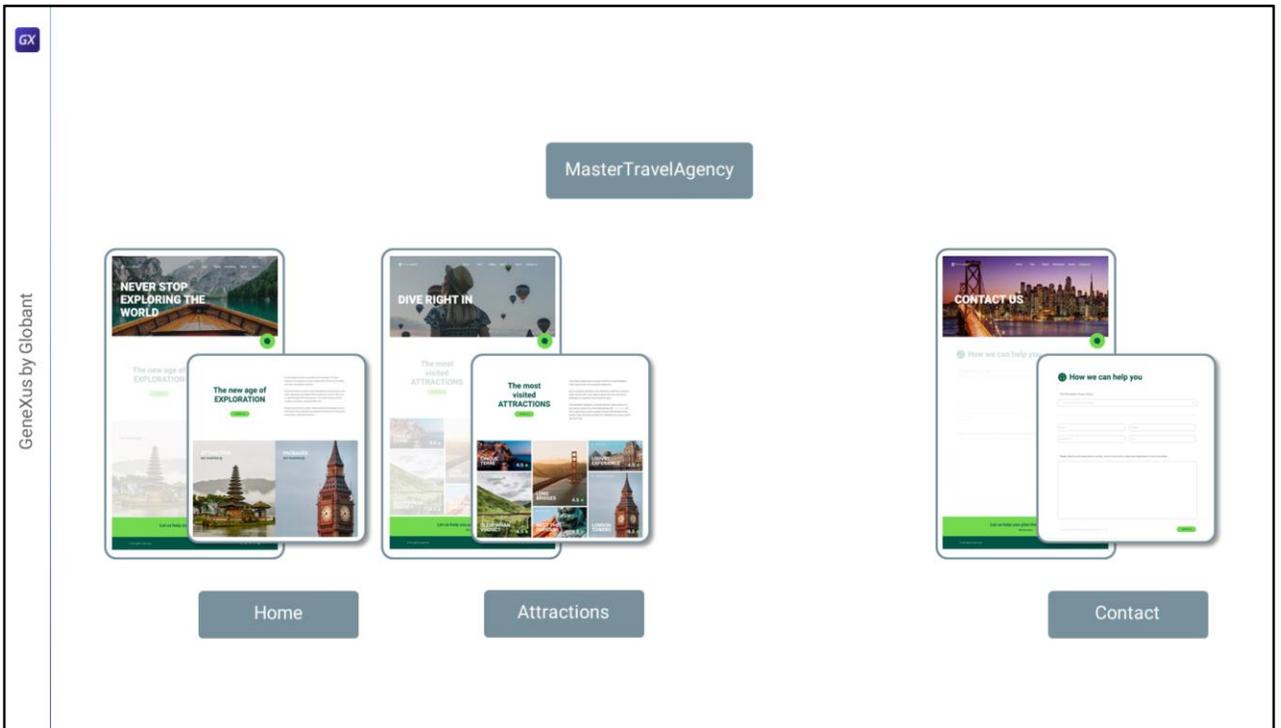


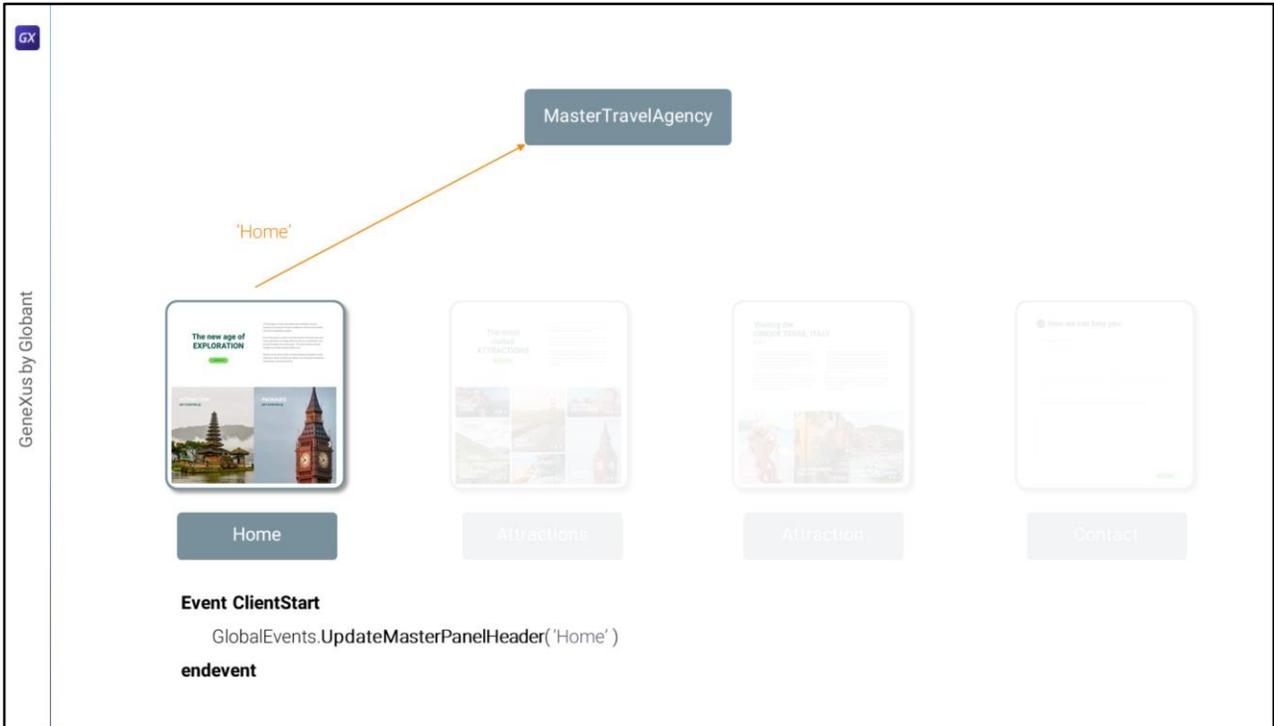
Master Panel: Header update and Navigations (part 2)



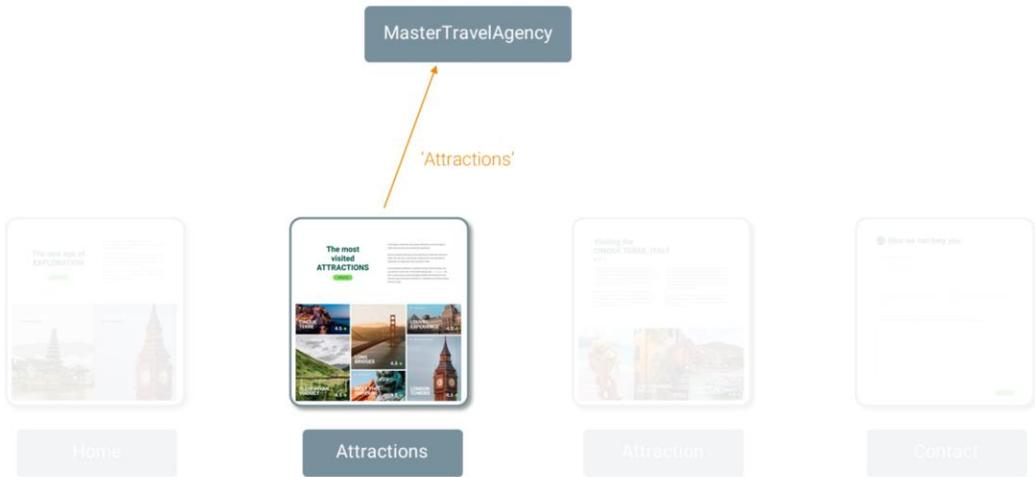
Cecilia Fernández



In the previous video, we implemented the correct navigations and loading of the Master Panel Header...



...according to whether the Home was being loaded...



Event ClientStart

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

endevent

...or Attractions...

MasterTravelAgency

'Contact'



Home



Attractions



Attraction



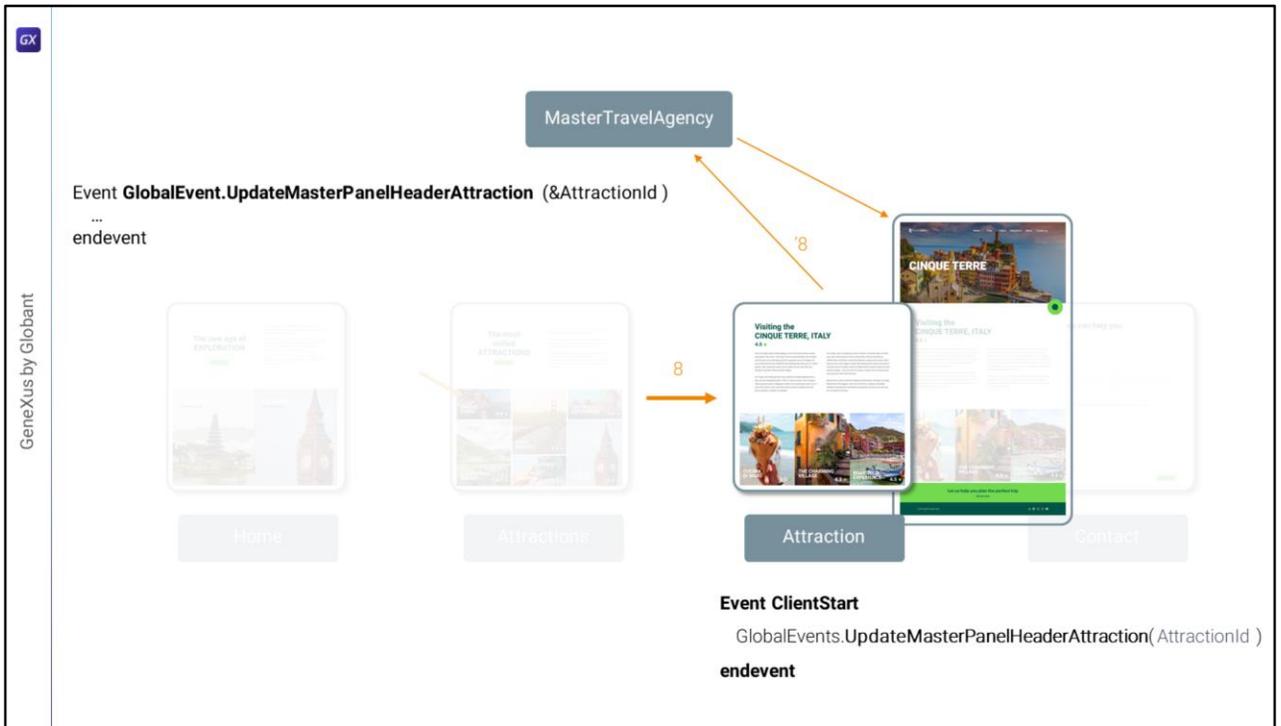
Contact

Event ClientStart

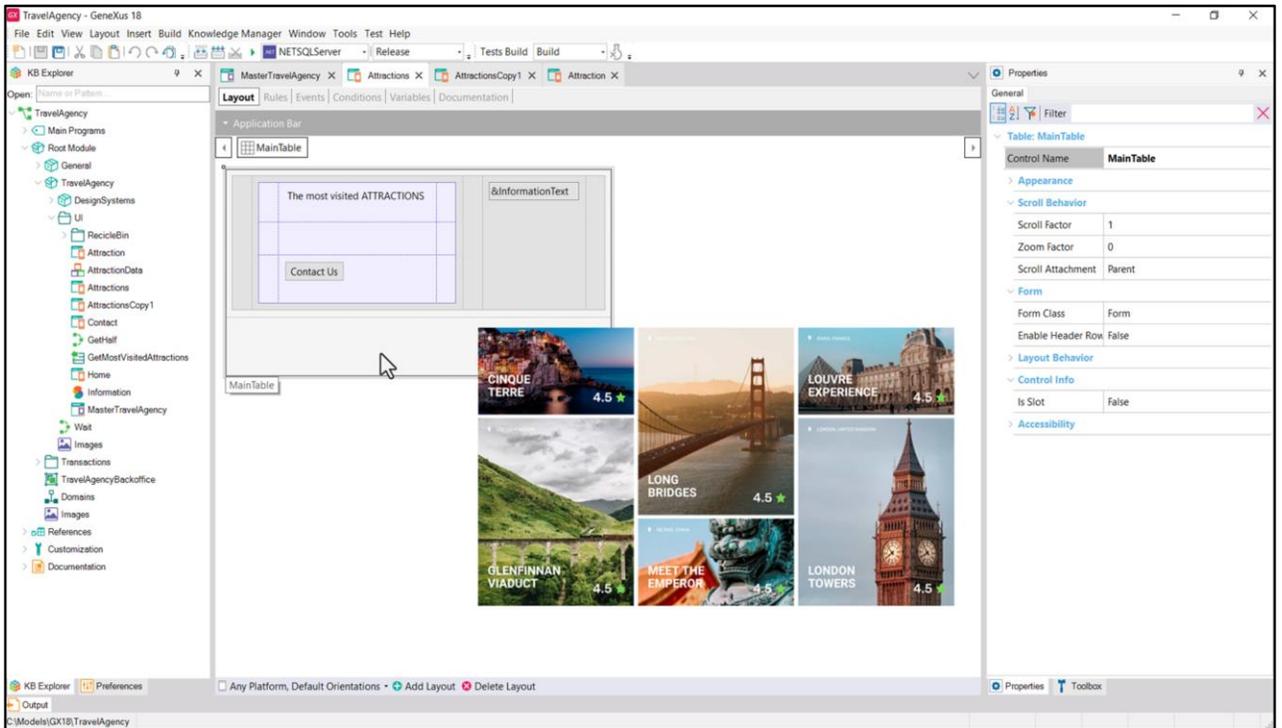
```
GlobalEvents.UpdateMasterPanelHeader('Contact')
```

endevent

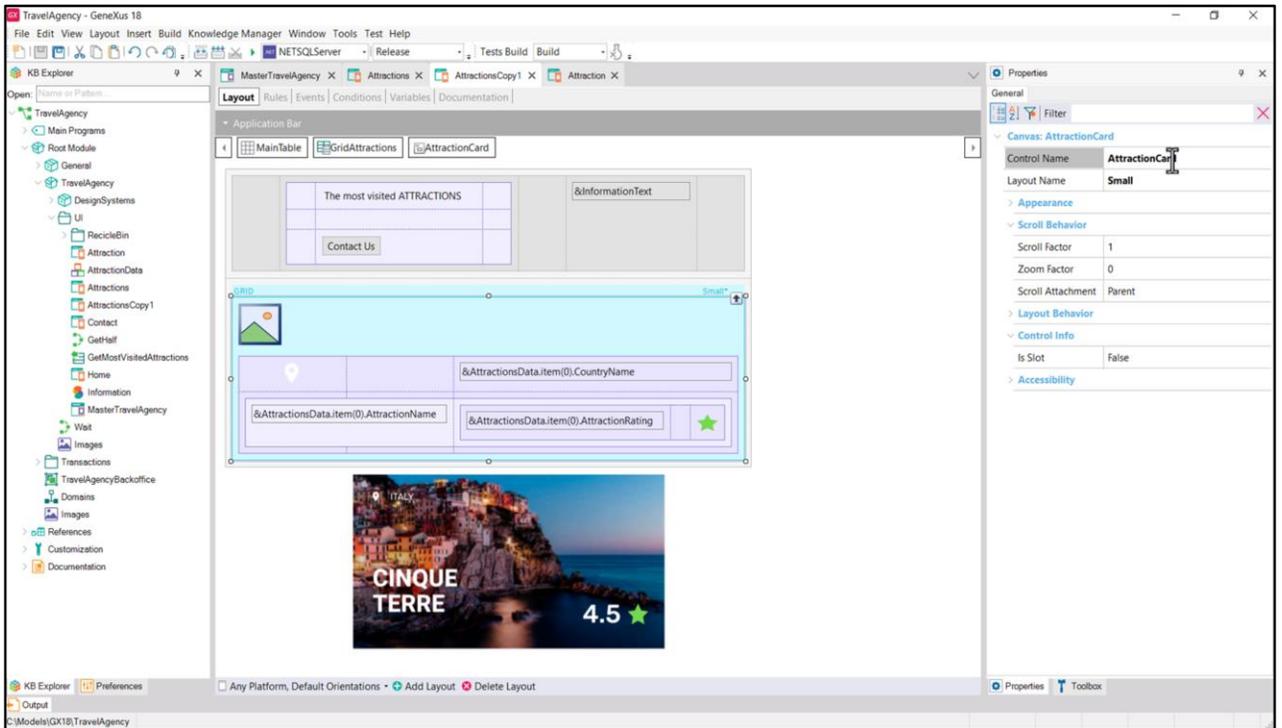
...or Contact...



We have yet to implement that the Master Panel loads in the Header the image and title of the attraction that the Attraction panel received by parameter when it was executed. And which was received when the user selected a tourist attraction in this grid.

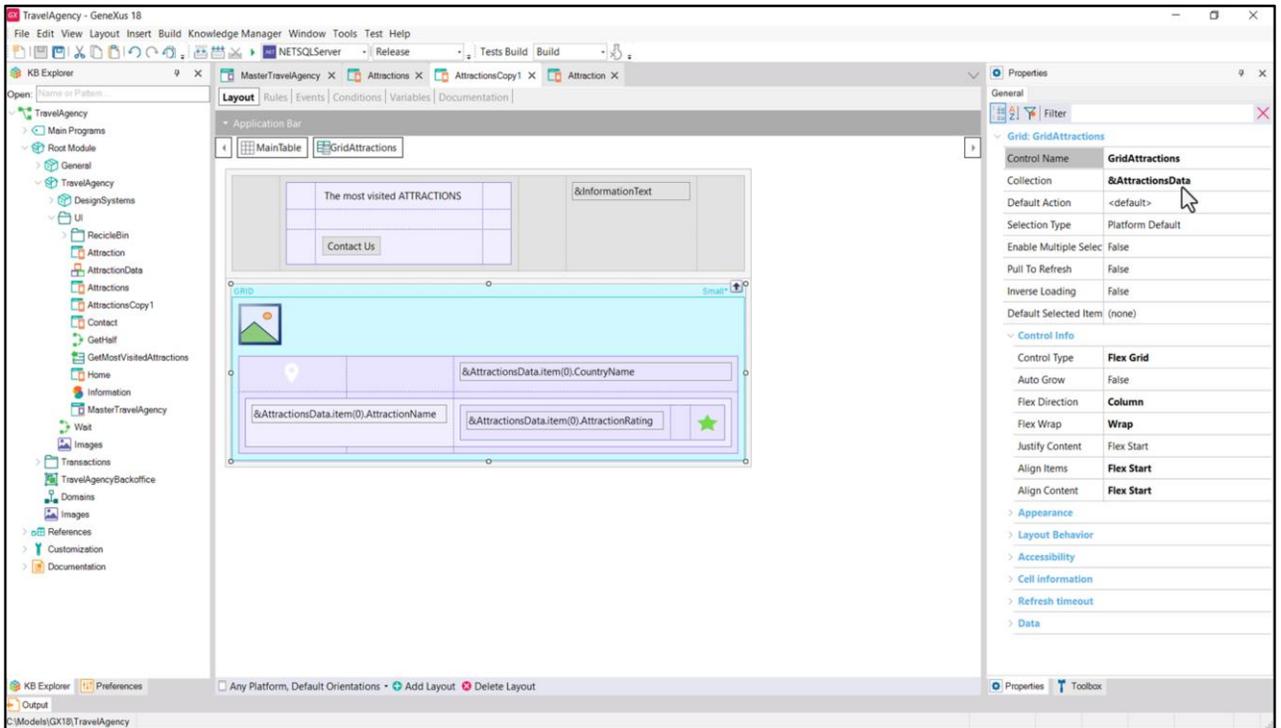


In order to implement this and test it, first I will need to add to the Attractions panel, in this second row, the carousel with the tourist attractions loaded from the database....

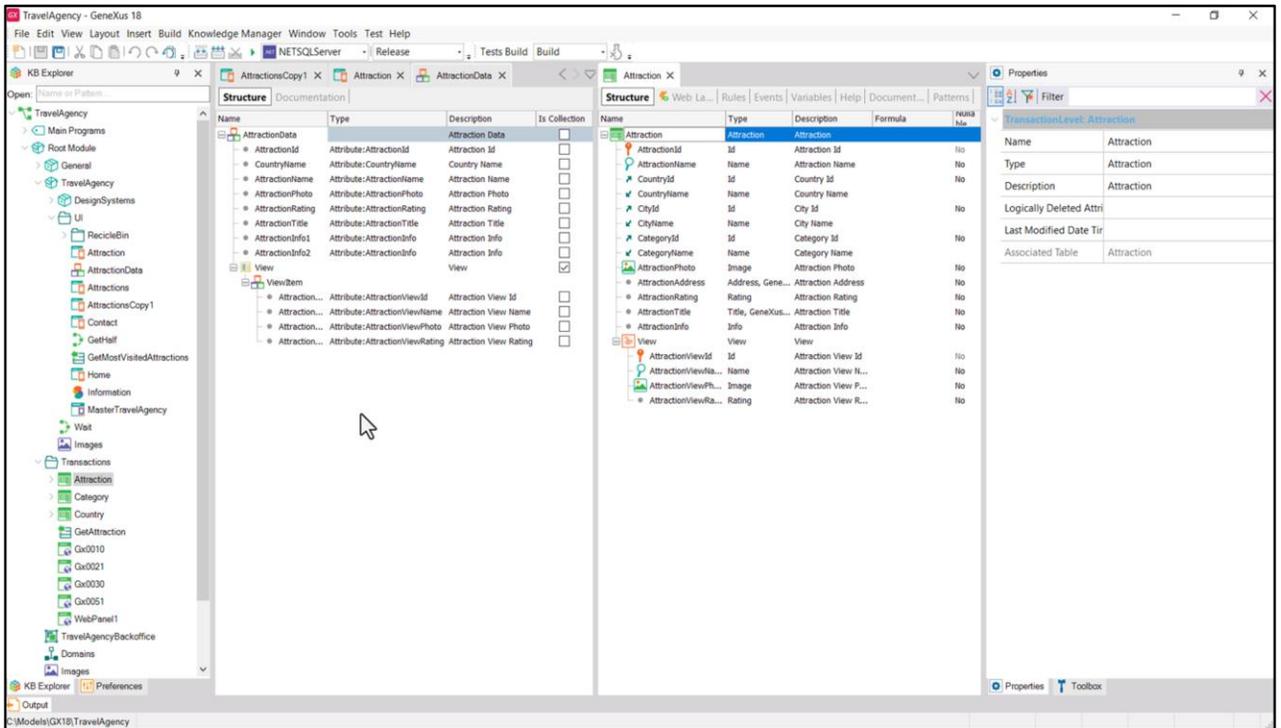


Here I made a copy of the panel and implemented it so that we don't waste time doing it now (in another video, we will see this implementation in detail, let's not worry about this). Just note that it is a grid, so it dynamically loads its items.

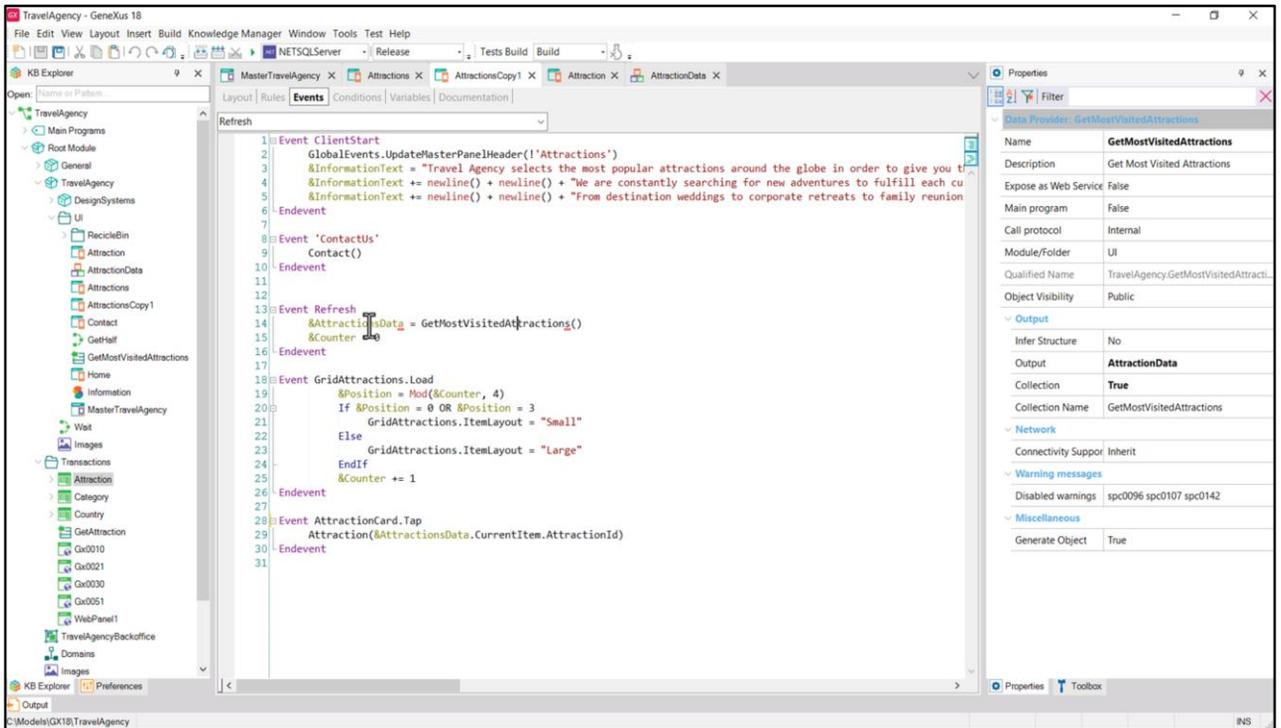
Each item is a canvas that I called AttractionCard, precisely to render each tourist attraction as a card.



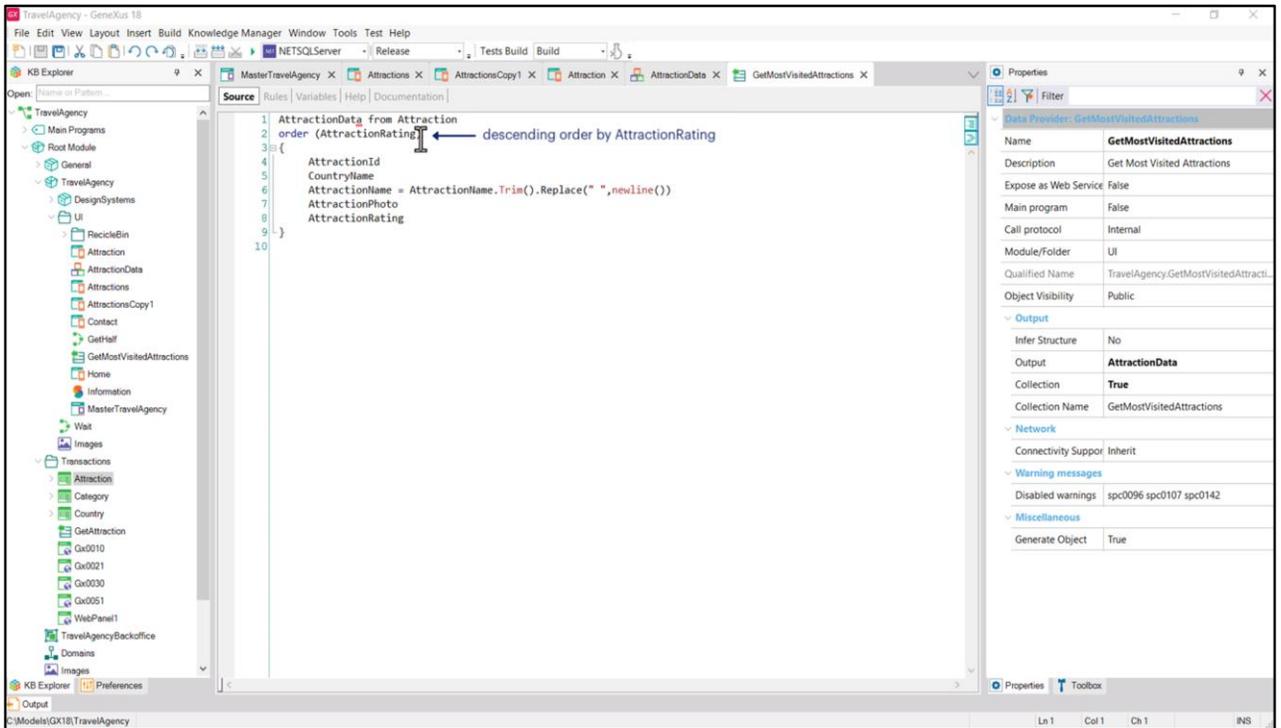
Actually there are many possible ways to implement this grid, here I chose one, where I load the grid items into this variable whose data type is a collection of this SDT.



See that it is almost identical to the structure of the Attraction transaction. I created it with the fields that I will need for the two panels Attractions and Attraction.

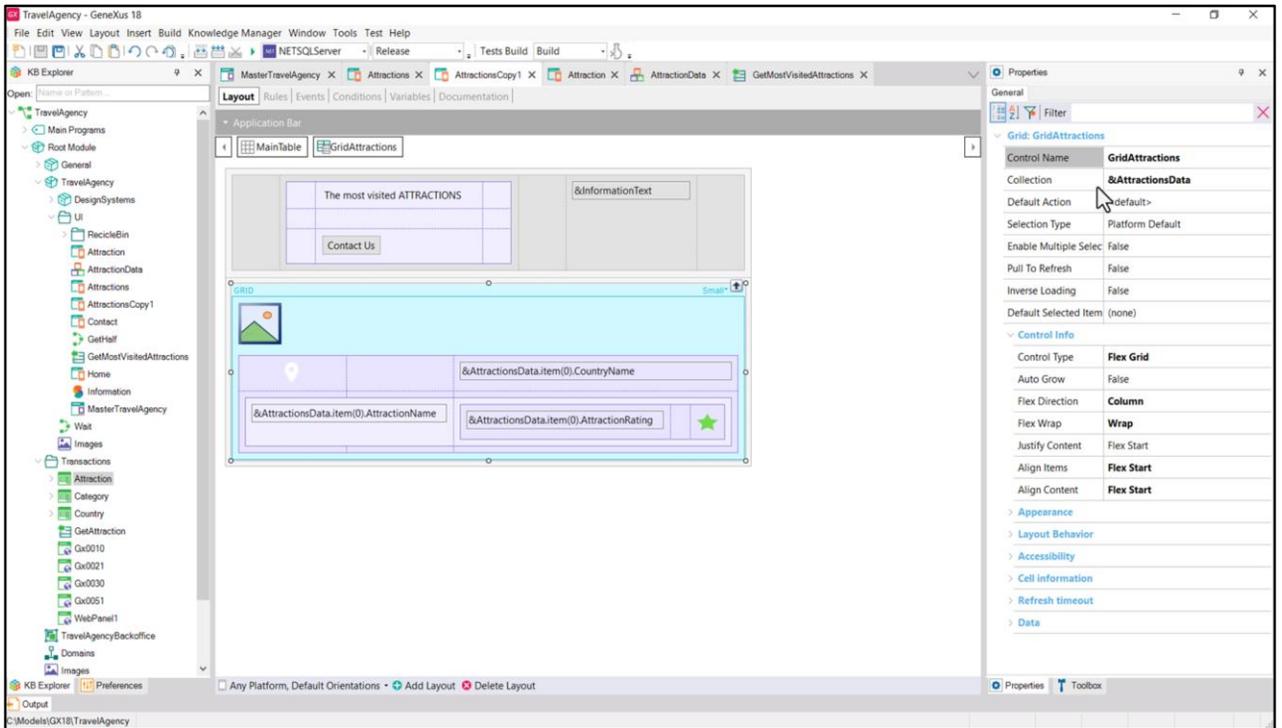


In short, what I do here is to call this data provider...

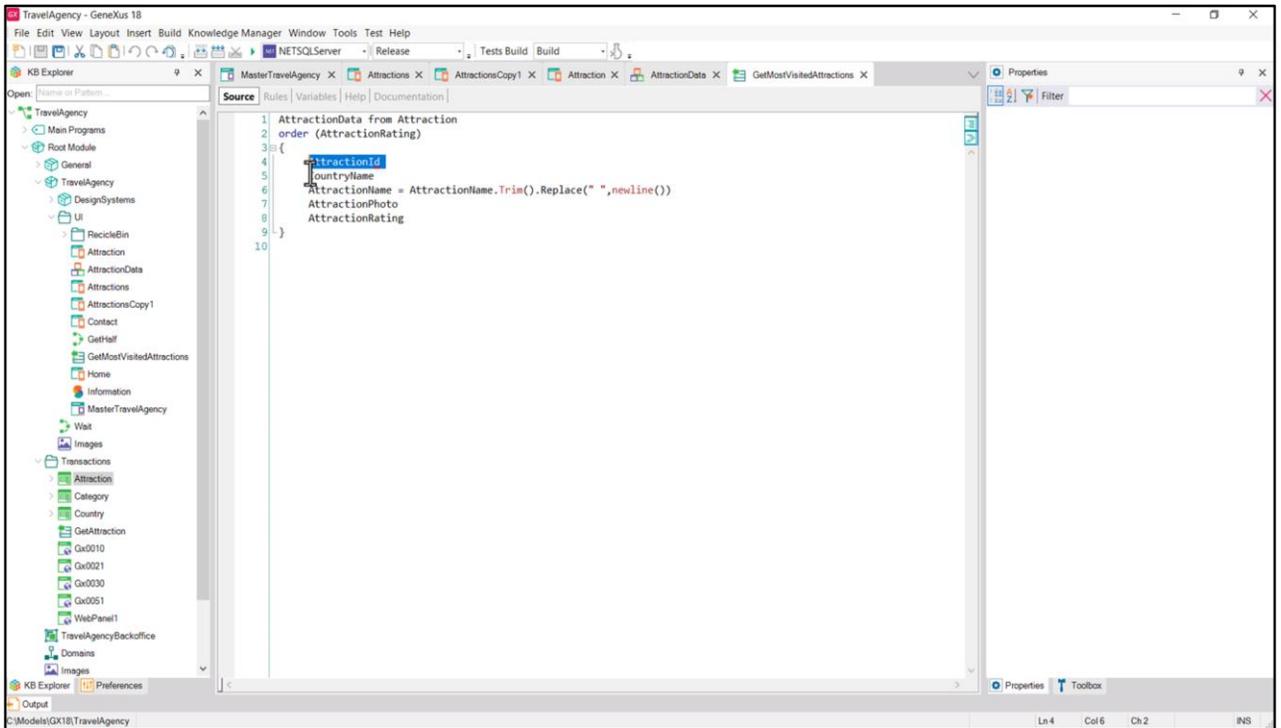


...that will return the SDT collection loaded from the database.

Here we are indicating the fields of each item that we are interested in getting to show in each Card. Note that we are not loading all the items, but only the ones we are interested in this time.

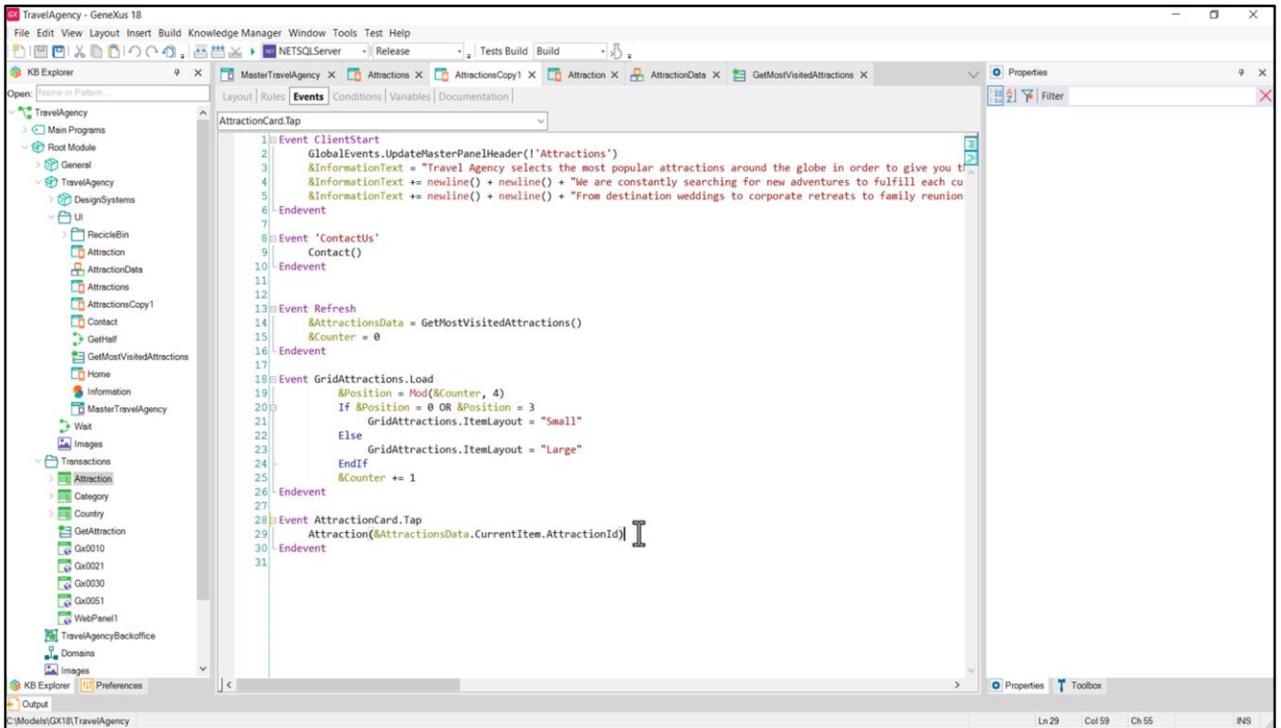


This collection will be the one automatically loaded in the grid, since I indicated it in its Collection property.

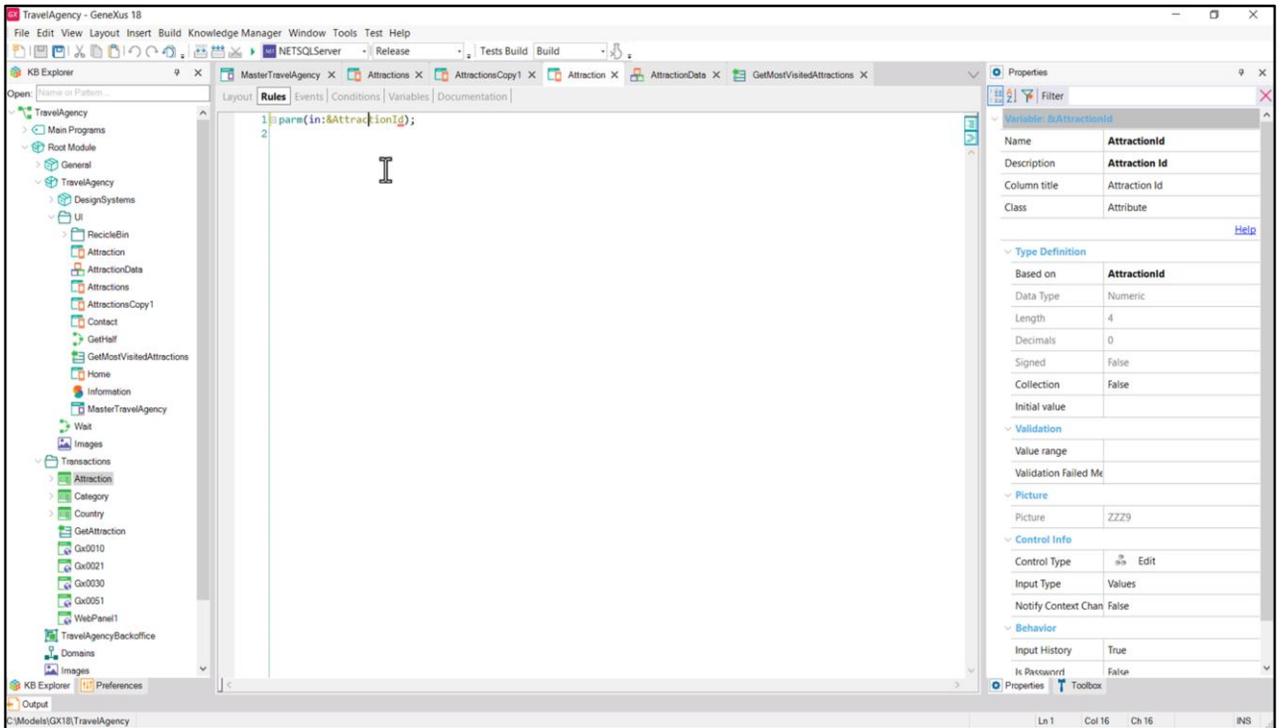


Once again, it is not important to understand this now. It is enough to know that this grid will be loaded from this SDT collection, which in turn will be loaded with the information we are interested in from the database.

In particular, let's see that each item in the collection will come with the AttractionId loaded.

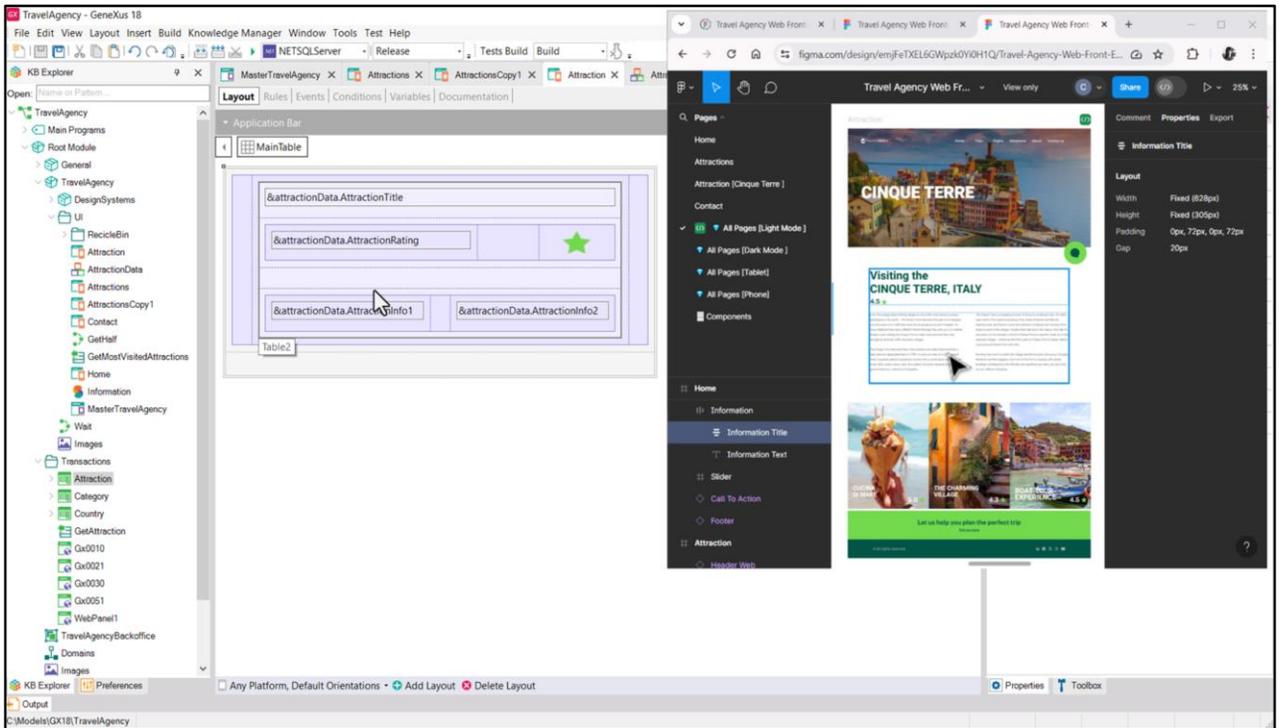


So what I did here was to program, on the canvas corresponding to each item of the grid, the tap event (which is the same as click for web). And there what I'm doing is to invoke the Attraction panel, passing it by parameter the value of AttractionId for the clicked on item.

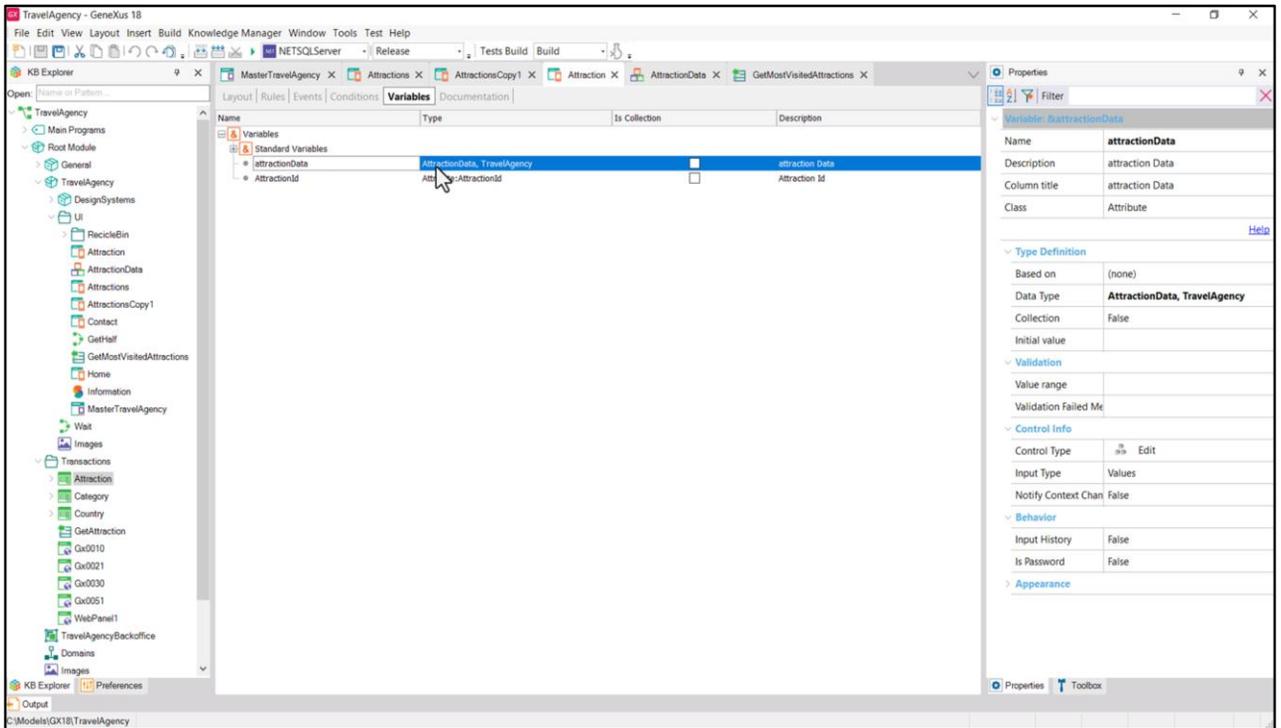


OK, with this we implemented the invocation.

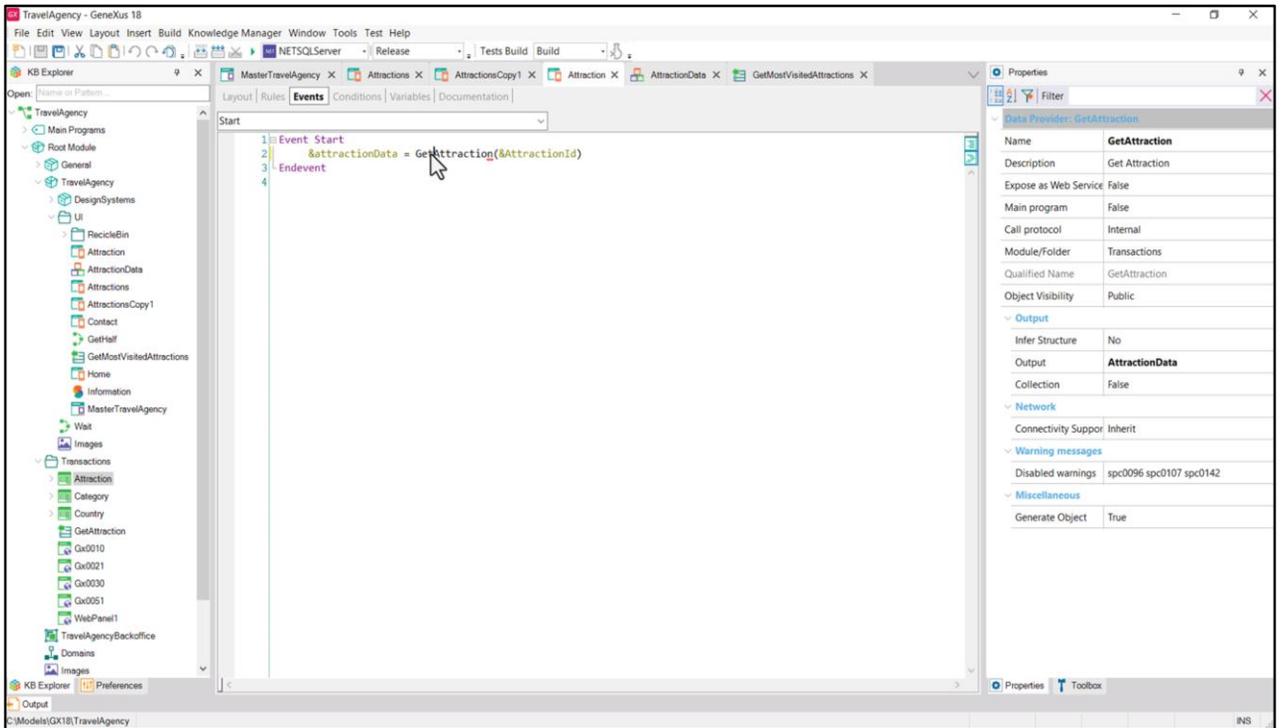
Then, in Attraction I already advanced a little bit too. Note that I defined the parm rule so that the panel can receive by parameter the identifier of the tourist attraction in this variable.



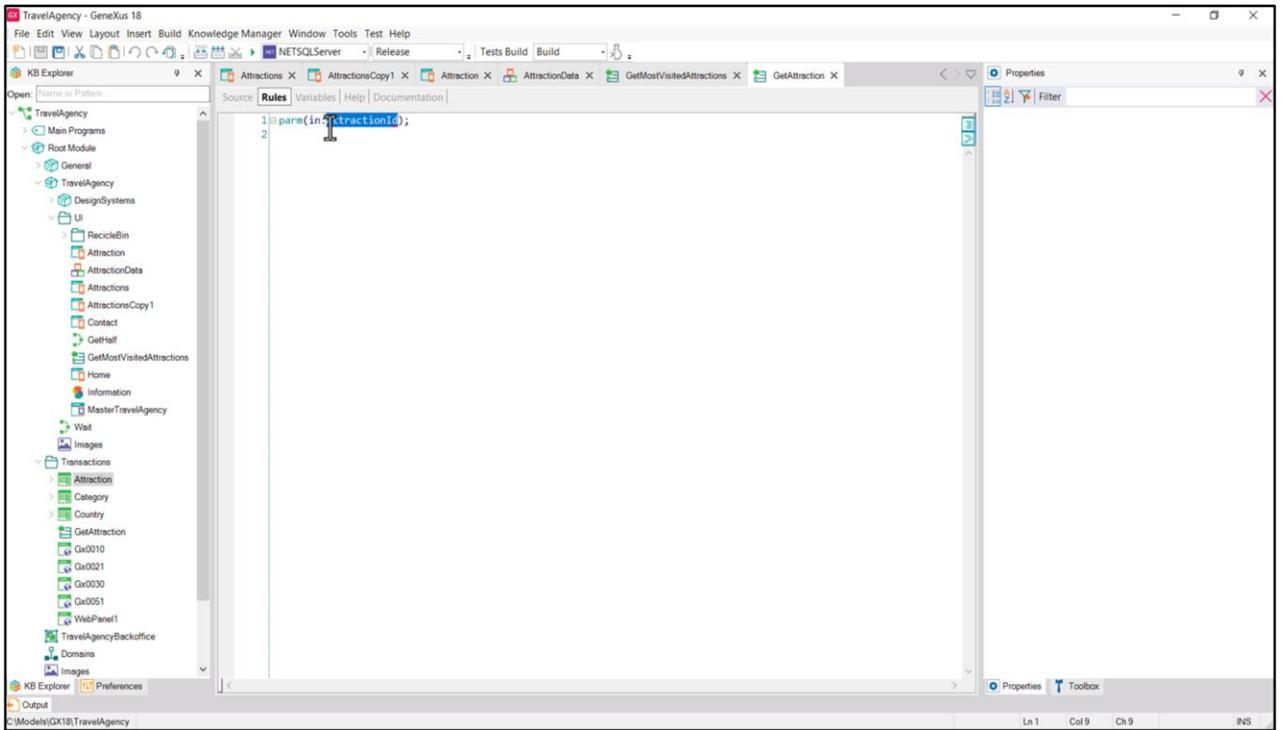
And for now, in its layout, I only implemented this part...



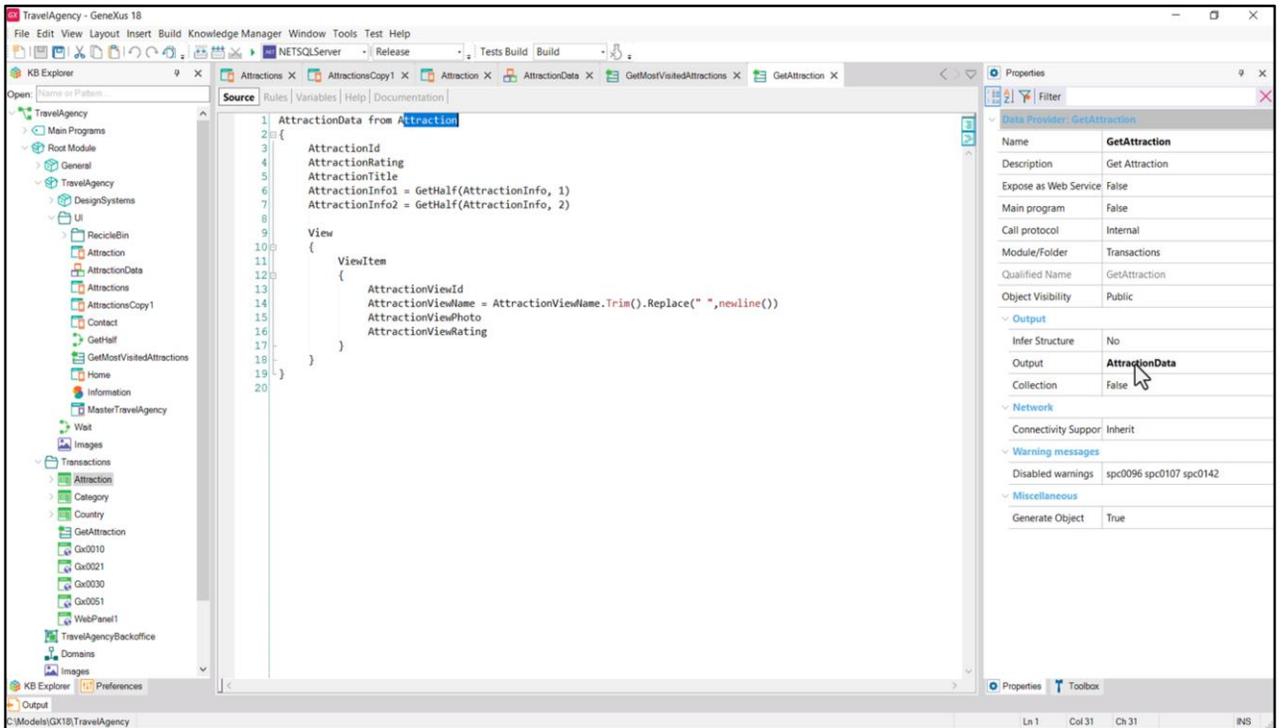
...for which I use another variable based on the same SDT, but this time the variable is not a collection.



And now I call another Data Provider to which I pass the attraction identifier received by parameter...



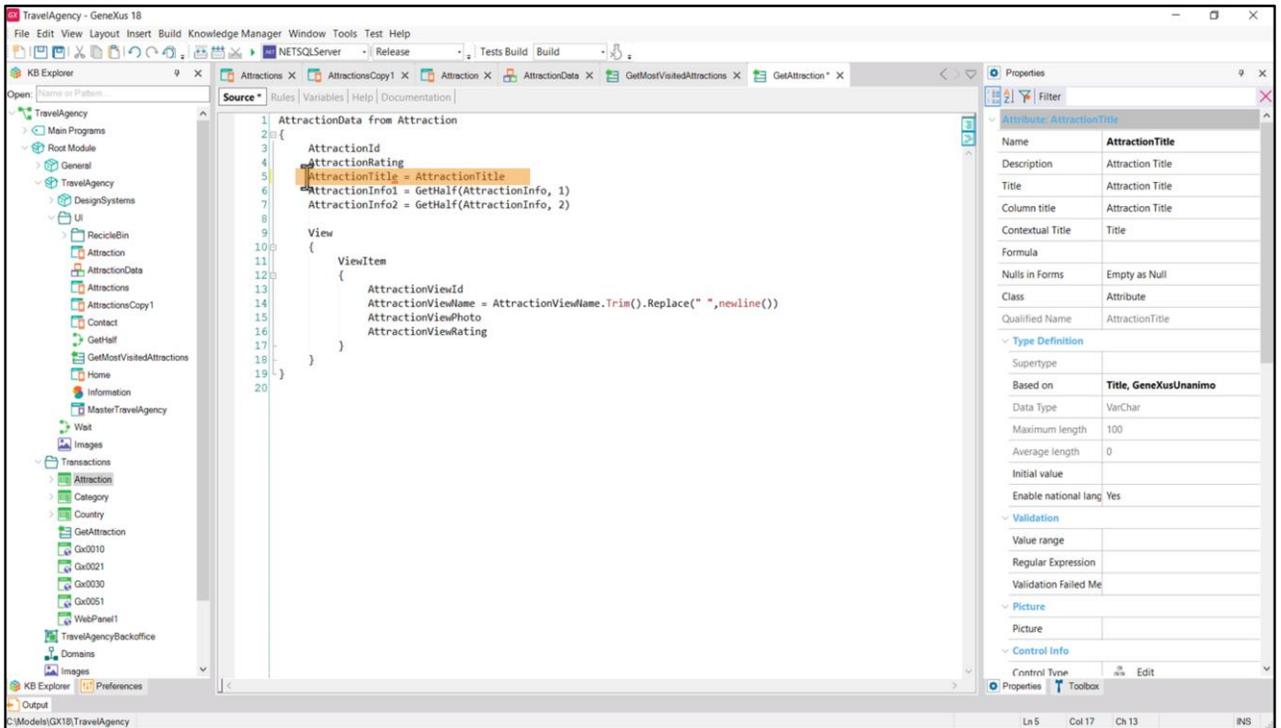
...and this Data Provider receives that value in the AttractionId attribute...



...so it will perform an automatic filter... when running through the Attraction base transaction in this group...

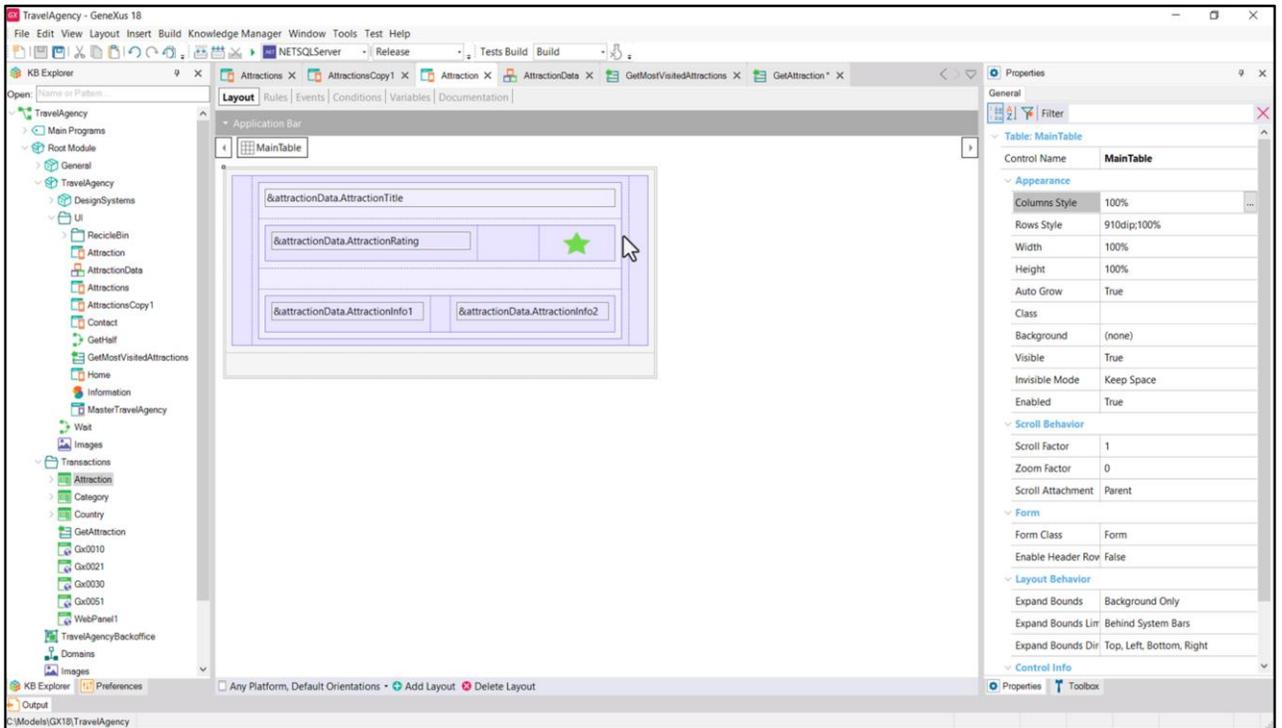
Because of the automatic filter, precisely, it will keep only the record of the Attraction table that has that parameter ID.

And I will load in the SDT that will return the Data Provider only what I need for the moment: the ID, the attraction rating, the title, the information divided in two blocks of text (to make this division I call a procedure that returns from the Info attribute the first half if I send 1, the second half if I send 2)...



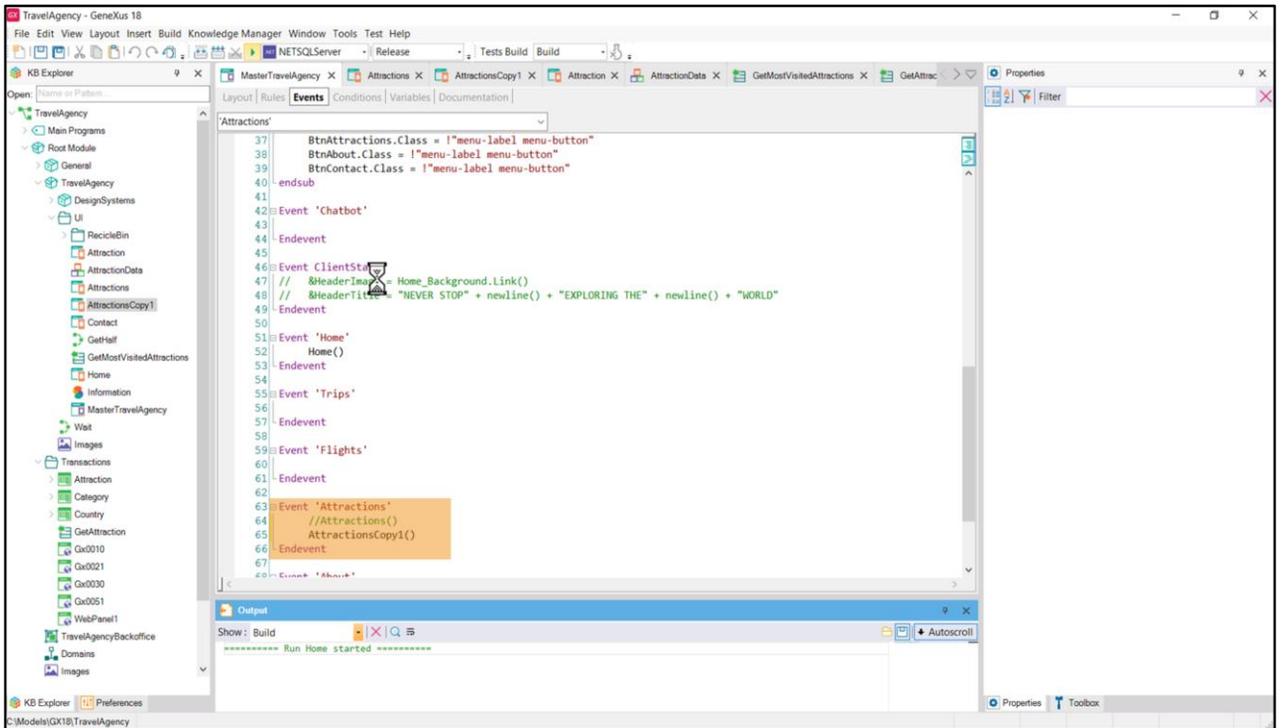
As a reminder to understand this code: since the SDT element and the base table attribute are called the same, I can use the shorthand notation, instead of having to explicitly do the assignment. For example, writing this is the same as this... But while the one on the left is the SDT element, the one on the right is the attribute of the database Attraction table.

The Data Provider will also return loaded items from this collection, but for now we will only use its first level...

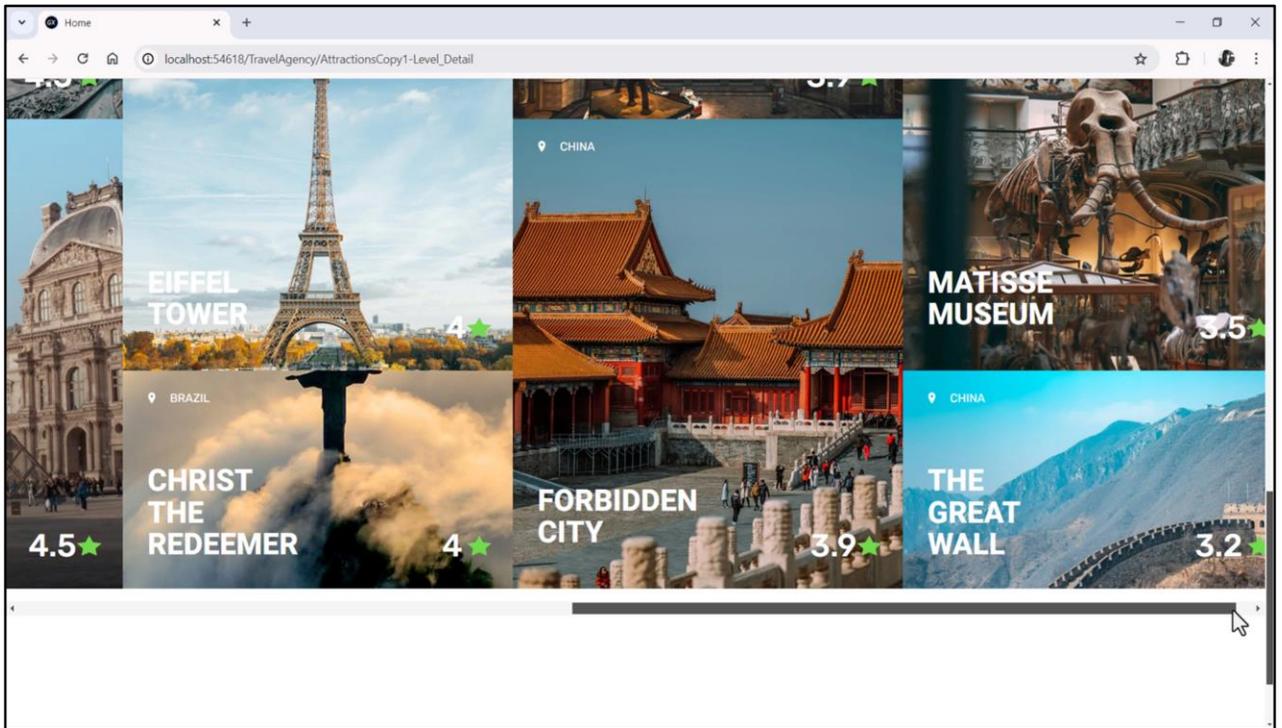


...which is what we use here in the Attraction layout.

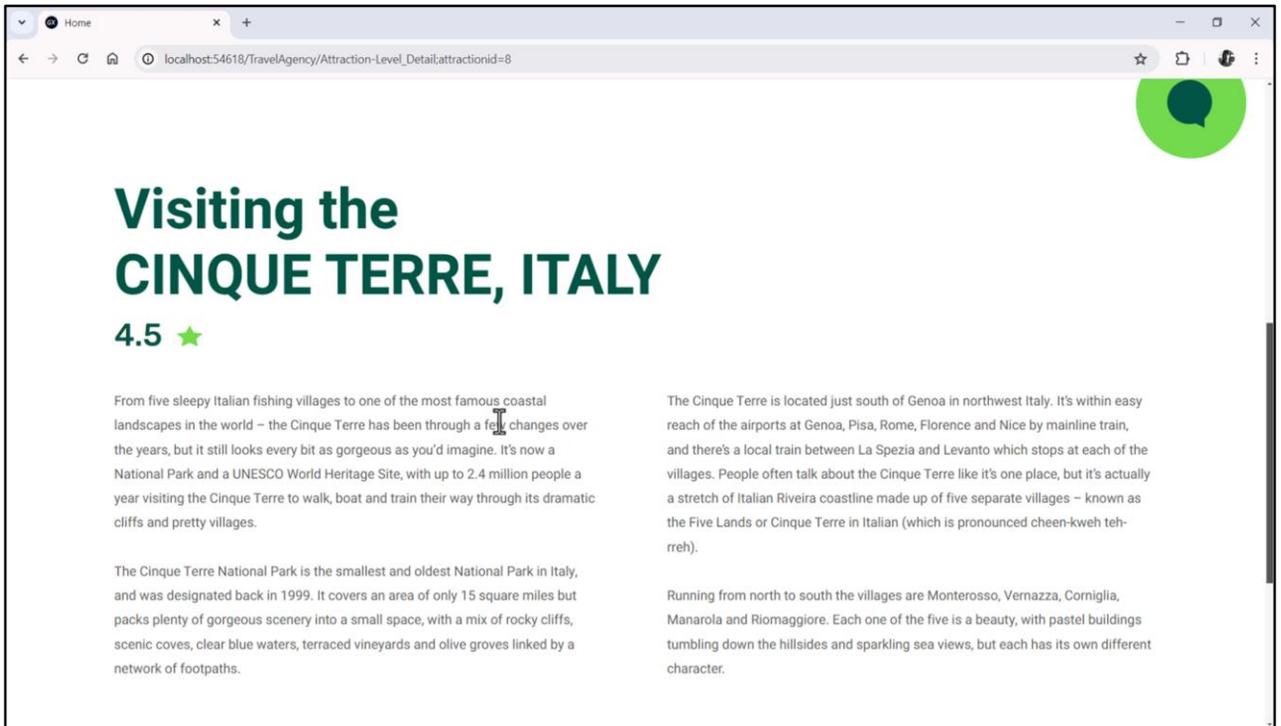
Before we implement the objective of this video, I'm going to run this so that we're clear on where we're starting from.



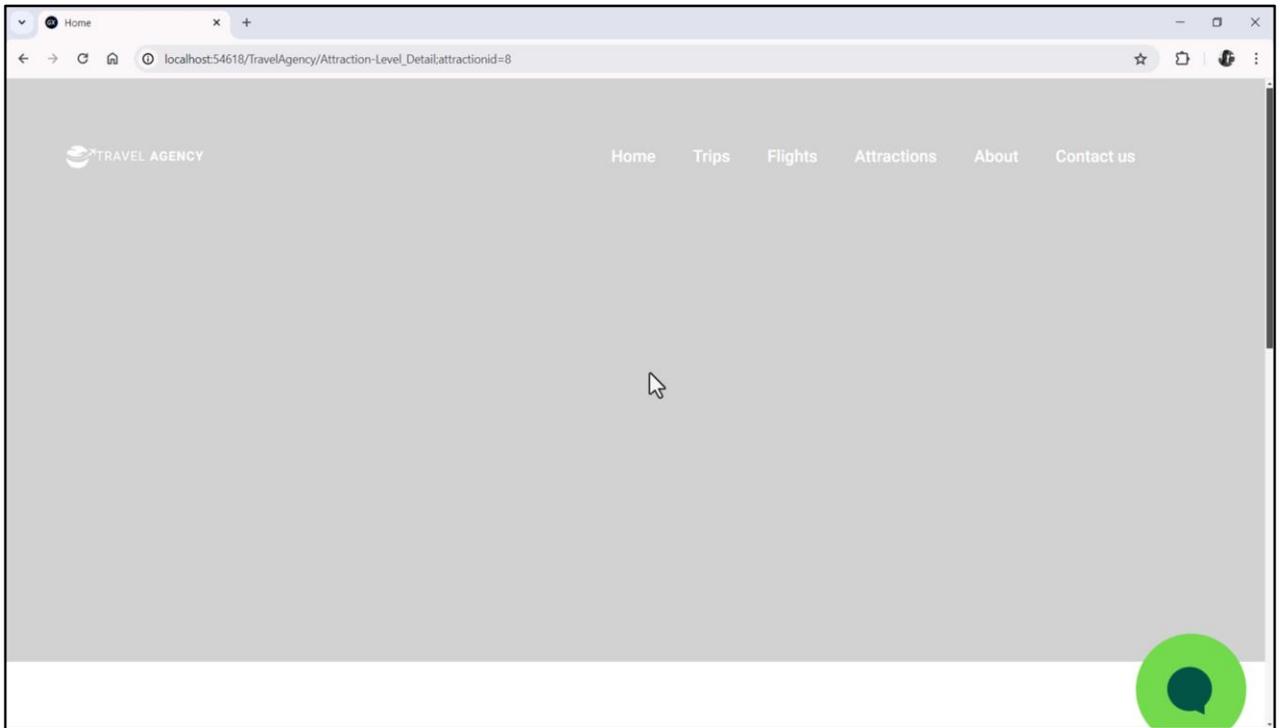
Here, in the Master Panel, in the event associated with the Attractions Menu button I'm calling the Attractions panel that doesn't have the grid. So what I'm going to do is to change the invocation for the panel that does have it. I run...



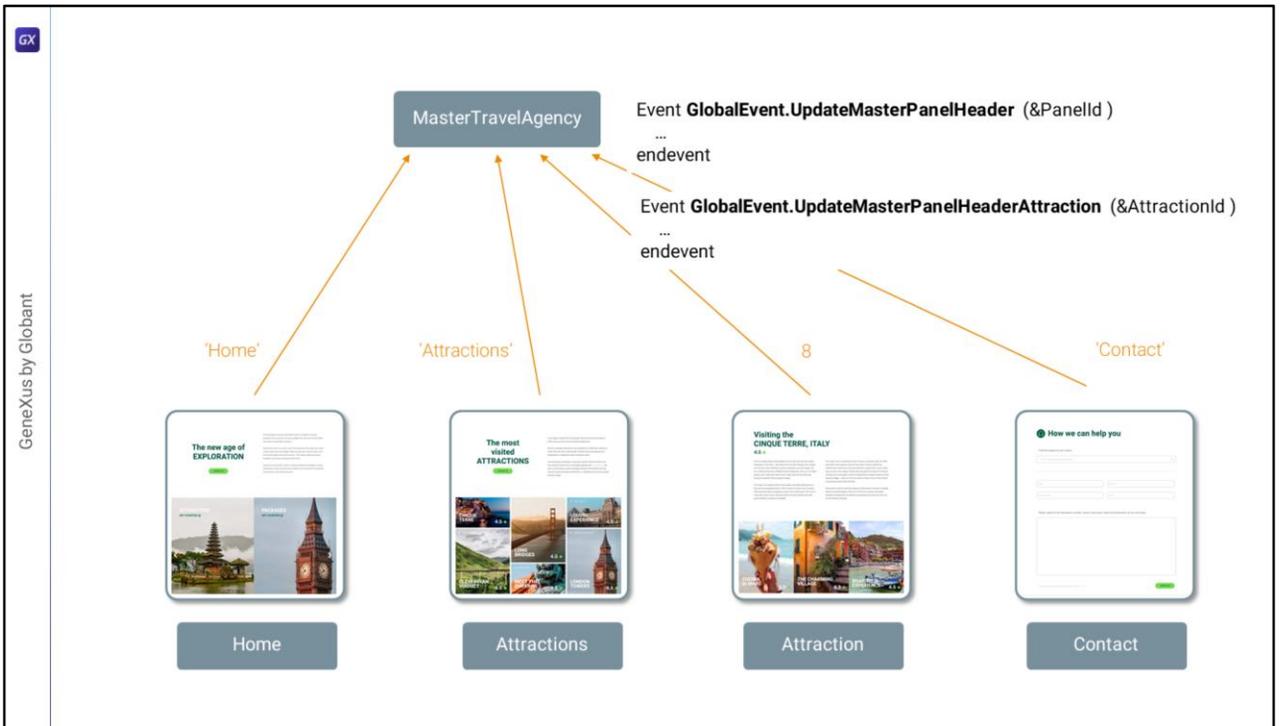
Here we have the Home, and if we go to the Attractions option we see the carousel of the attraction cards appear, with horizontal scroll...



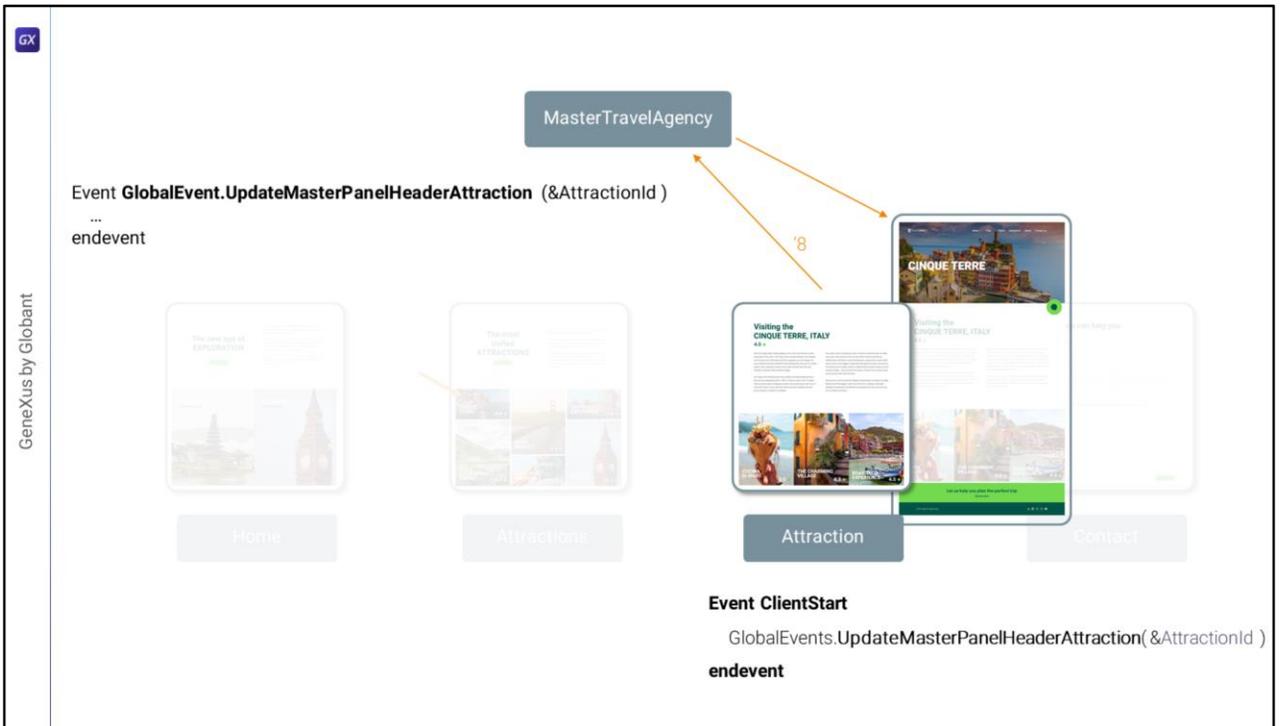
And if we click on an attraction, or tap, which is the same... it takes us to the Attraction panel, which has loaded the information of this part, the only one we have implemented so far.



But we don't see in the Master Panel Header neither the picture of the attraction nor its name. This is what we wanted to focus on in this video, so let's get to it.



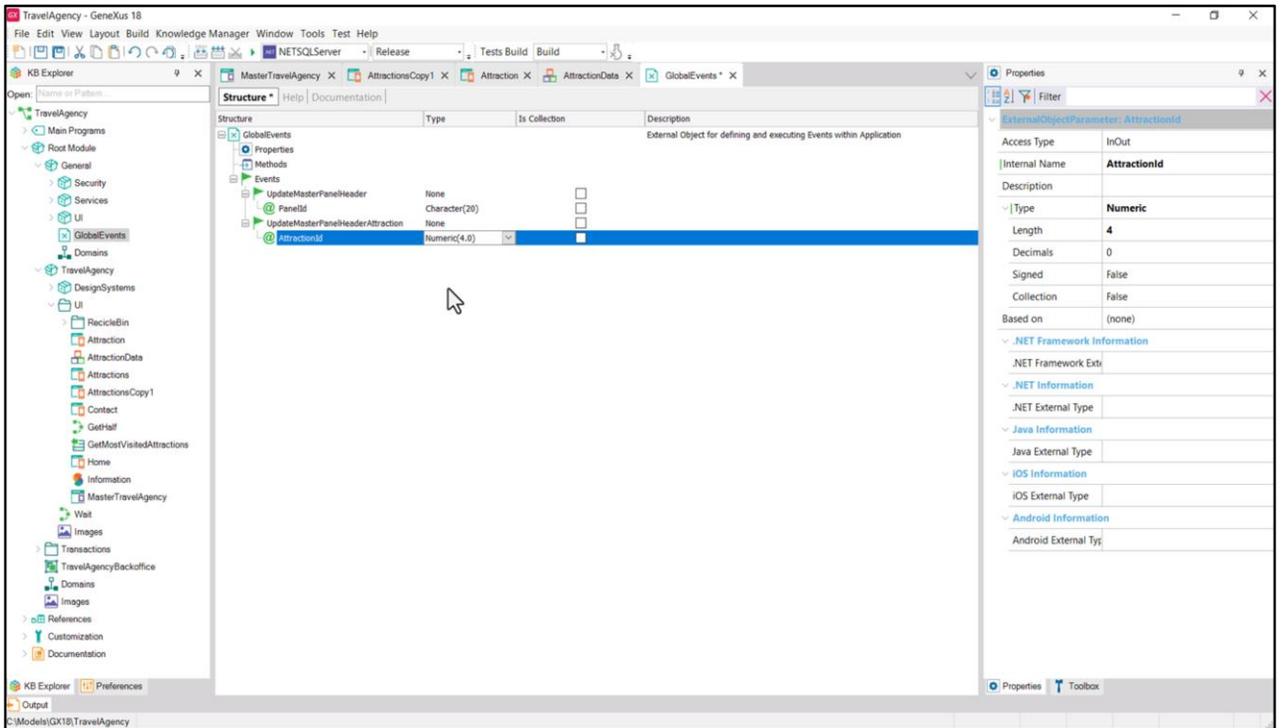
In the previous video we had decided to use two global events, one for the Home, Attractions and Contact panels, and a second global event for Attraction.



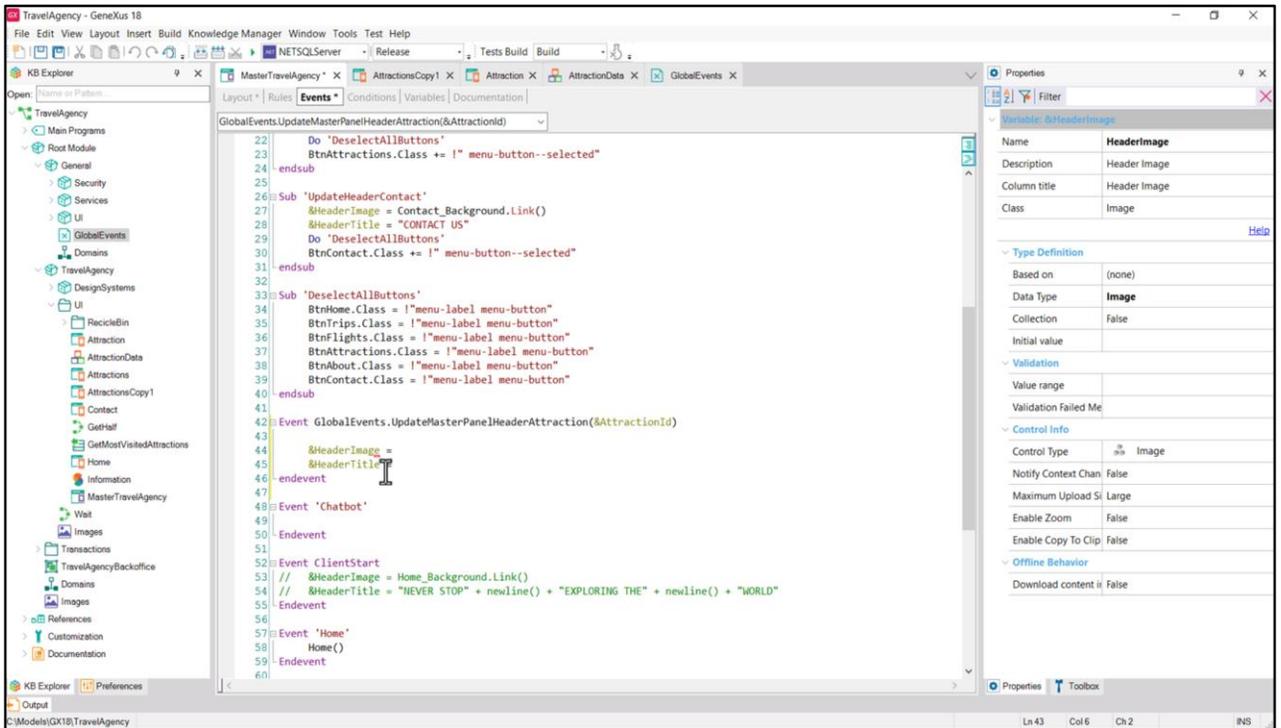
Then we will trigger it from the ClientStart of Attraction, passing it the ID of the attraction.

And we will have to implement it in the Master Panel, with its logic.

This is what we will do now in GeneXus.



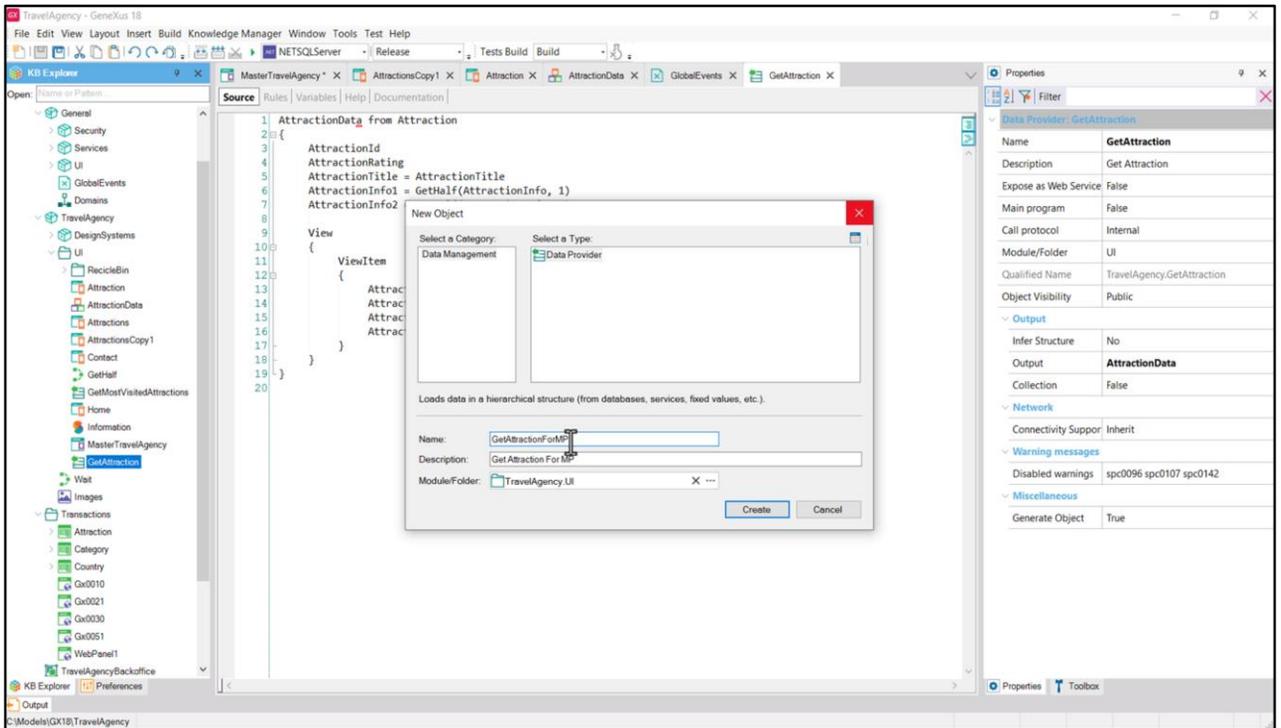
I open the GlobalEvents object and add this new event... with AttractionId parameter...



Now I go to the Master Panel, and add it in the events section, wherever I want.

I define the `&AttractionId` variable in this object (remember that the variables are local to each object).

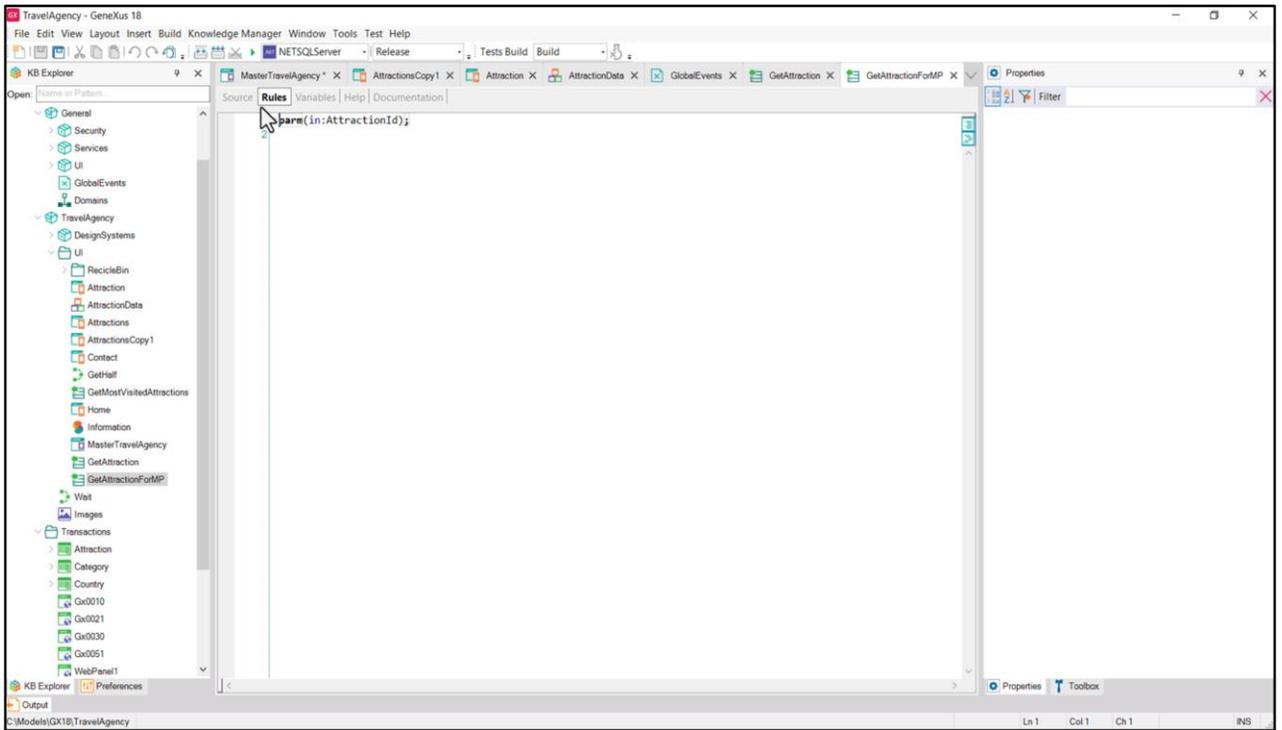
OK. I need to get from the database the name of the attraction and its photo, to assign them to these two variables of the Header of the Master Panel layout.



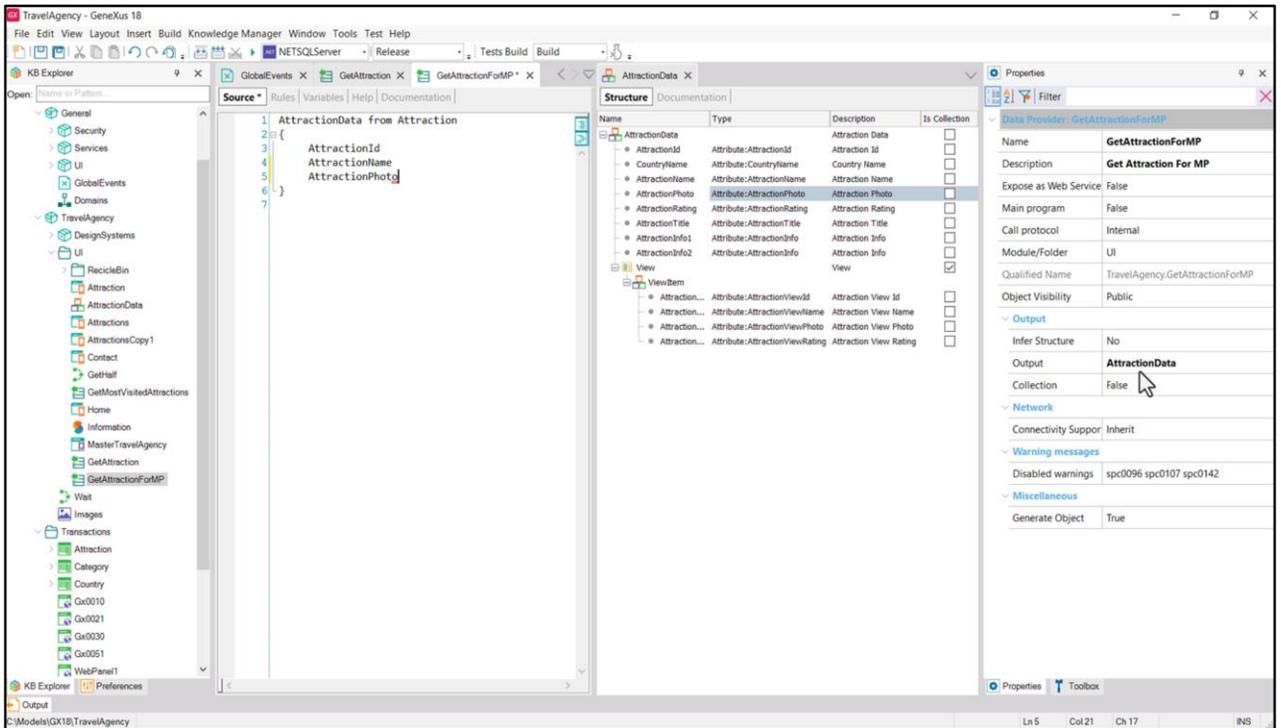
If I want to be extremely efficient I can do a Save as of the Data Provider that I used in Attraction to bring the attraction information, in order to return only these two values loaded... and everything else empty.

It was saved by mistake in the Transaction folder, so I'll move it to this other one.

I do the Save as, assign it this name...

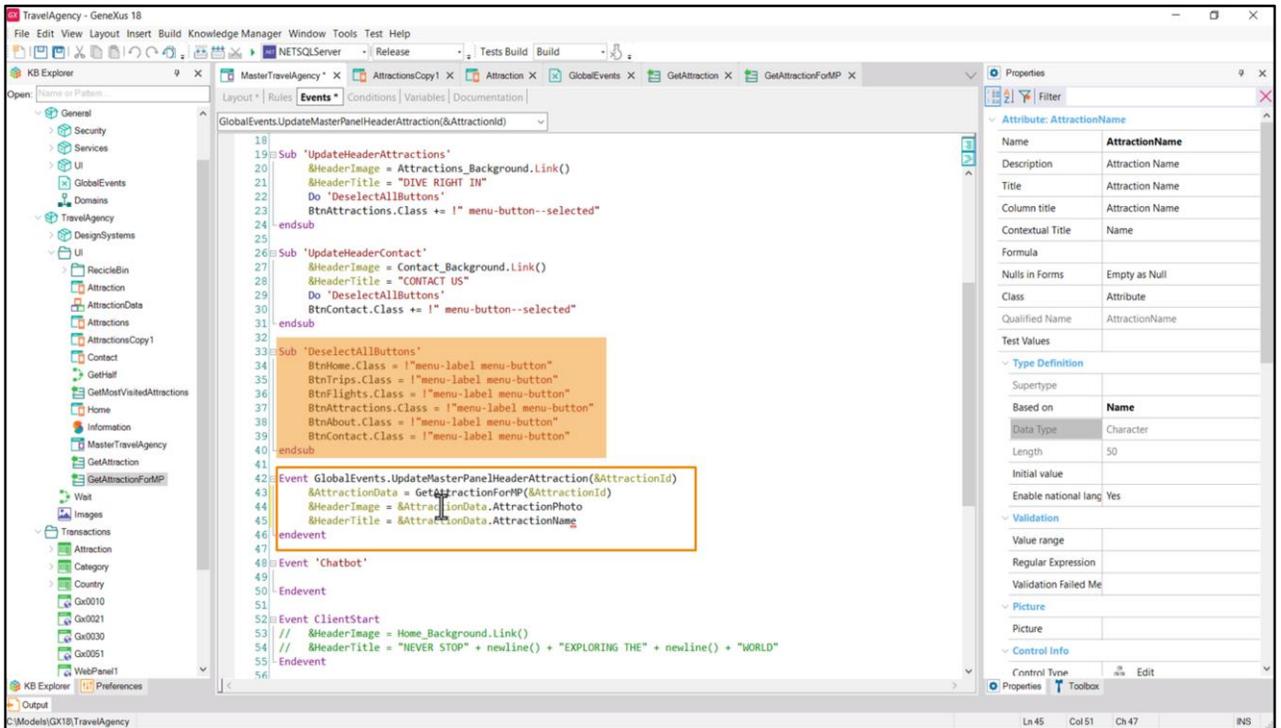


It will receive the parameter in the AttractionId attribute...



...and in the source we will remove all these assignments and leave only the loaded elements that we are interested in... `AttractionName` and `AttractionPhoto`.

It will return this SDT loaded only with these three elements. The rest will be empty.

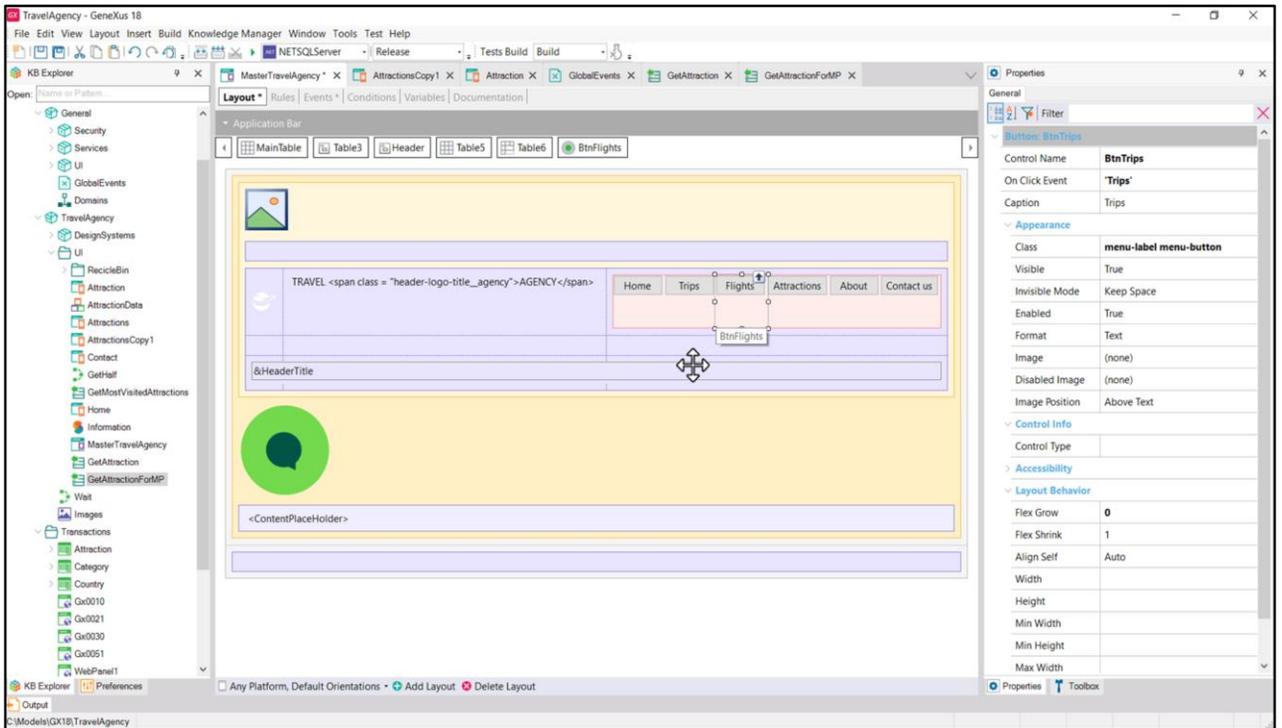


And then, in the Master Panel, we assign to a variable of the SDT data type the result of executing the data provider, passing as parameter the value of this &AttractionId variable that is the parameter that receives the global event when it is fired.

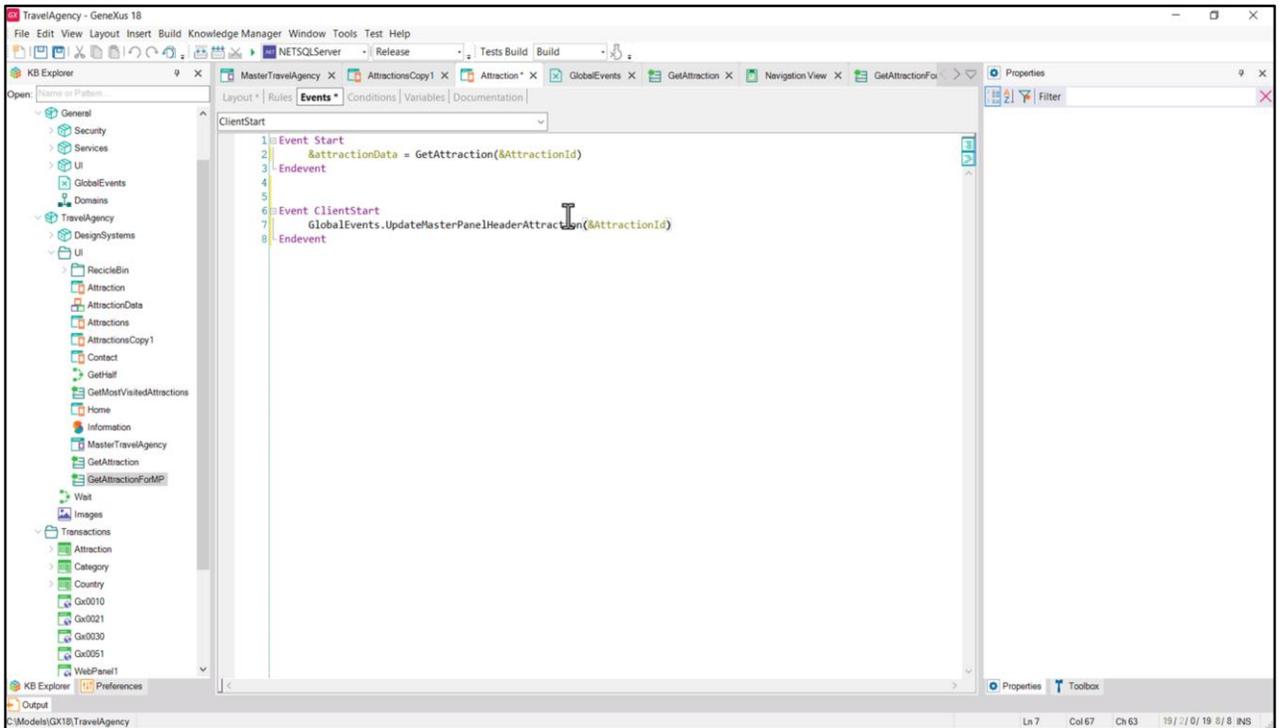
Next, to the &HeaderImage variable of the layout we assign the AttractionPhoto element of the SDT.

And to &HeaderTitle the AttractionName element.

What about the menu buttons? Should we call this subroutine to deselect them?

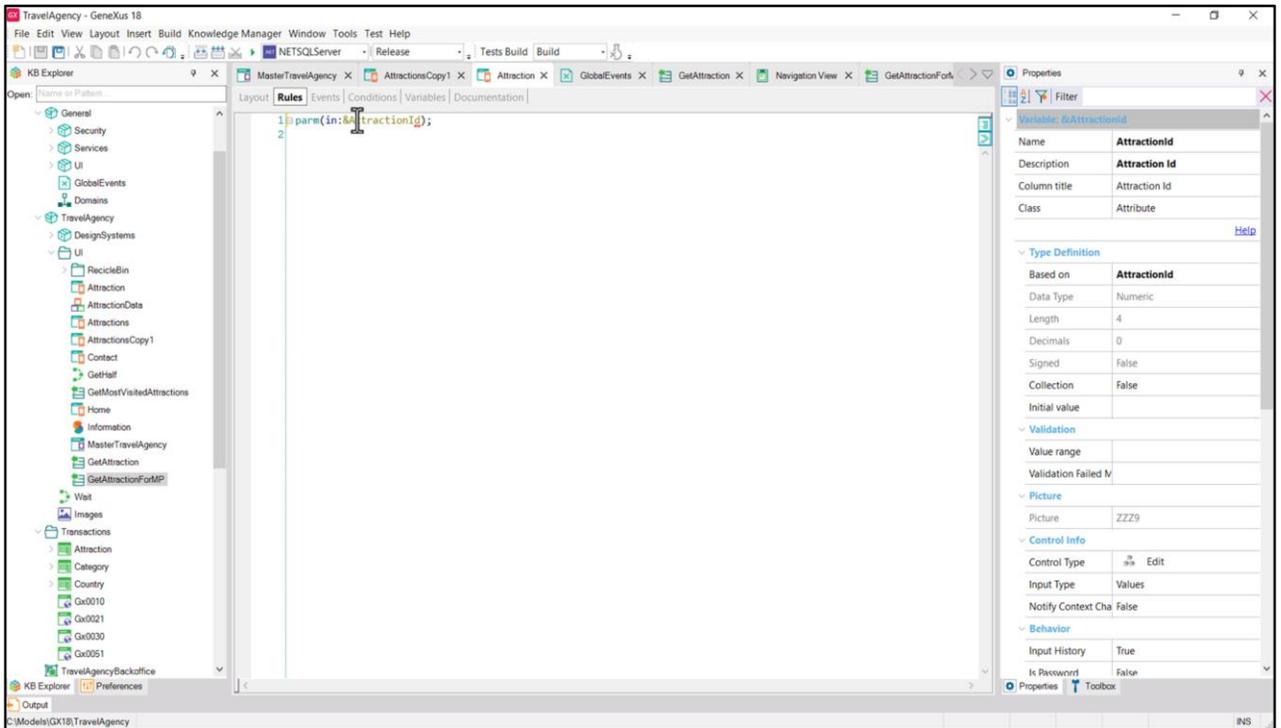


Actually, it is not necessary because these same classes are already statically determined for the buttons.



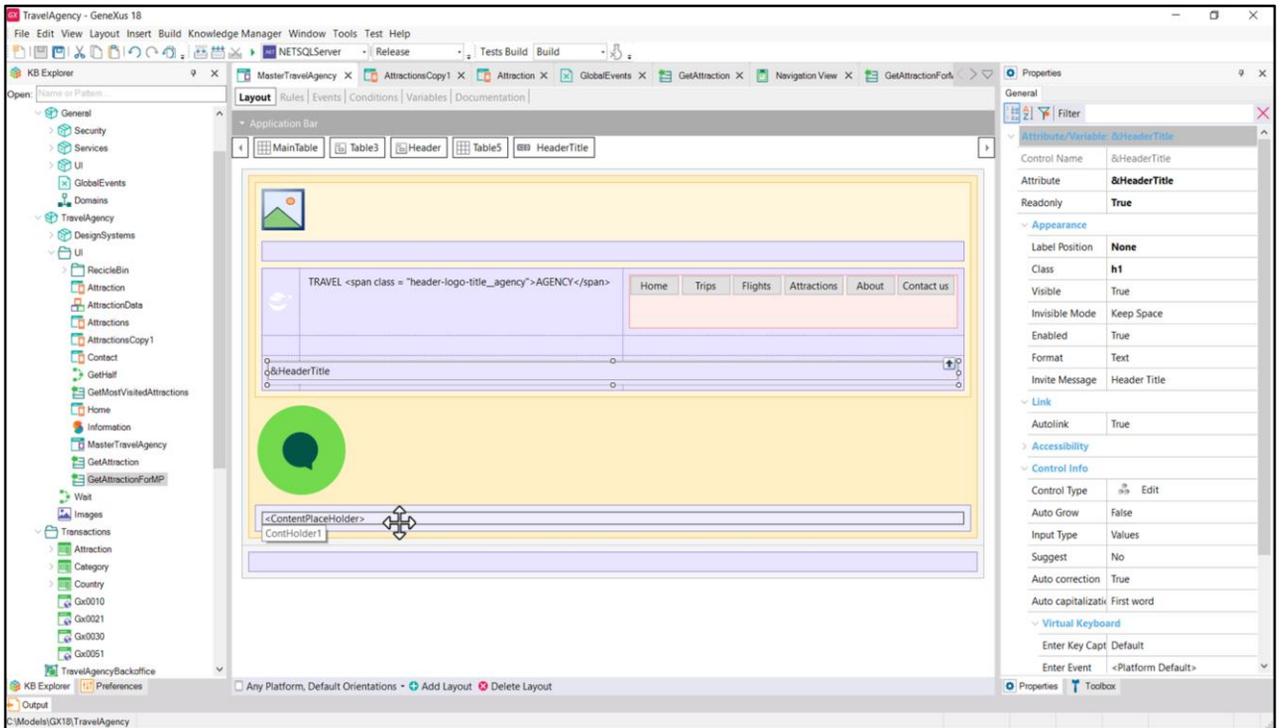
The only thing we need now is to fire this event from the Attraction panel.

So in the ClientStart event... we invoke the global event that we want, passing it as parameter this variable, the one that receives the Attraction panel. And from which its layout is loaded.



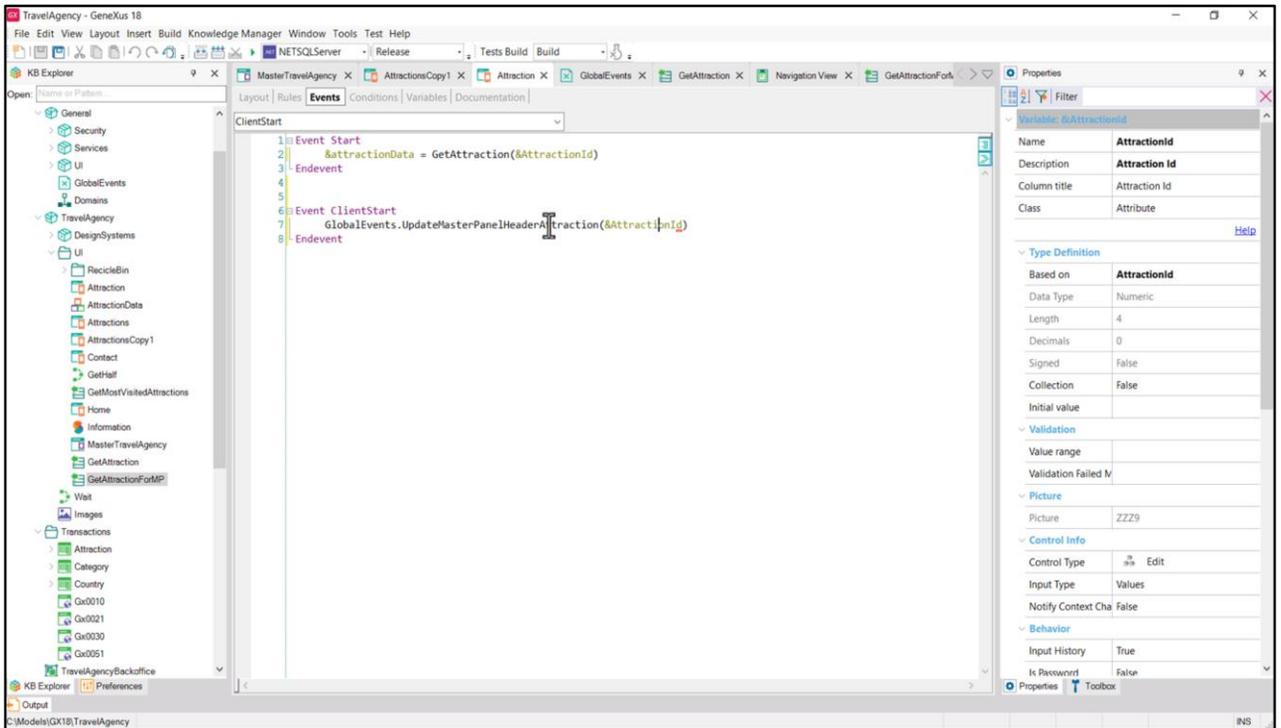
In short, let's review everything.

When the Attraction panel is executed, it receives in this variable the ID of the attraction in question.



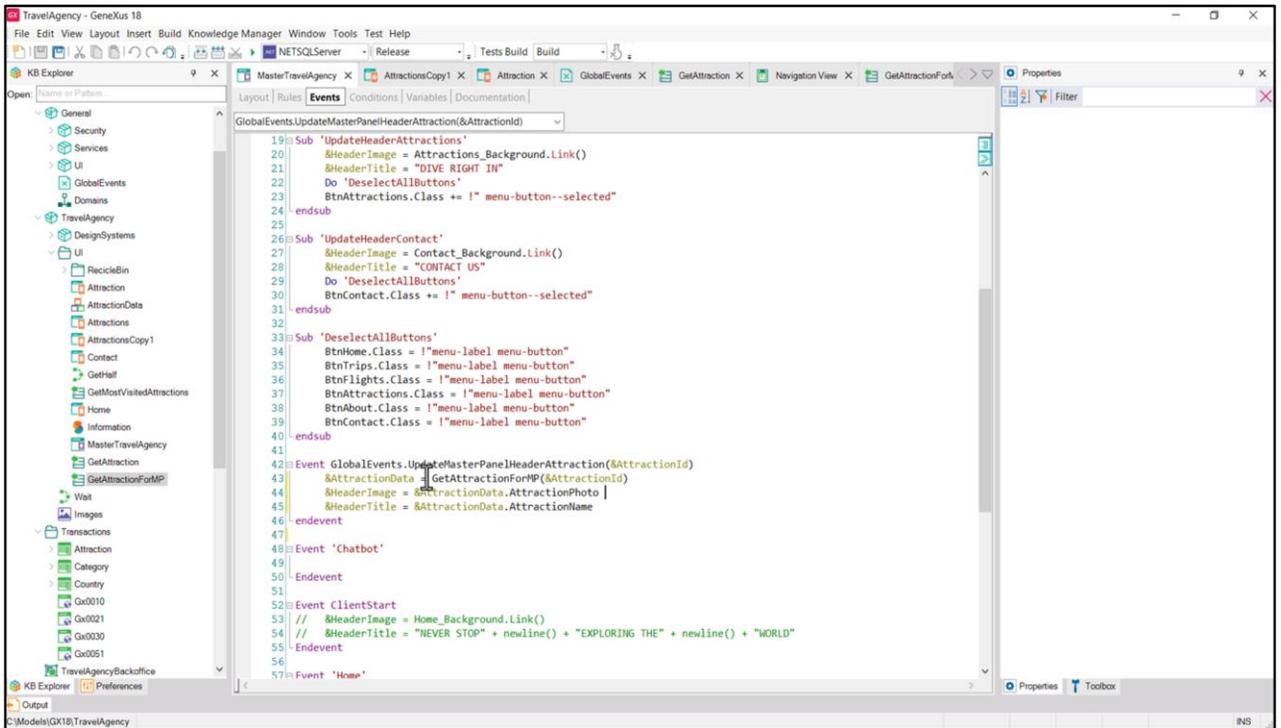
The first thing it does, as we knew, is to execute the Master Panel, and its ClientStart system event, which in this case is empty. And it renders the Master Panel for the first time. At that moment the header variables will be empty.

Then it is executed to load the Attraction panel itself in the ContentPlaceHolder.



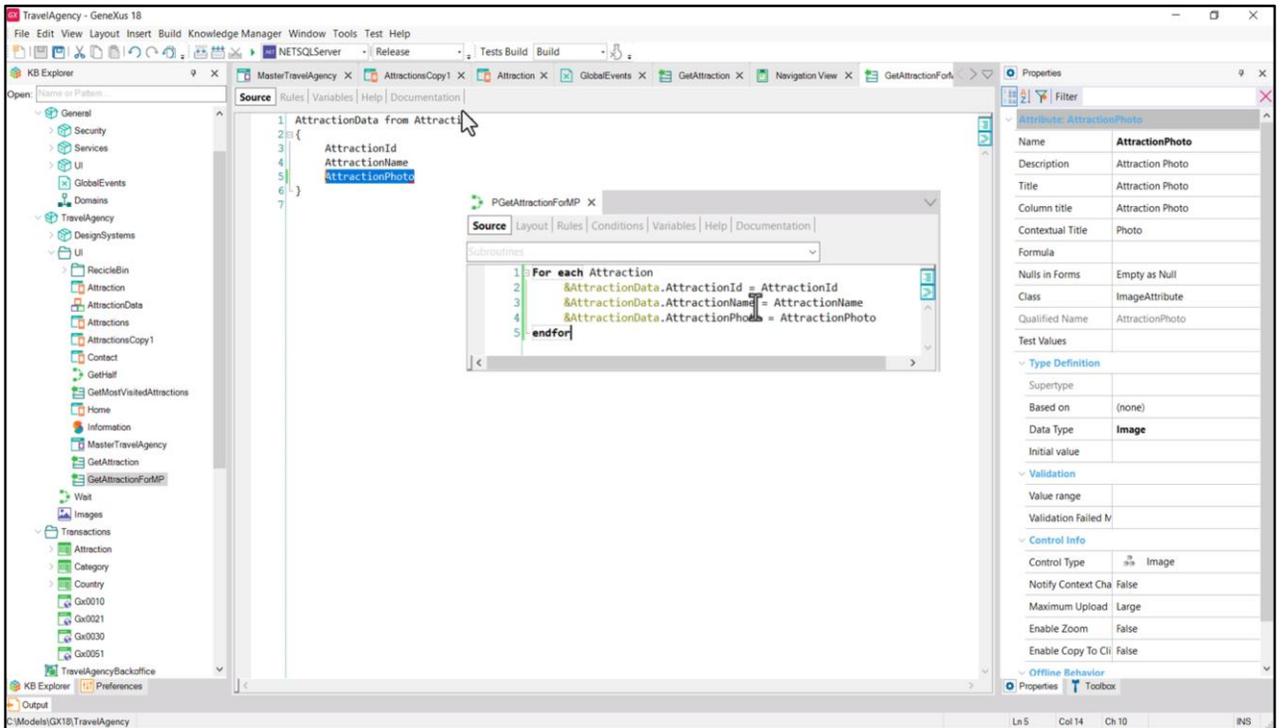
The static properties of the layout controls are assigned and then the ClientStart event is executed. Here is where the global event of the Master Panel is fired, which, as we said in the previous video, is asynchronous.

This event is already sent the attraction ID received by parameter.

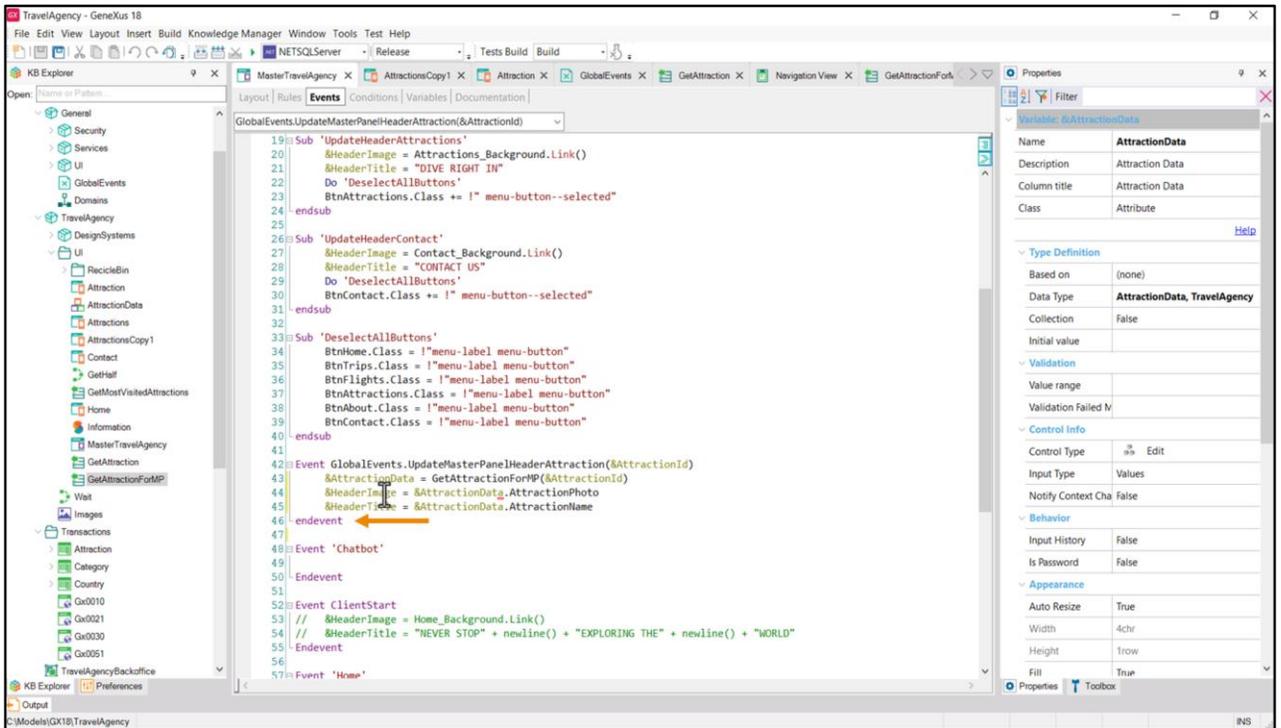


And here it executes this in the Master Panel. Here is where the image and the title are loaded from the SDT that returned this loaded data provider.

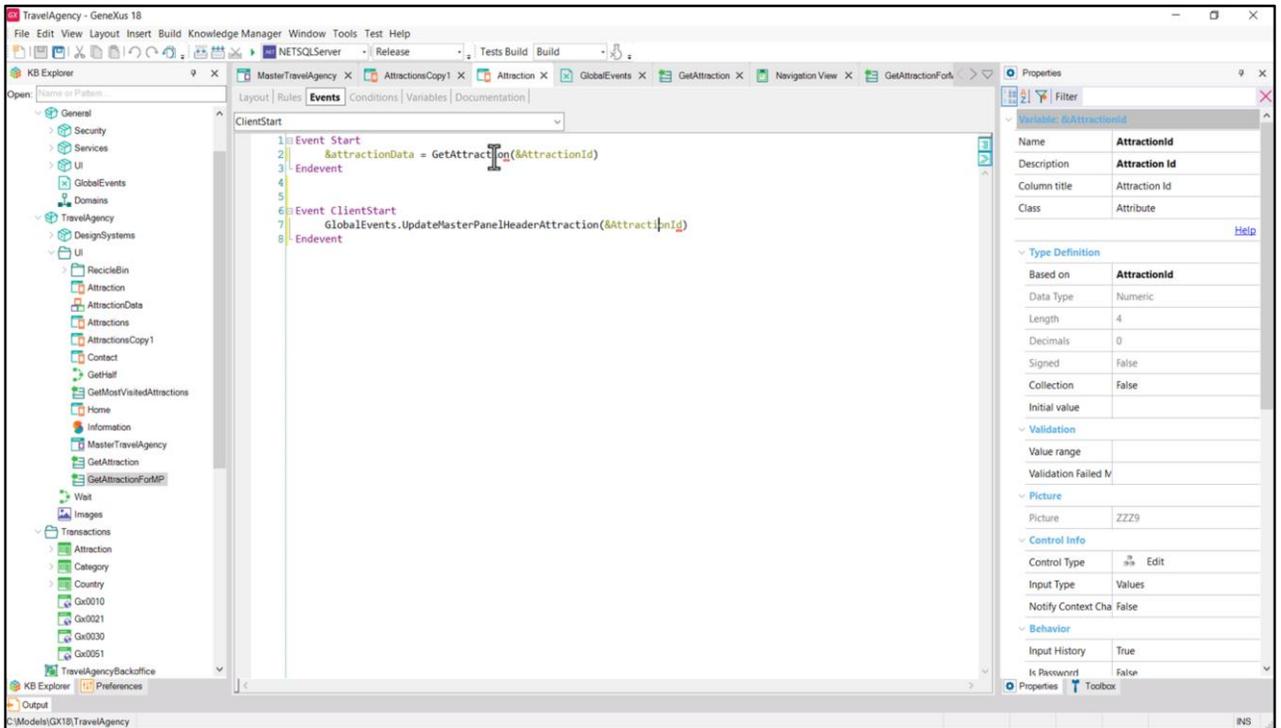
Please note that a bug has just been fixed that caused a DataProvider that loaded an image to do it incorrectly. If when you run this you don't see the image running, it is because you have a version with the bug.



In that case, I recommend that you to use a procedure to load the SDT instead of a Data Provider, and there you won't have any problems.

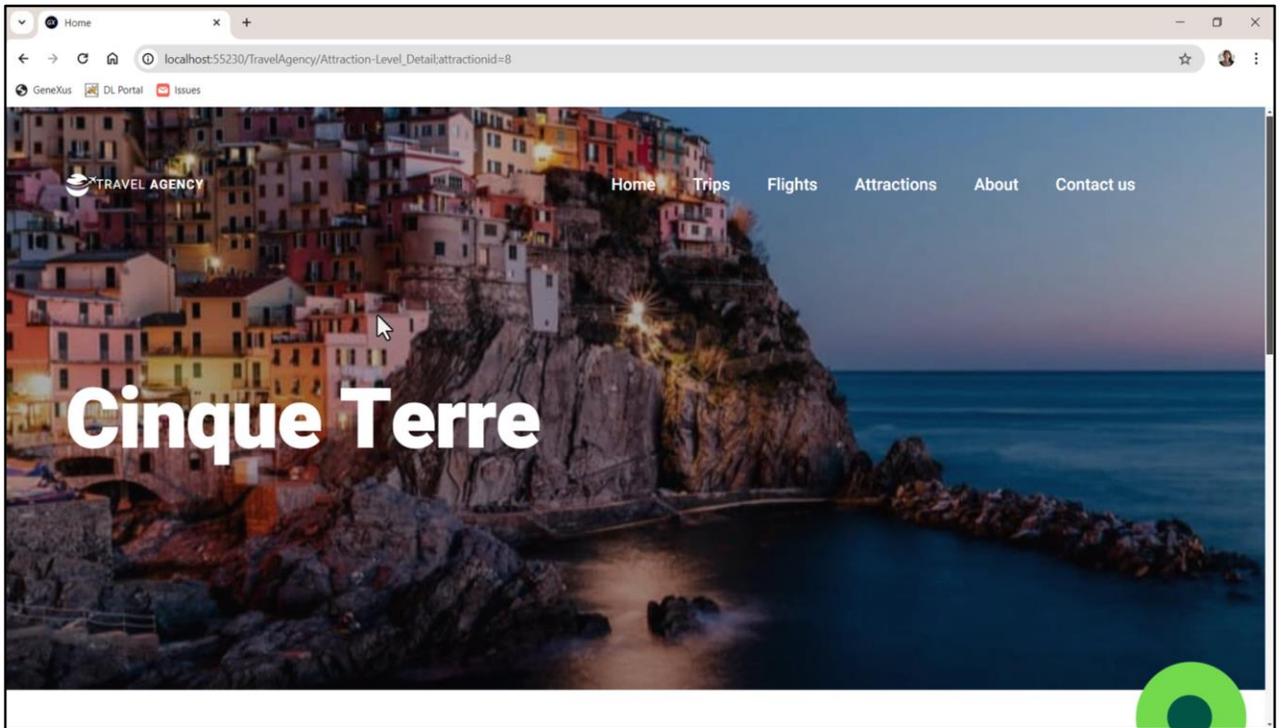


Well, here the image and title of the Master Panel Header will be rendered.

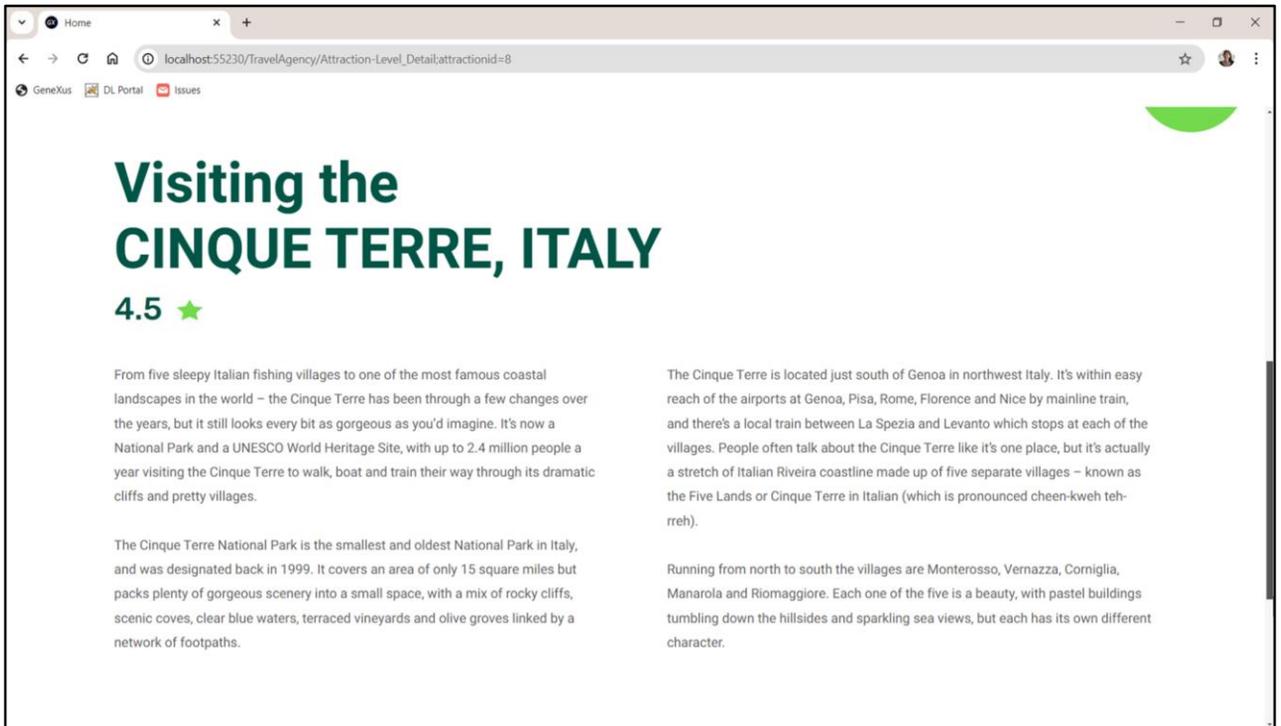


Meanwhile, in Attraction it will have already continued with the following system event, Start. There this other Data Provider is called that brings the necessary information of the attraction with this ID. And that is the one that will be rendered in the layout. Remember that the Refresh and Load events are empty in this panel.

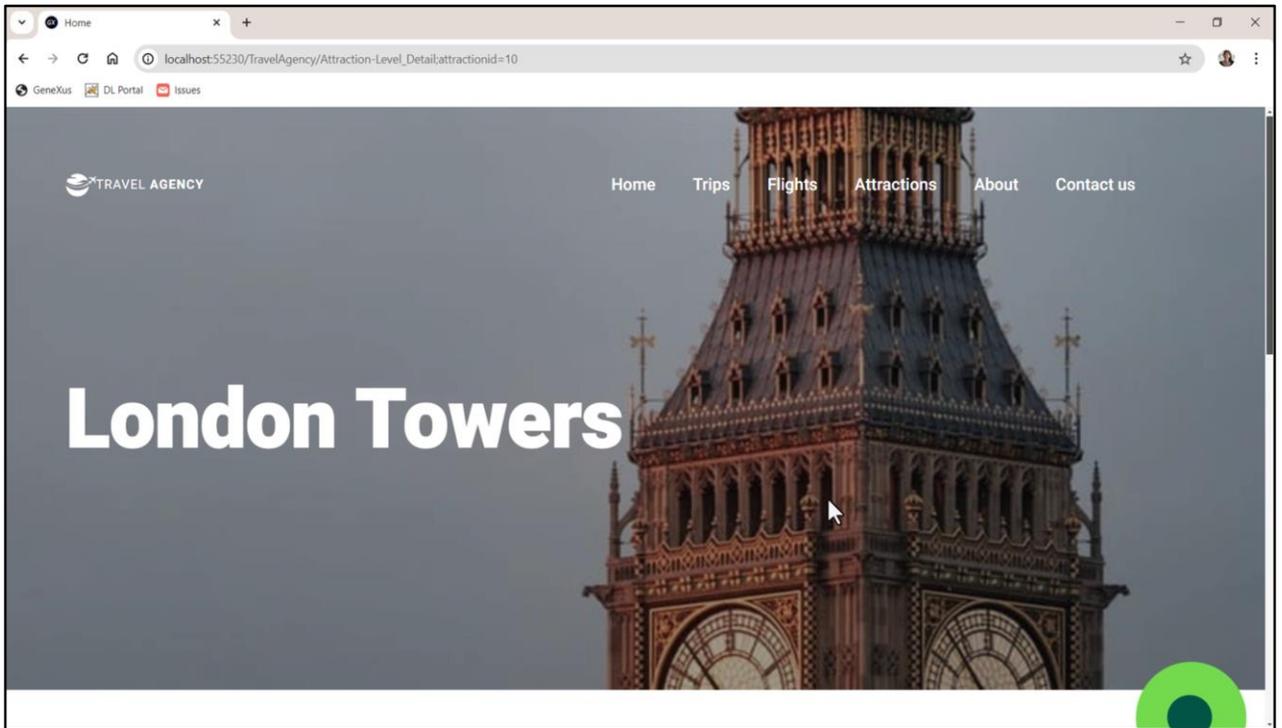
OK, we execute.



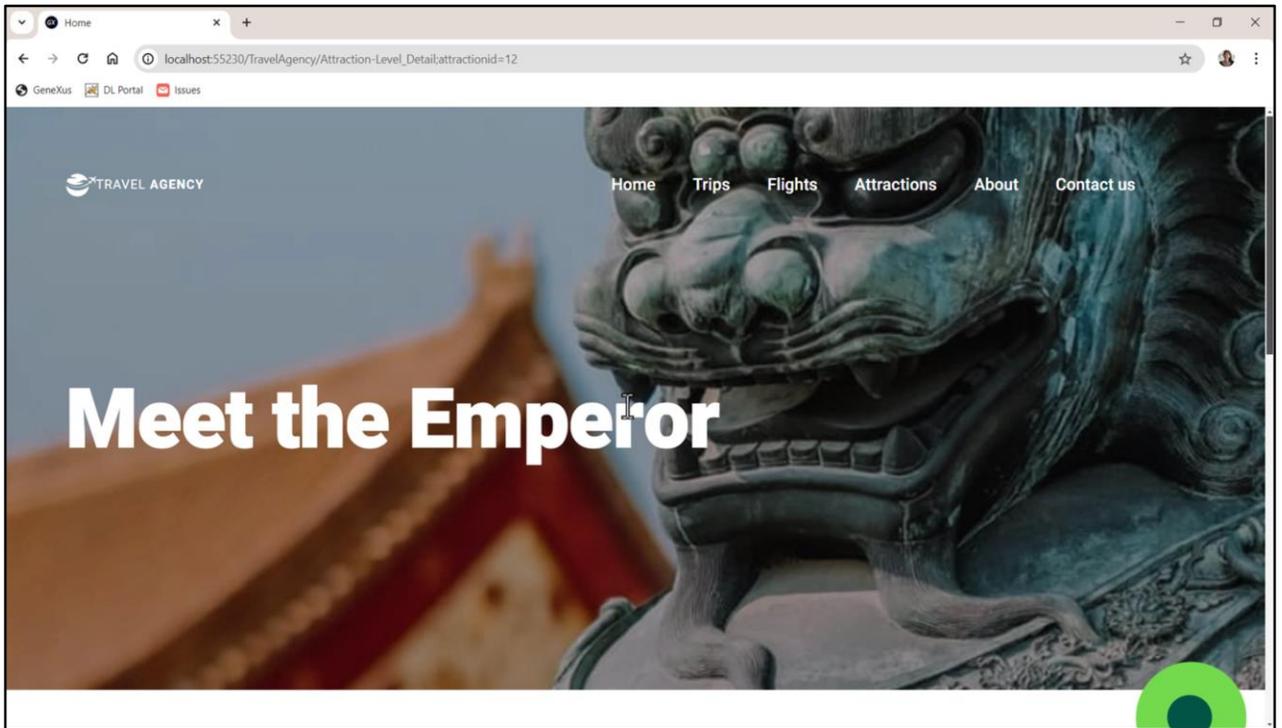
We go to Attractions... we choose the first one in the carousel... and we see how the photo and title are loaded correctly...



...and the specific content of the attraction page.

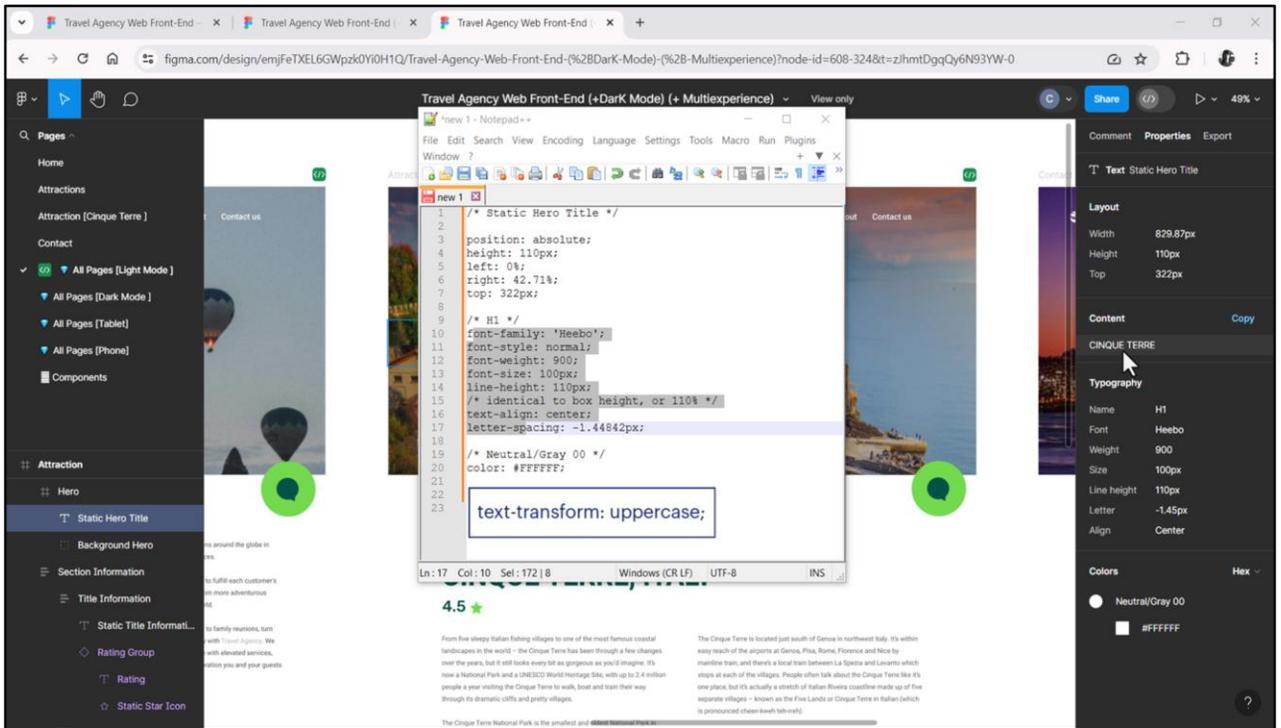


If now we are going to choose another one... also...

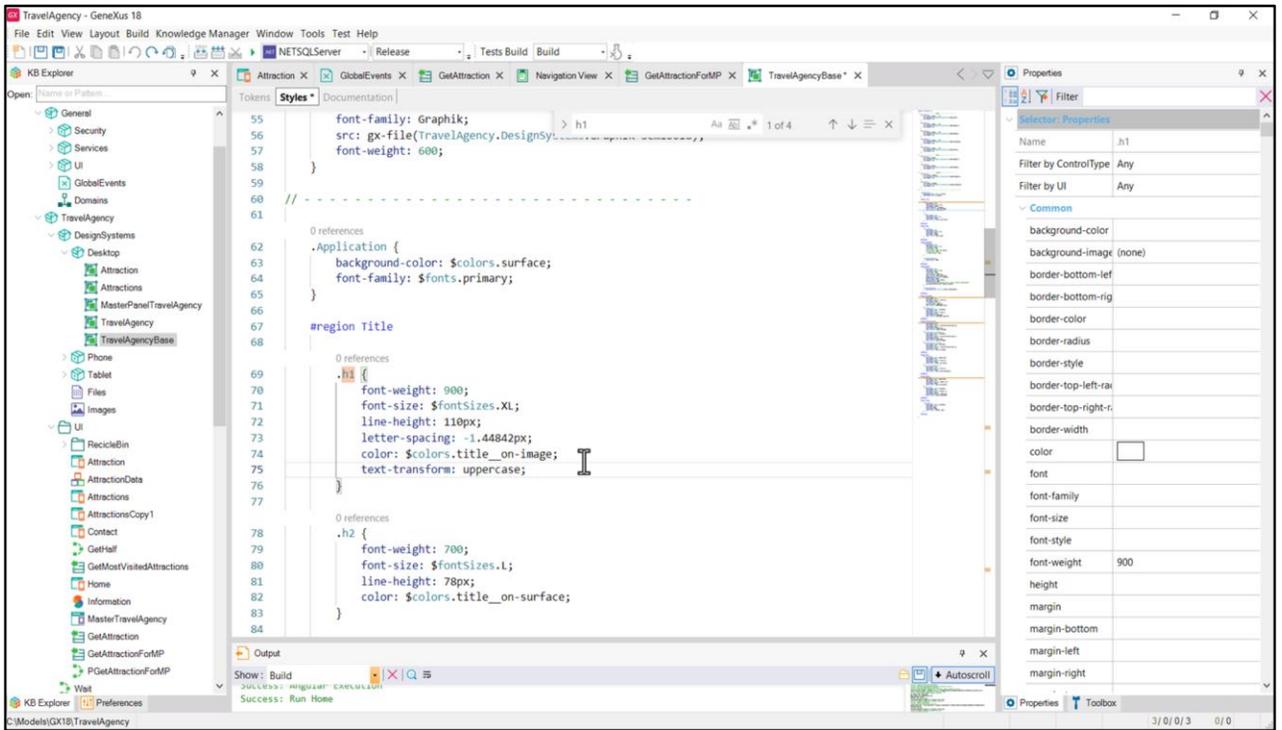


We choose another one... and also...

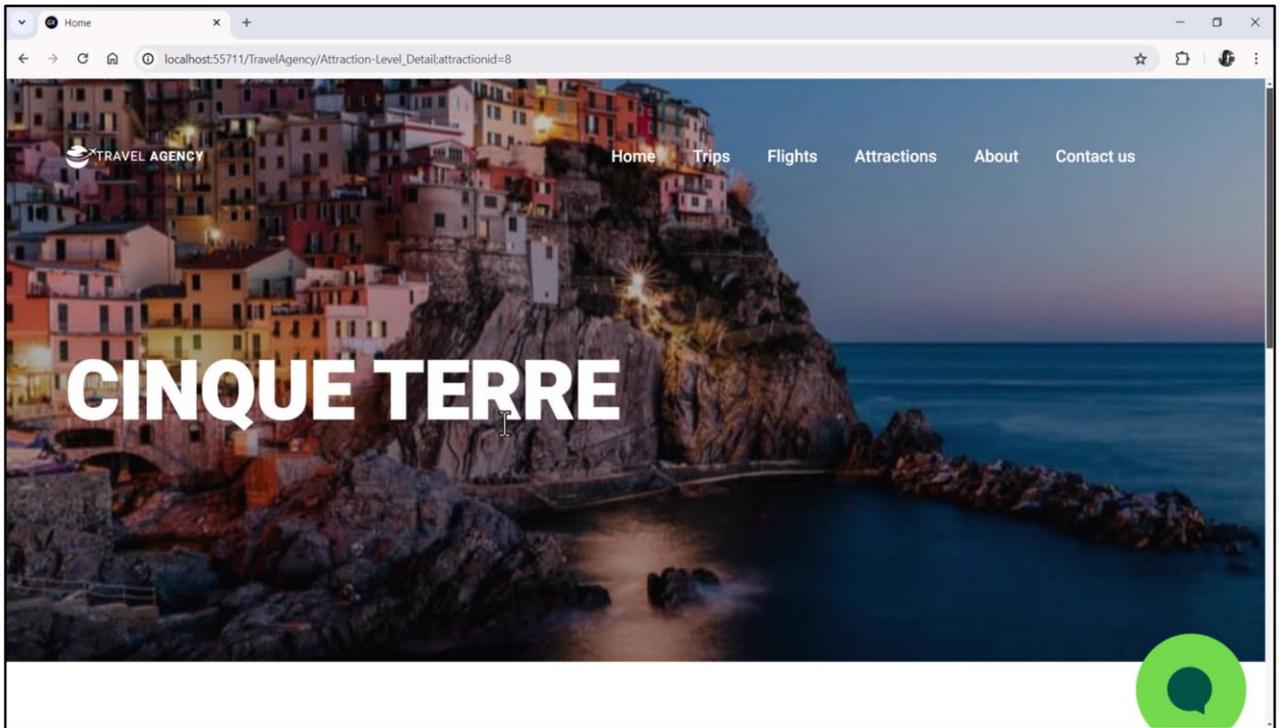
This title should always be shown in uppercase, regardless of how it is entered in the variable.



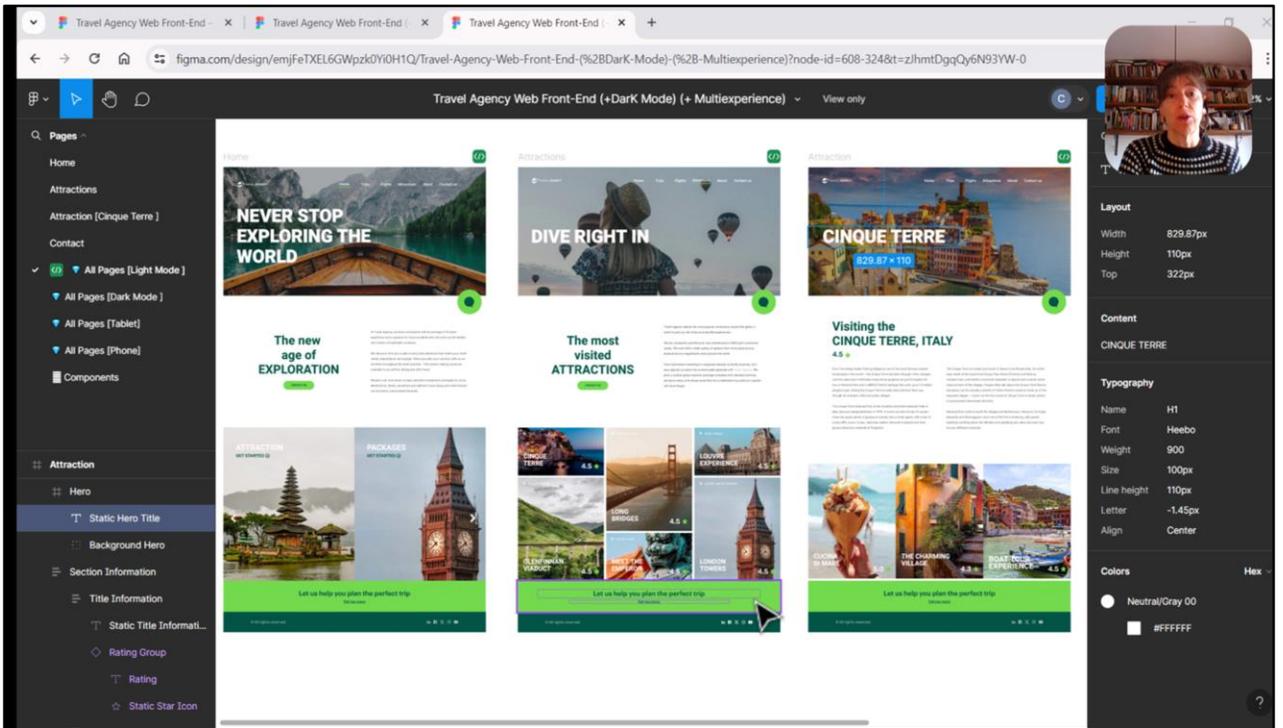
Clearly Chechu didn't notice this when she created the typography style... because she entered the texts in uppercase... so she didn't add the property text-transform: uppercase.



Let's add it to the h1 class.



Well, now it's OK.



With this I close the implementation of the Master Panel.

It remains to be done, and I leave it as a task for you, to implement the footer. All this part below, which is very simple with all the knowledge we have so far.

What we are going to do in the following video is the implementation, in detail, of this carousel. It is a carousel that takes, as we saw, the information from the database, so it is the most complex of the three. That's why I'm going to focus on this one. The others are going to be quite simple after understanding how to implement and design this carousel.

OK, see you in the next video.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com