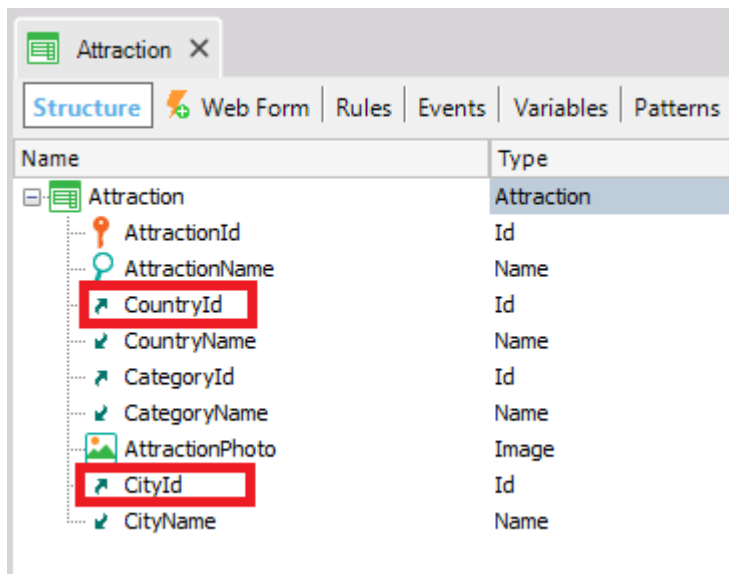
El objetivo es que cualquier persona pueda consultar desde su Smart Device todas las ciudades que la agencia ofrece visitar, las excursiones disponibles, las principales atracciones turísticas, etc.

Vayamos entonces a GeneXus, y repasemos nuestro diseño de transacciones.

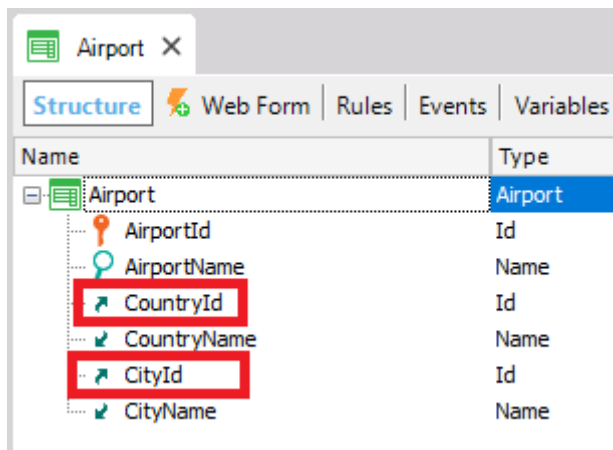
Contamos con la transacción Country, con su conjunto de ciudades propias:



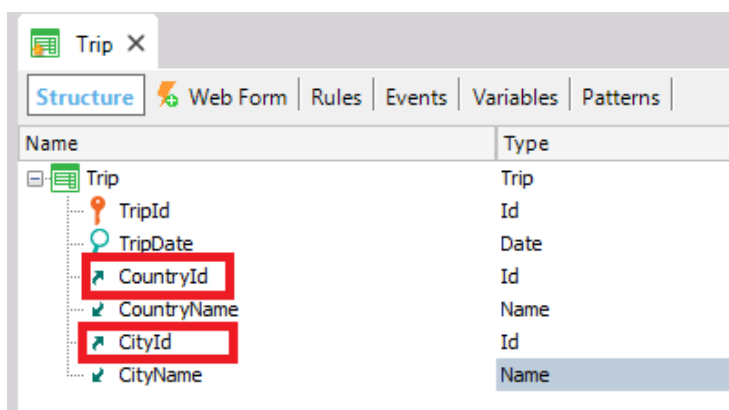
La transacción Attraction, que tiene a los atributos CountryId y CityId como llaves foráneas:



La transacción Airport que tiene también a CountryId y CityId como llaves foráneas:



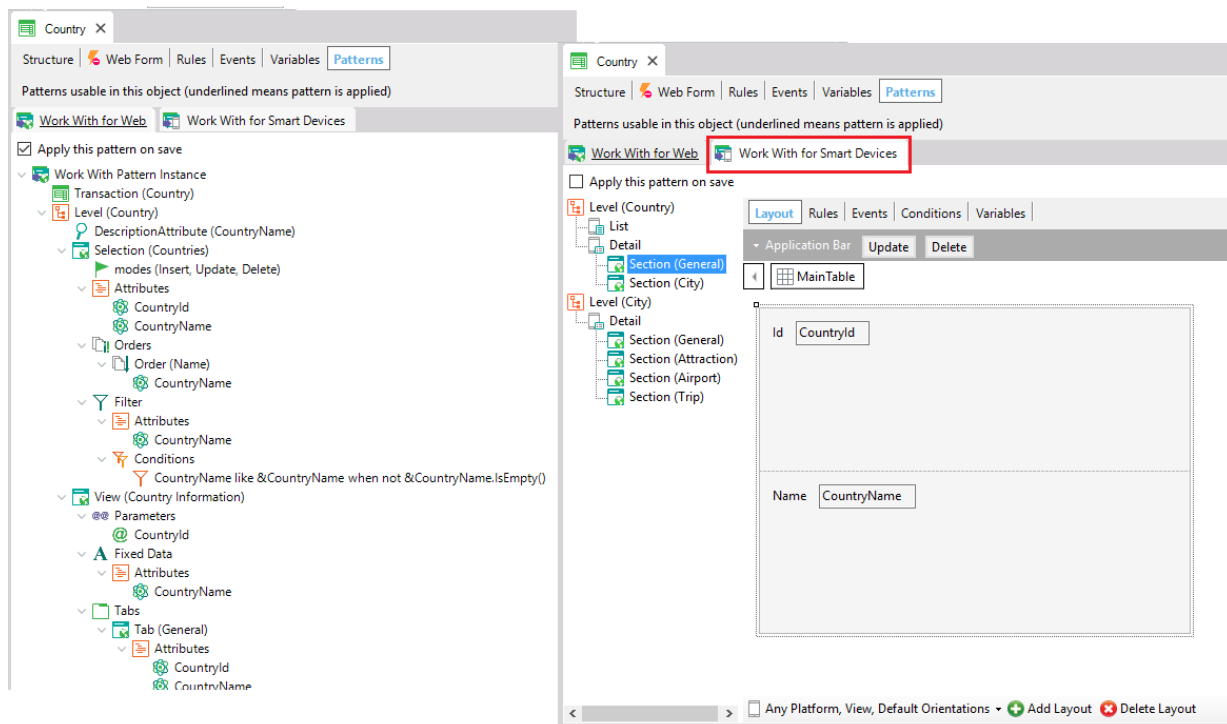
Y la transacción Trip que tiene también a CountryId y CityId como llaves foráneas:



Bien. Recordemos que en videos anteriores hemos aprendido sobre el pattern Work With “for web” y lo hemos aplicado a algunas transacciones.

Vayamos entonces a la transacción Country, seleccionemos la sección: Patterns.

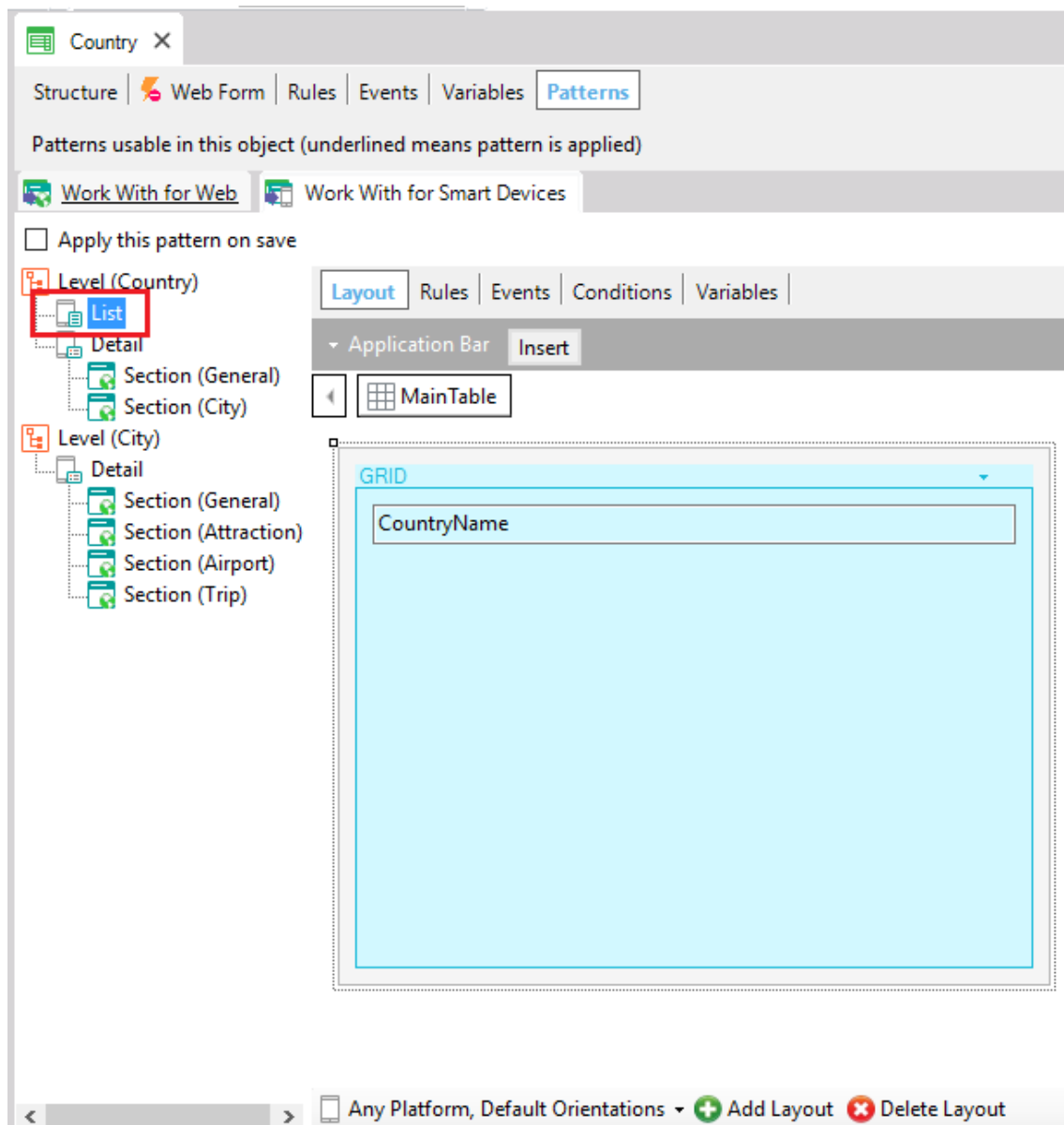
Y observemos que además de la aplicación del pattern Work With para web, GeneXus también ofrece la aplicación del mismo patrón, pero para Smart Devices.



Seleccionemos Work With for Smart Devices...

Y pasemos a observar los nodos del árbol de la izquierda, y lo que vemos a la derecha.

Bajo el nodo correspondiente al nivel Country, aparece el nodo List. Si hacemos click sobre él, vemos que corresponde a una grilla que tiene insertado el atributo CountryName.



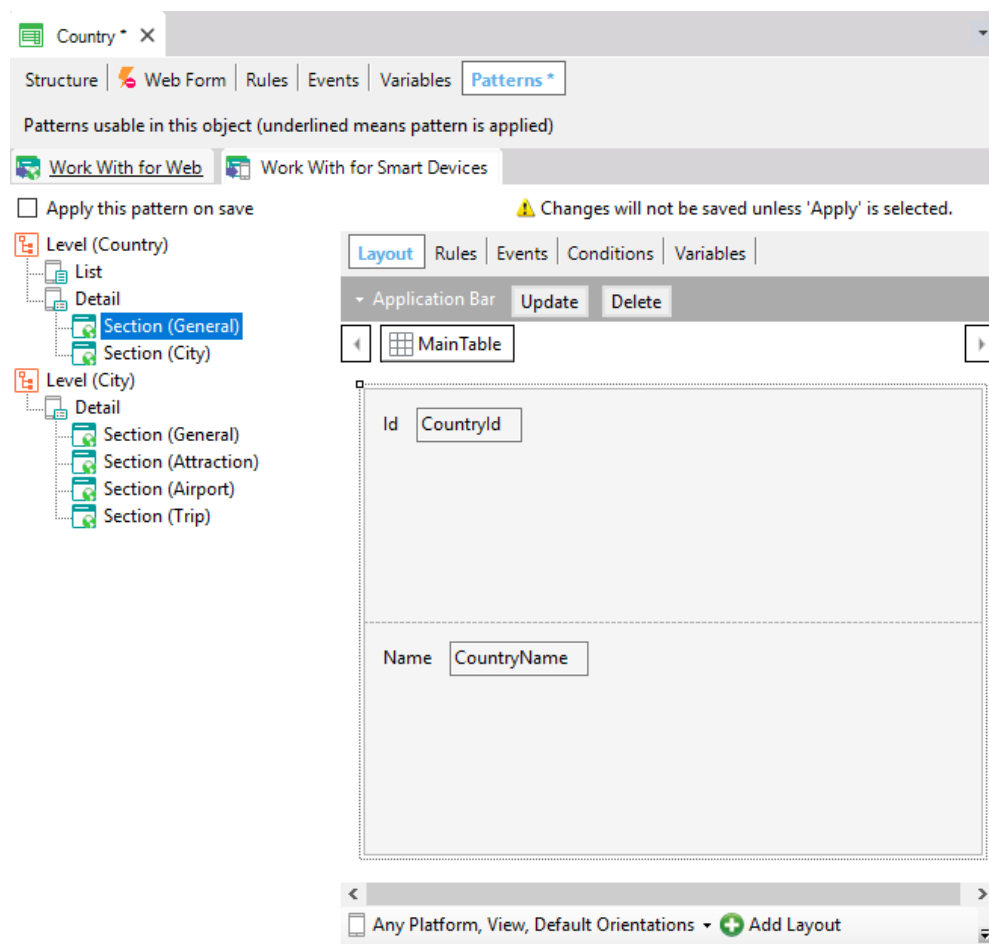
El nodo denominado “List” en este patrón, corresponde a la grilla que listará los datos de cada registro en el panel que se generará. Por lo tanto la grilla en el panel que se generará, mostrará **la lista** de nombres de los países registrados.

Vimos que en este caso entonces, se muestra ya el Layout para el nodo List, en lugar de ver la lista de atributos a ser incluidos en la grilla, debajo del nodo, como era el caso del pattern Work With for Web.

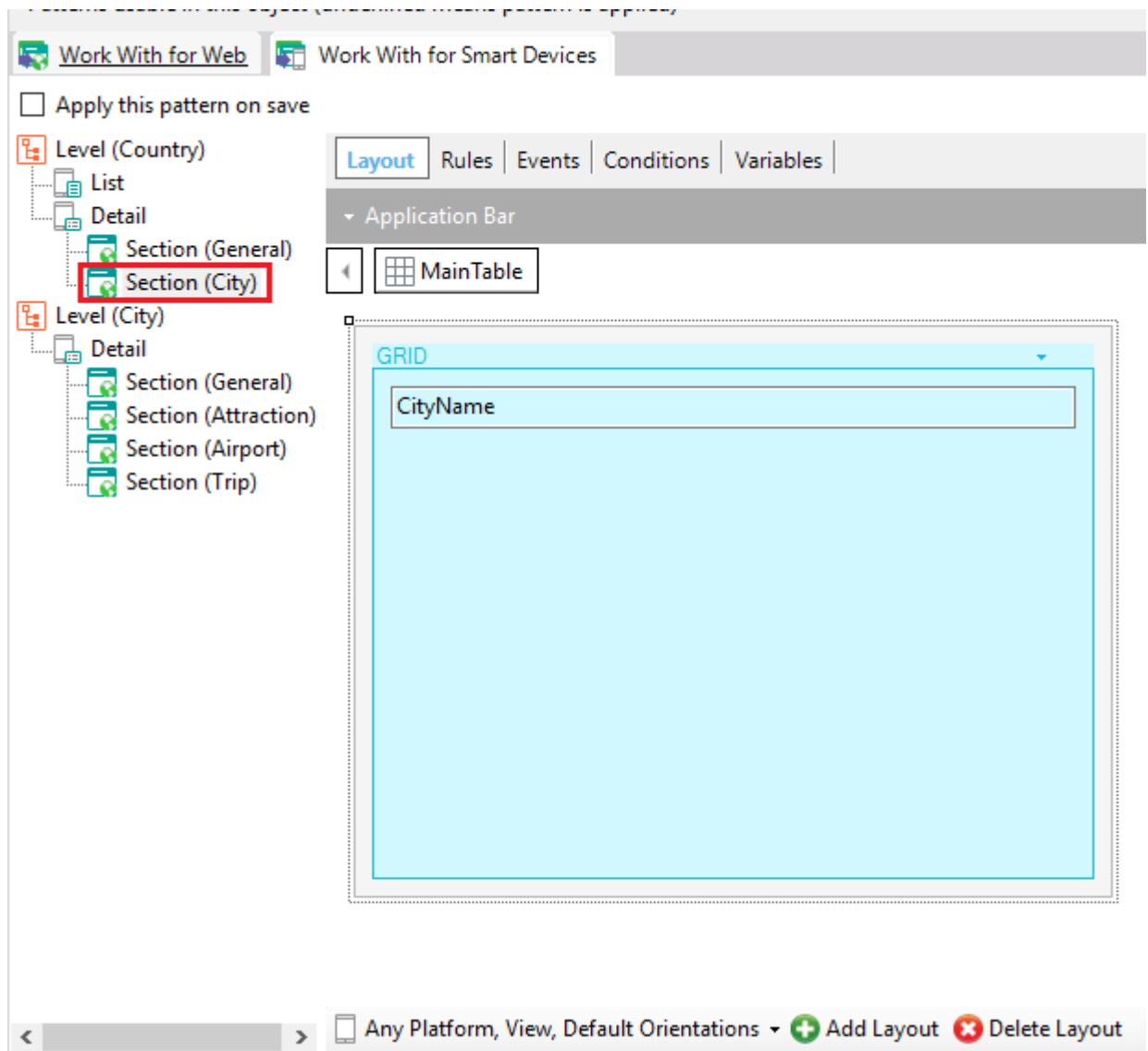
Pasemos ahora a observar el nodo Detail. Podemos asociar el término Detail a lo que en el patrón Work With for web conocíamos por “View”. O sea, a ver los detalles de un registro en particular de la lista.

El nodo Detail se compone de dos secciones: General y City.

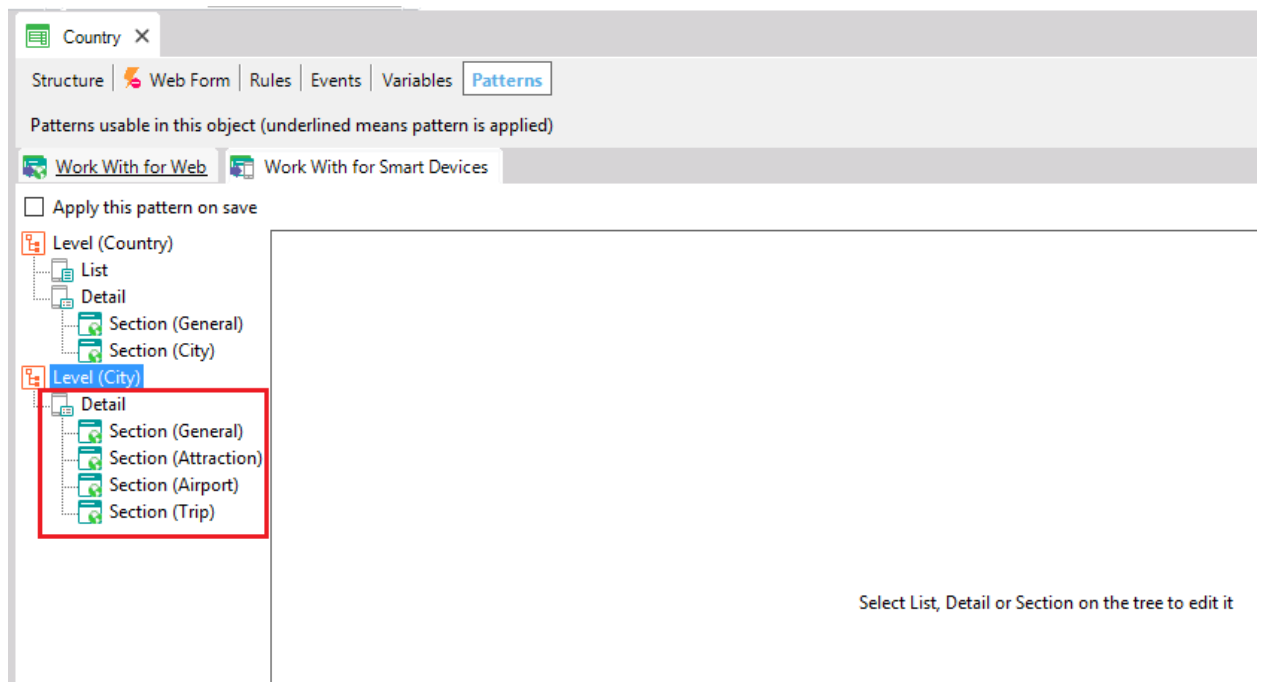
Al igual que la funcionalidad implementada por el patrón Work With for Web, la sección General mostrará los datos asociados al país.



... y la sección City mostrará todas sus ciudades

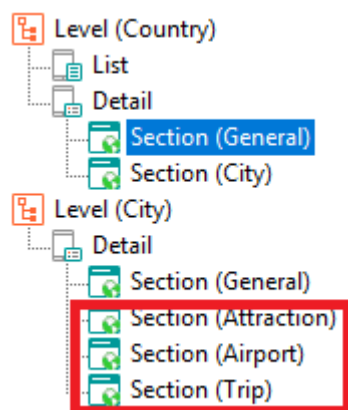


Ahora observemos este nodo: Level(City):

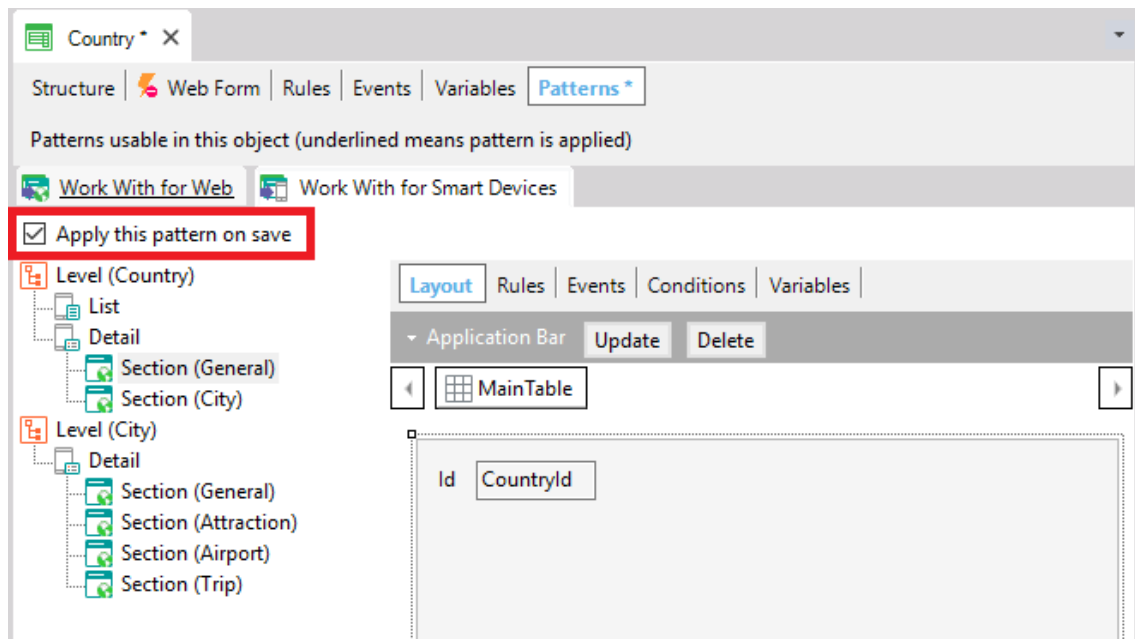


Dado que la transacción Country tiene dos niveles, así como hemos visto recién el nodo correspondiente al primer nivel, este nodo corresponde al segundo nivel.

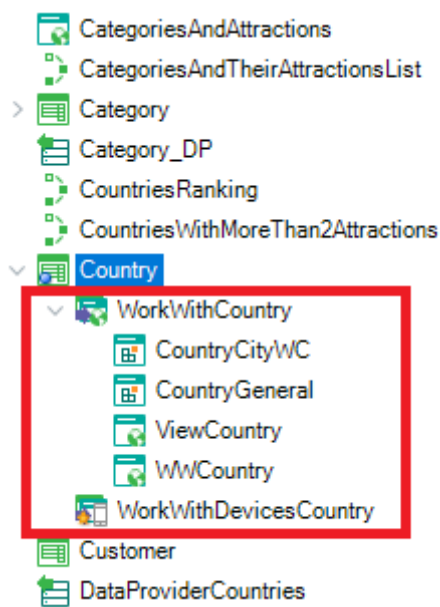
Bajo el mismo, vemos la sección General, tal como está aquí también... y luego hay una sección por cada relación 1-N que tienen las ciudades. Es decir, cada ciudad tiene N Atracciones, N aeropuertos y N excursiones:



Vamos a aplicar este patrón para ver en ejecución lo que genera y ofrece. Solamente debemos clicar la casilla Apply this pattern on save... (tal como lo hicimos para aplicar el pattern Work With for Web)



Y observemos que bajo la transacción Country, vemos ahora también el objeto WorkWithDevicesCountry:

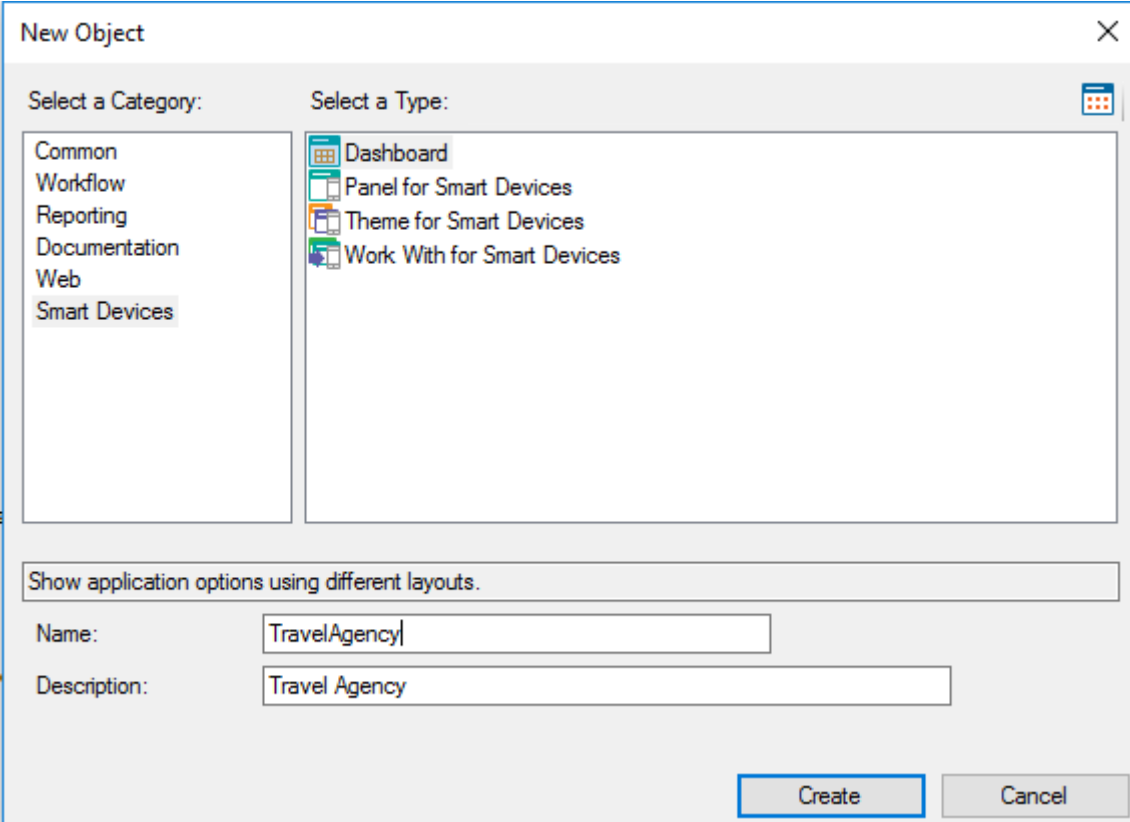


A diferencia de cuando aplicamos el patrón para web, aquí no vemos un conjunto de objetos generados, sino solamente el objeto WorkWithDevicesCountry.

Tenemos que crear ahora un menú de acceso que esté en el dispositivo para que dicho menú acceda a la funcionalidad generada por el patrón. Para eso vamos a crear un objeto GeneXus de tipo Dashboard.

Los objetos Dashboard entonces nos permiten crear menús de acceso que conformen nuestra aplicación para Smart Devices.

Seleccionamos New object, y dentro de la categoría Smart Devices, seleccionamos: Dashboard. Le damos el nombre: TravelAgency...



New Object

Select a Category:

- Common
- Workflow
- Reporting
- Documentation
- Web
- Smart Devices

Select a Type:

- Dashboard
- Panel for Smart Devices
- Theme for Smart Devices
- Work With for Smart Devices

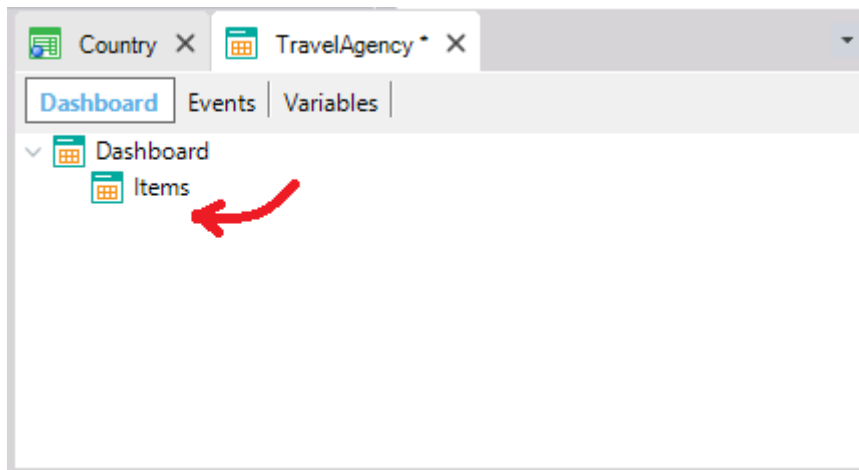
Show application options using different layouts.

Name: TravelAgency

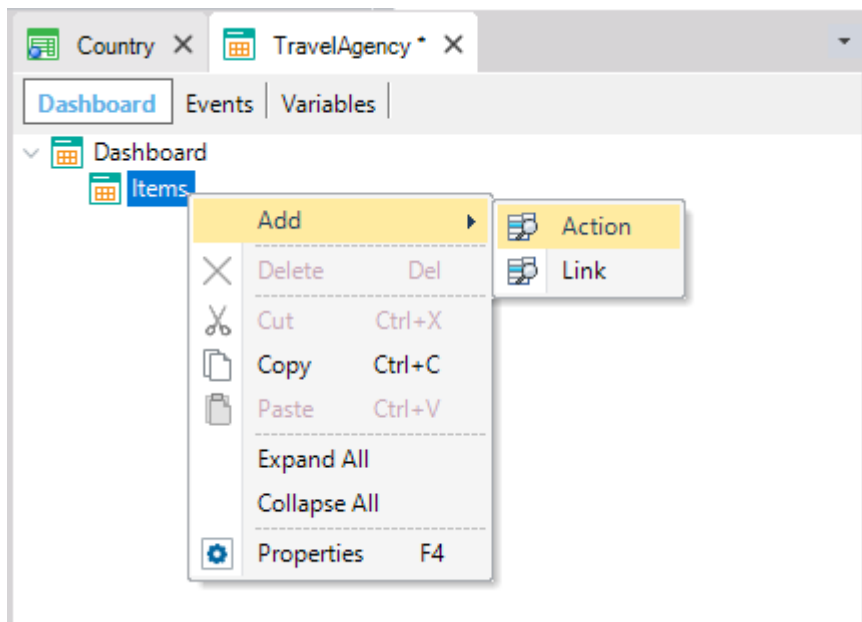
Description: Travel Agency

Create Cancel

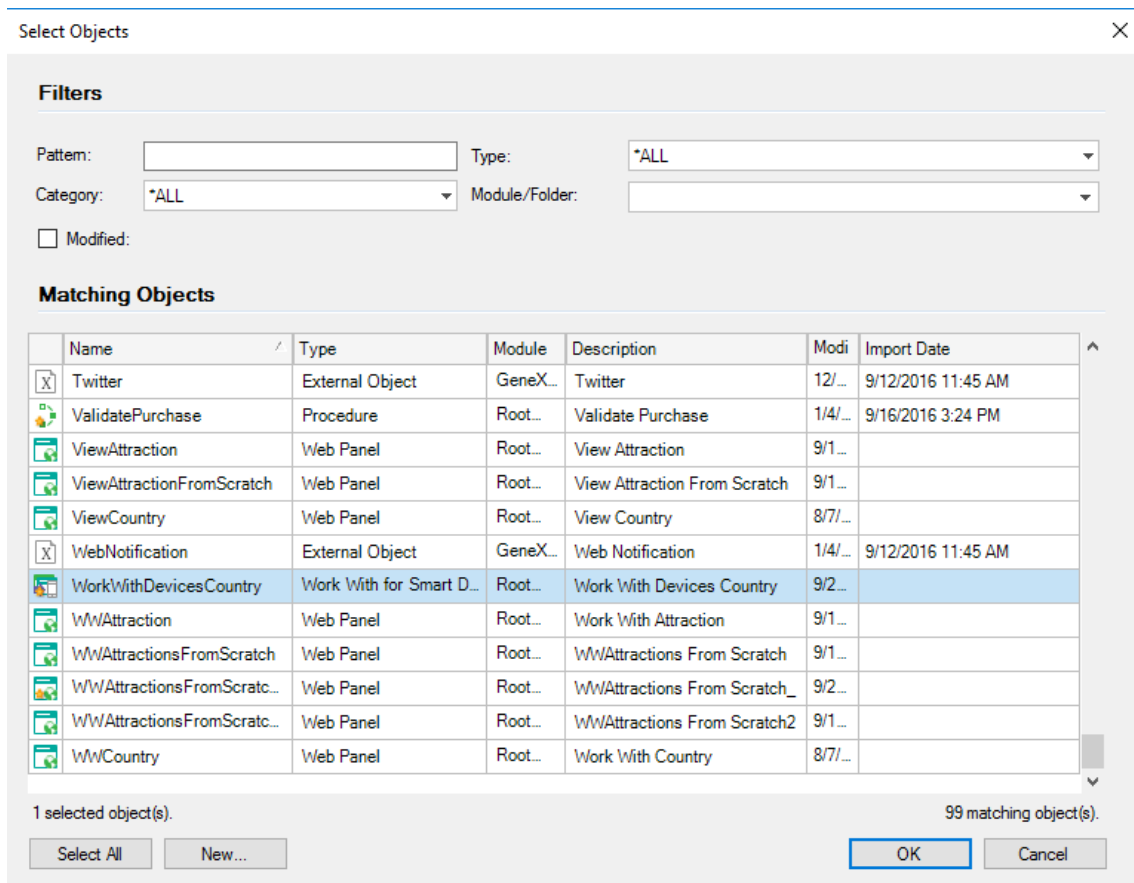
Aquí se agregan las opciones que incluirá el menú:



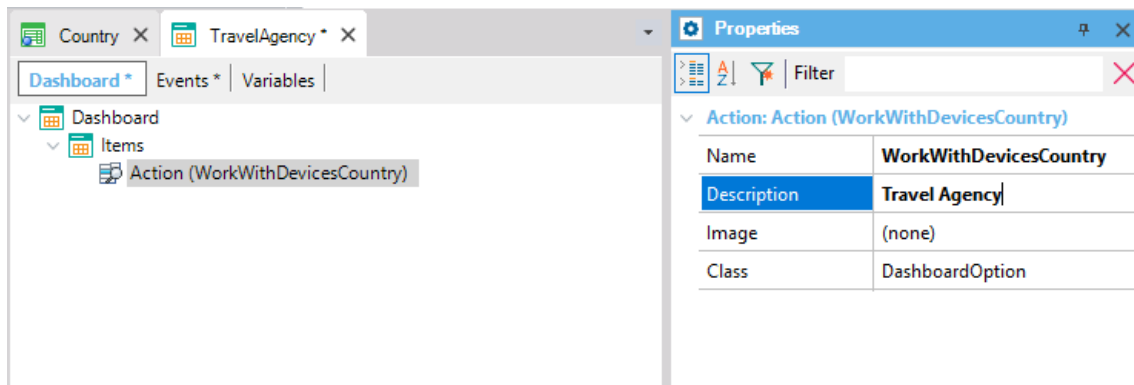
Presionamos clic derecho sobre el nodo Item y seleccionamos Add / Action:



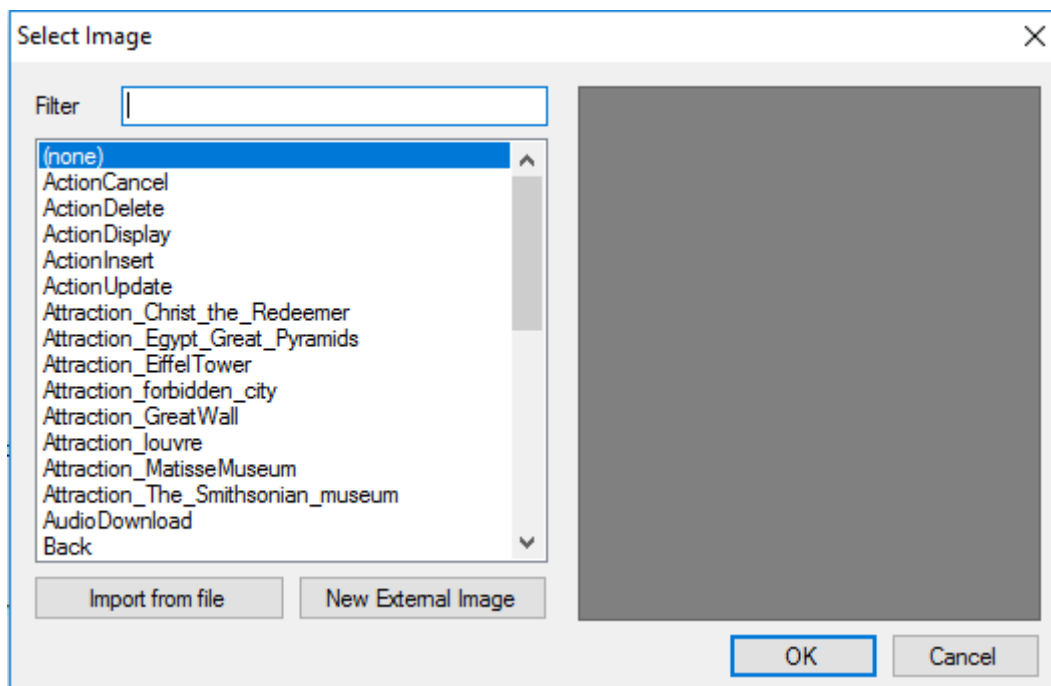
En el cuadro elegimos el objeto WorkWithDevicesCountry...



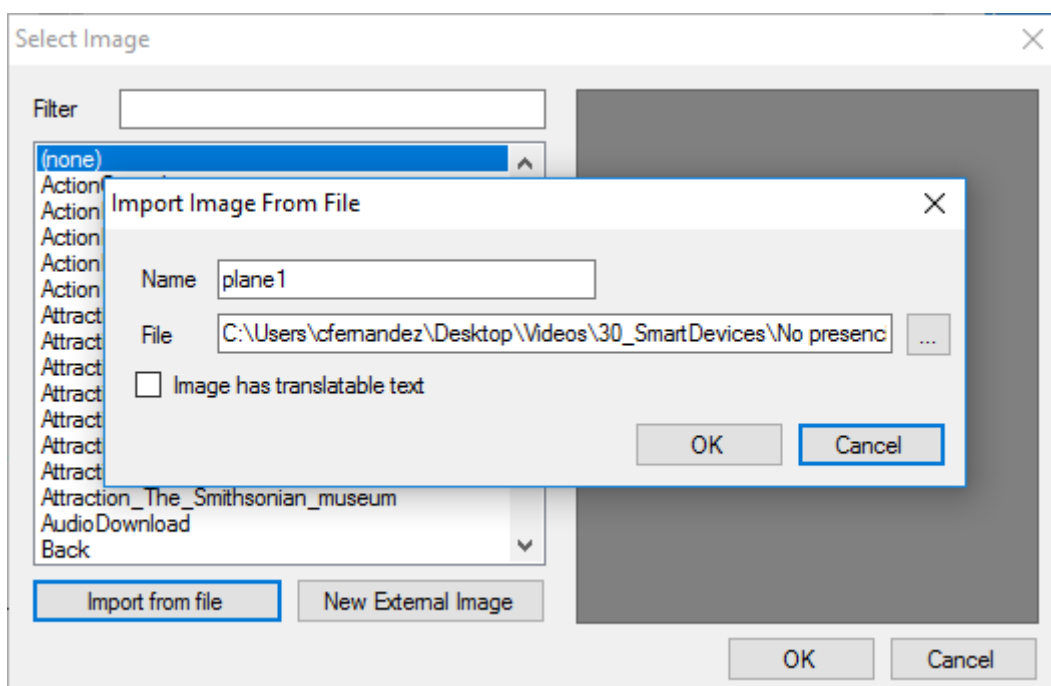
En las propiedades del objeto Dashboard ponemos la descripción: TravelAgency:

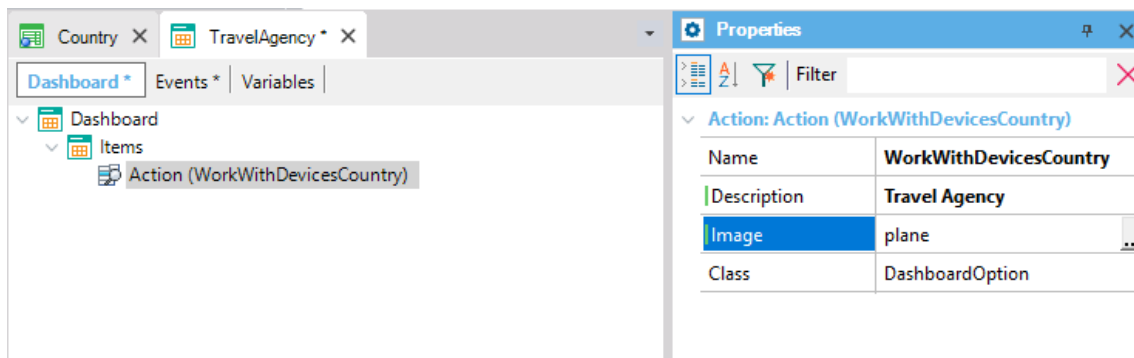


y podemos también elegir una imagen...



Si no nos gusta ninguna de las imágenes que disponemos por defecto dentro de nuestra base de conocimiento, podemos seleccionar una imagen desde un archivo e importarlo...























Bien. Ahora por el hecho de haber creado dentro de nuestra base de conocimiento, objetos propios para Smart Devices, veremos que al presionar F5 automáticamente se ejecutará el emulador para Android... pudiendo también acceder a nuestra aplicación web desde el Developer menú...


Presionemos F5...

Vemos que ahora el Developer menu muestra, además de las opciones de acceso a la aplicación web, un código QR generado automáticamente para poder descargar la aplicación en un 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](#)



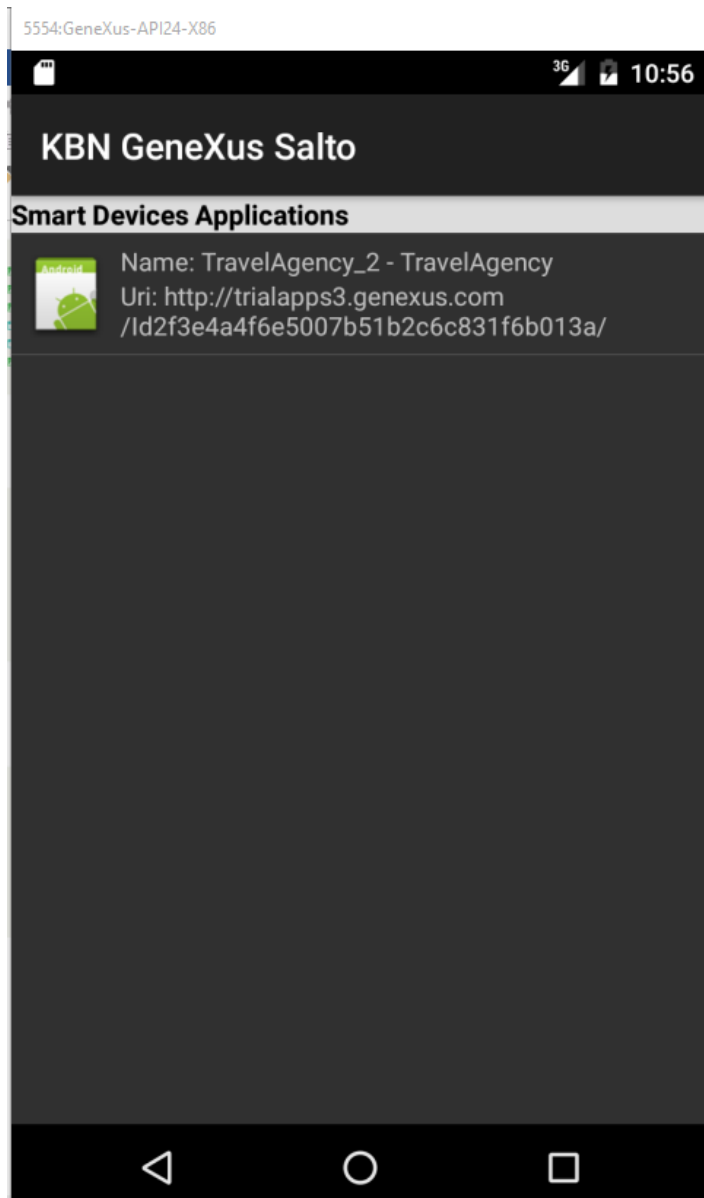
Knowledge Base Navigator
The KBN app makes native iOS online apps prototyping simple. [Download](#)

Install Android Apps

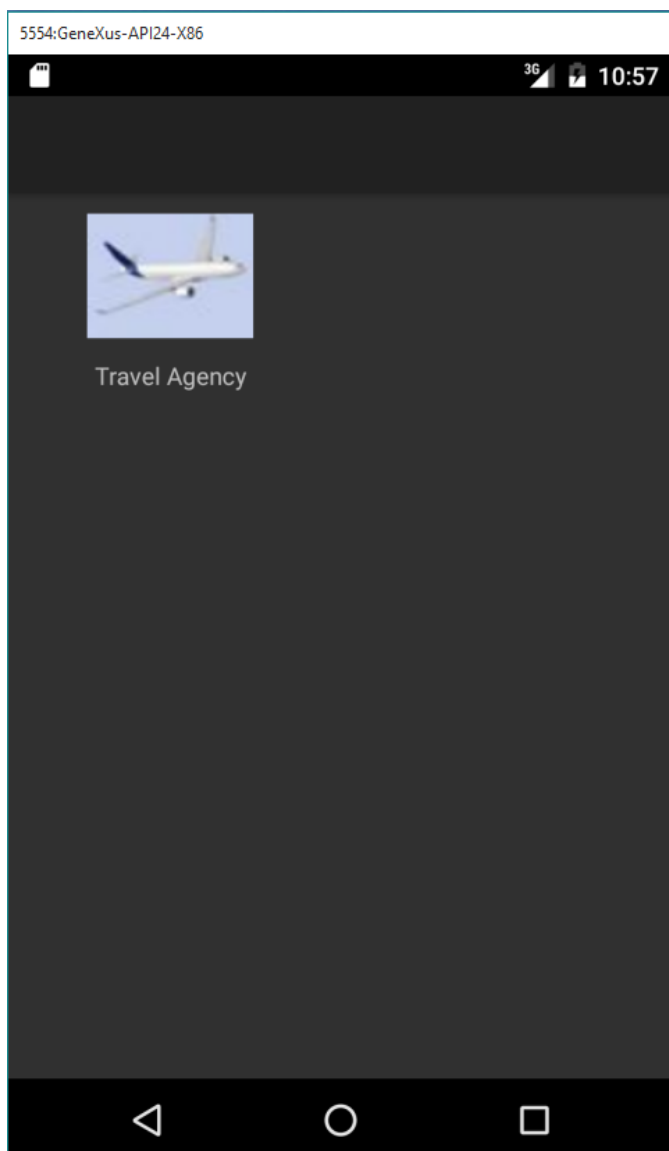


TravelAgency
[Download](#)

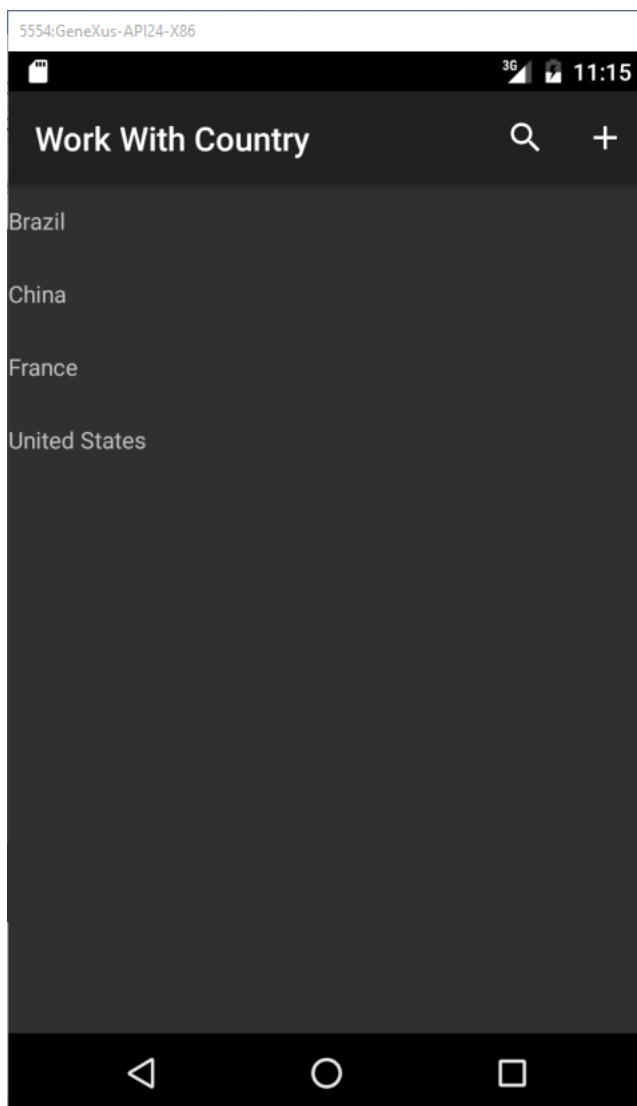
Dejemos que levante ahora el emulador de Android...



Click aquí. Y así se ve el Dashboard en ejecución con la imagen establecida...

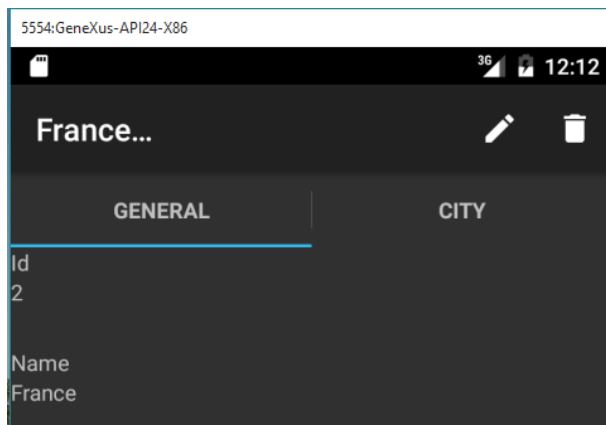


la seleccionamos... y vemos todos los países registrados...

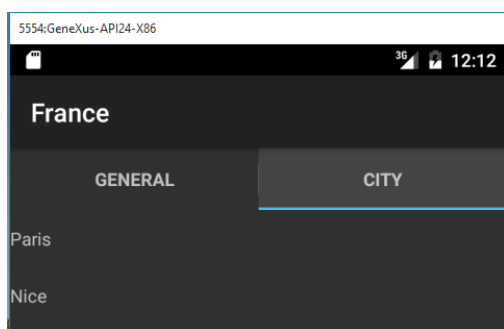


...o lo que veíamos como List.

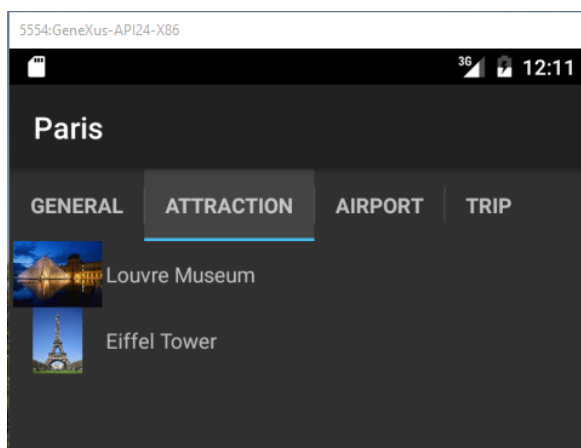
Elegimos por ejemplo Francia... vemos su información general:



Y sus ciudades:



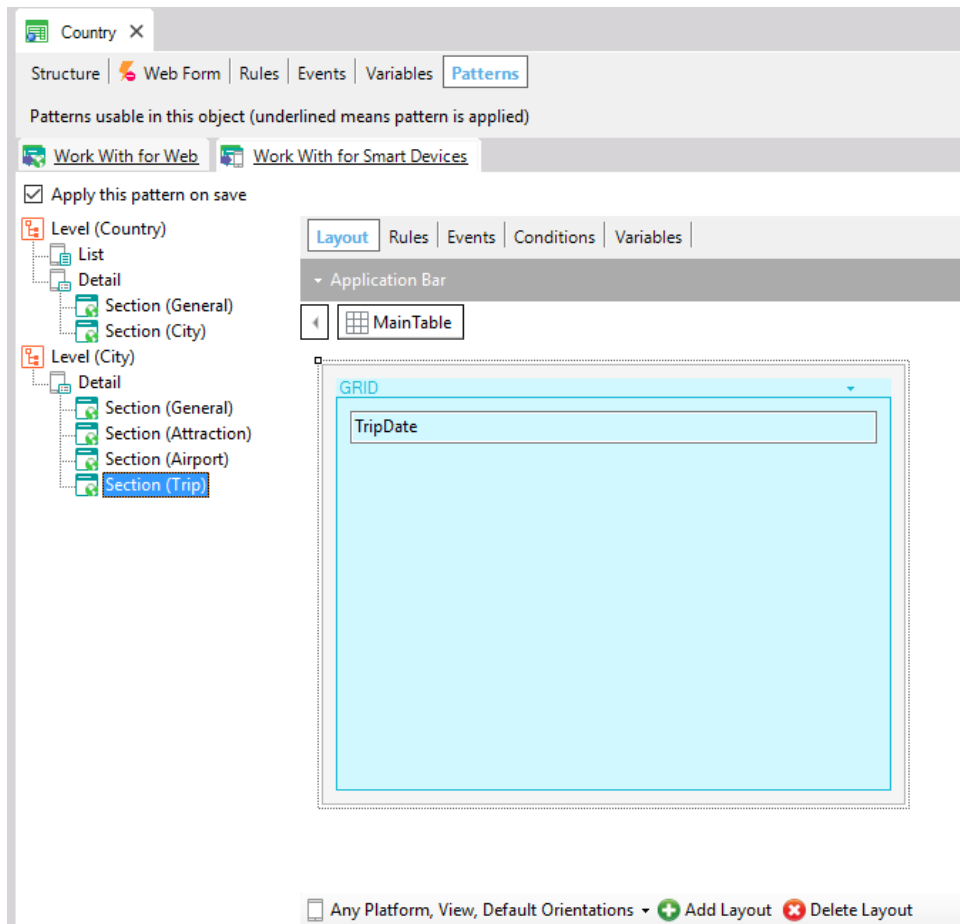
Elegimos Paris.... Y aquí vemos la sección general... y su información relacionada... sus atracciones, aeropuertos... y excursiones:



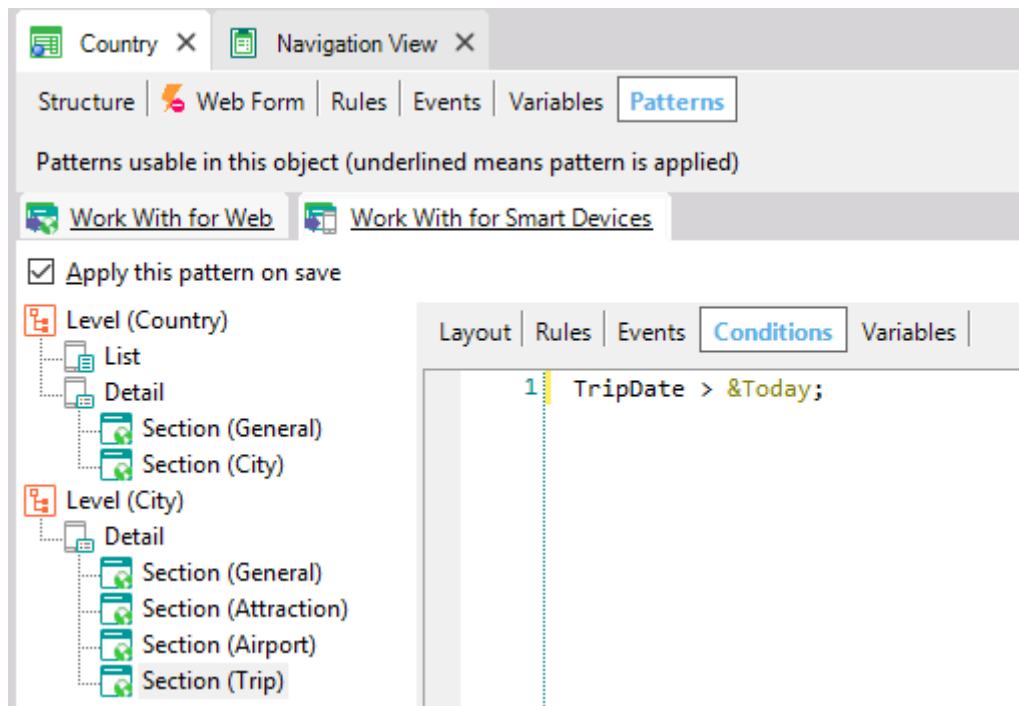
Volvamos a GeneXus para personalizar un poco la aplicación del patrón...

Por ejemplo, supongamos que queremos que solamente se visualicen las excursiones con fecha posterior a la actual, o sea, las excursiones por realizarse...

Así que hacemos click sobre la sección Trip:



Y en las Conditions ponemos: TripDate >&Today;



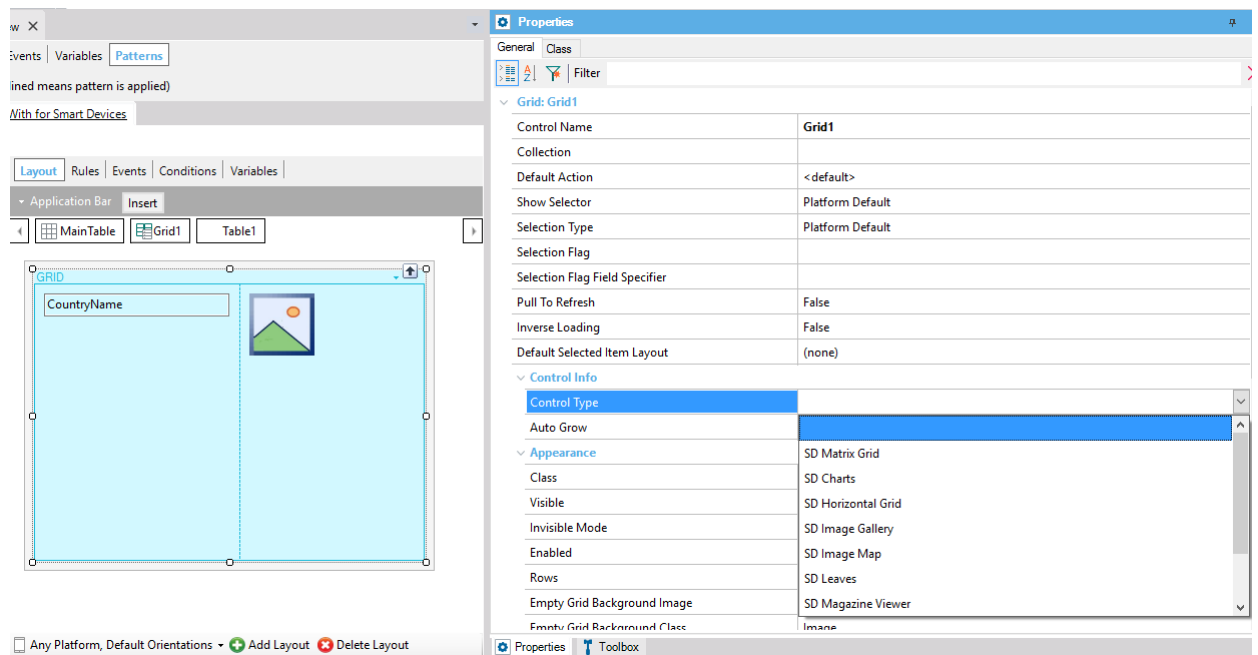
Bien. Ahora propondremos que la lista de países se vea de una forma más atractiva, no simplemente como una lista de nombres.

Podría ser por ejemplo que se vean las banderas, como una galería de imágenes, cada una con el nombre del país, y al seleccionar una bandera se acceda a la información relacionada.

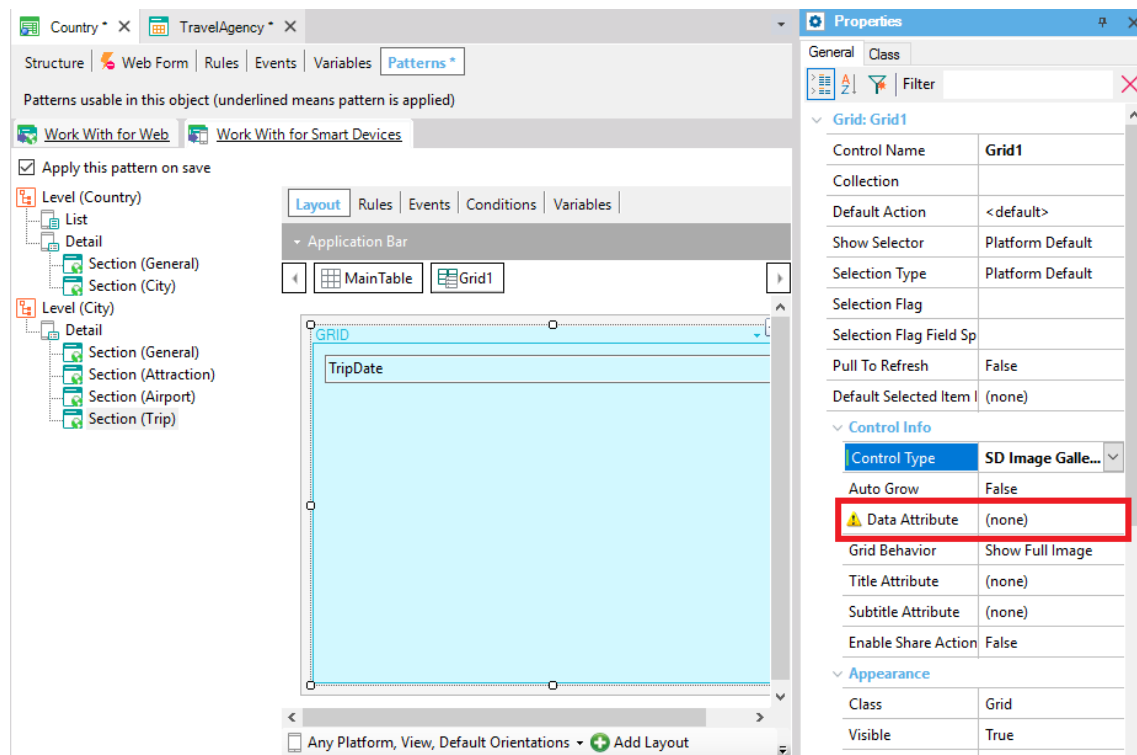
Hemos definido entonces en la transacción Country, un nuevo atributo: CountryFlag de tipo Image... y ya hicimos la reorganización física y carga de imágenes, para ganar tiempo.

Volvamos ahora al patrón Work With for Smart Devices de Country, para definir ahora que la lista de países se visualice como una galería de banderas:

En la propiedad Control Type del grid, seleccionamos SD Image Gallery:



En la propiedad Data Attribute:

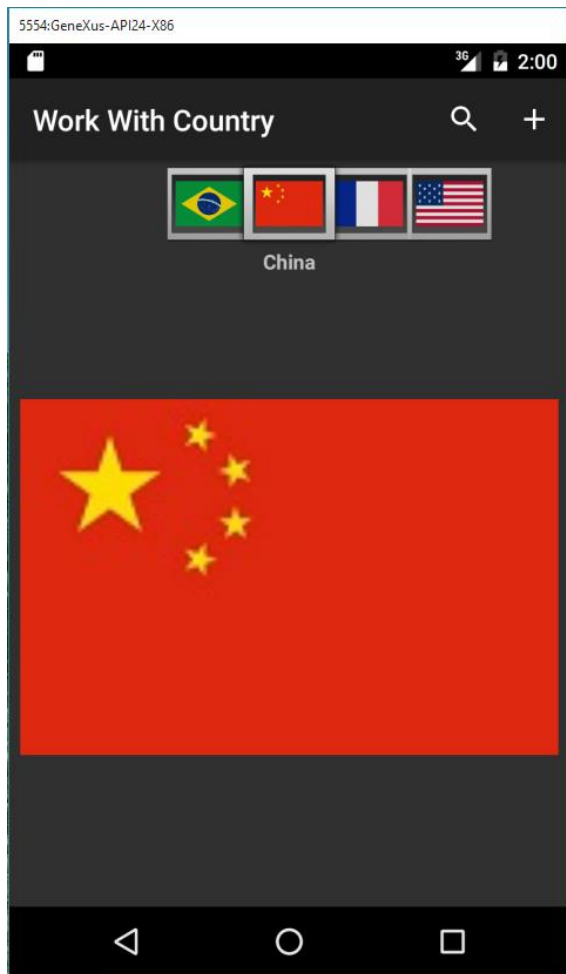


...indicamos el atributo que contiene la imagen a visualizar, en este caso es: CountryFlag, y luego en la propiedad: Title Attribute podemos indicar un atributo cuyo valor sea el título de la imagen. Seleccionamos CountryName.

Llegados a este punto, veamos todas estas modificaciones en ejecución.

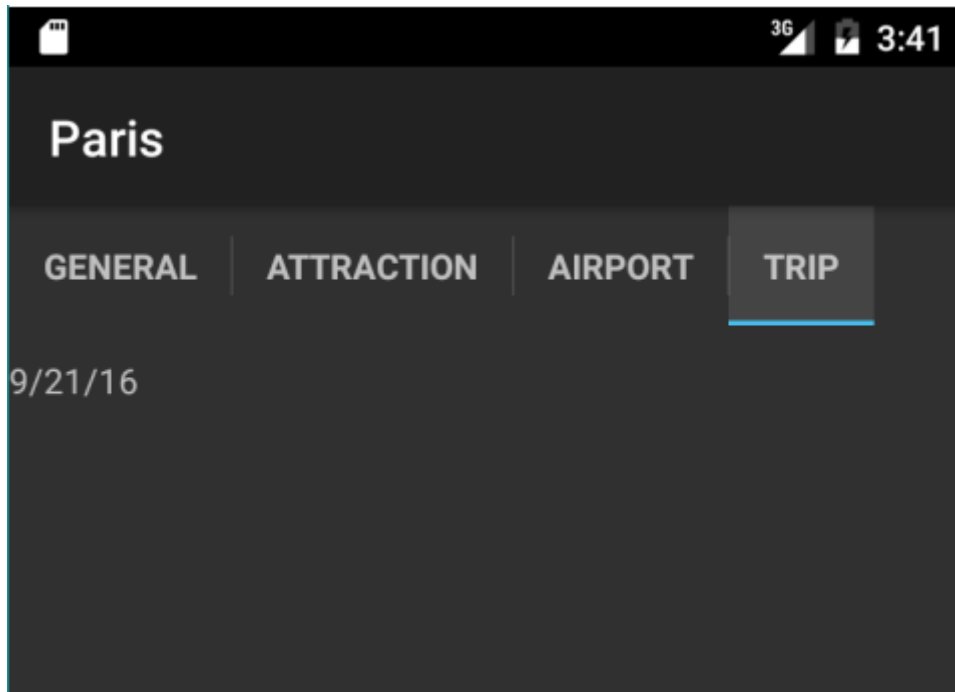
Presionamos nuevamente F5:

Abre el emulador, seleccionamos la URL de KB y vemos nuestra imagen de la aplicación, que está asociada al dashboard... y ahora vemos la lista de países como galería de banderas:



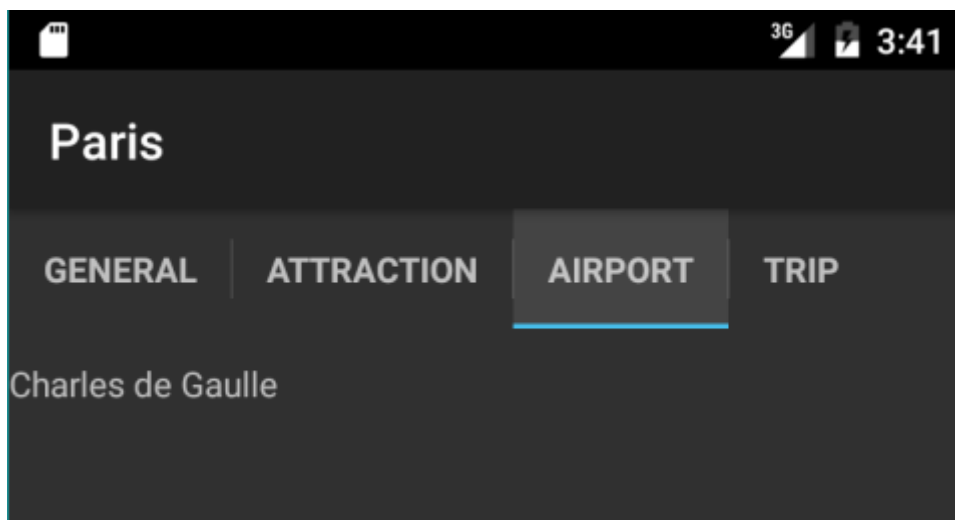
Seleccionamos Francia, y vemos su información.

Elegimos París... y para París, su lista de excursiones.



Y las excursiones que se visualizan tienen efectivamente fecha posterior a la actual.

Detengámonos ahora en los aeropuertos... se ve solamente el nombre, pero sería de utilidad para los usuarios de la aplicación poder visualizar también el mapa con la ubicación de los aeropuertos.

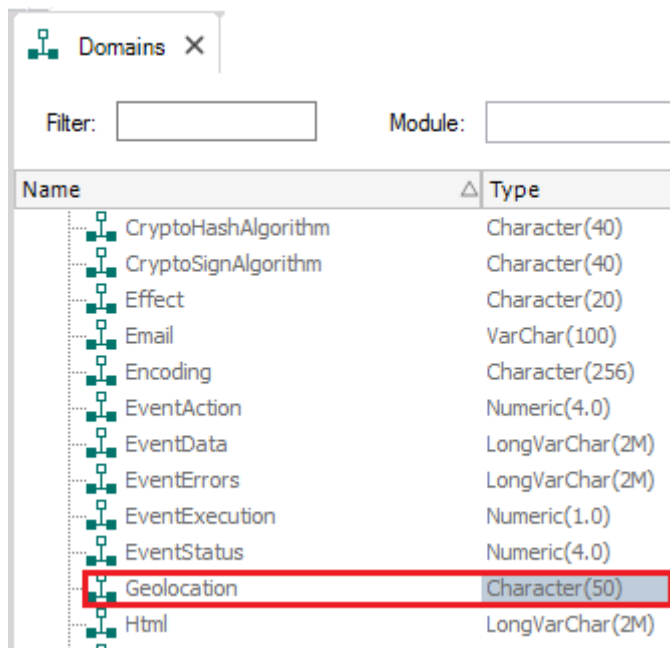


Volvamos a GeneXus...

Vamos a definir un nuevo atributo en la transacción Airport que nos permita almacenar la información que necesitamos.

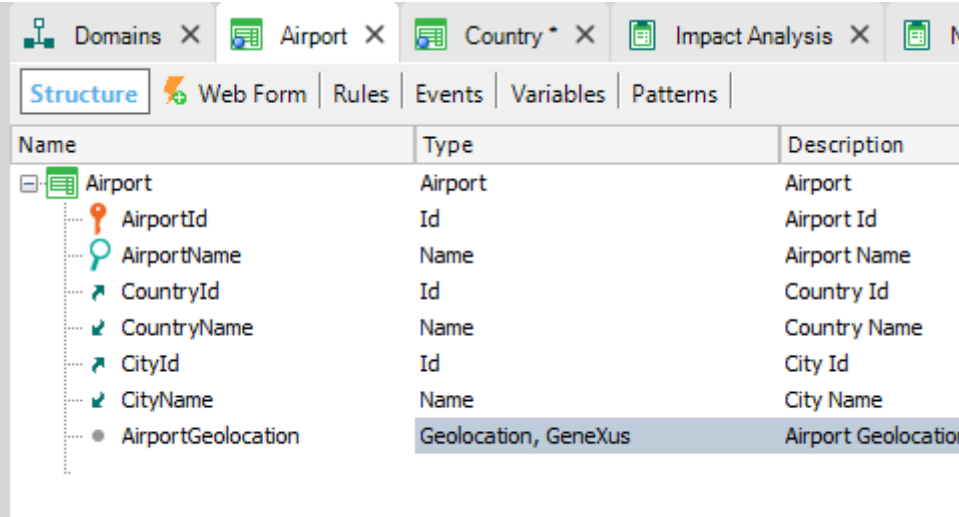
Dicha información debe estar basada en algún tipo de dato que permita que se genere la información geográfica necesaria para que en el dispositivo se visualice el mapa con la ubicación.

Para eso GeneXus nos ofrece el dominio Geolocation:



Se trata de un tipo de dato que permite administrar la información geográfica en términos de latitud y longitud.

El atributo se llama entonces AirportGeolocation, y ya hemos reorganizado y cargado la latitud y longitud correspondiente a la ubicación de cada aeropuerto. Con esta información en nuestra base de datos, el dispositivo mostrará el mapa con la correspondiente ubicación.

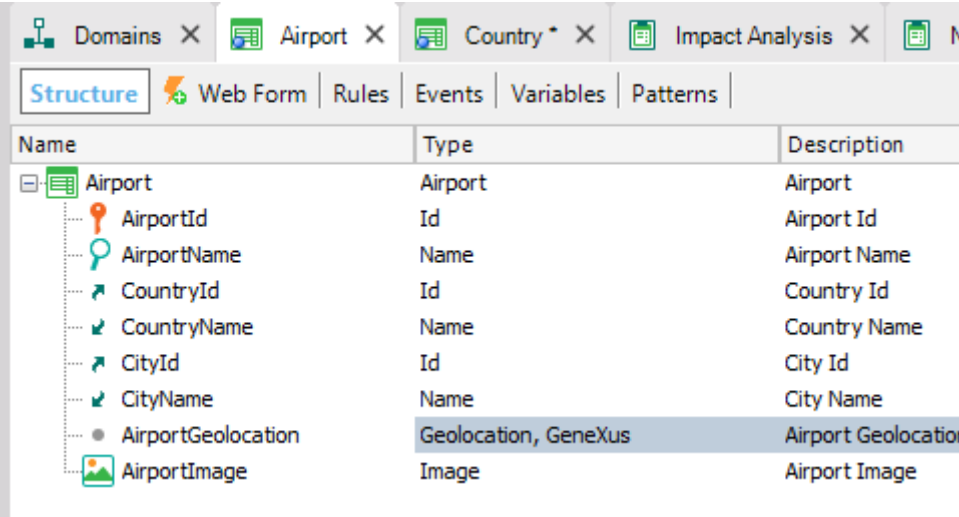


The screenshot shows the 'Structure' tab in the GeneXus IDE. The 'Airport' entity is expanded, showing its attributes. The 'AirportGeolocation' attribute is highlighted, showing its type as 'Geolocation, GeneXus' and its description as 'Airport Geolocation'.

Name	Type	Description
Airport	Airport	Airport
AirportId	Id	Airport Id
AirportName	Name	Airport Name
CountryId	Id	Country Id
CountryName	Name	Country Name
CityId	Id	City Id
CityName	Name	City Name
AirportGeolocation	Geolocation, GeneXus	Airport Geolocation

Y para que la aplicación sea aún más atractiva, agregaremos una imagen de cada aeropuerto.

Así que hemos creado también el atributo AirportImage y cargado las fotos de los aeropuertos:



The screenshot shows the 'Structure' tab in the GeneXus IDE. The 'Airport' entity is expanded, showing its attributes. The 'AirportImage' attribute is highlighted, showing its type as 'Image' and its description as 'Airport Image'.

Name	Type	Description
Airport	Airport	Airport
AirportId	Id	Airport Id
AirportName	Name	Airport Name
CountryId	Id	Country Id
CountryName	Name	Country Name
CityId	Id	City Id
CityName	Name	City Name
AirportGeolocation	Geolocation, GeneXus	Airport Geolocation
AirportImage	Image	Airport Image

Antes de ejecutar la aplicación, verifiquemos en el Work With for Devices que esté presente toda la información tal como deseamos:

Seleccionemos la sección Airport...

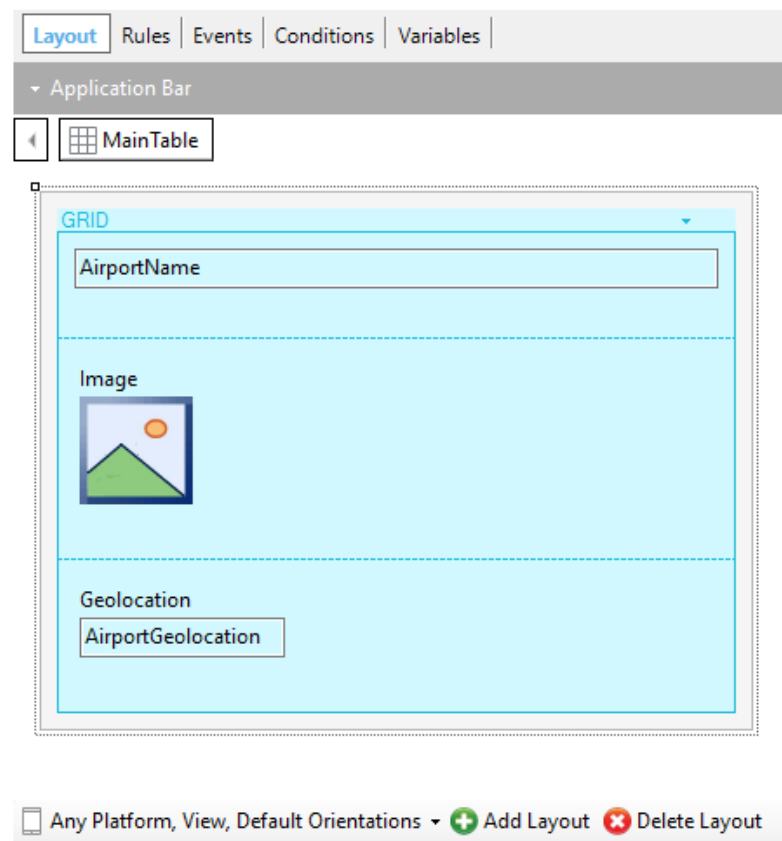
Vamos a personalizar un poco la vista...

Y para aprovechar mejor el espacio en la pantalla del dispositivo, vamos a ubicar los controles en forma vertical en lugar de horizontal que es como se visualiza por defecto... Lo hacemos en el grid:

Agreguemos los datos de geo localización para que se pueda ver el mapa con la ubicación

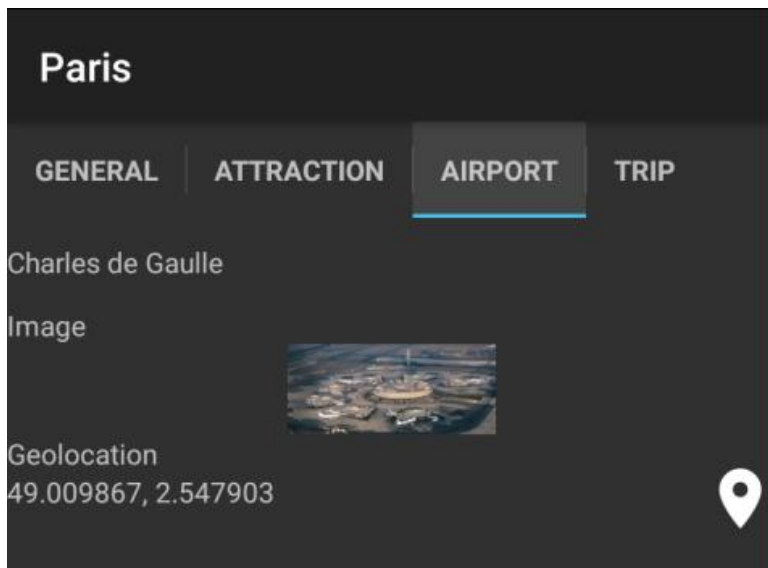
Y en la propiedad Label position de cada atributo indicamos que la posición de la etiqueta sea Top

Insertamos el atributo AirportGeolocation y modificamos el caption de la etiqueta para que solamente diga Geolocation.



Ahora sí, veamos la aplicación nuevamente en ejecución:

Seleccionamos Francia, Paris, Airport, y vemos lo siguiente:



Para ver el mapa presionemos aquí:



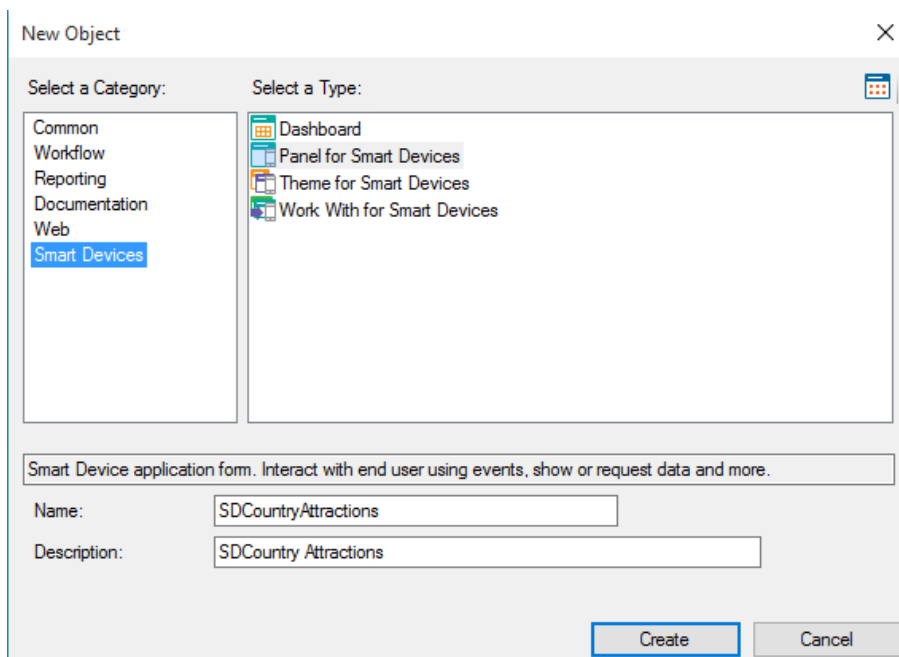
Y ahora por último, supongamos que la agencia nos solicita que en la pantalla con la galería de países, se ofrezca también un botón que al presionarlo nos muestre algo así:



Es decir, todos los países listados y para cada uno de ellos sus atracciones registradas.

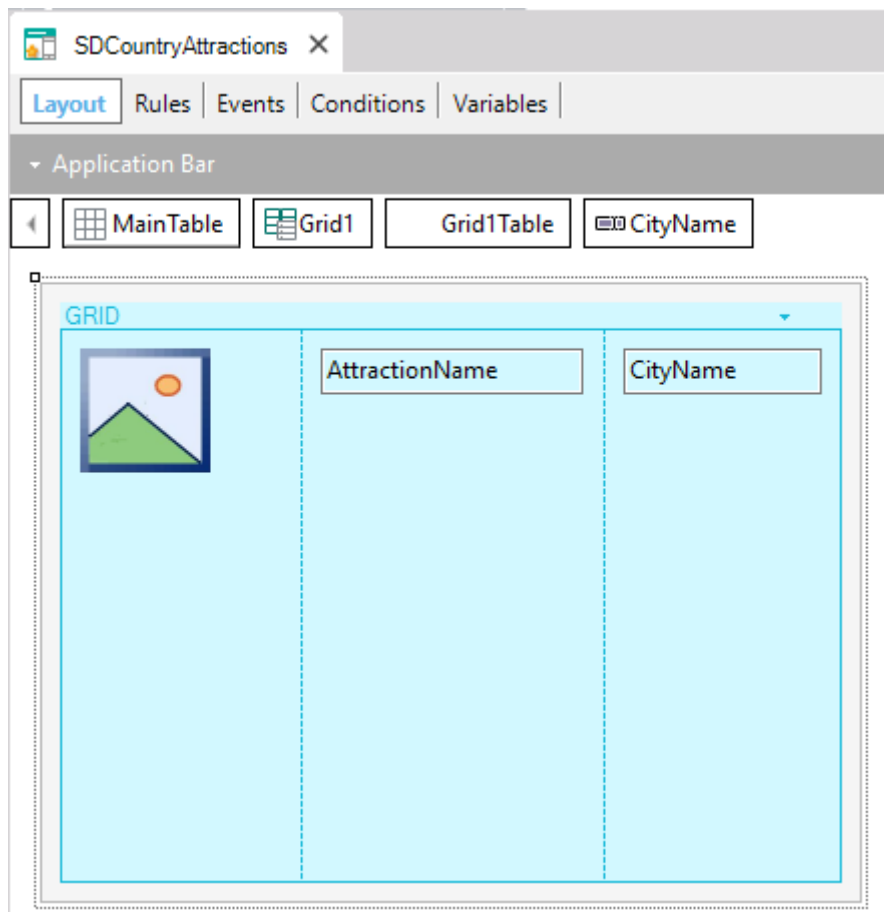
De esta forma el usuario final podrá tener una visión general de cuáles y la cantidad de atracciones ofrece cada país.

Nuevamente en GeneXus, creamos un nuevo objeto, esta vez de tipo Panel for Smart Devices, que es una especie de web panel, pero para Smart Devices:



Desde la toolbox arrastramos un grid, e insertamos los atributos AttractionName, AttractionImage y CityName.

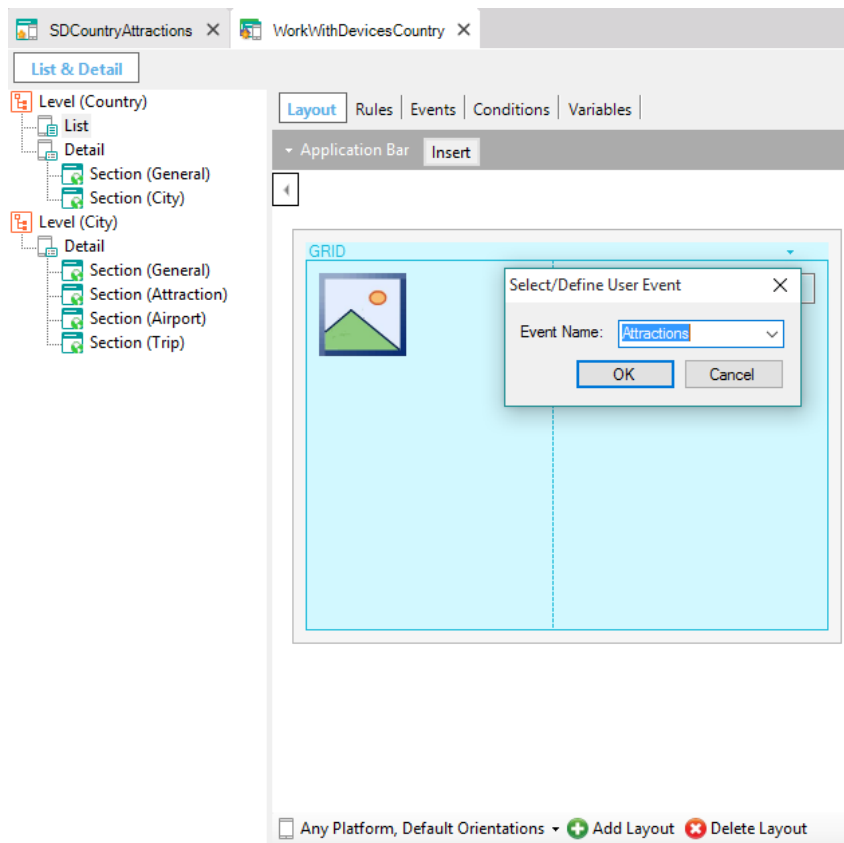
Verificamos que no se muestren las etiquetas. Para eso editamos las propiedades de cada atributo y en la propiedad Label Position ponemos None



Para que el usuario final pueda acceder a esta información desde la pantalla principal, vamos a agregar el botón que habíamos mencionado en la sección List de países del objeto Work With for Smart Devices.

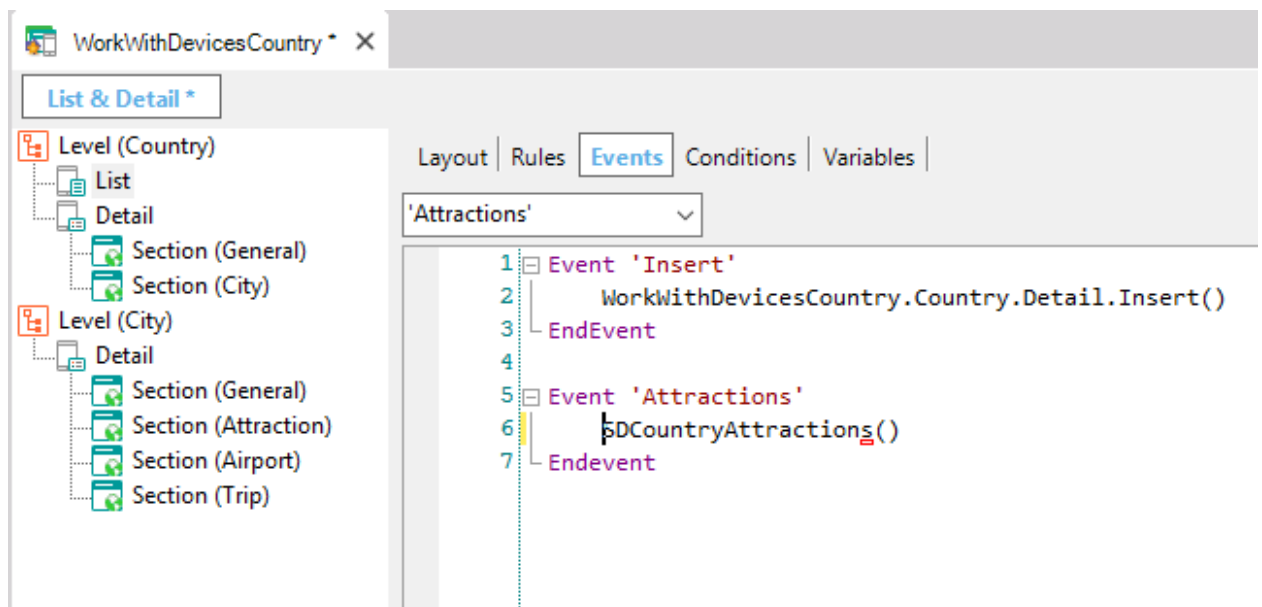
Así que desde la toolbox arrastramos un botón sobre la application bar.

En el event name del botón ponemos el nombre: Attractions



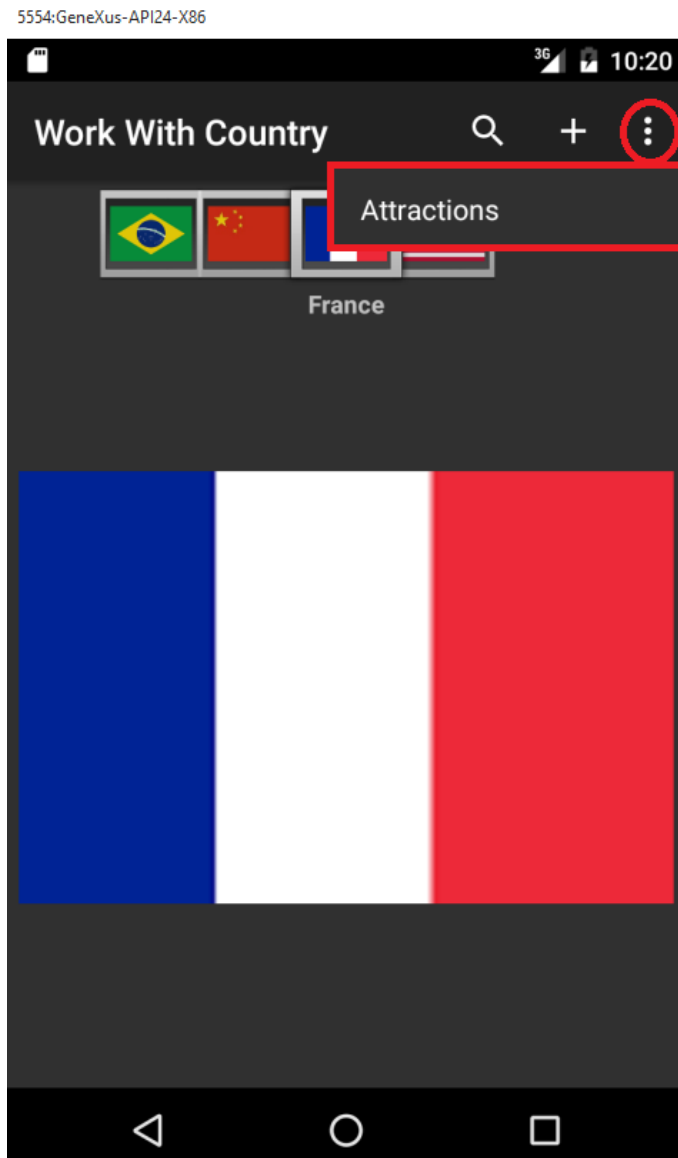
Hacemos doble clic sobre el botón y nos posicionamos en la sección de eventos.

Desde KB Explorer arrastramos el objeto SDCountriesAttractions y completamos el llamado con dos paréntesis, tal como hemos hecho en la aplicación web.

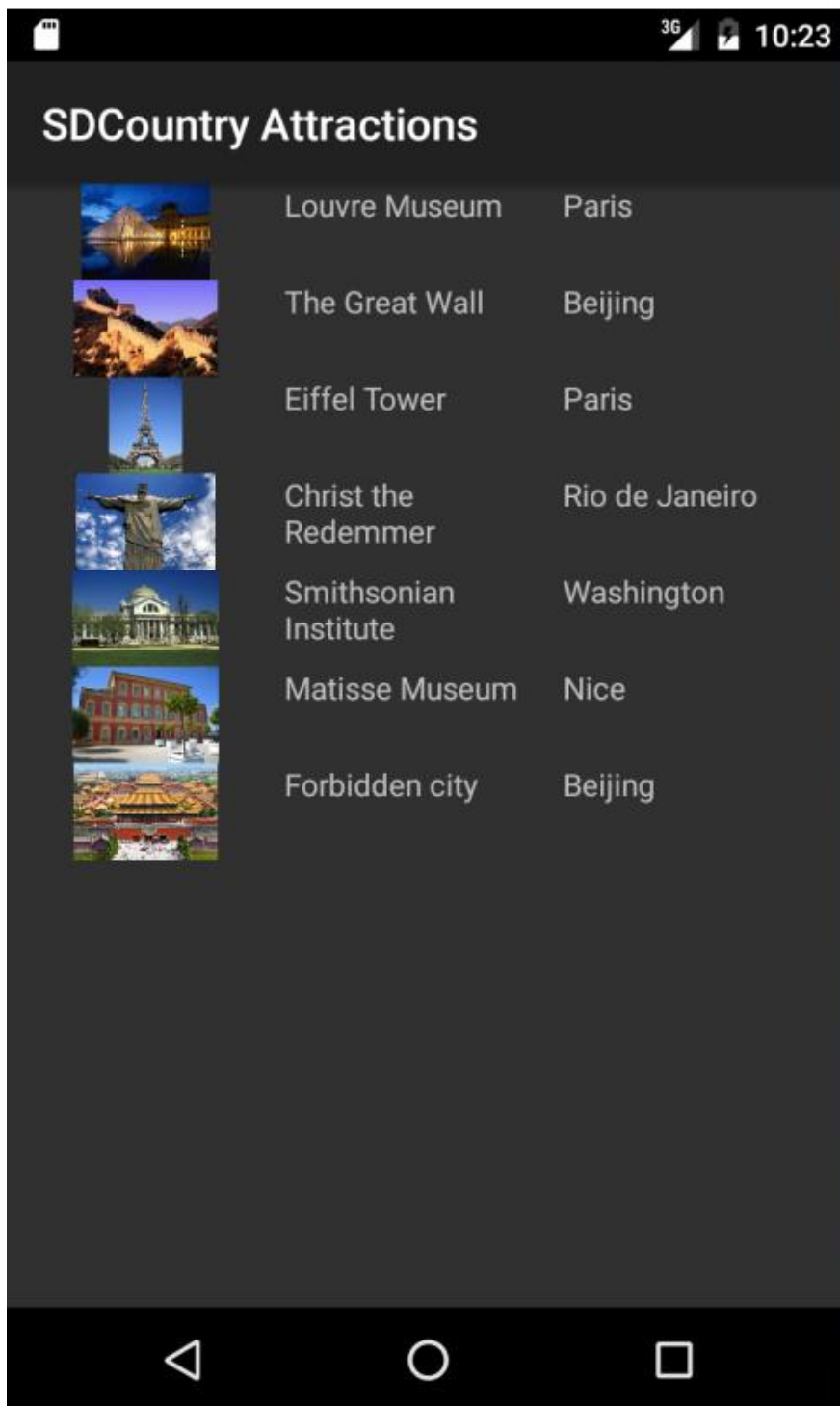


Presionemos F5...

En la pantalla List, si seleccionamos el botón menú del emulador, se ofrece la nueva opción:



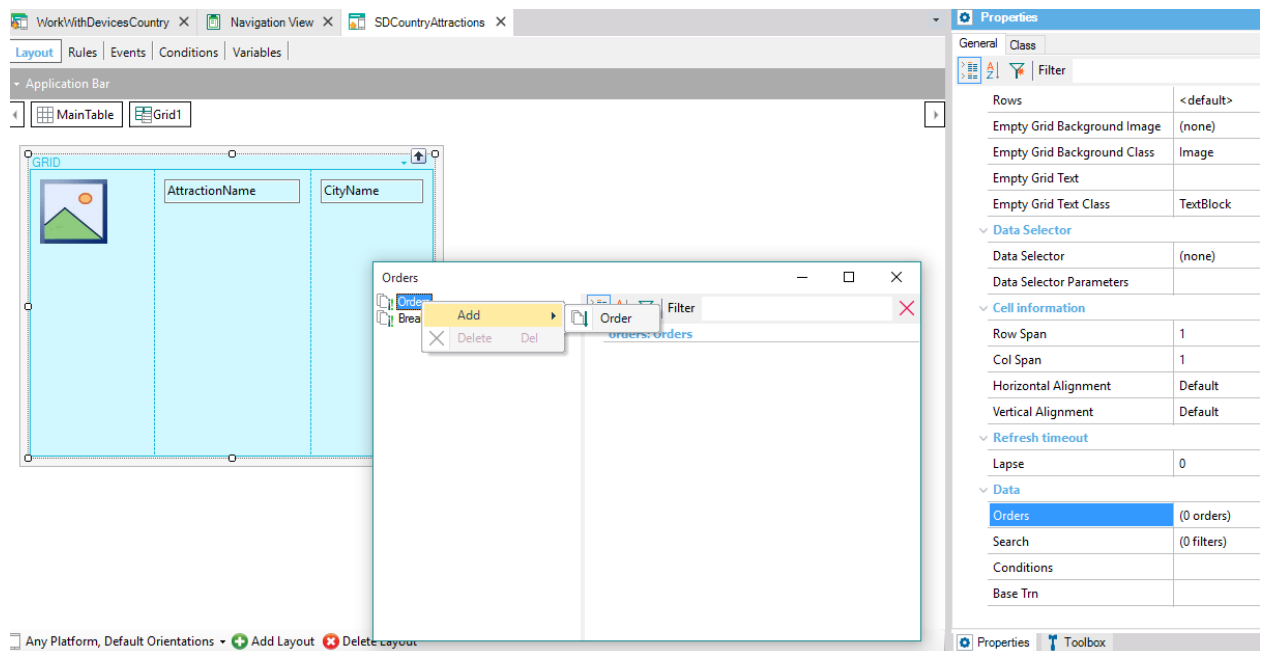
Seleccionamos el botón Attractions... y vemos todas las atracciones registradas:



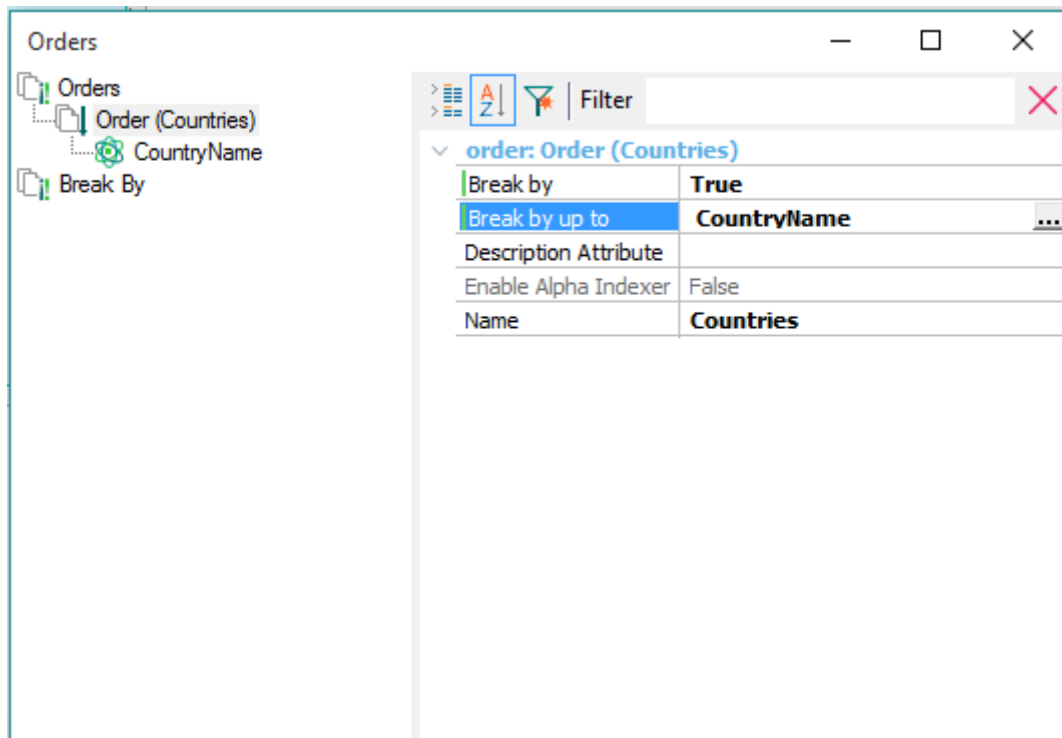
En este listado las ciudades se repiten, por ende los países se repiten también aunque no los estemos visualizando. Pretendíamos otro formato, así que vamos a agrupar entonces las atracciones por país.

Volvamos a GeneXus---

Seleccionamos el Grid para ver sus propiedades y observemos esta propiedad Orders. Desde aquí vamos a definir un nuevo orden de nombre Countries.



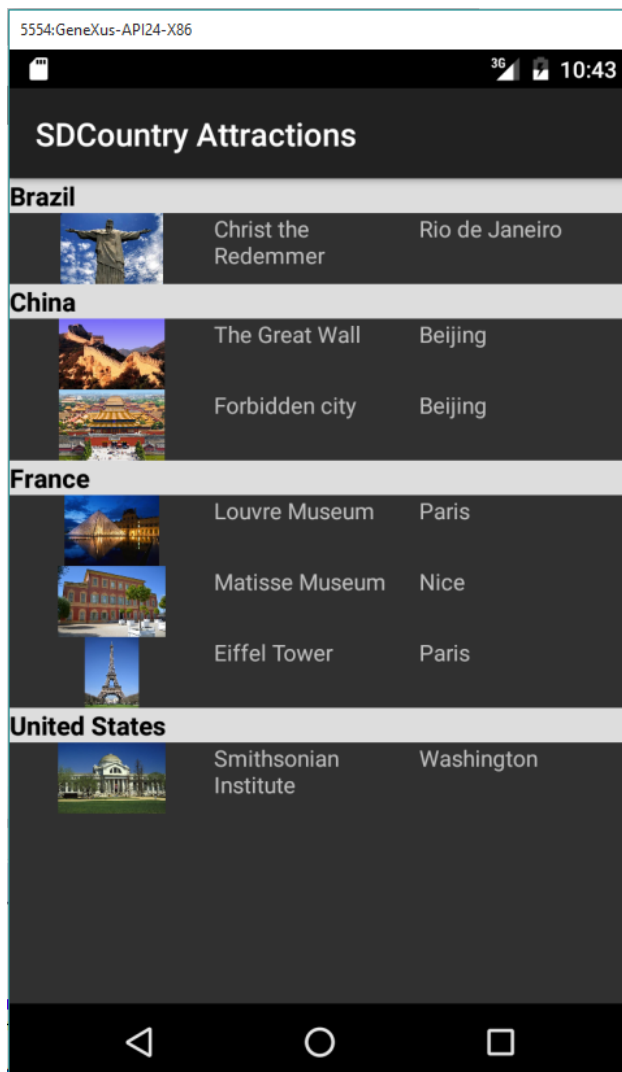
Hacemos clic con el botón derecho Add Attribute y seleccionamos el atributo CountryName ya que es el atributo seleccionado para el orden. Observemos las propiedades de este Order definido. En la propiedad Break by elegimos True. Indicamos que el corte se hará por CountryName y de esta forma de establecerá el agrupamiento:



y en la propiedad Description Attribute elegimos también CountryName para que se visualice el nombre del país.

Ejecutemos nuevamente la aplicación, presionemos F5...

Ejecutamos nuevamente la aplicación... presionamos el botón Attractions y vemos las atracciones agrupadas ahora por país:



De esta forma hemos desarrollado un pequeño módulo para smart devices con el fin de ofrecer a los usuarios finales una opción más de acceso a la información que maneja la agencia de viajes.