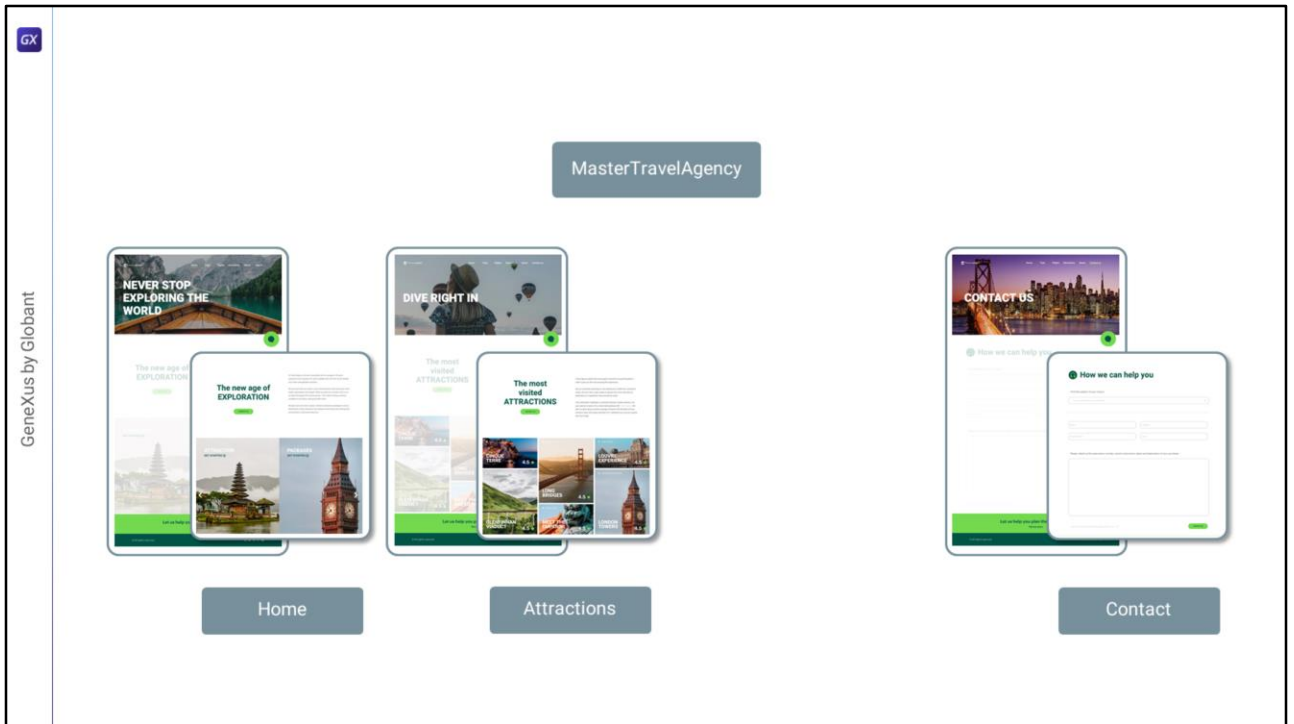


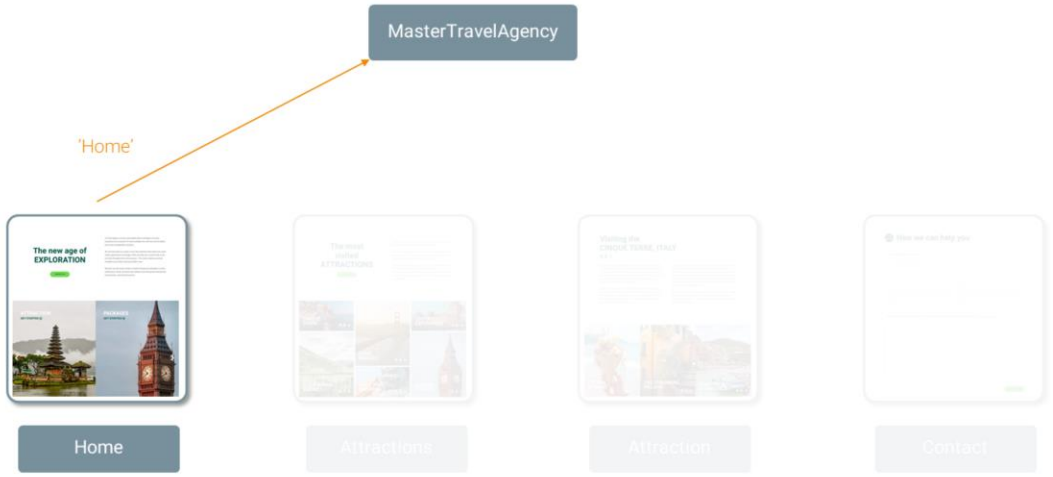
Master Panel: Header update and Navigations (part 2)



Cecilia Fernández



En el video anterior implementamos las navegaciones y carga correcta del Header del Master Panel...

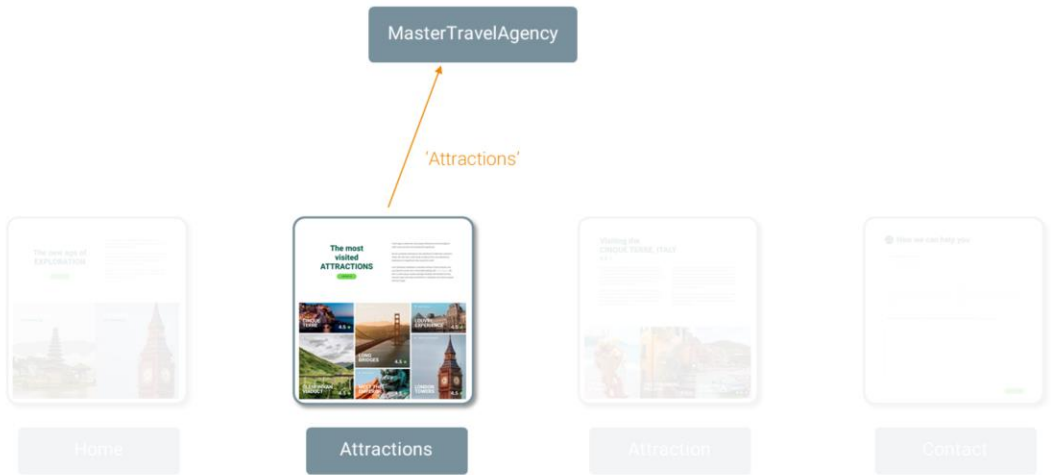


Event ClientStart

GlobalEvents.UpdateMasterPanelHeader('Home')

endevent

...de acuerdo a si se estaba cargando el Home...



Event ClientStart

```
GlobalEvents.UpdateMasterPanelHeader('Attractions')
```

endevent

...Attractions...

MasterTravelAgency

'Contact'



Home



Attractions



Attraction



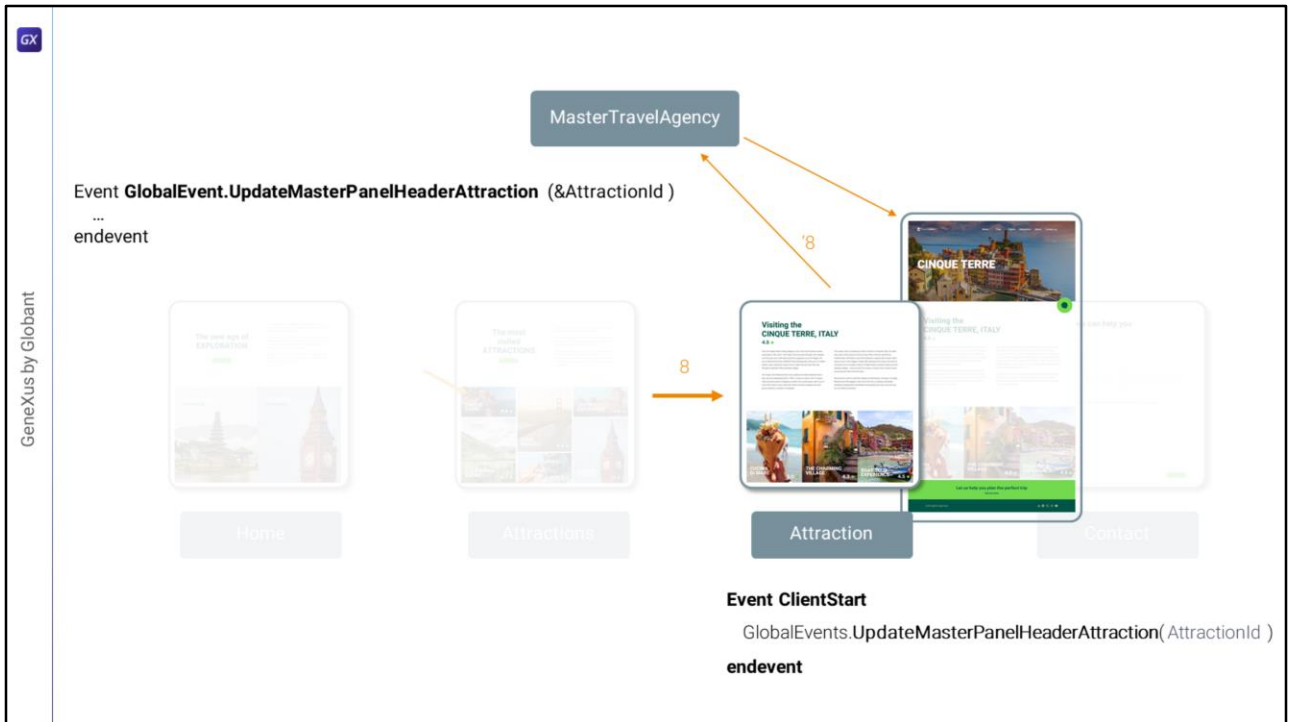
Contact

Event ClientStart

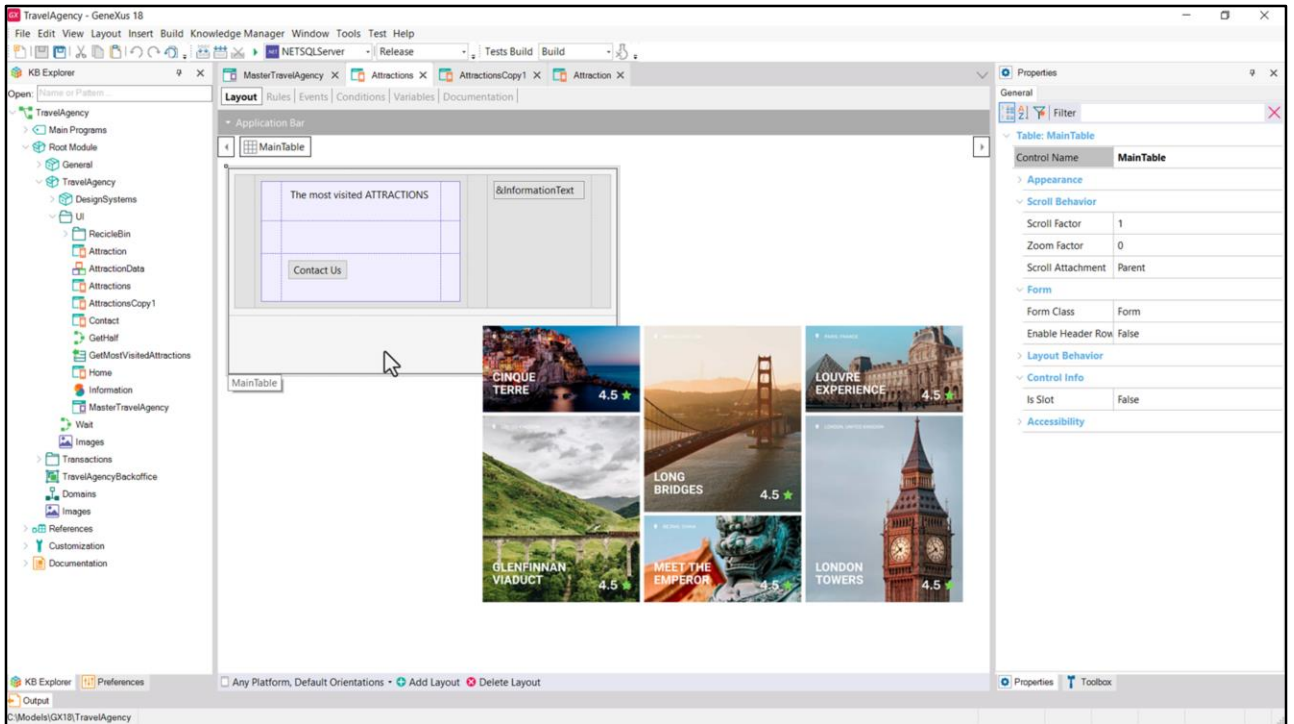
```
GlobalEvents.UpdateMasterPanelHeader('Contact')
```

endevent

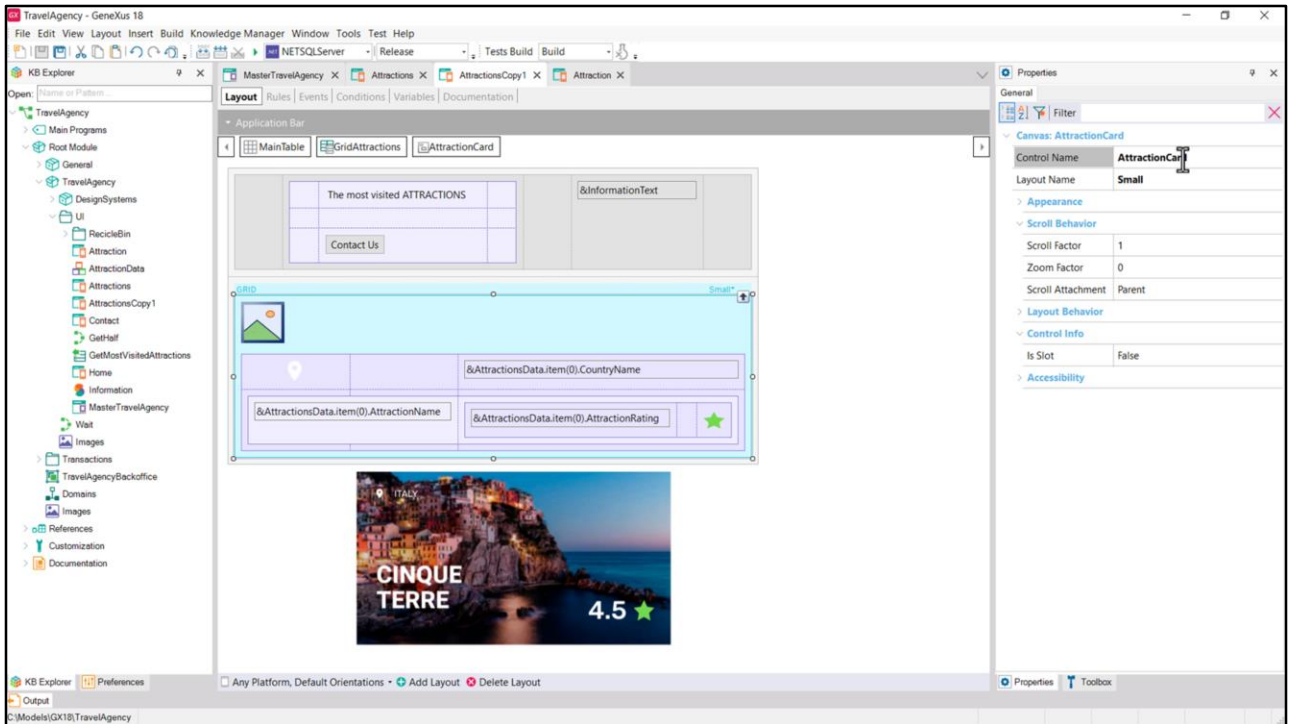
...o Contact...



Nos quedó pendiente implementar que el Master Panel cargue en el Header imagen y título de la atracción que recibió por parámetro el panel Attraction al ejecutarse. Y que lo recibió cuando el usuario seleccionó una atracción turística en este grid.

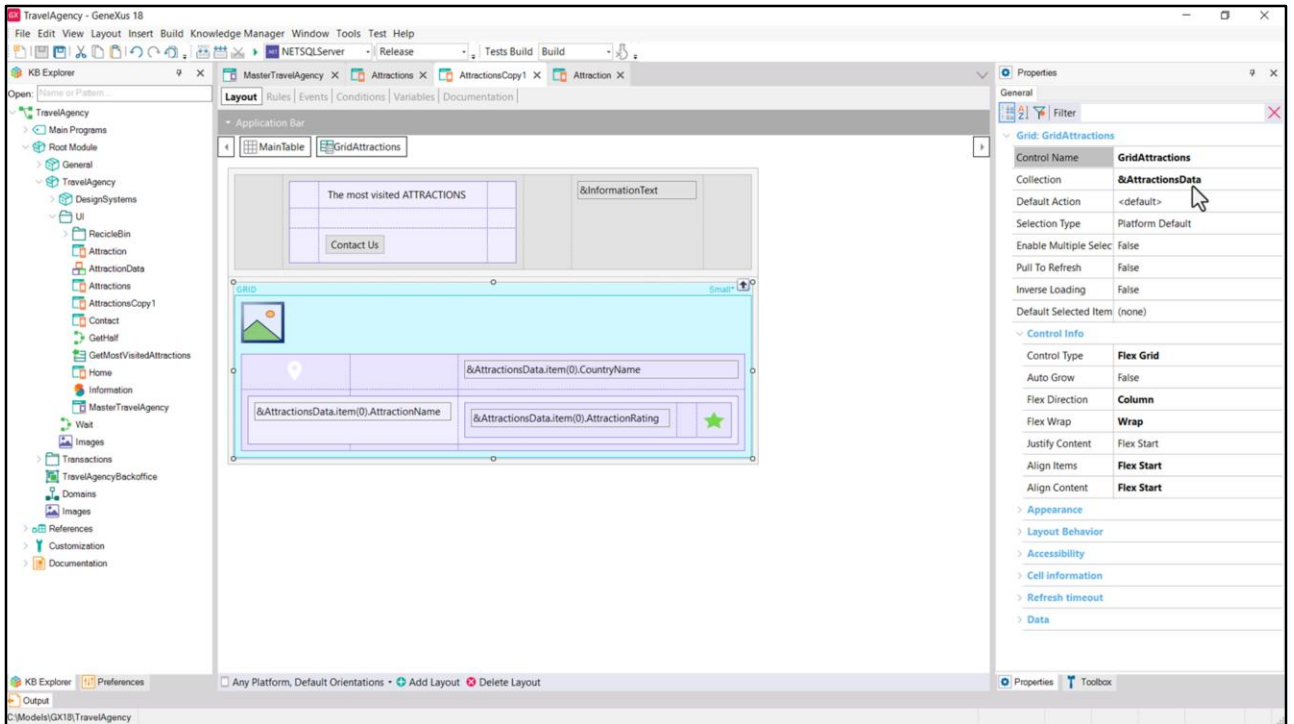


Para poder implementar esto y probarlo, entonces, necesitaré primero agregar al panel Attractions, en esta segunda fila, el carrusel con las atracciones turísticas cargadas de la base de datos...

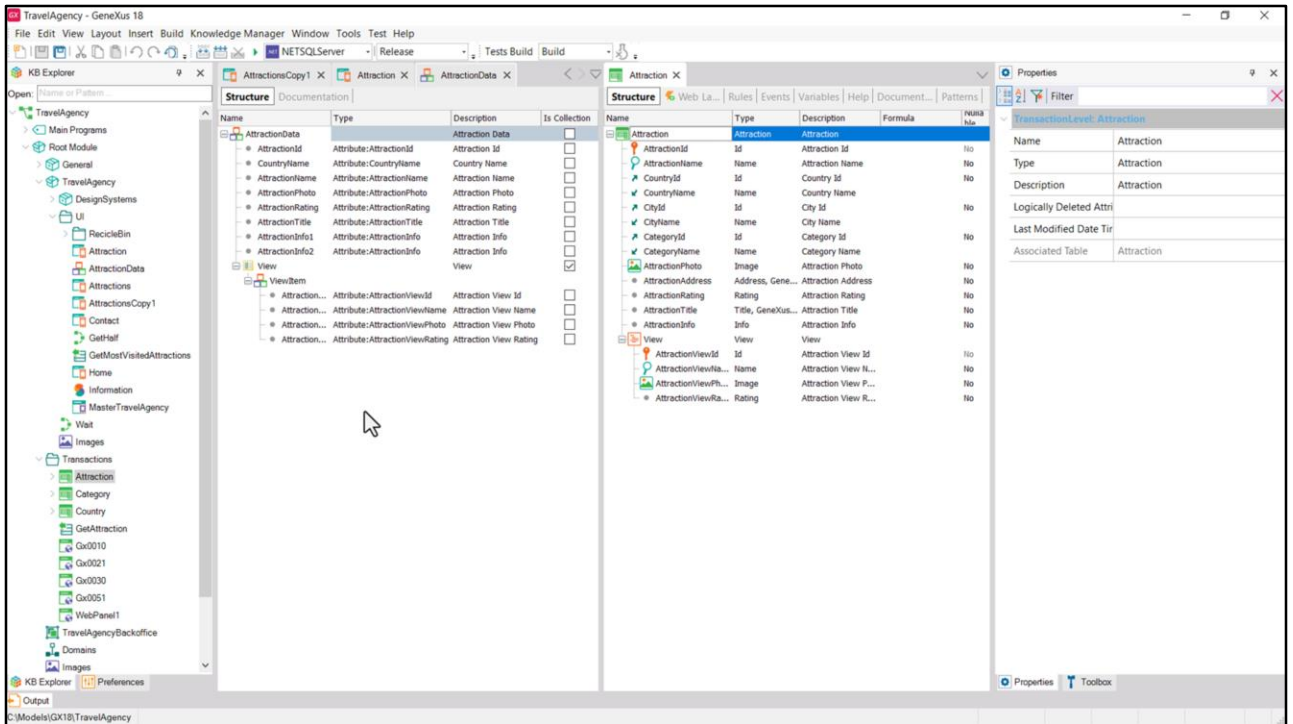


Aquí hice una copia del panel y lo implementé para que no perdamos tiempo ahora haciéndolo (en otro video veremos esta implementación en detalle, no nos preocupemos en este momento). Sólo repararemos en que se trata de un grid, por lo que carga dinámicamente sus ítems.

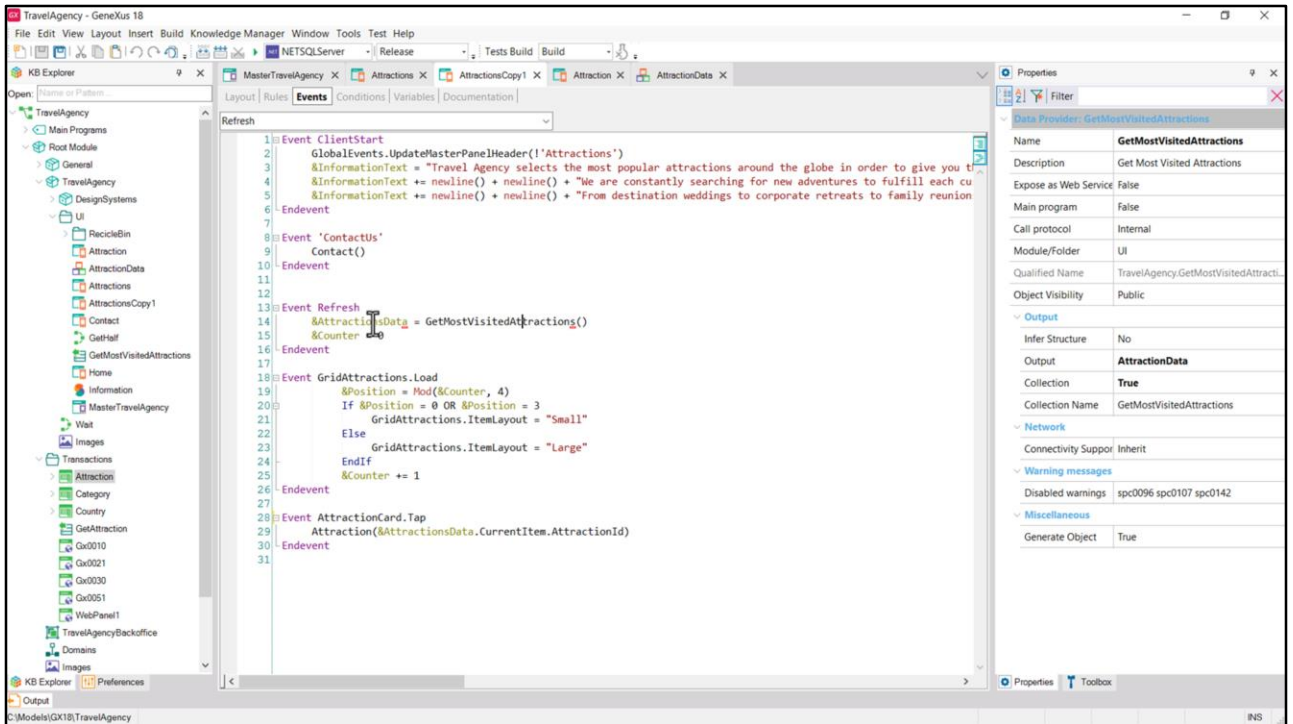
Cada ítem se trata de un canvas al que llamé AttractionCard, justamente para renderizar cada una de las atracciones turísticas como una card.



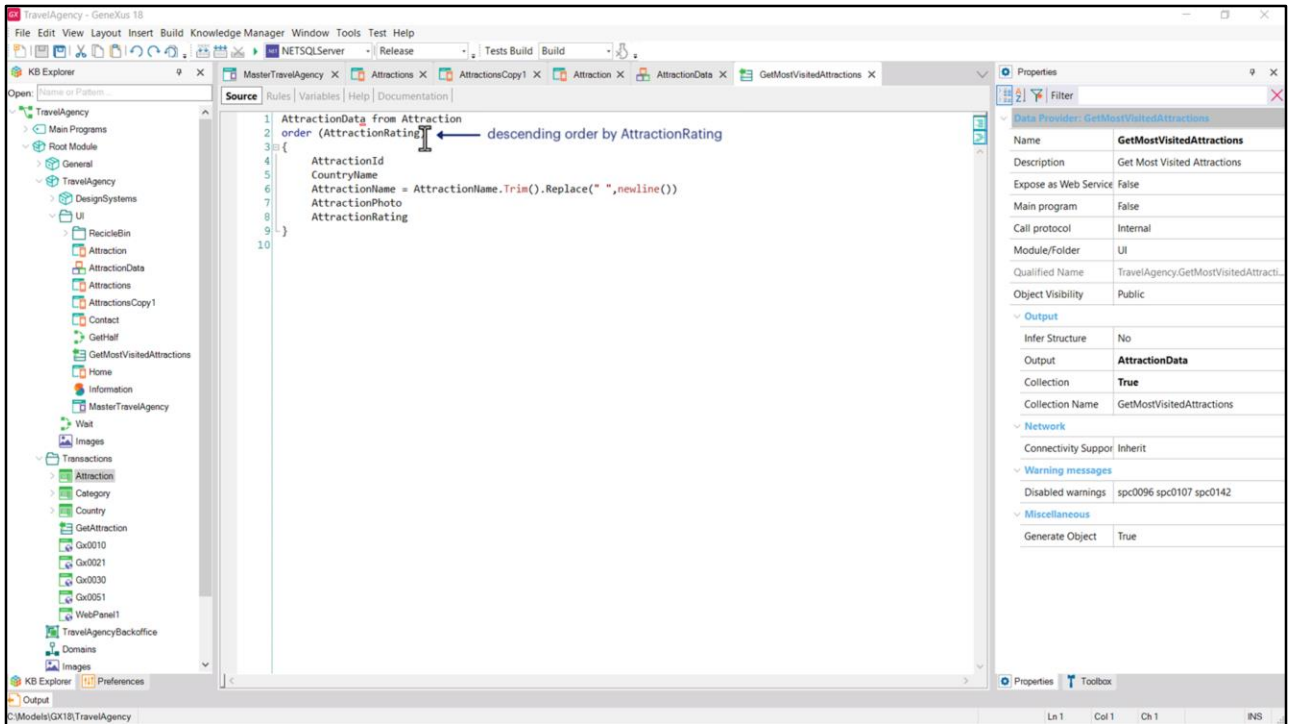
En verdad hay muchas maneras posibles de implementar este grid, aquí elegí una, donde cargo los ítems del grid en esta variable que es de tipo de datos una colección de este SDT.



Vean que es casi idéntico a la estructura de la transacción Attraction. Lo creé con los campos que voy a necesitar para los dos paneles Attractions y Attraction.

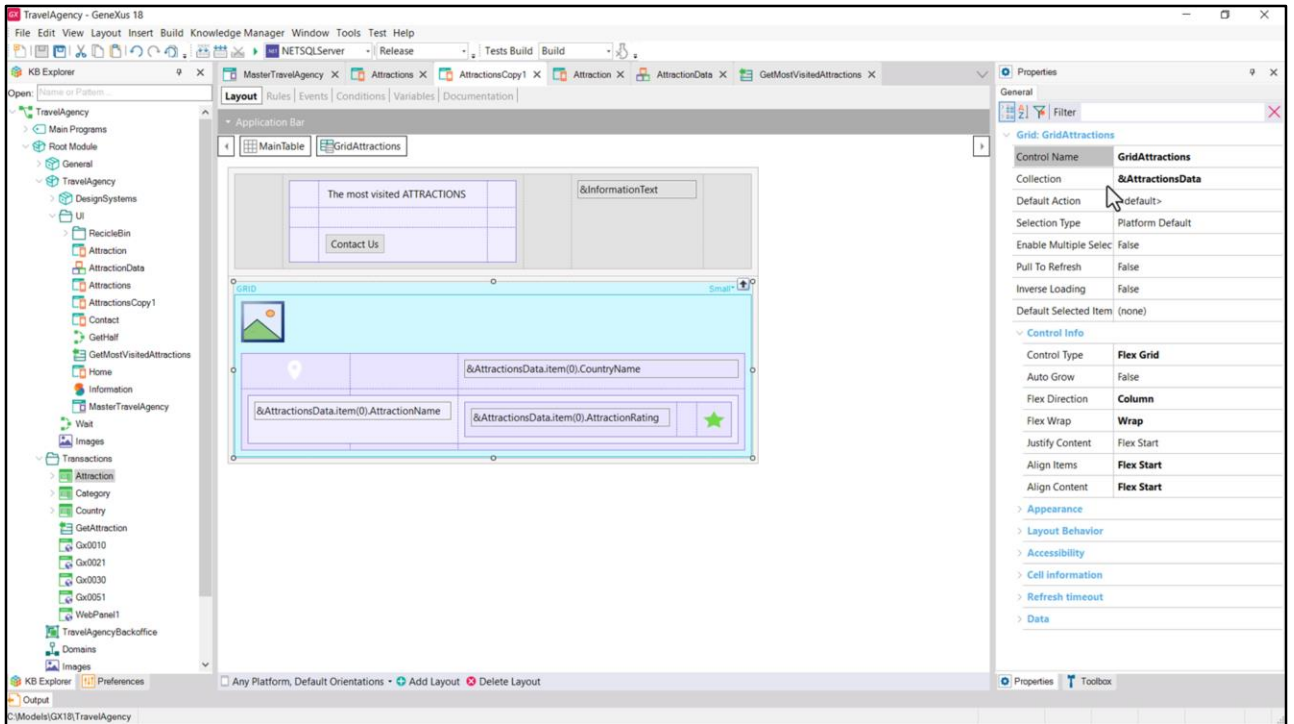


En definitiva lo que hago aquí es llamar a este data provider...

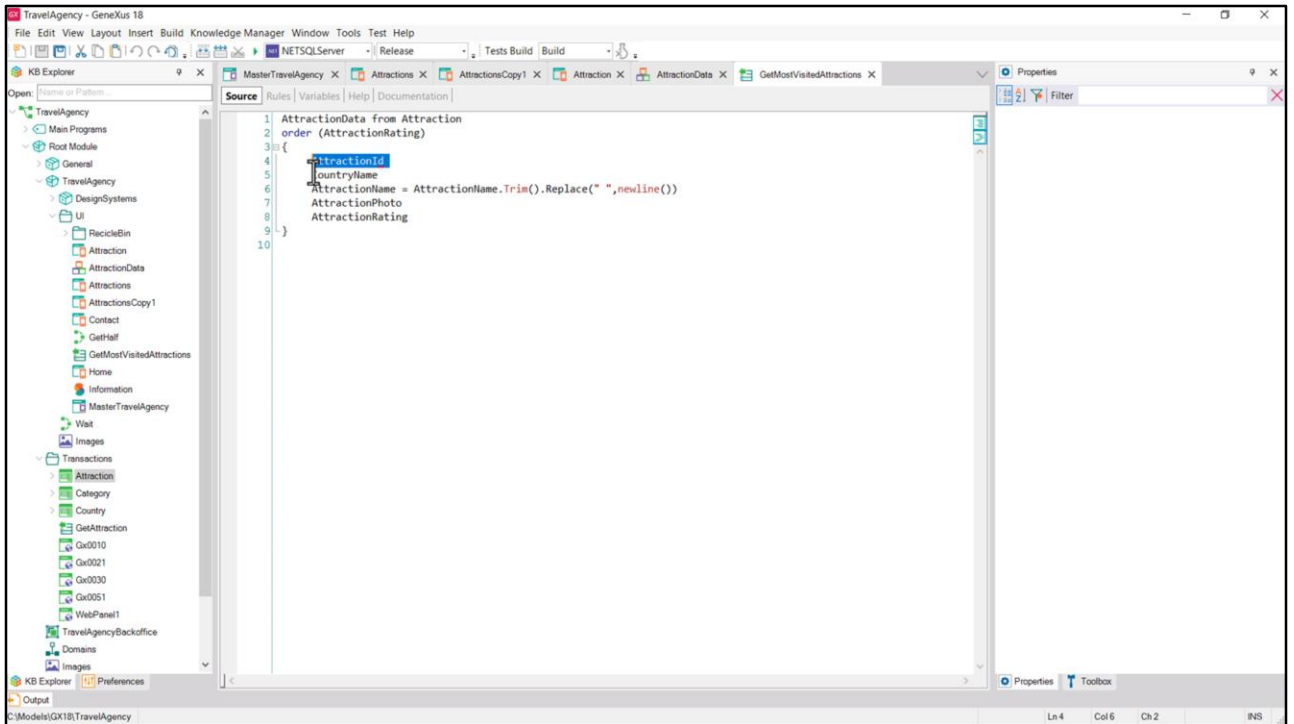


...que devolverá la colección del SDT cargada a partir de la base de datos.

Aquí le estamos indicando los campos de cada ítem que nos interesa obtener para mostrar en cada Card. Vean que no estamos cargando todos los elementos, sino sólo los que nos interesan en esta oportunidad.



Esta colección será la que se cargará automáticamente en el grid, puesto que así lo indiqué en su propiedad Collection.

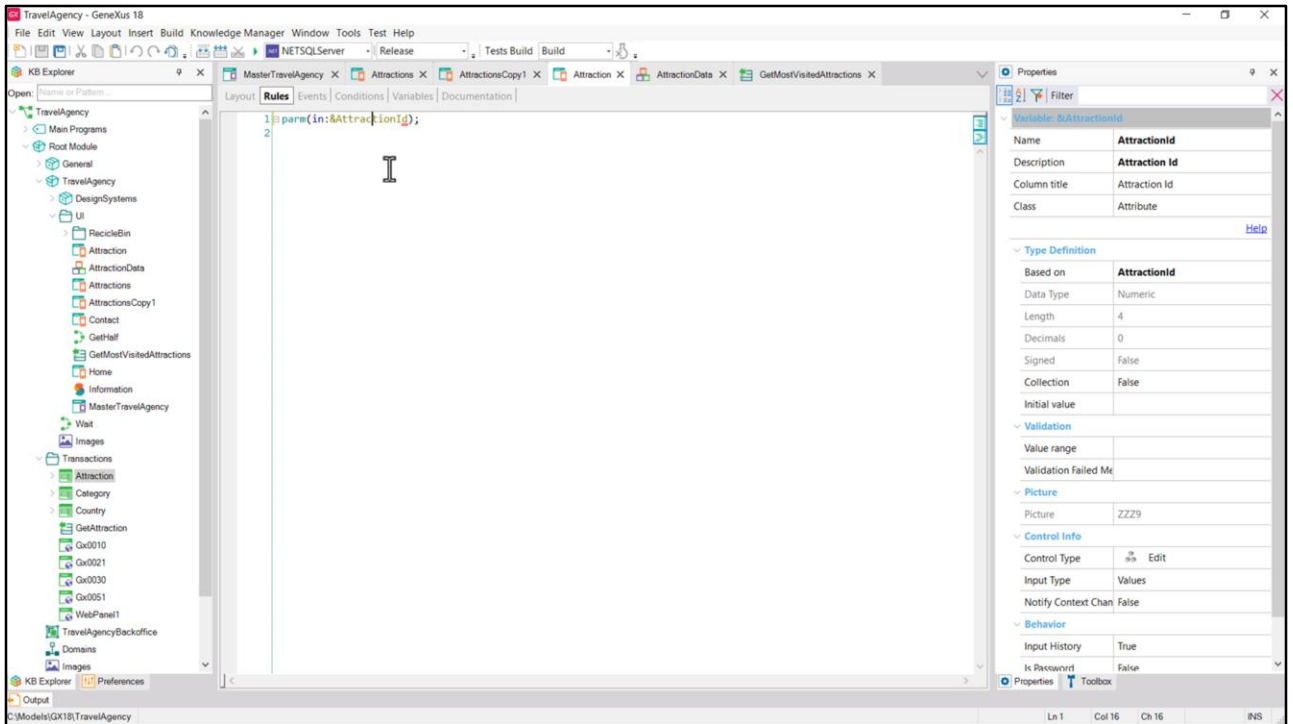


Repito que no importa ahora entender esto. Alcanza con saber que este grid se cargará a partir de este SDT colección que a su vez estará cargado con la información que nos interesaba de la base de datos.

En particular veamos que cada ítem de la colección vendrá con el AttractionId cargado.

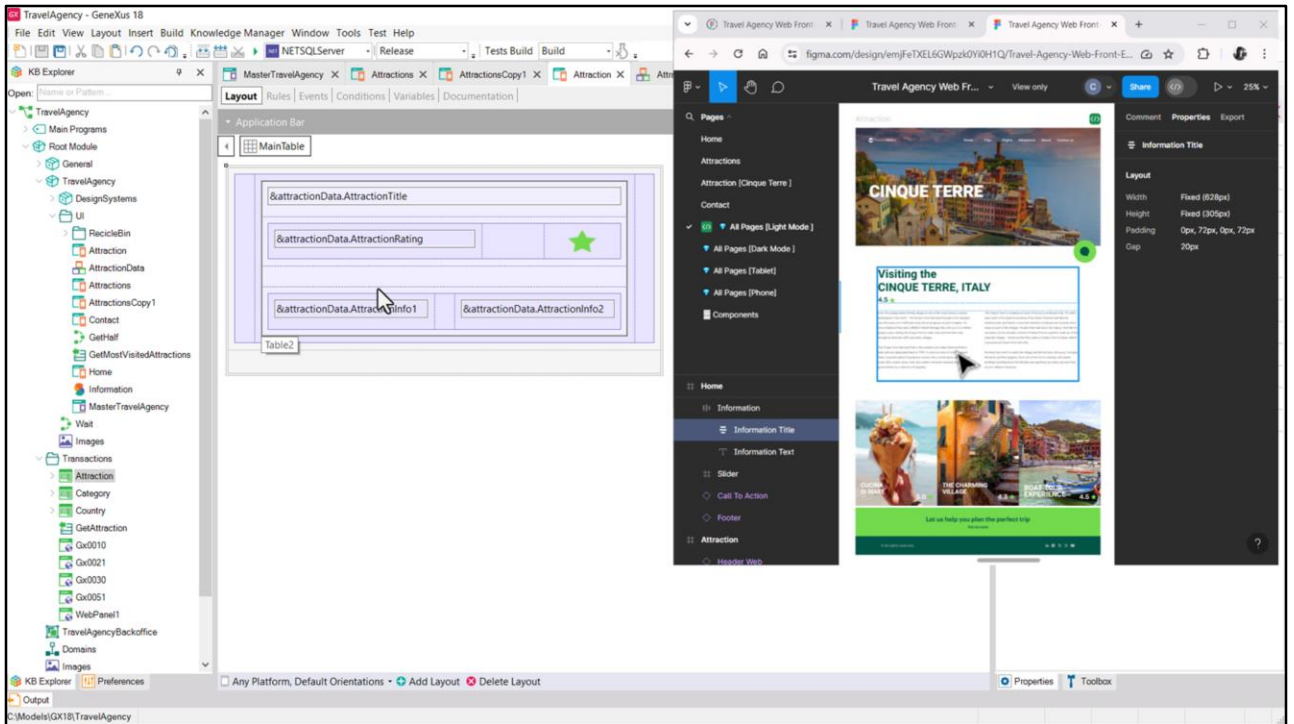
```
1 Event ClientStart
2   GlobalEvents.UpdateMasterPanelHeader(! 'Attractions')
3   &InformationText = "Travel Agency selects the most popular attractions around the globe in order to give you the
4   &InformationText += newline() + newline() + "We are constantly searching for new adventures to fulfill each cu
5   &InformationText += newline() + newline() + "From destination weddings to corporate retreats to family reunion
6 Endevent
7
8 Event 'ContactUs'
9   Contact()
10 Endevent
11
12
13 Event Refresh
14   &AttractionsData = GetMostVisitedAttractions()
15   &Counter = 0
16 Endevent
17
18 Event GridAttractions.Load
19   &Position = Mod(&Counter, 4)
20   If &Position = 0 OR &Position = 3
21     GridAttractions.ItemLayout = "Small"
22   Else
23     GridAttractions.ItemLayout = "Large"
24   EndIf
25   &Counter += 1
26 Endevent
27
28 Event AttractionCard.Tap
29   Attraction(&AttractionsData.CurrentItem.AttractionId)
30 Endevent
31
```

Entonces lo que hice aquí fue programar sobre el canvas que corresponde a cada ítem del grid, el evento tap (que es lo mismo que el click para web). Y allí lo que estoy haciendo es invocar al panel Attraction, pasándole por parámetro para el ítem cliqueado, el valor de AttractionId.

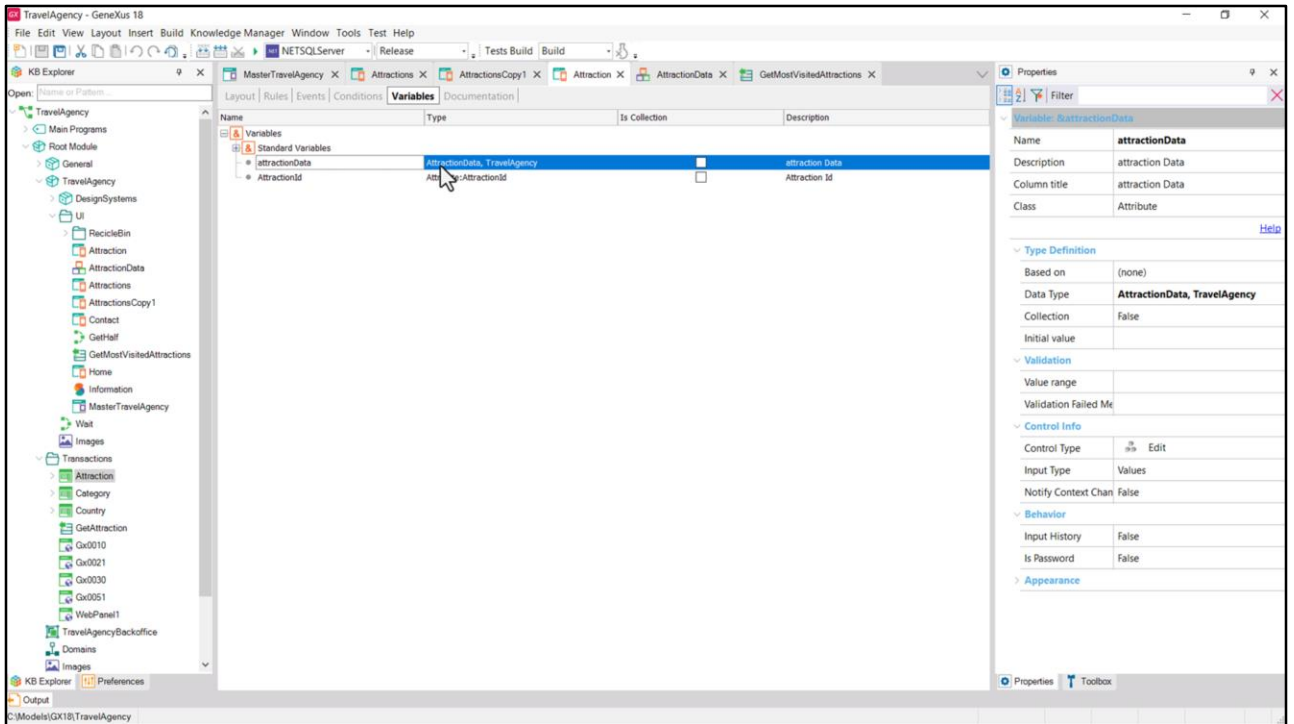


Bien, con esto implementamos la invocación.

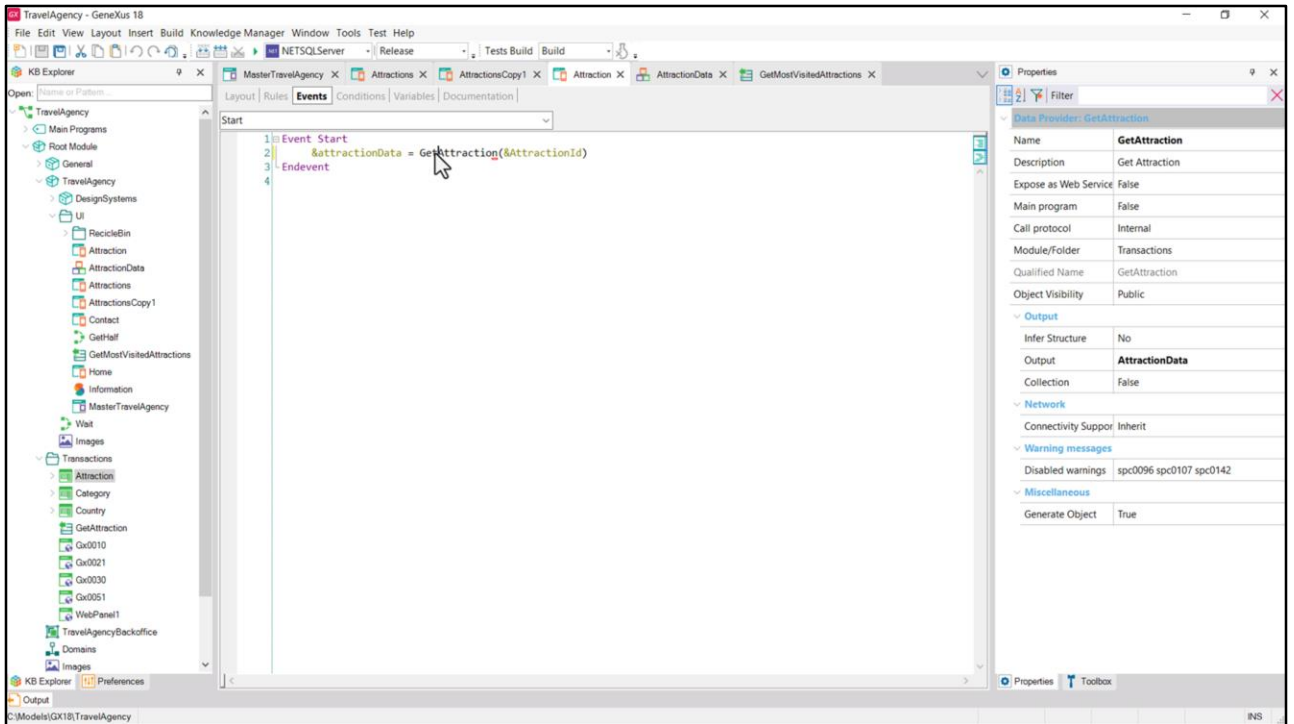
Luego, en Attraction ya avancé un poco también. Vean que definí la regla parm para que el panel pueda recibir por parámetro en esta variable el identificador de la atracción turística.



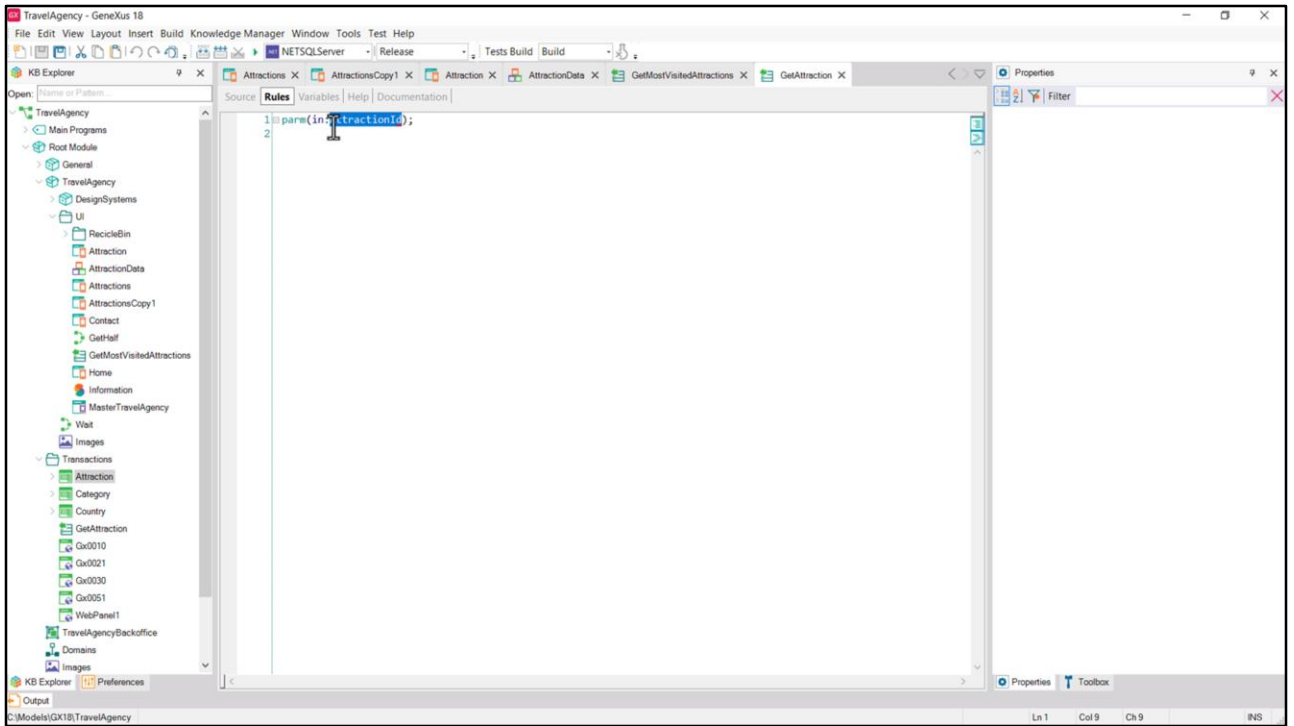
Y por ahora, en su layout solamente implementé esta parte...



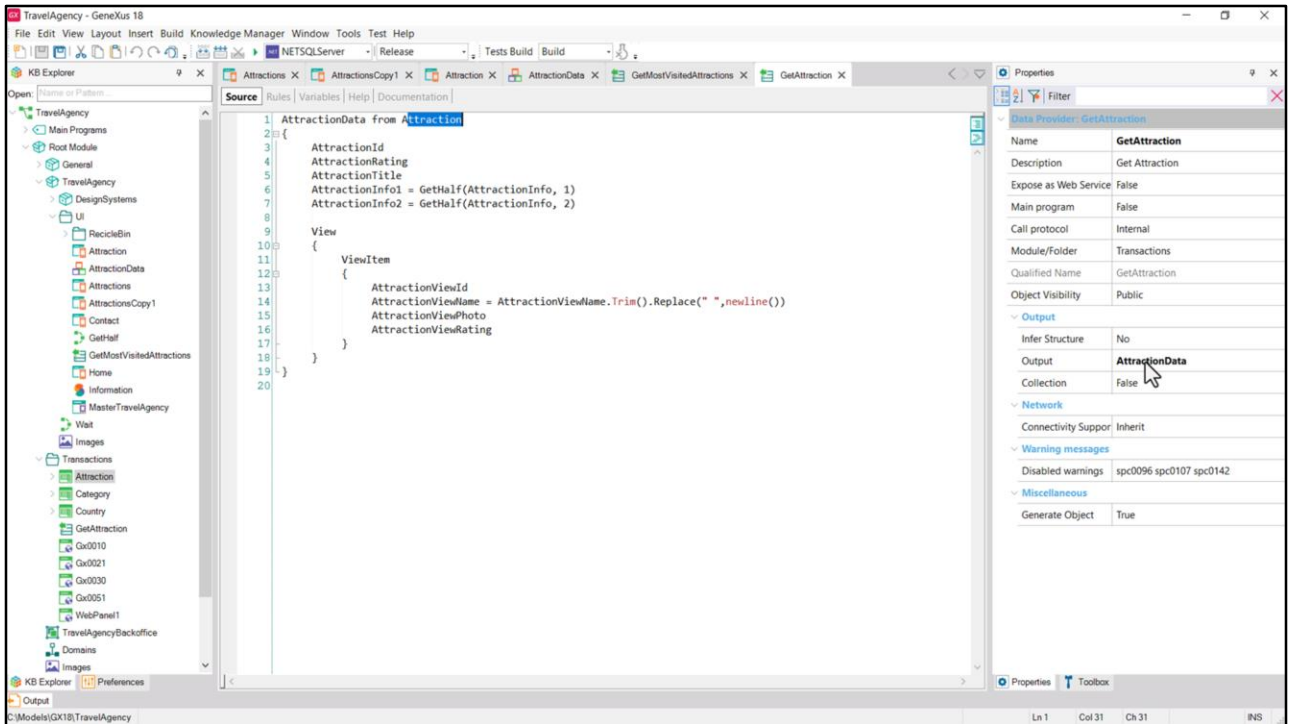
...para la cuál utilizo otra variable basada en el mismo SDT, pero esta vez la variable no es colección.



Y ahora llamo a otro Data Provider al que le paso el identificador de la atracción recibido por parámetro...



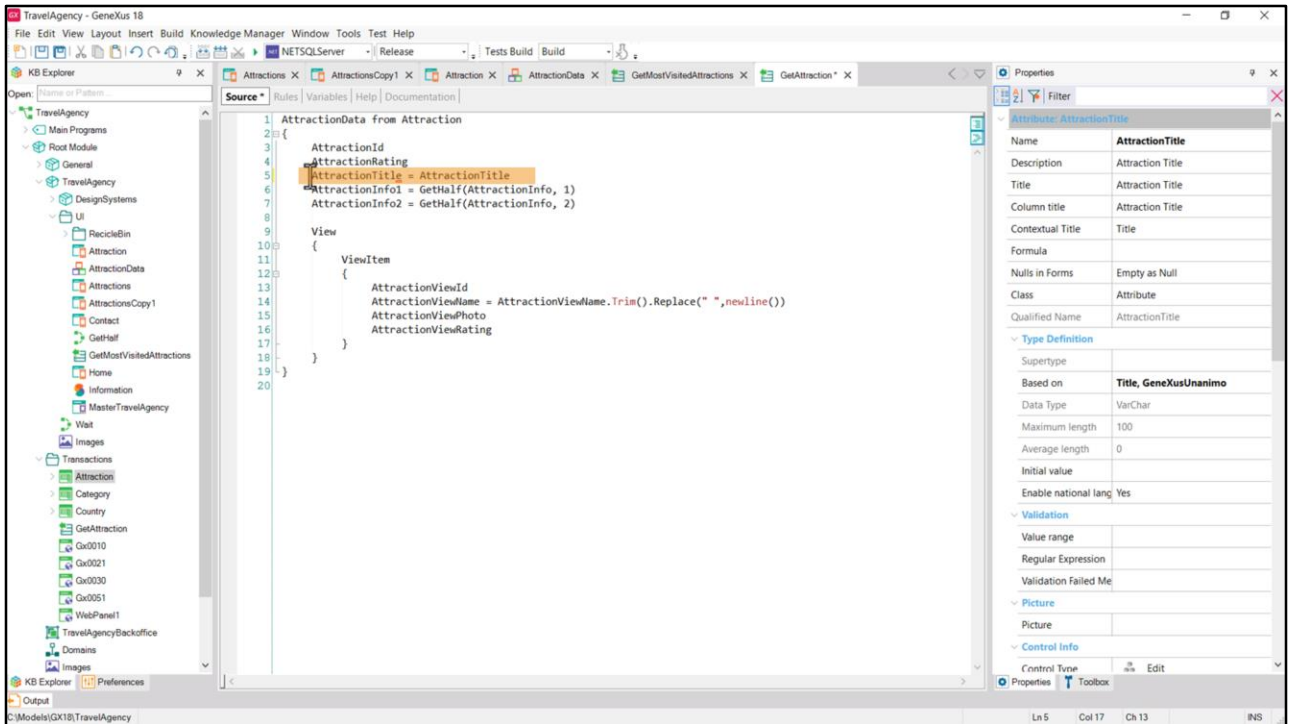
...y este Data Provider recibe ese valor en el atributo AttractionId...



...por lo que realizará un filtro automático...al recorrer en este grupo la transacción base Attraction...

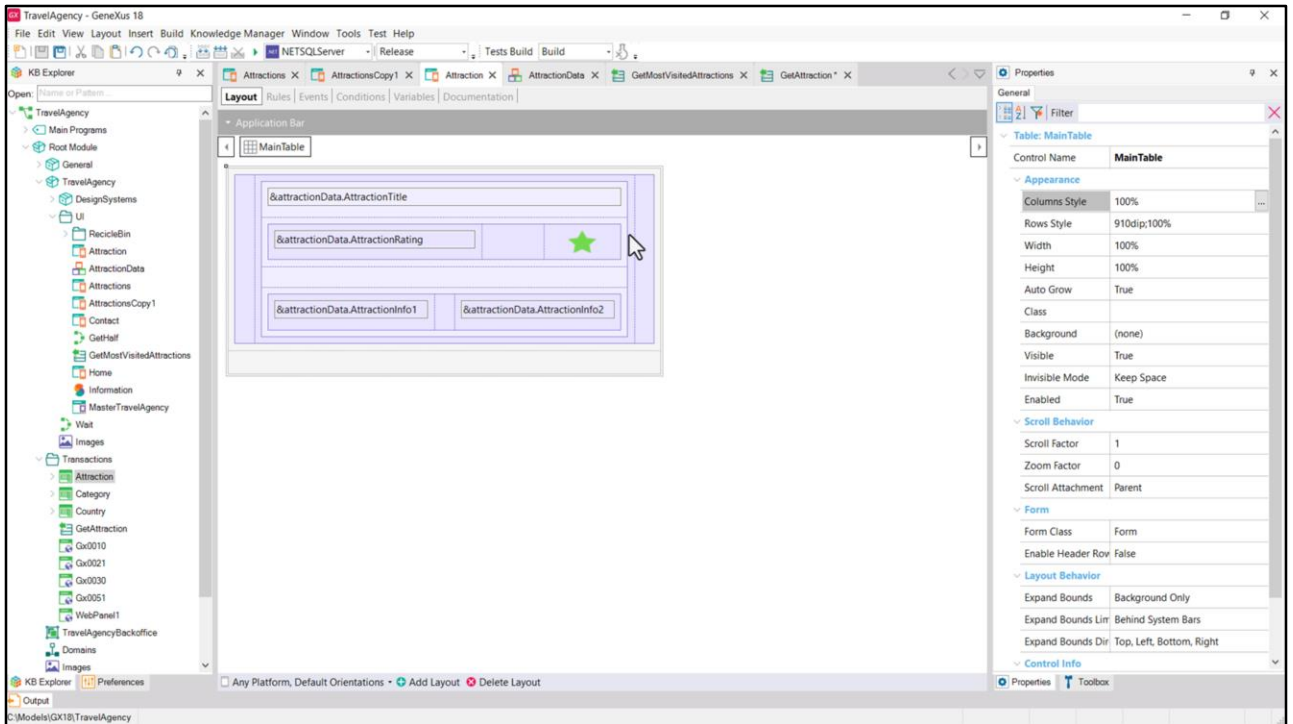
Por el filtro automático, justamente, se quedará solamente con el registro de la tabla Attraction que tiene ese id del parámetro.

Y lo que haré será cargar en el SDT que devolverá el Data Provider solamente lo que necesito por el momento: el id, el rating de la atracción, el título, la información dividida en dos bloques de texto (para hacer esa división llamo a un procedimiento que me devuelve del atributo de Info la primera mitad si envió 1, la segunda mitad si envió 2)...



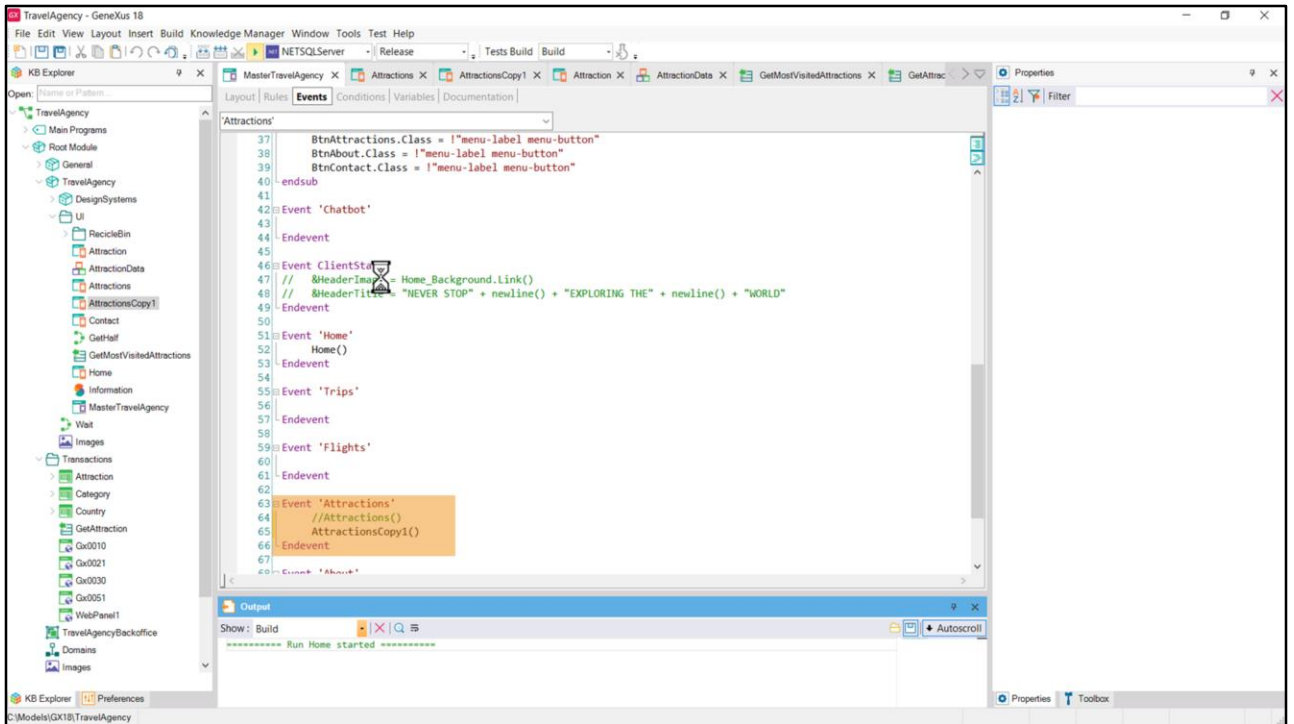
Como recordatorio para entender este código: como el elemento del SDT y el atributo de la tabla base se llaman igual, entonces puedo usar la notación abreviada, en lugar de tener que hacer explícitamente la asignación. Por ejemplo, escribir esto es lo mismo que esto otro... Pero mientras el de la izquierda es el elemento del SDT, el de la derecha es el atributo de la tabla Attraction de la base de datos.

El Data Provider también devolverá cargados los ítems de esta colección, pero por ahora solo utilizaremos su primer nivel...

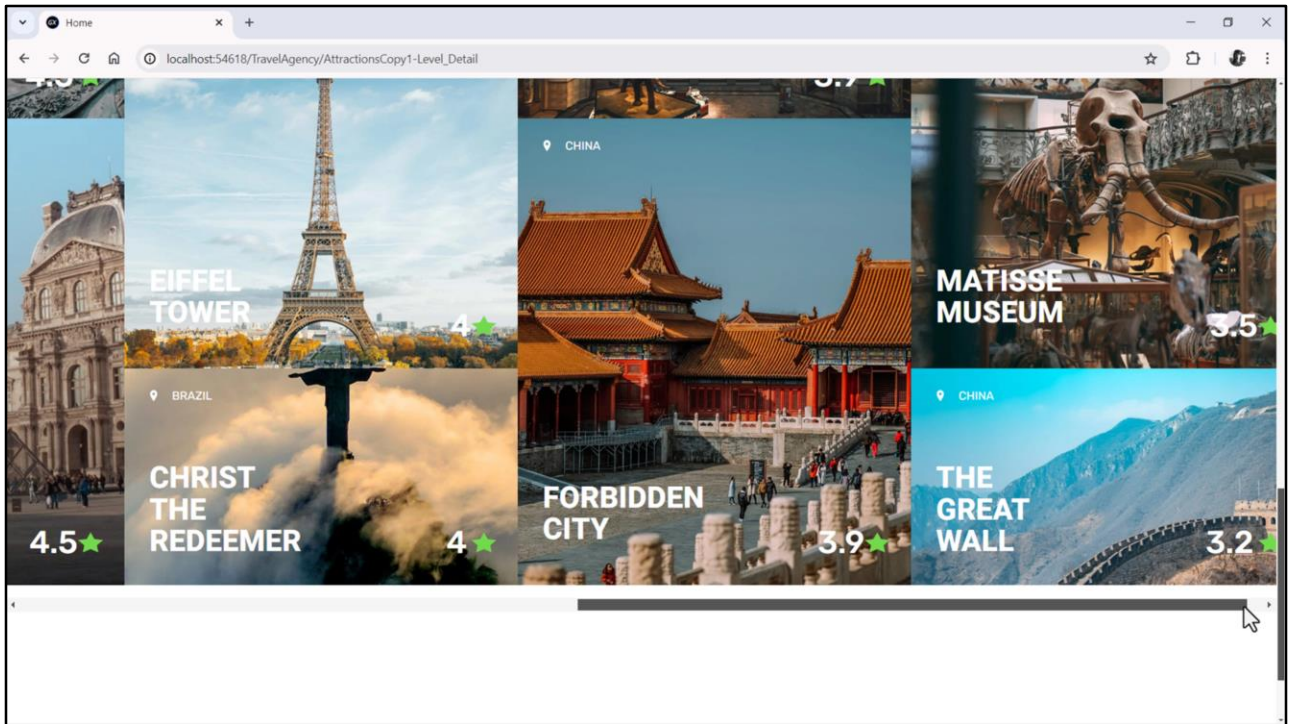


...que es lo que utilizamos aquí en el layout de Attraction.

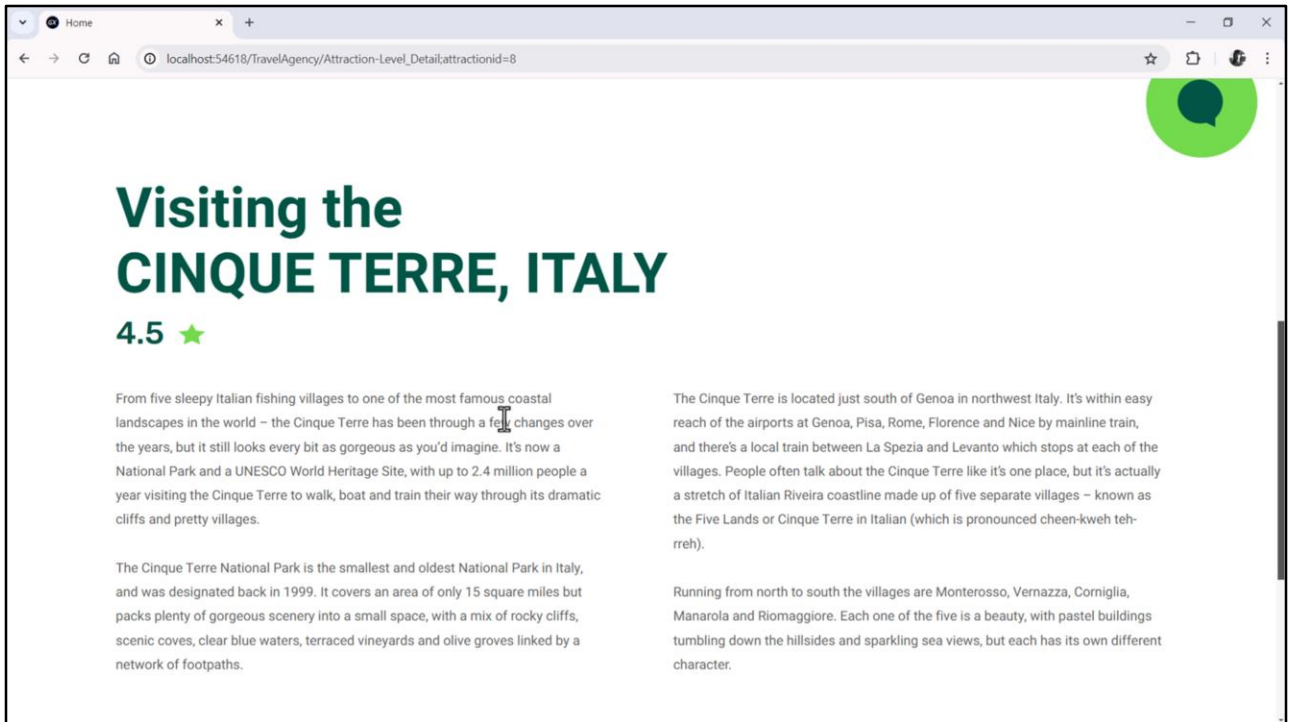
Antes de implementar lo que es el objetivo de este video, voy a ejecutar esto para que nos quede bien claro de dónde partimos.



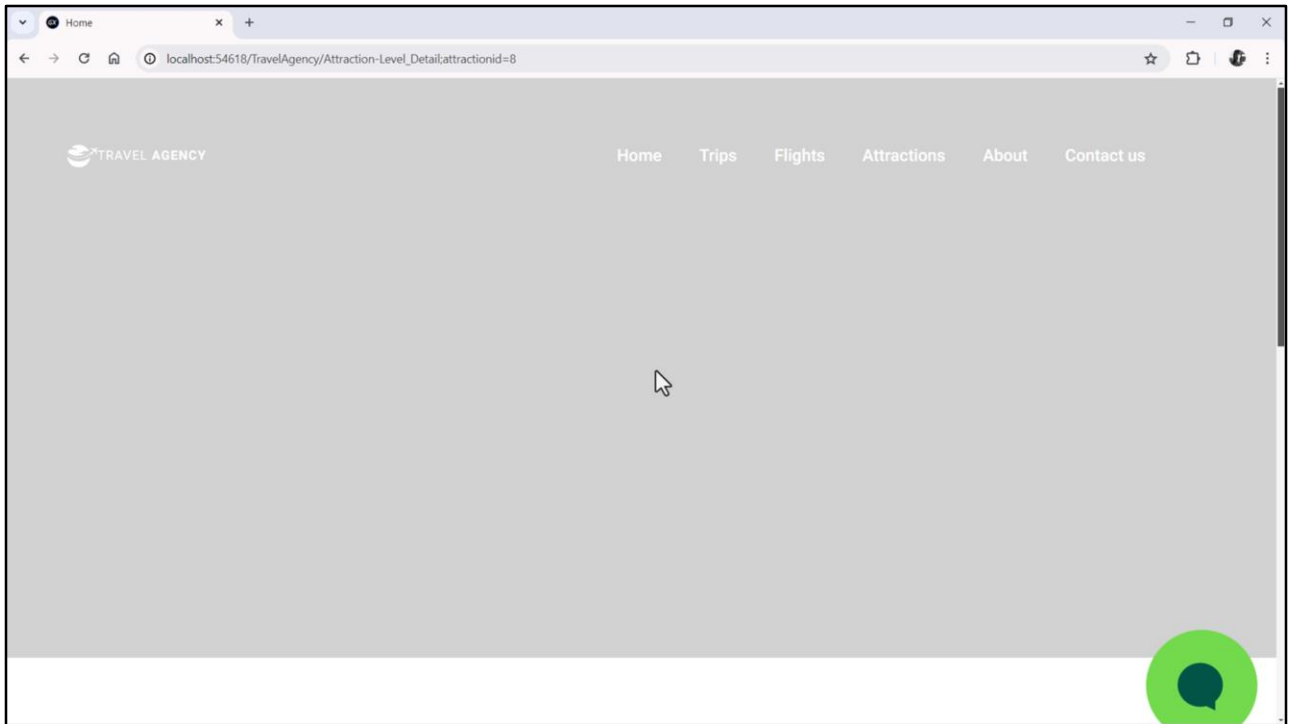
Aquí, en el Master Panel, en el evento asociado al botón del Menú Attractions estoy llamado al panel Attractions que no tiene el grid. Así que lo que voy a hacer es cambiar la invocación por el panel que sí lo tiene. Ejecuto...



Aquí tenemos el Home, y si vamos a la opción de Attractions vemos aparecer el carrusel de las cards de atracciones, con scroll horizontal...



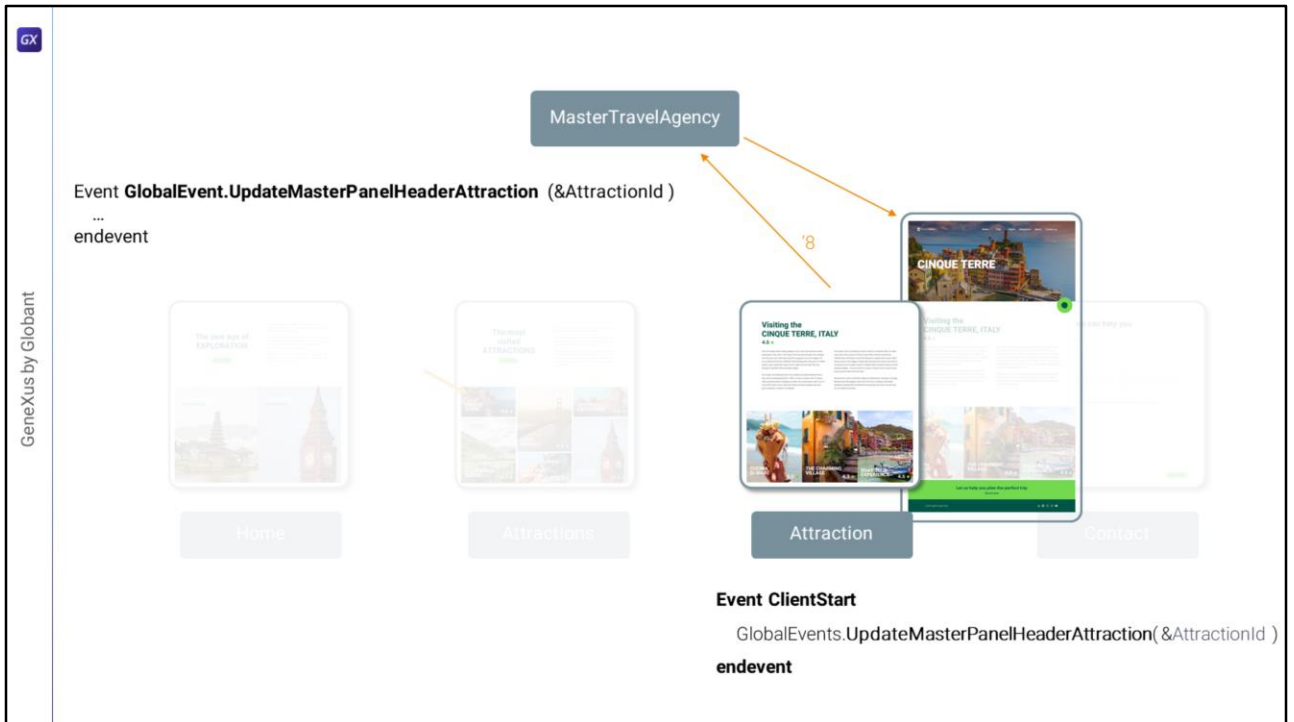
Y si hacemos clic sobre una atracción, o tap, que es lo mismo... nos lleva al panel Attraction, que tiene cargada la información de esta parte que es la única que implementamos hasta el momento.



Pero no vemos en el Header del Master Panel ni la foto de la atracción ni su nombre. Era en esto en lo que nos queríamos concentrar en este video, así que vamos a ello.



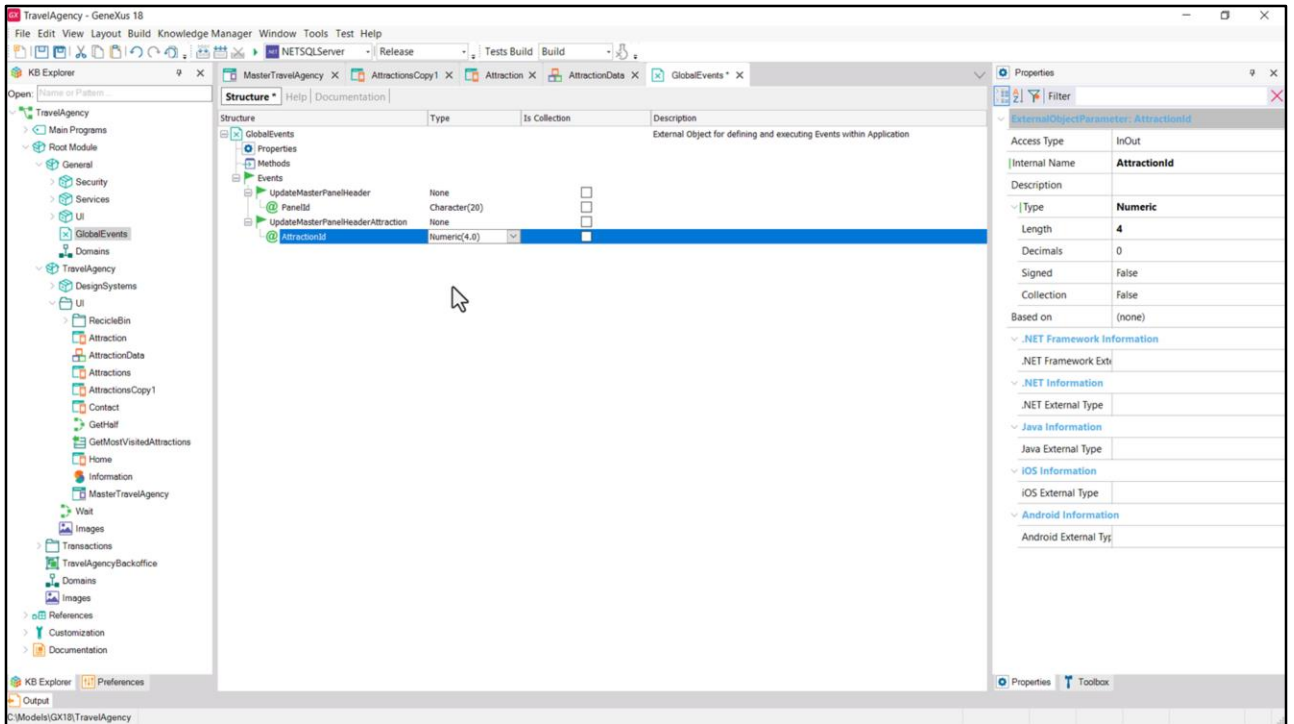
En el video anterior habíamos decidido utilizar dos eventos globales, uno para los paneles Home, Attractions y Contact, y un segundo evento global para Attraction.



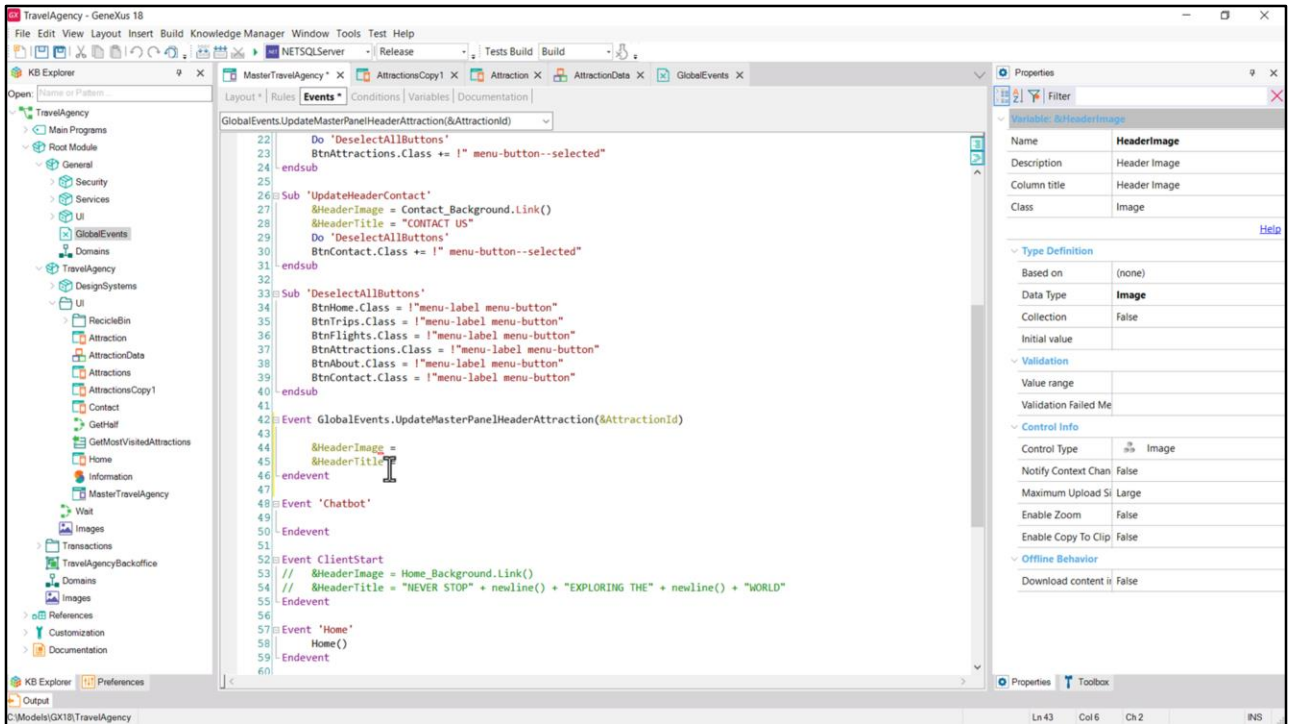
Luego lo dispararemos de el ClientStart de Attraction, pasándole el id de la atracción.

Y tendremos que implementarlo en el Master Panel, con su lógica.

Es lo que haremos ahora en GeneXus.



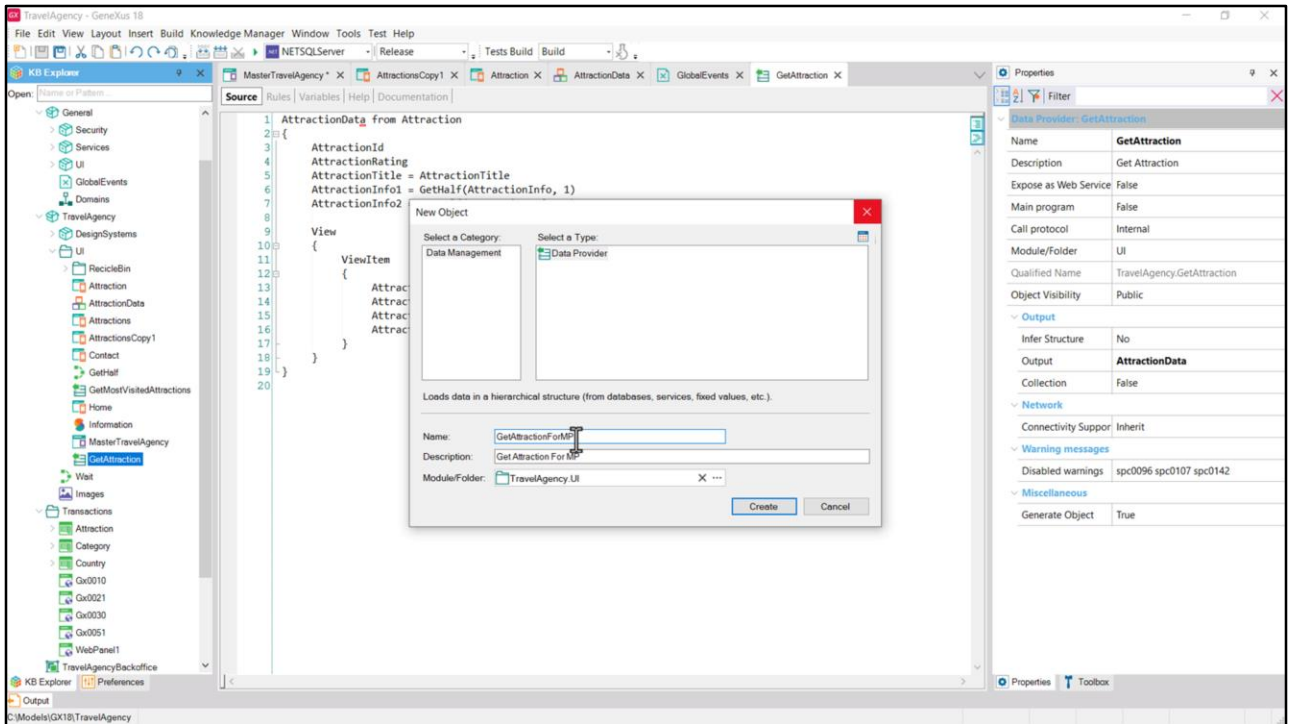
Abro el objeto GlobalEvents y agrego este nuevo evento... con parámetro AttractionId...



Ahora voy al Master Panel, y lo agrego en la sección de eventos, donde más me guste.

Defino la variable &AttractionId en este objeto (recordemos que las variables son locales a cada objeto).

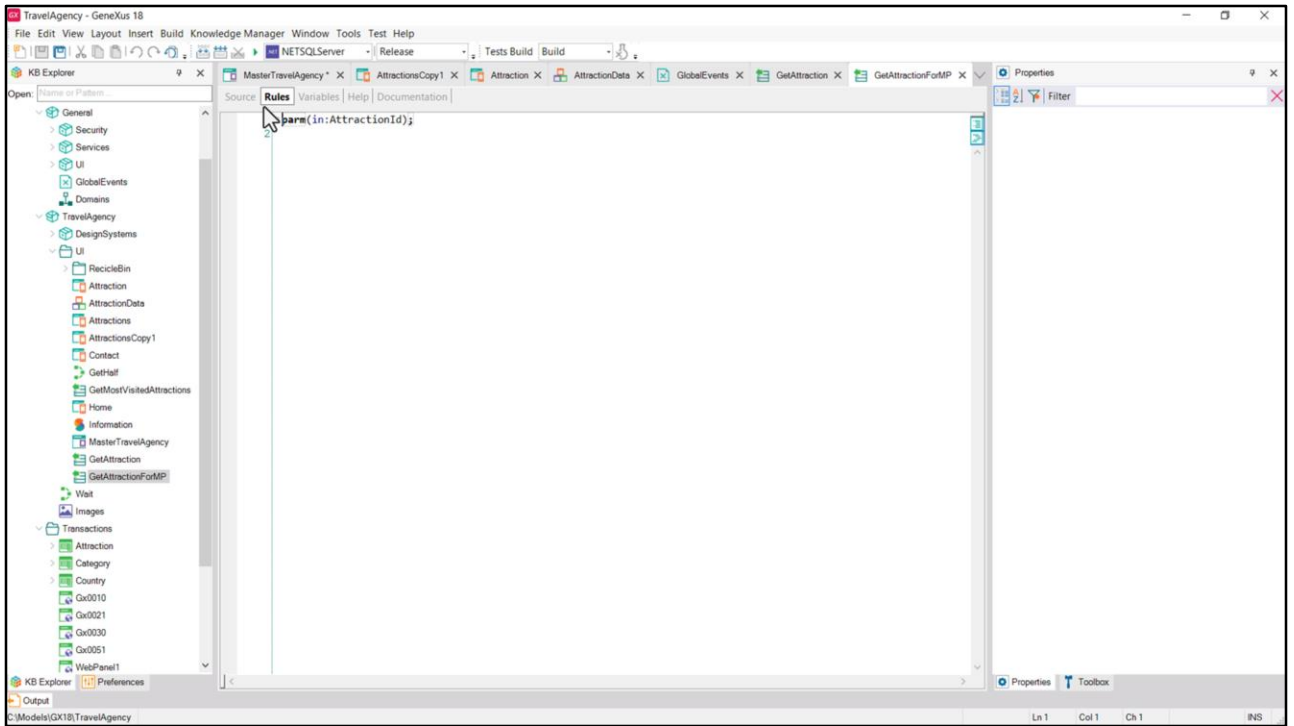
Bien. Necesito traer de la base de datos el nombre de la atracción y su foto, para asignárselas a estas dos variables del Header del layout del Master Panel.



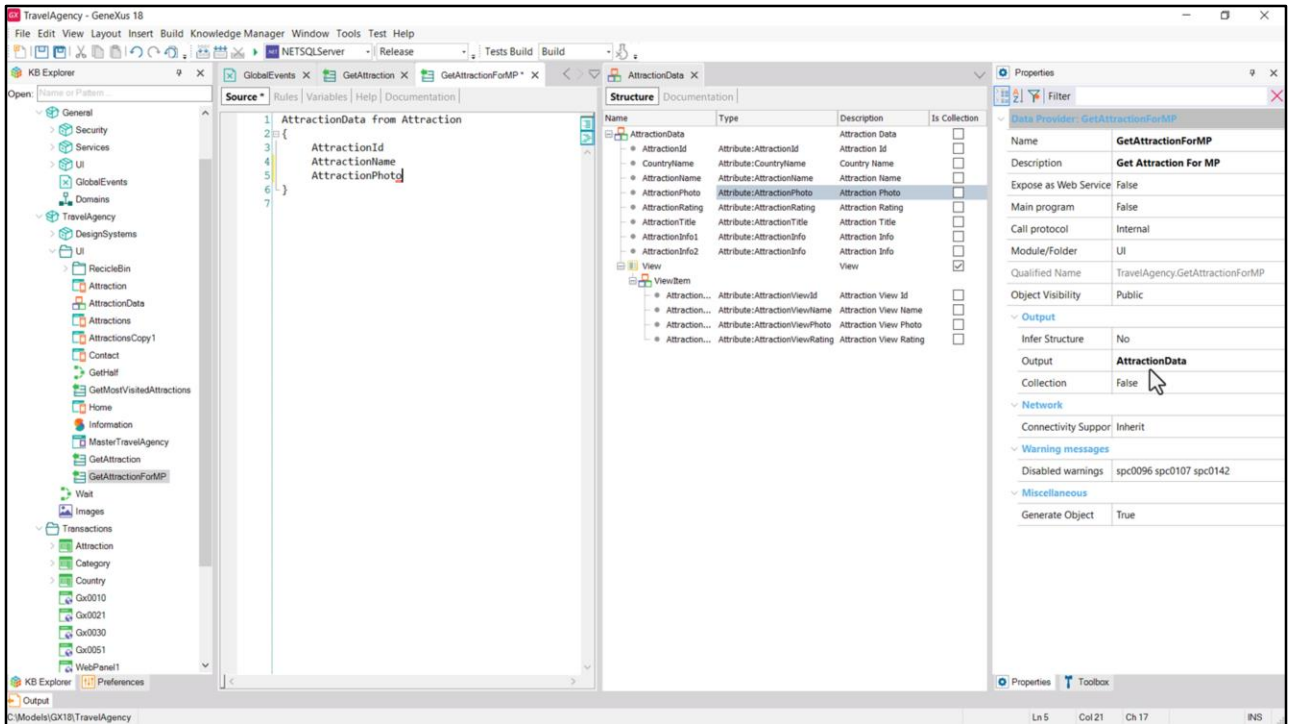
Si quiero ser extremadamente eficiente puedo hacer un Save as del Data Provider que utilice en Attraction para traer la info de la atracción, de modo de devolver cargados únicamente estos dos valores... y todo lo demás vacío.

Me quedó erróneamente en el folder Transaction, así que lo voy a mover para este otro.

Hago el Save as, le asigno este nombre...

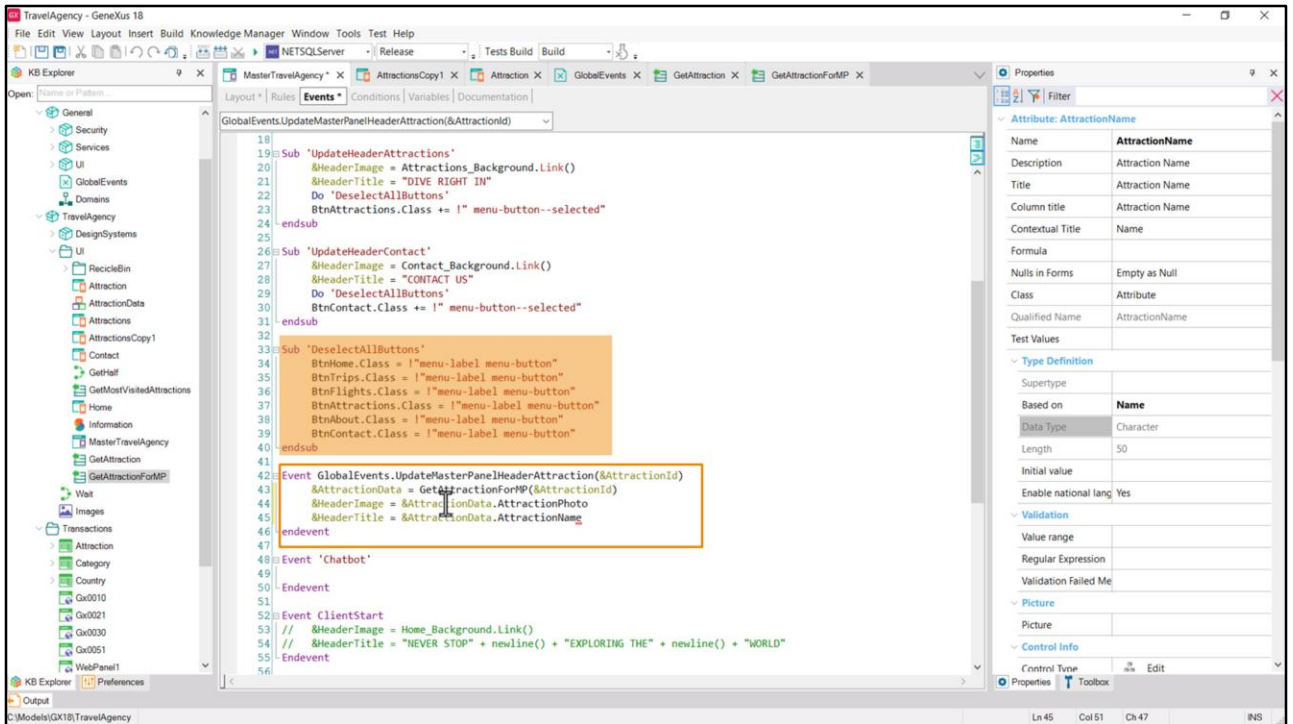


Recibirá el parámetro en el atributo AttractionId...



...y en el source quitaremos todas estas asignaciones y sólo dejaremos cargados los elementos que nos interesan... AttractionName y AttractionPhoto.

Devolverá este SDT cargado solamente con estos tres elementos. Lo demás todo vacío.

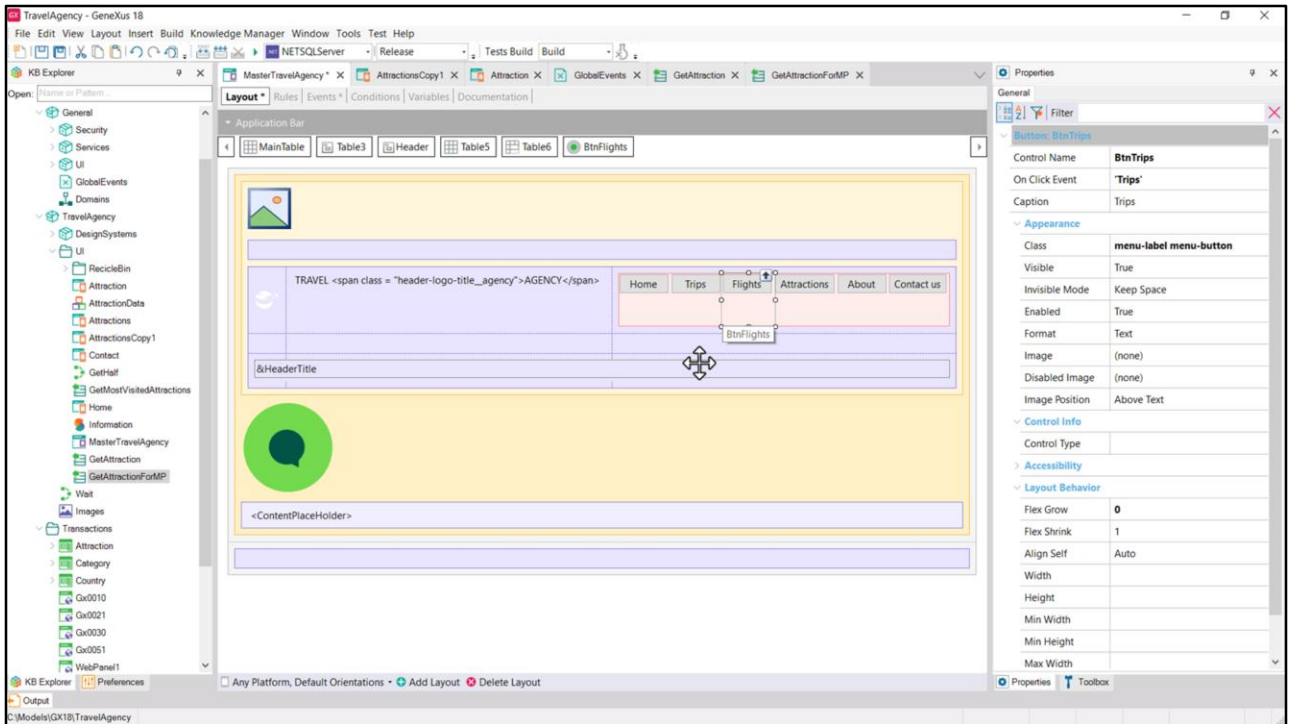


Y luego en el Master Panel a una variable del tipo de datos del SDT le asignamos el resultado de ejecutar el data provider, pasándole como parámetro el valor de esta variable &AttractionId que es el parámetro que recibe el evento global cuando es disparado.

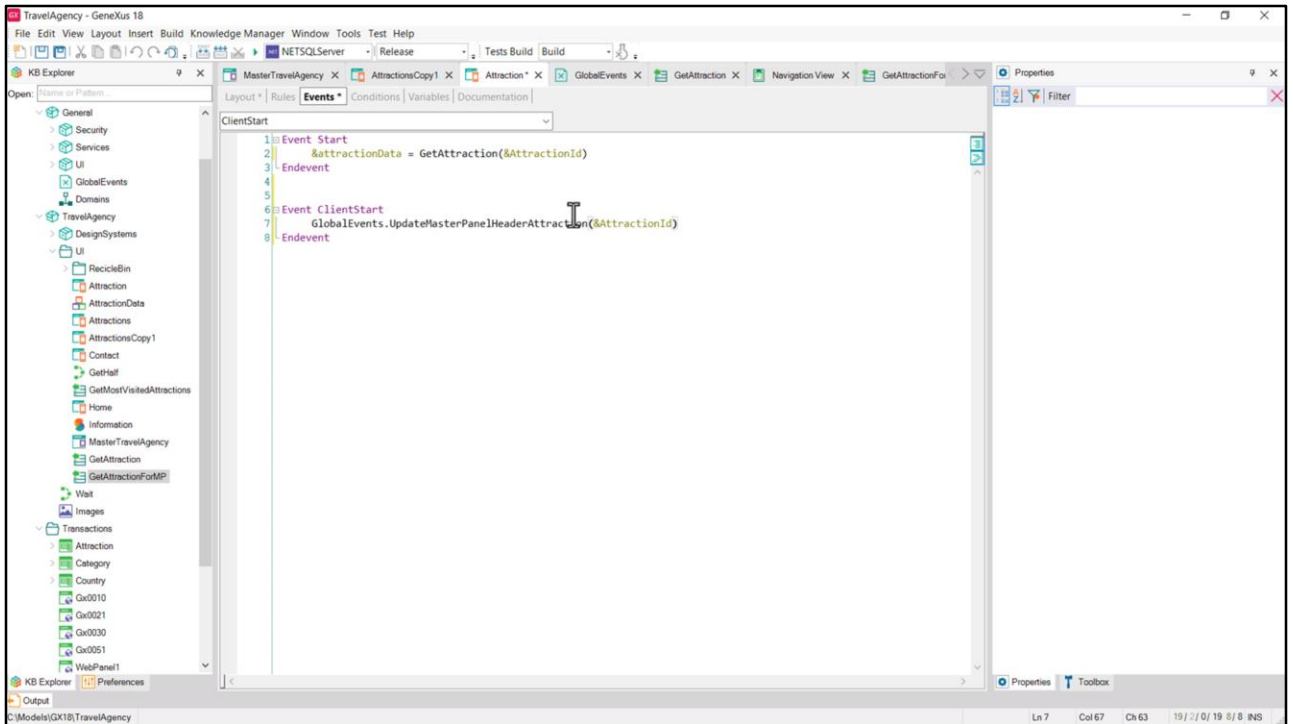
Luego a la variable &HeaderImage del layout le asignamos del SDT el elemento AttractionPhoto.

Y a &HeaderTitle el elemento AttractionName.

¿Y qué pasa con los botones del menú? ¿Deberíamos llamar a esta subrutina para deseleccionarlos?

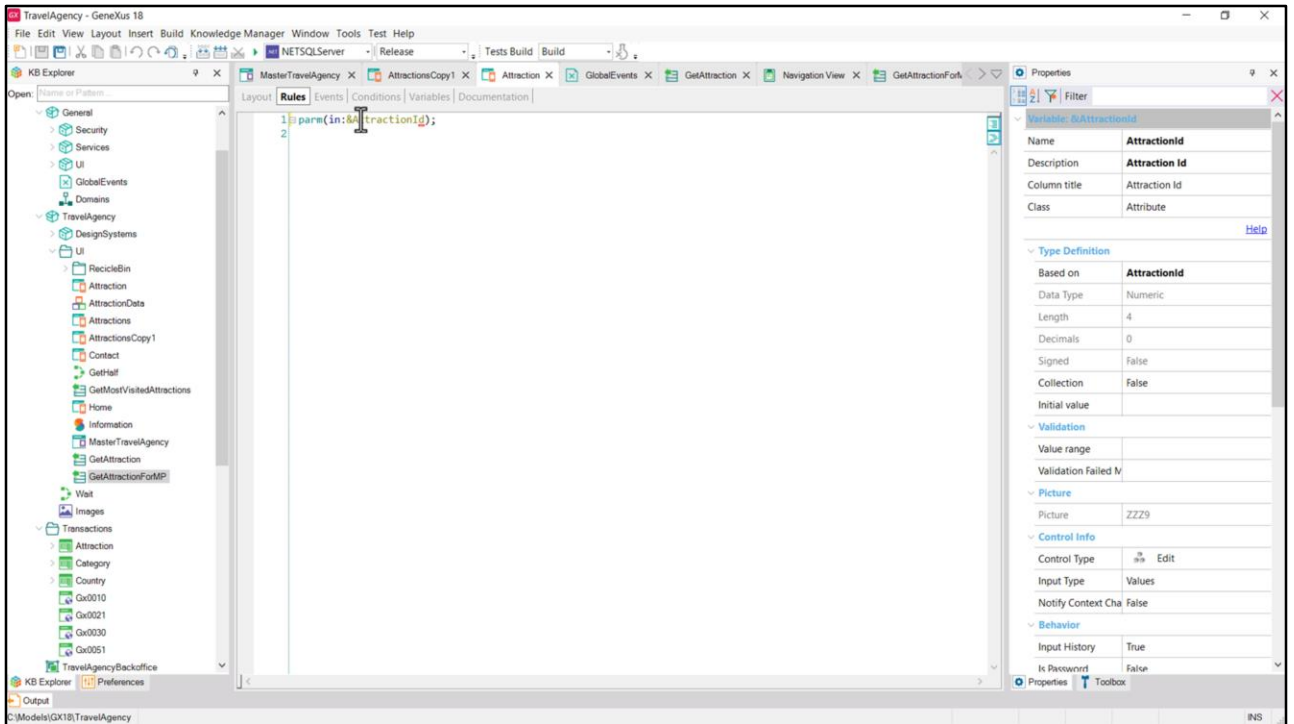


En realidad no es necesario porque estas mismas clases ya las tienen determinadas los botones en forma estática.



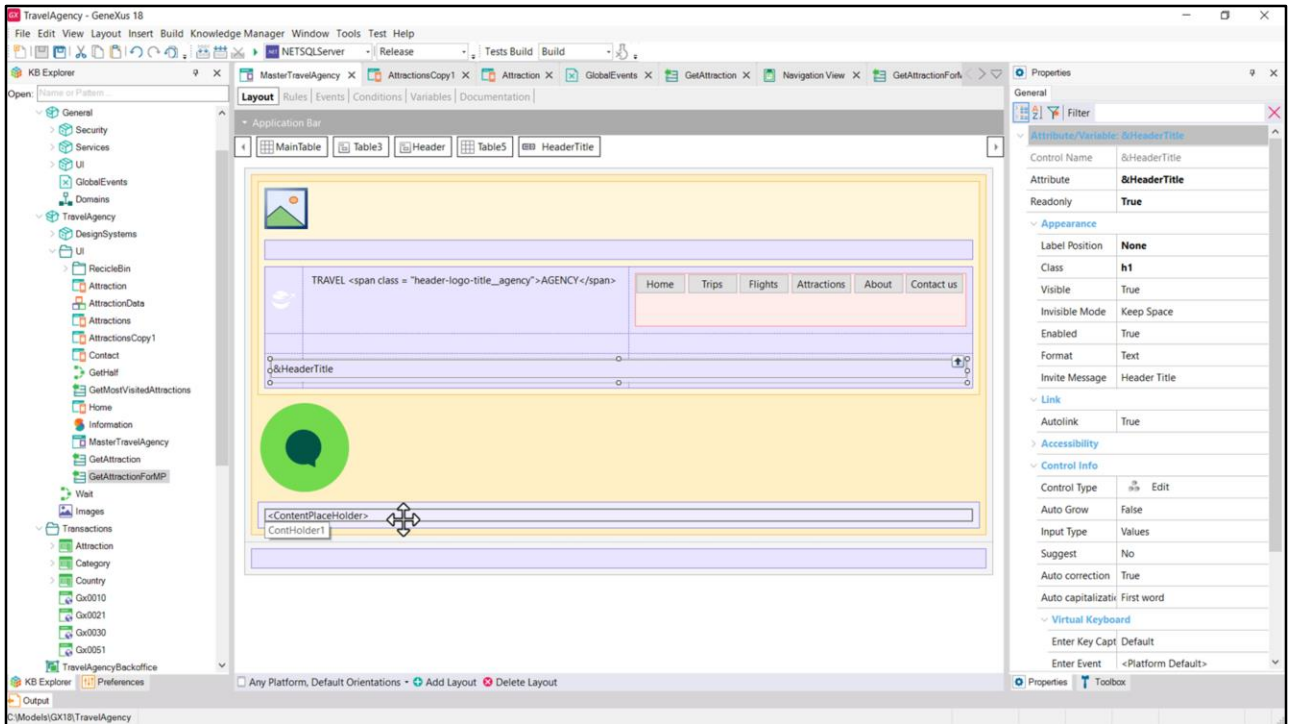
Lo único que nos está haciendo falta ahora es disparar este evento desde el panel Attraction.

Así que en el evento ClientStart... invocamos al evento global que queremos, pasándole como parámetro esta variable, la que recibe el panel Attraction. Y a partir de la cual se carga su layout.



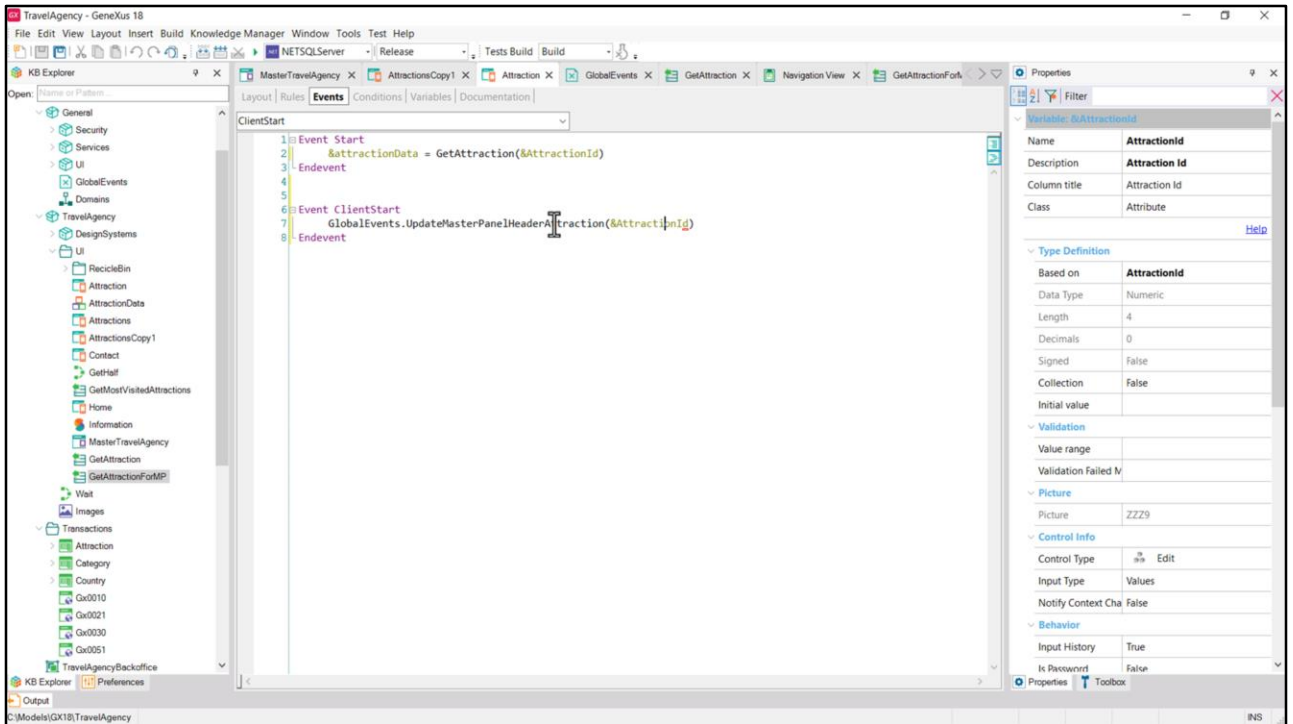
En definitiva, repasemos todo.

Cuando el panel Attraction es ejecutado recibe en esta variable el id de la atracción de la que se tratará.



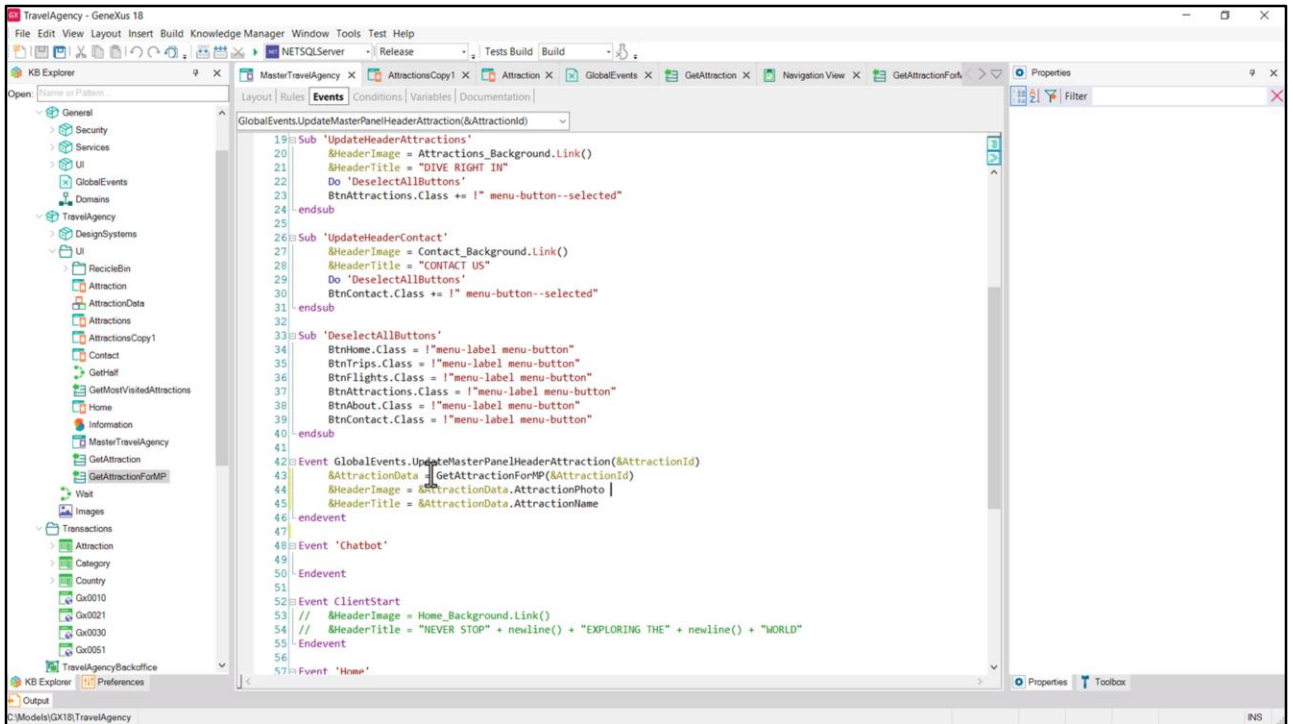
Lo primero que hace, como sabemos, es ejecutar el Master Panel, y su evento del sistema ClientStart, que en este caso está vacío. Y renderiza por primera vez el Master Panel. En ese momento las variables del header quedarán vacías.

Luego se ejecuta para cargarse en el ContentPlaceHolder el panel Attraction propiamente dicho.



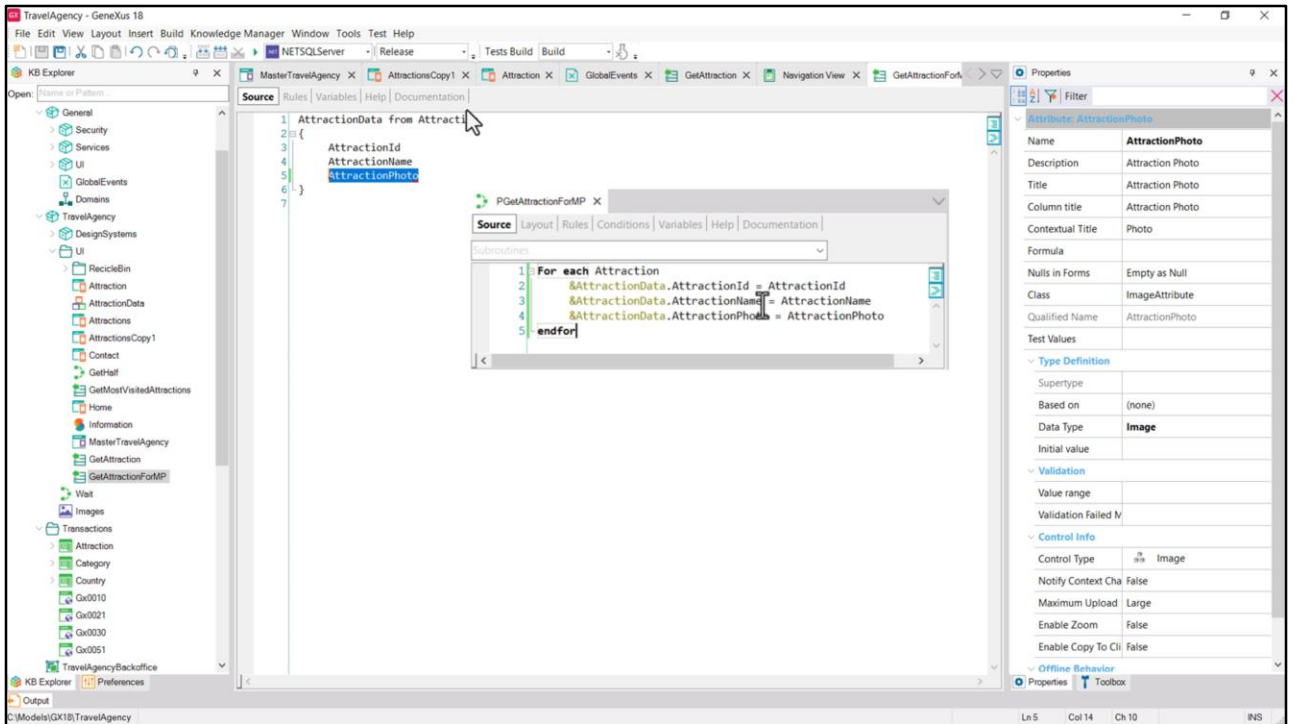
Se asignan las propiedades estáticas de los controles del layout y a continuación se ejecuta el evento ClientStart. Aquí es que se dispara el evento global del Master Panel, que, como dijimos en el video anterior, es asíncrono.

Ya se le envía a este evento el id de atracción recibido por parámetro.



Y aquí es que ejecuta en el Master Panel esto. Aquí es donde se carga la imagen y el título a partir del SDT que devolvió cargado este data provider.

Ya les advierto que acaba de arreglarse un bug que hacía que un DataProvider que cargaba una imagen no lo hiciera bien. Si cuando ejecuten esto les sucede que no vean en ejecución la imagen, entonces es que tienen una versión con el bug.



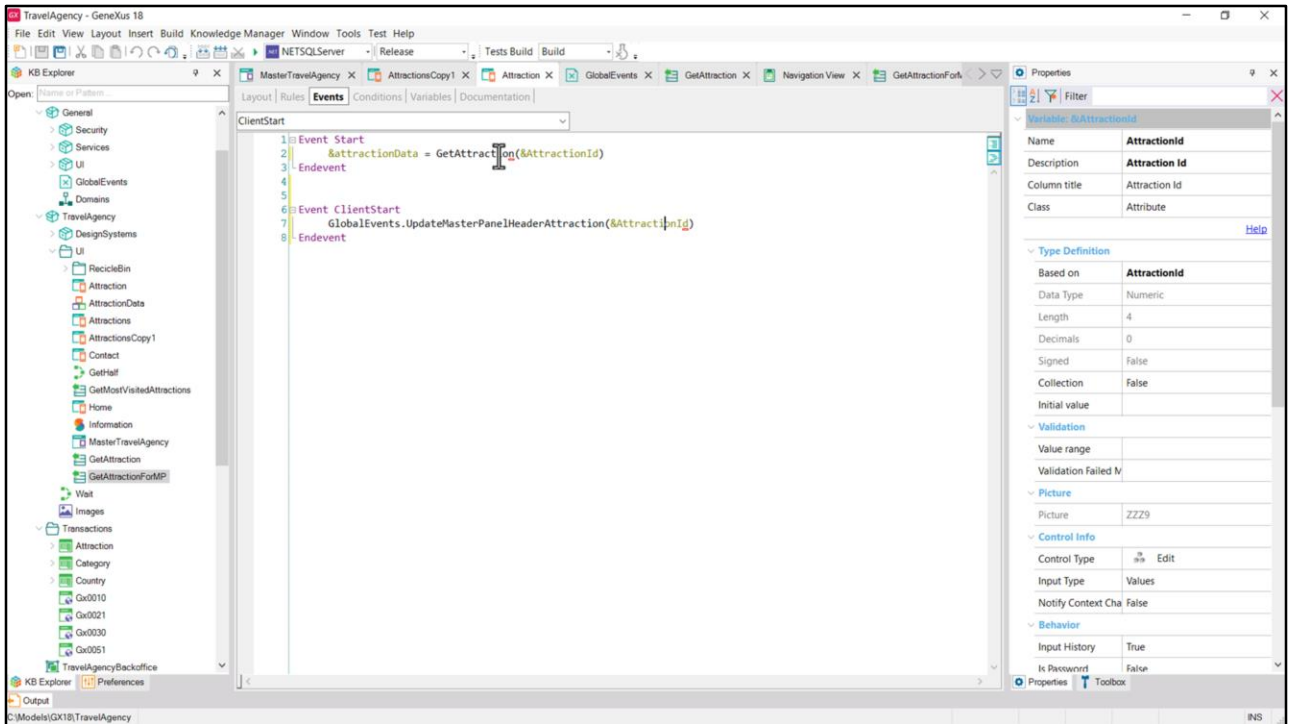
En ese caso les recomiendo utilizar un procedimiento para cargar el SDT en lugar de un Data Provider, y allí no tendrán ningún problema.

The screenshot shows the Genexus 18 IDE with the following components:

- KB Explorer:** A tree view on the left showing the project structure, including folders like 'General', 'Security', 'Services', 'UI', 'GlobalEvents', 'Domains', 'TravelAgency', 'DesignSystems', 'UI', 'RecycleBin', 'Attraction', 'AttractionData', 'Attractions', 'AttractionsCopy1', 'Contact', 'GetHalf', 'GetMostVisitedAttractions', 'Home', 'Information', 'MasterTravelAgency', 'GetAttraction', 'GetAttractionForMP', 'Wait', 'Images', 'Transactions', 'Attraction', 'Category', 'Country', 'Gx0010', 'Gx0021', 'Gx0030', 'Gx0051', 'TravelAgencyBackoffice', and 'TravelAgencyBackoffice'.
- Code Editor:** The main window displays the code for the 'Events' tab. The code includes several subroutines and an event handler:

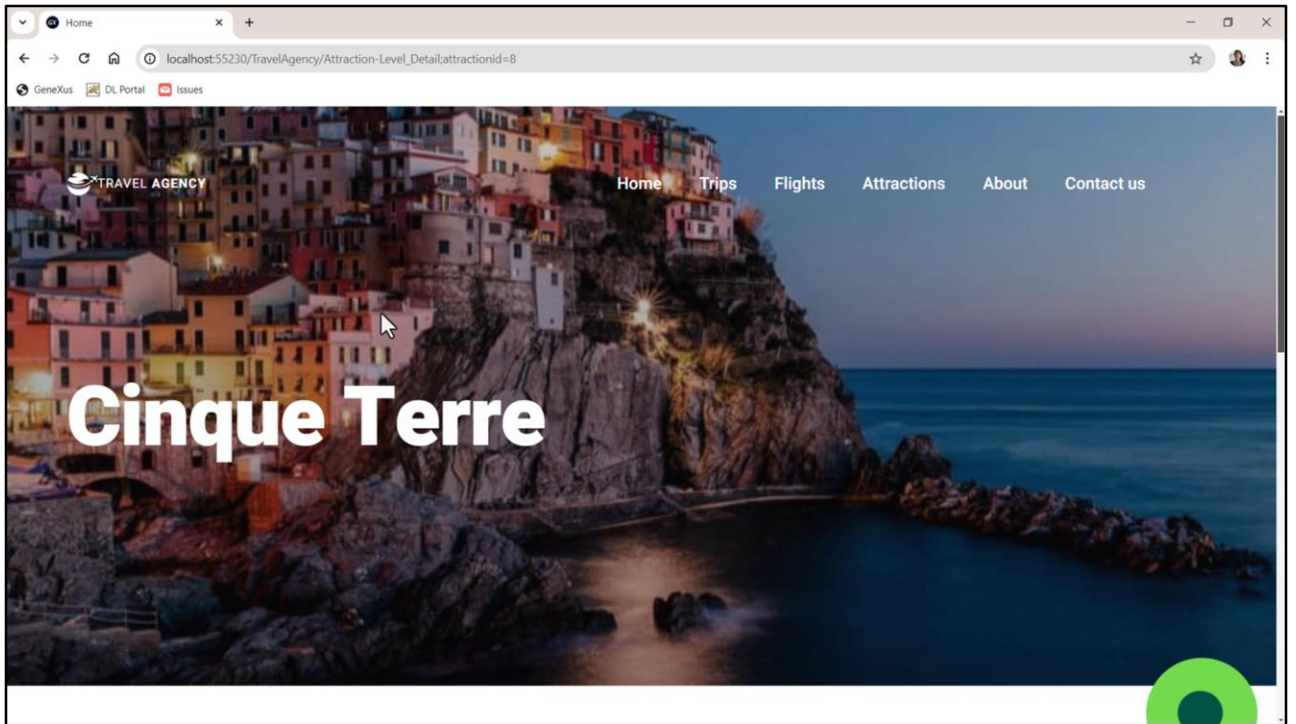
```
19 Sub 'UpdateHeaderAttractions'  
20   &HeaderImage = Attractions_Background.Link()  
21   &HeaderTitle = "DIVE RIGHT IN"  
22   Do 'DeselectAllButtons'  
23     BtnAttractions.Class += "! menu-button--selected"  
24   EndSub  
25  
26 Sub 'UpdateHeaderContact'  
27   &HeaderImage = Contact_Background.Link()  
28   &HeaderTitle = "CONTACT US"  
29   Do 'DeselectAllButtons'  
30     BtnContact.Class += "! menu-button--selected"  
31   EndSub  
32  
33 Sub 'DeselectAllButtons'  
34   BtnHome.Class = "!menu-label menu-button"  
35   BtnTrips.Class = "!menu-label menu-button"  
36   BtnFlights.Class = "!menu-label menu-button"  
37   BtnAttractions.Class = "!menu-label menu-button"  
38   BtnAbout.Class = "!menu-label menu-button"  
39   BtnContact.Class = "!menu-label menu-button"  
40 EndSub  
41  
42 Event GlobalEvents.UpdateMasterPanelHeaderAttraction(&AttractionId)  
43   &AttractionData = GetAttractionForMP(&AttractionId)  
44   &HeaderImage = &AttractionData.AttractionPhoto  
45   &HeaderTitle = &AttractionData.AttractionName  
46 EndEvent  
47  
48 Event 'Chatbot'  
49  
50 EndEvent  
51  
52 Event ClientStart  
53 // &HeaderImage = Home_Background.Link()  
54 // &HeaderTitle = "NEVER STOP" + newline() + "EXPLORING THE" + newline() + "WORLD"  
55 EndEvent  
56  
57 Event 'Home'
```
- Properties Window:** On the right, the 'Properties' window is open for the variable '&AttractionData'. It shows the following details:
 - Name:** &AttractionData
 - Description:** Attraction Data
 - Column title:** Attraction Data
 - Class:** Attribute
 - Type Definition:** Based on (none)
 - Data Type:** AttractionData, TravelAgency
 - Collection:** False
 - Initial value:**
 - Validation:** Value range, Validation Failed M
 - Control Info:** Control Type Edit, Input Type Values, Notify Context Cha False
 - Behavior:** Input History False, Is Password False
 - Appearance:** Auto Resize True, Width 4chr, Height 1row, Fill True

Bueno, entonces, aquí se renderizará imagen y título del Header del Master Panel.

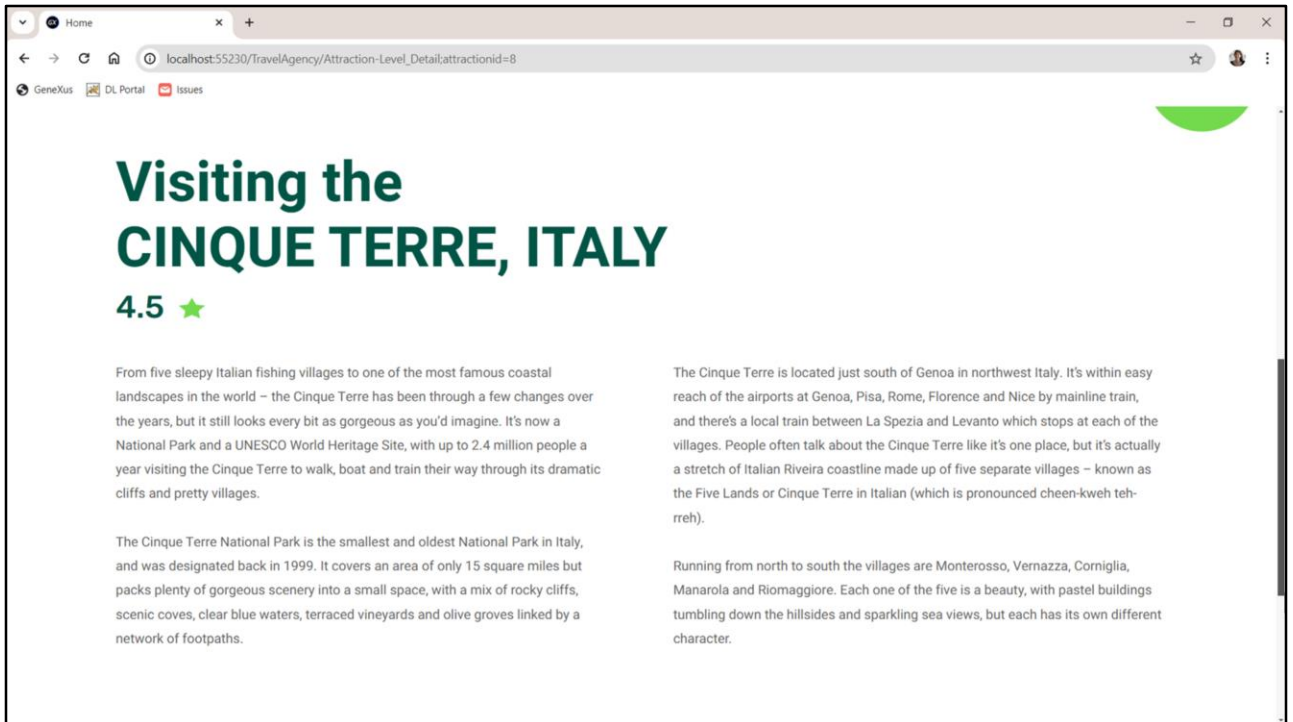


Y por otro lado, en Attraction se habrá continuado ya con el evento del sistema siguiente, el Start. Allí se llama a este otro Data Provider que trae cargada la información necesaria de la atracción de este id. Y esa es la que se renderizará en el layout. Recordemos que los eventos Refresh y Load están vacíos en este panel.

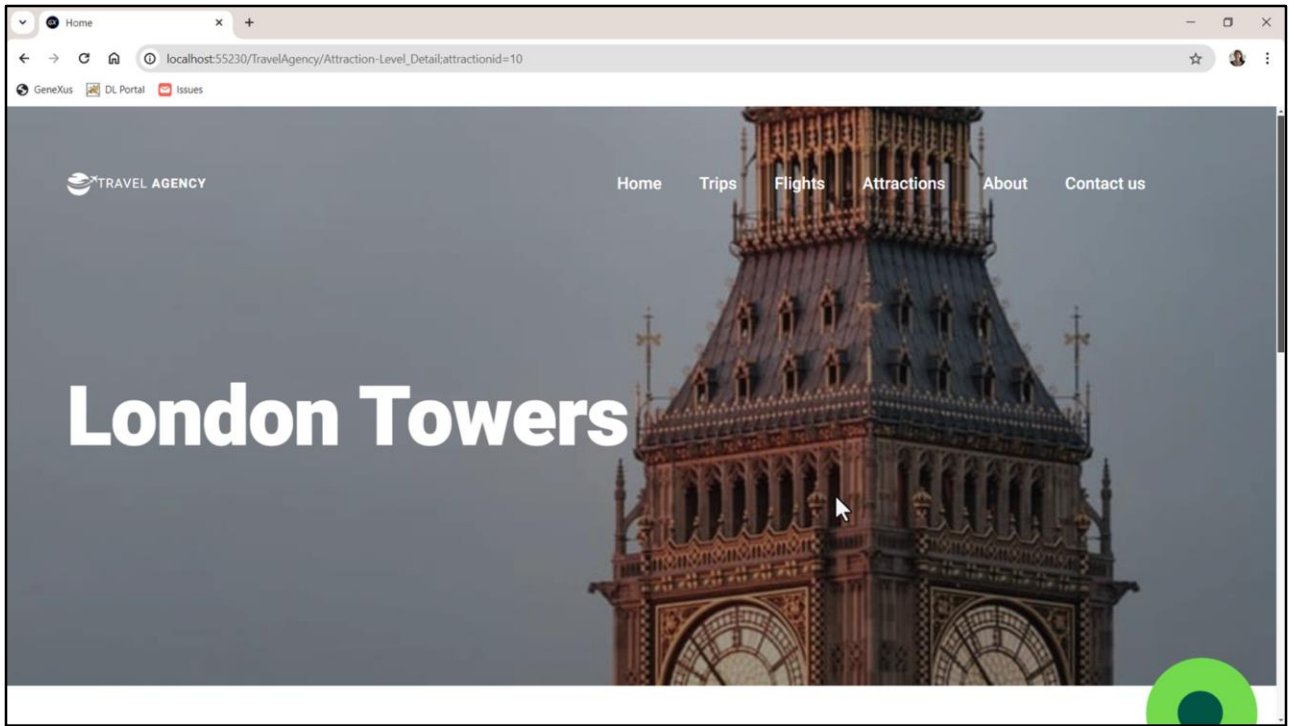
Bien, ejecutamos.



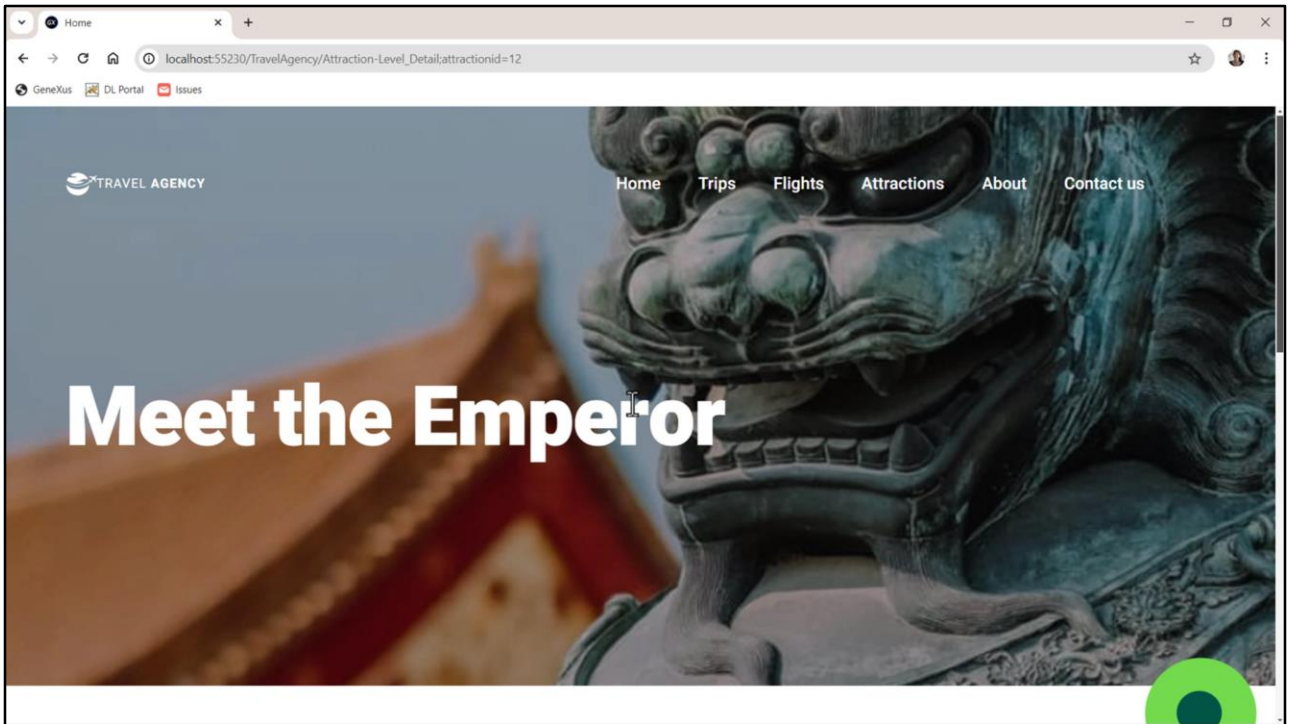
Vamos a Attractions... elegimos la primera del carrusel... y vemos cómo se carga correctamente foto y título...



...y el contenido específico de la página de la atracción.

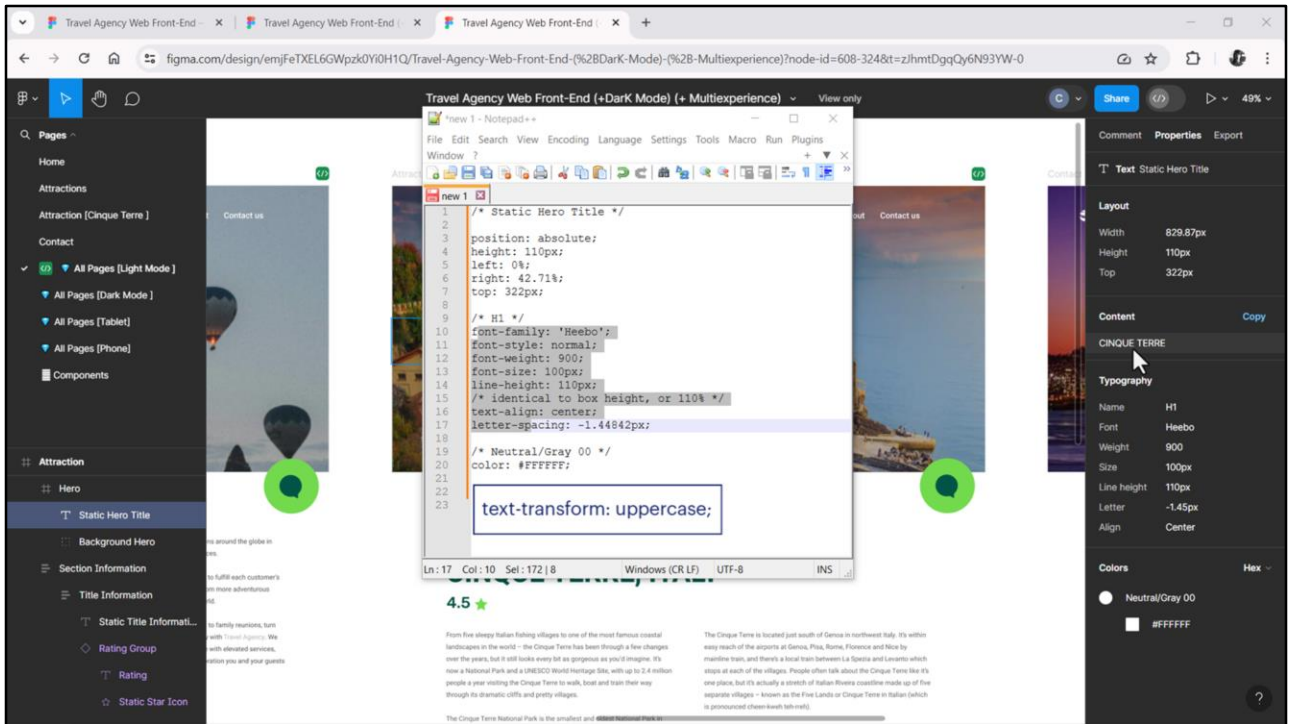


Si ahora vamos a elegir otra... también...

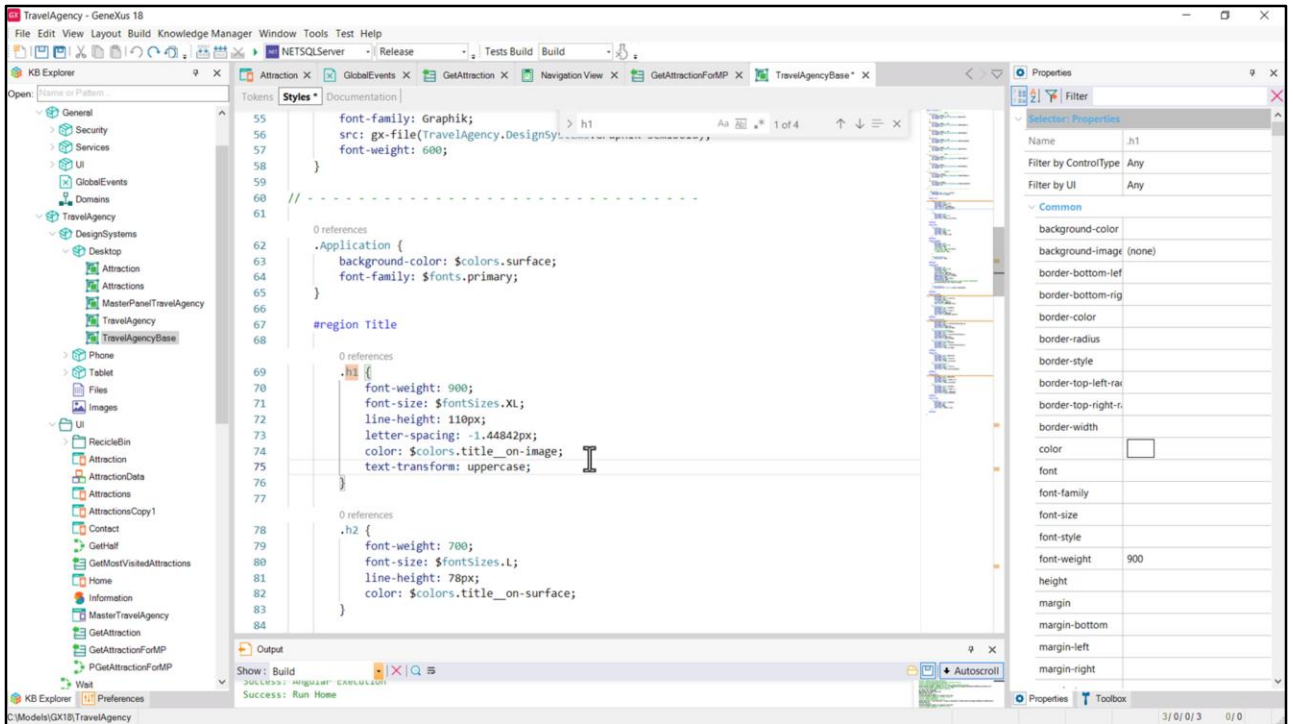


Elegimos otra... y también....

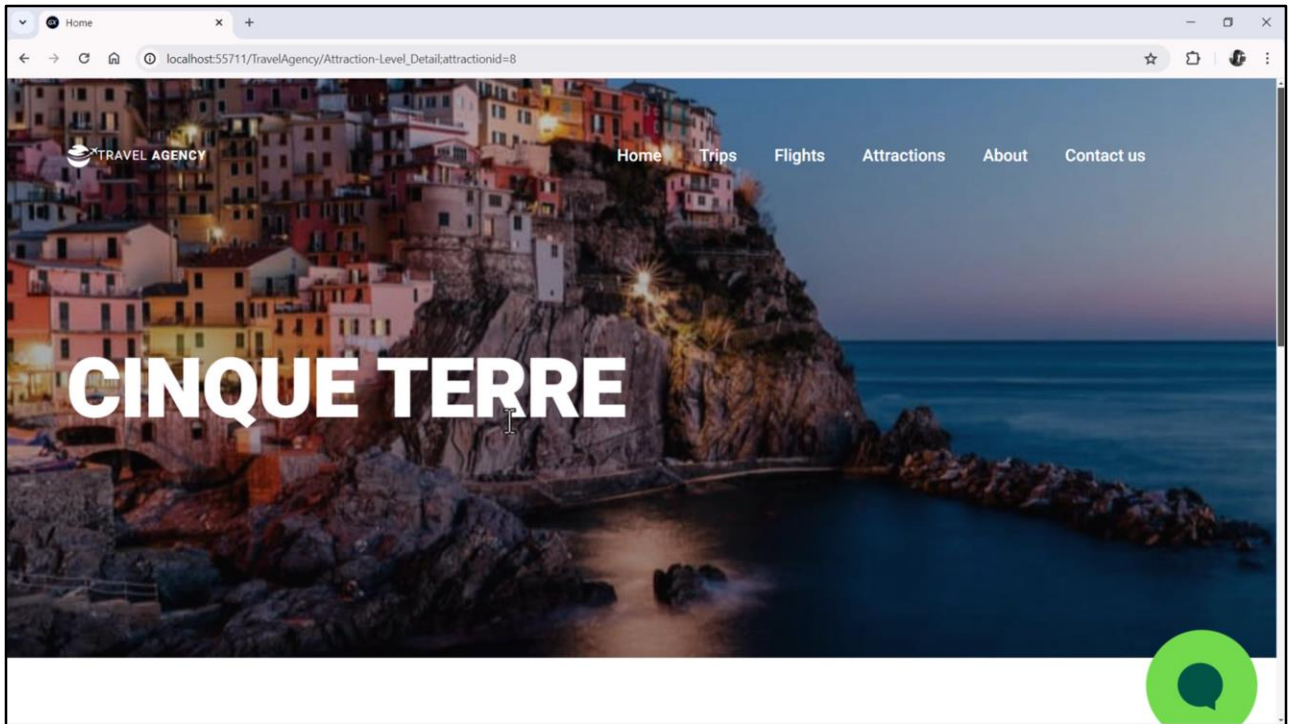
Aunque, este título debería mostrarse en mayúsculas siempre, independientemente de cómo esté ingresado en la variable.



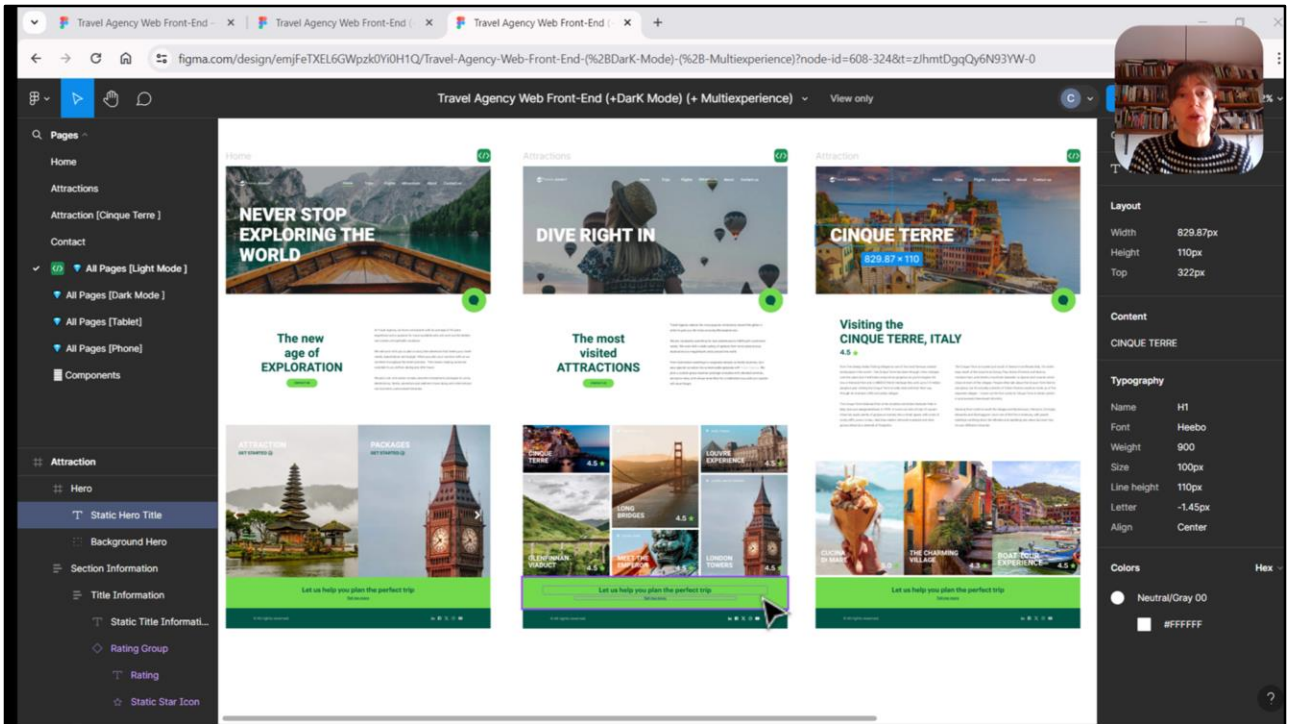
Evidentemente Chechu no se percató cuando creó el estilo tipográfico...porque ingresó los textos en mayúsculas...entonces no agregó la propiedad `text-transform: uppercase`.



Vayamos a agregarla a la clase h1.



Bueno, ahora sí. Bien.



Bueno, y con esto doy por cerrada la implementación del Master Panel.

Queda por hacer, y se los dejo como tarea para ustedes, implementar el footer. Toda esta parte de abajo, que es muy sencillo con todo el conocimiento que tenemos hasta el momento.

A lo que nos vamos a dedicar en el video que sigue es a la implementación, ya ahora sí en detalle, de este carrusel. Es un carrusel que se toma, como vimos, la información de la base de datos, por tanto es el más complejo de los tres que hay. Por eso me voy a detener en este. Los demás van a ser bastante sencillos a partir de entender, entonces, cómo implementar y diseñar este carrusel.

Bueno, los espero en el próximo video.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com