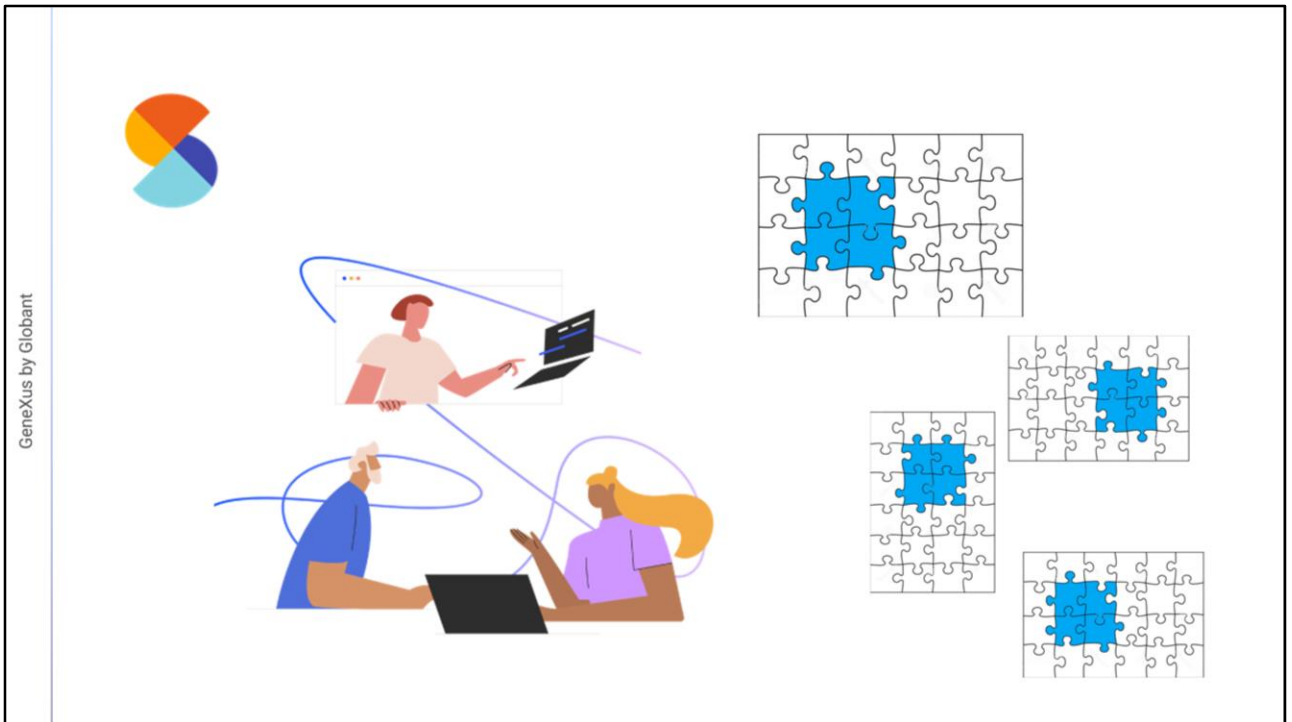


Stencil object



Vanesa Fernández



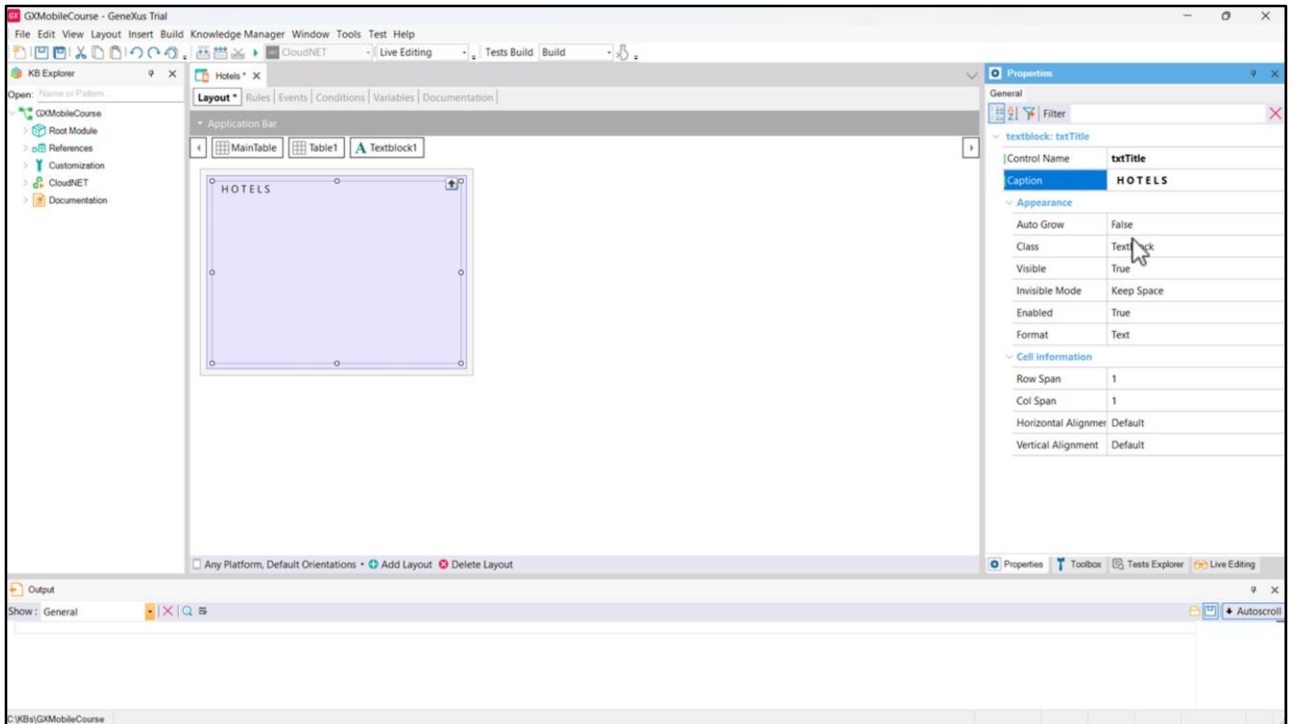
Stencils are part of the work of designers and developers to allow them to share designs comprising controls of the UI (with no associated code) and reuse them “as a unit” in different objects.

It is possible to define several Stencils with the same controls, though distributed or presented in different ways. This enables the possibility to have the same design in different objects, but with variants. They help in defining standards and adding consistency by simplifying the implementation of Design Systems.

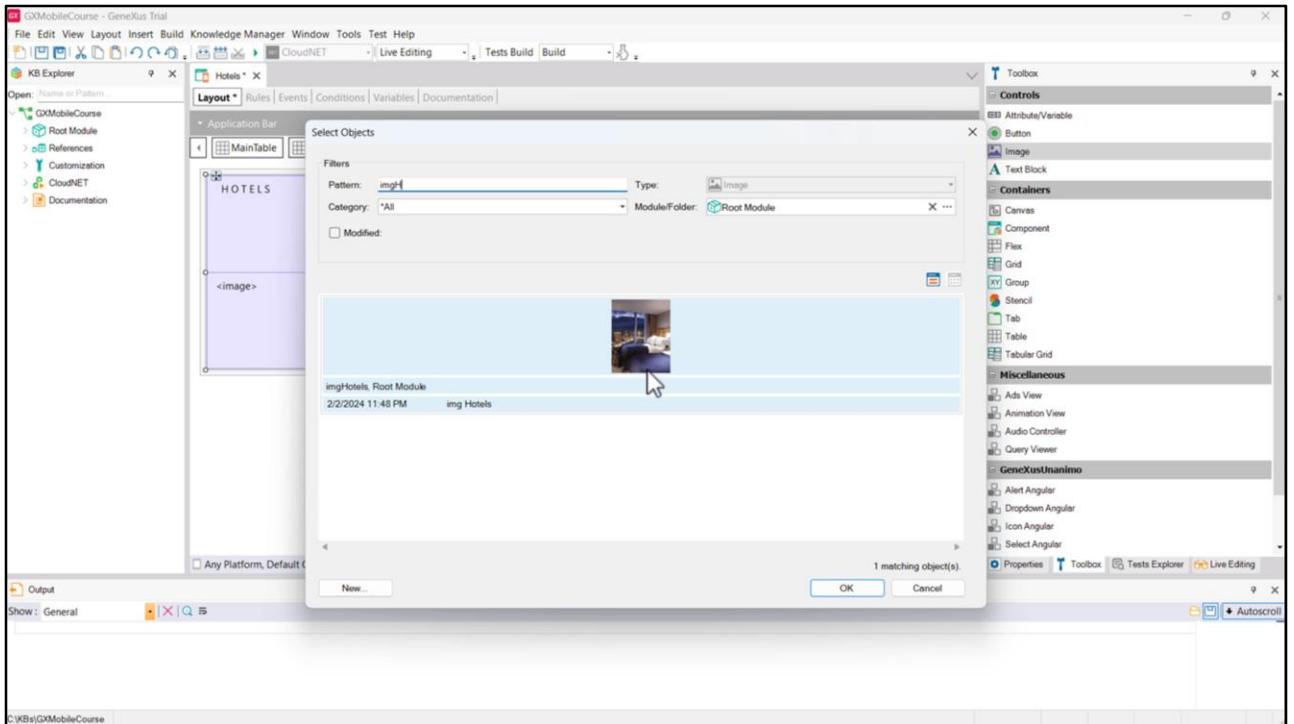
Stencils may be of aid in various scenarios: one of them is the reuse of the visual design of a table control in several panels by wrapping it as a Stencil.



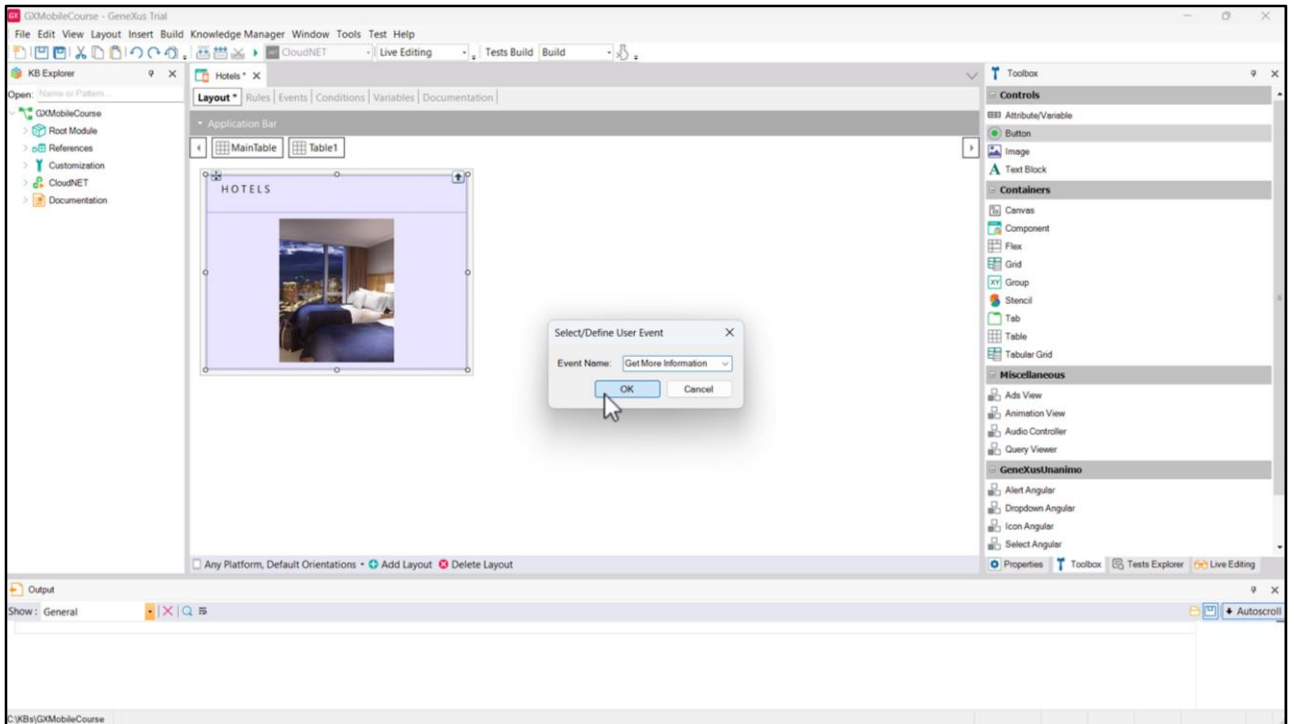
Let's suppose that the travel agency requests 3 screens: one as the entry for Hotel information, another one for Attractions, and the third one for Packages.



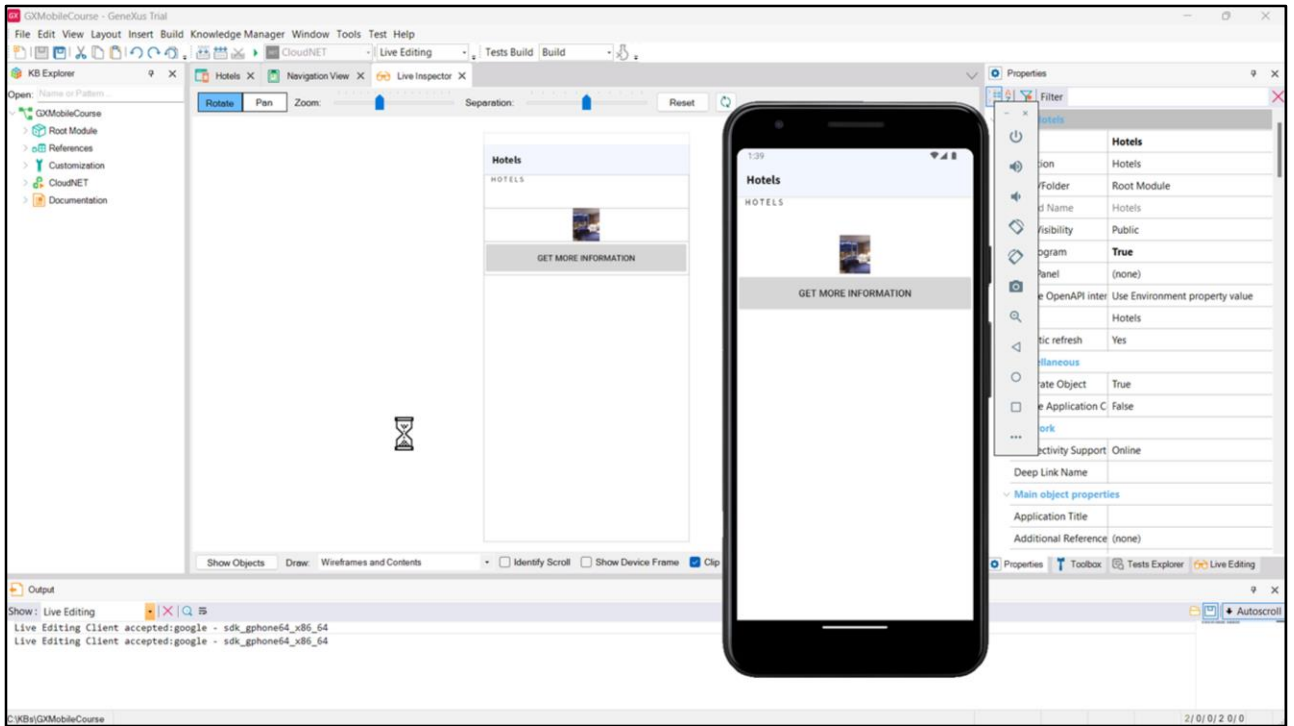
Let's start by creating the first Panel and call it Hotels. Then add a table that will be used to distribute the information, and in that table we drag a Textblock called txtTitle, whose label will be *HOTELS*...



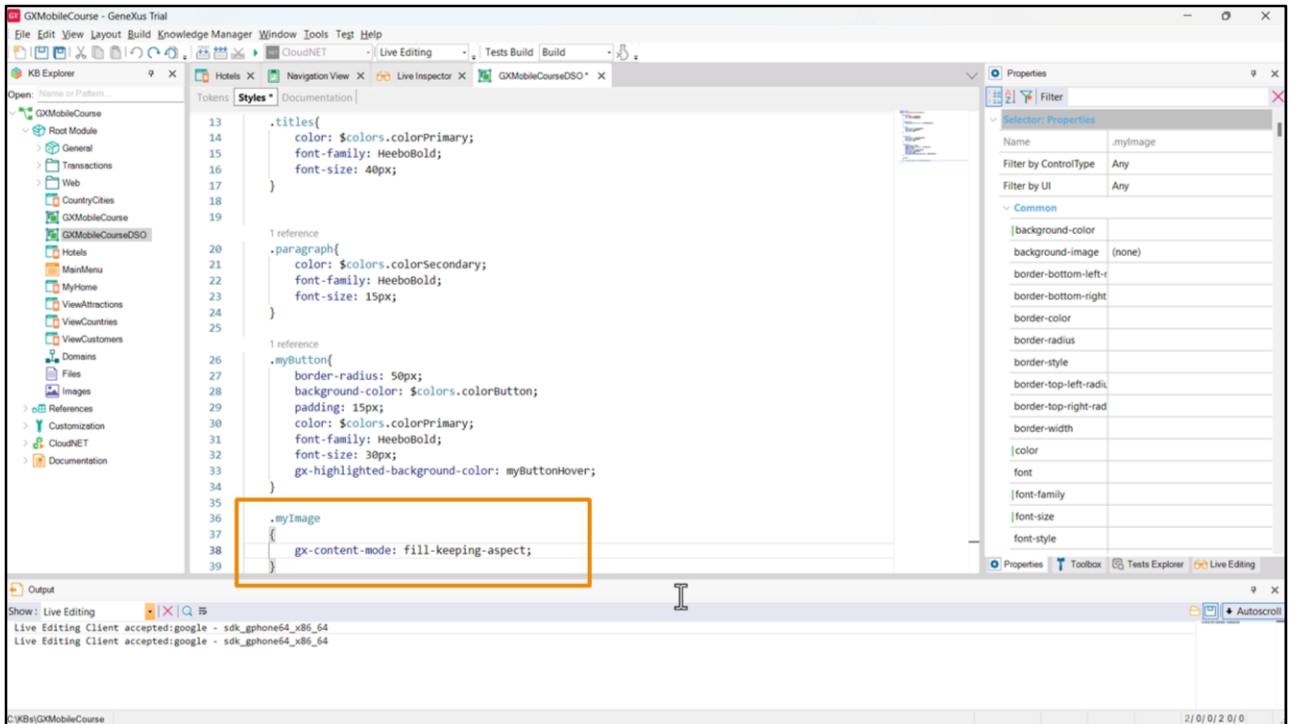
... then an image, where we load the imgHotels file ...



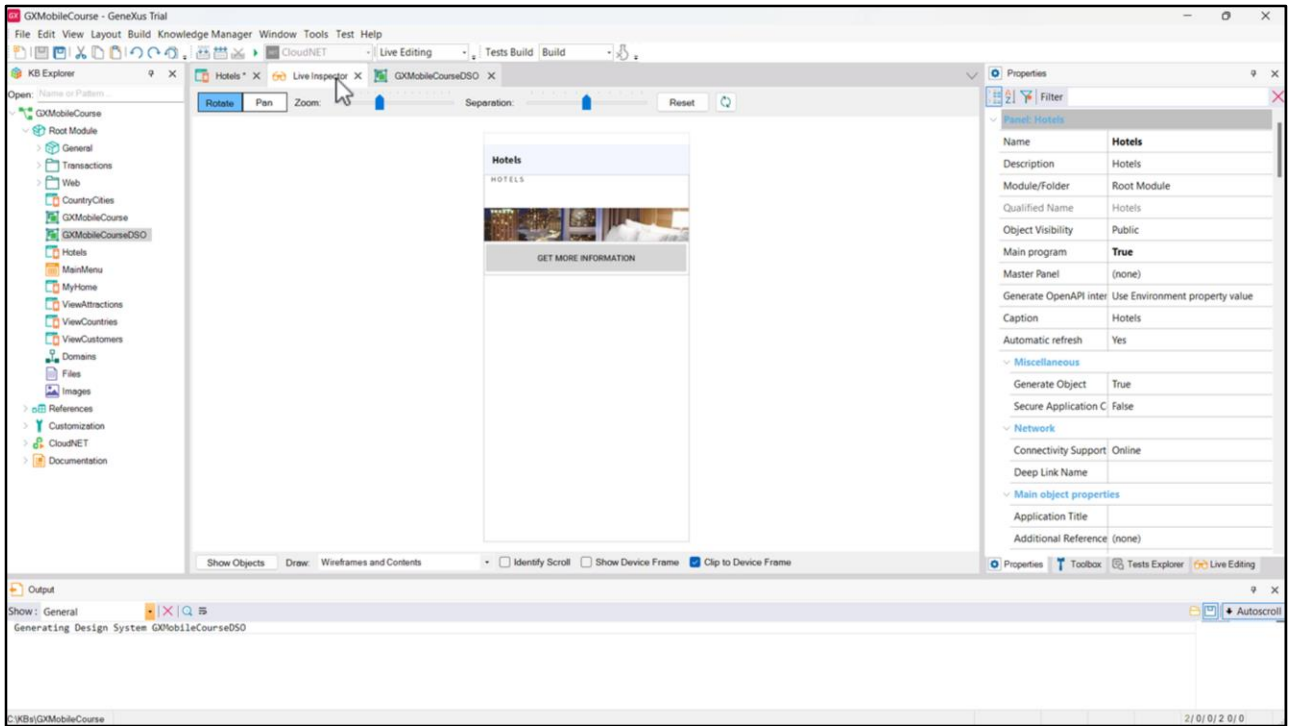
... and a button, whose associated event and label will be *Get More Information*.



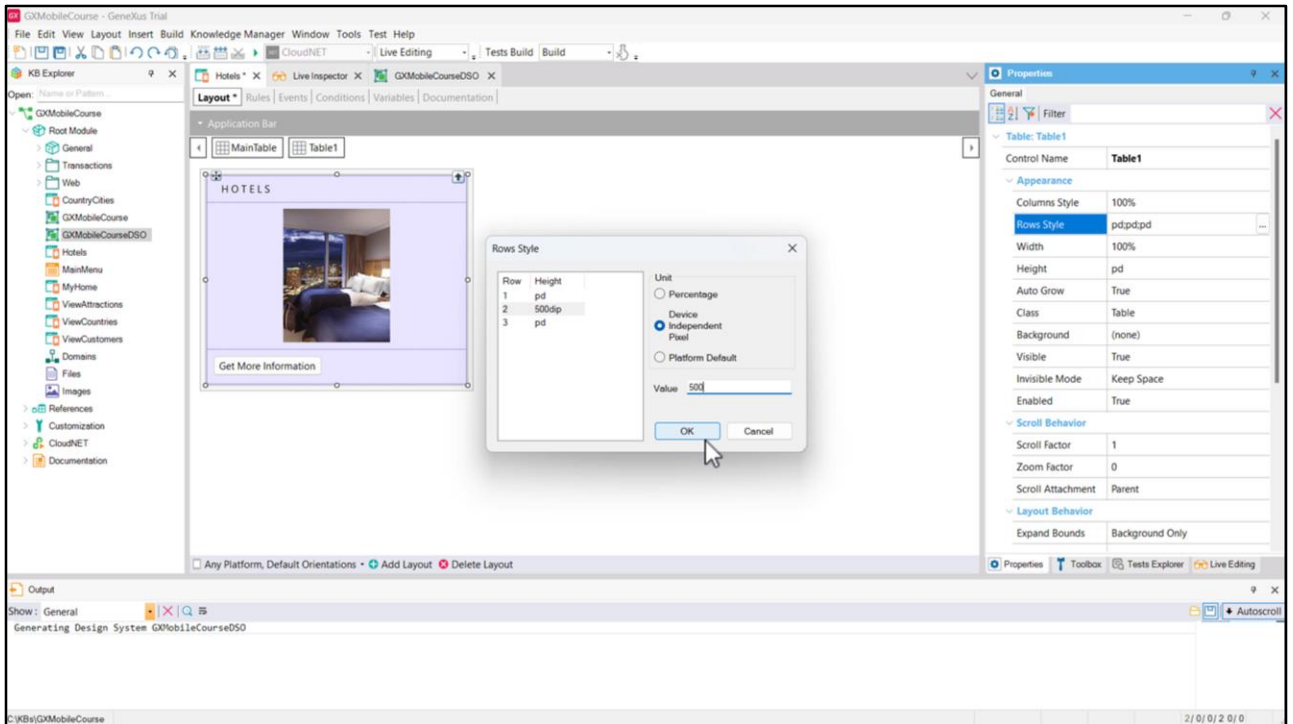
Obviously, there is no design here, so, let's get to it.



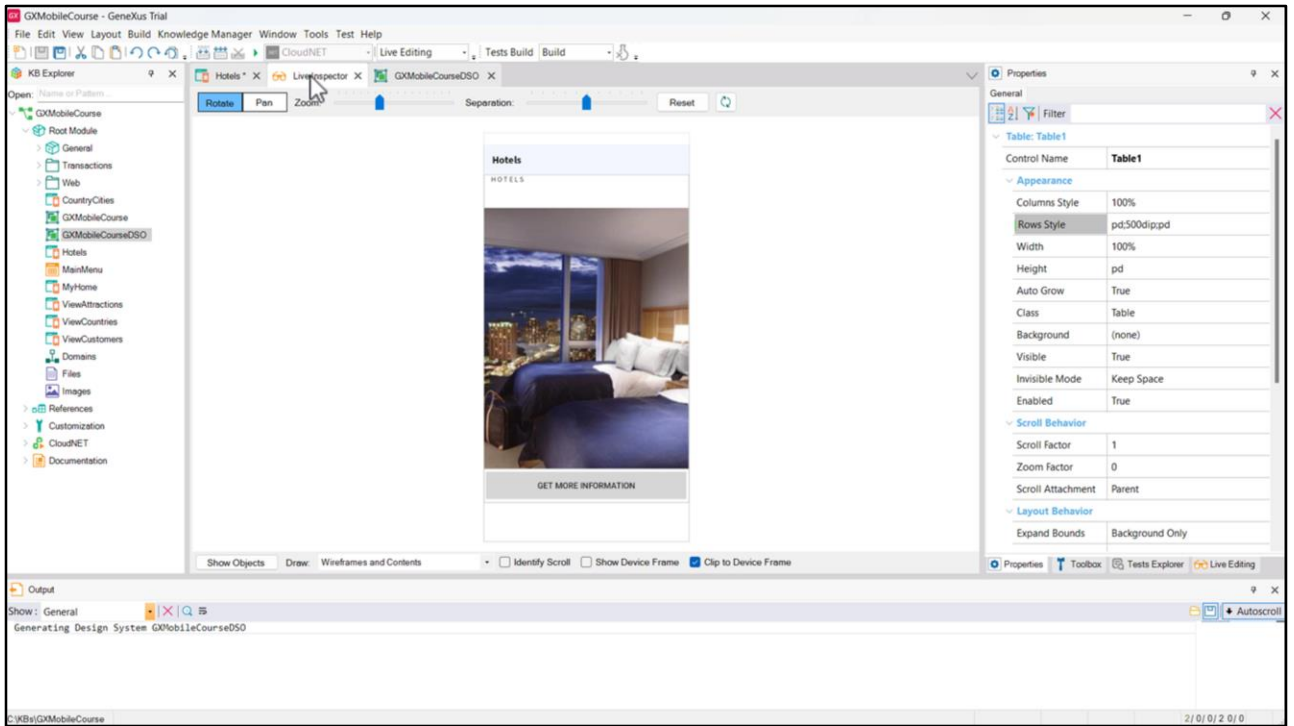
We start by working with the image. Let's create a class in the Design System Object, to enable us to change its viewing mode by scaling it so that it takes up all the space available in its container. To do this, we will use the `gx-property` `gx-content-mode`, and the value `fill-keeping-aspect`;



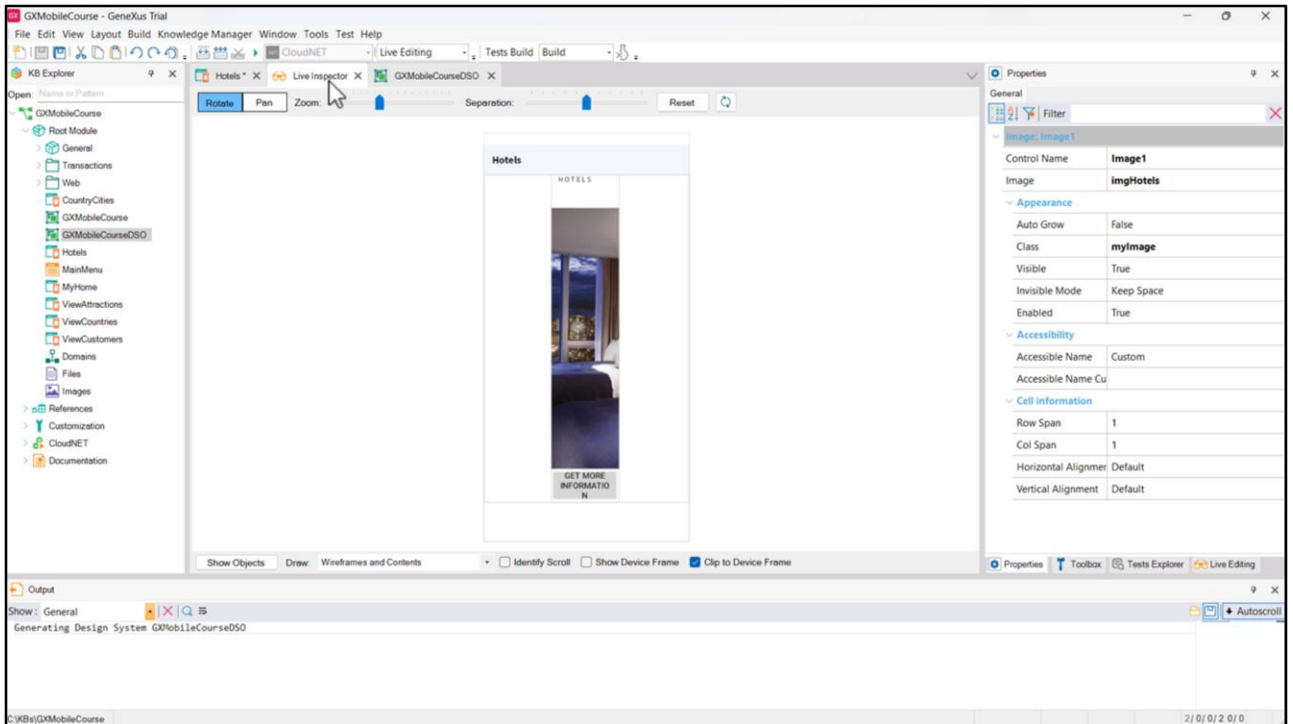
The image has been scaled, but not in its entirety, because the width of the row that contains it is not enough to show it in full.



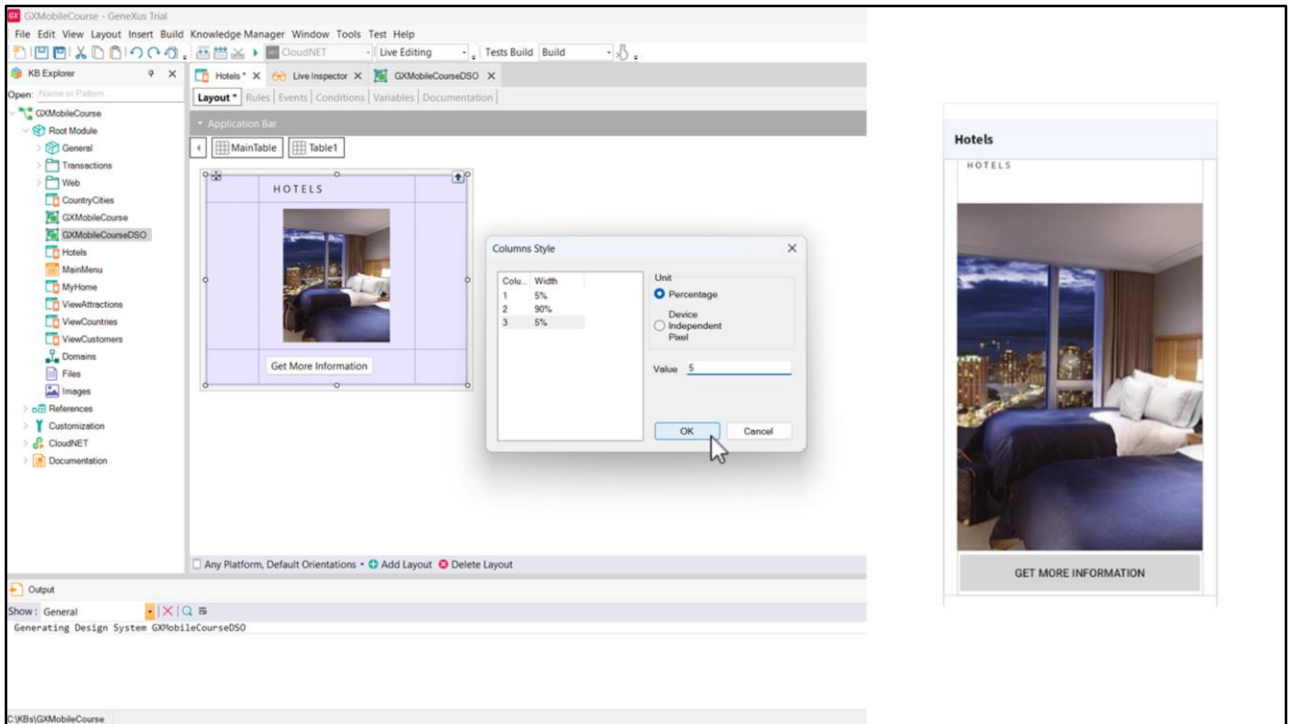
Let's change the value of the Rows Style property for the row's second table to 500 dips.



And there it is.

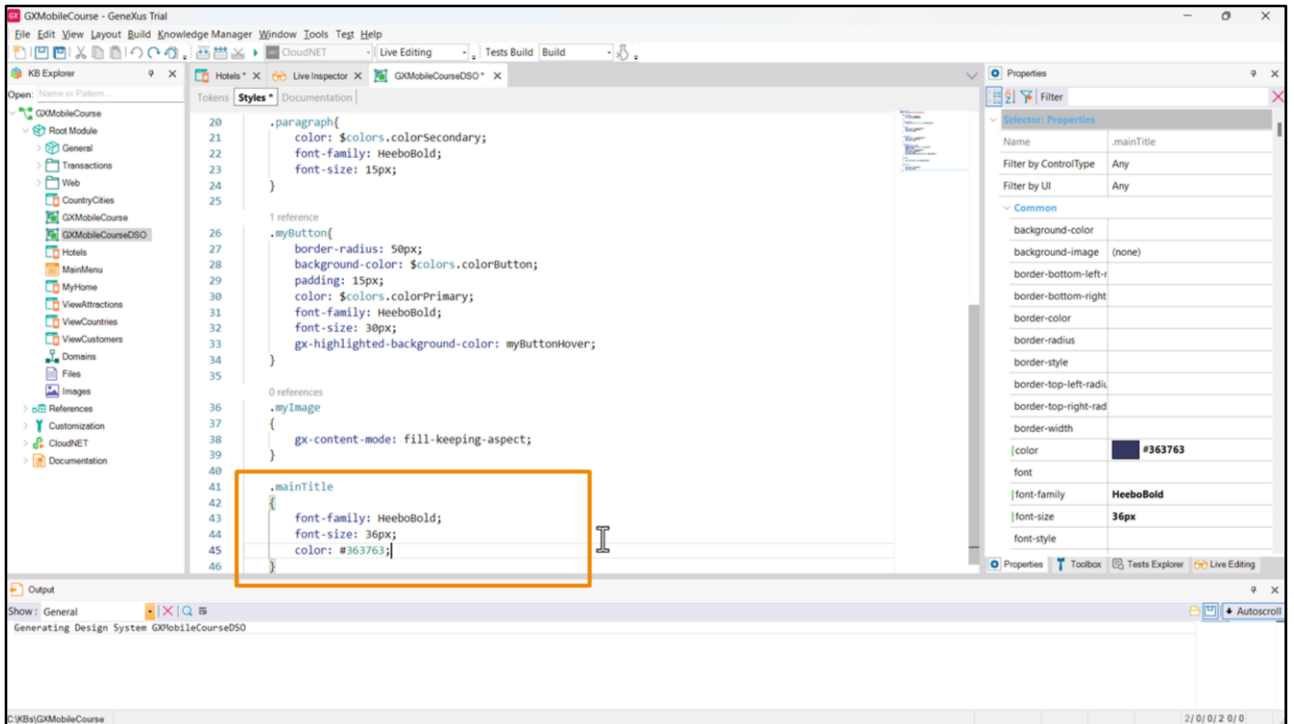


We add spaces on the left and on the right of the content in order to present the information more properly. To do that, we go to the table and add the two columns: automatically, the column is divided into 3 equal columns. However, this is not what we need, because we only wanted a minimum spacing to separate the content from the Panel's borders, with content column occupying the larger space.

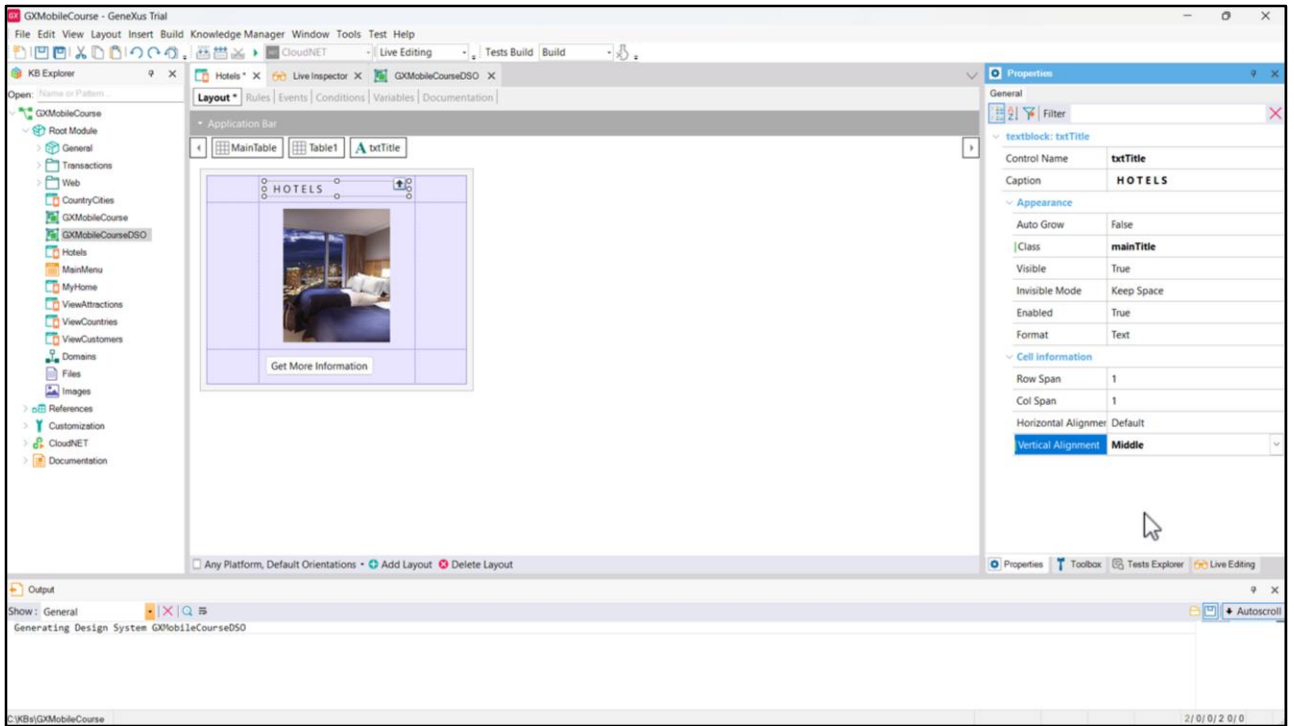


So we configure the table's Columns Style property for the first and third columns to occupy the remaining 5% of the total space, leaving the column in the middle, with the content, to occupy 90%.

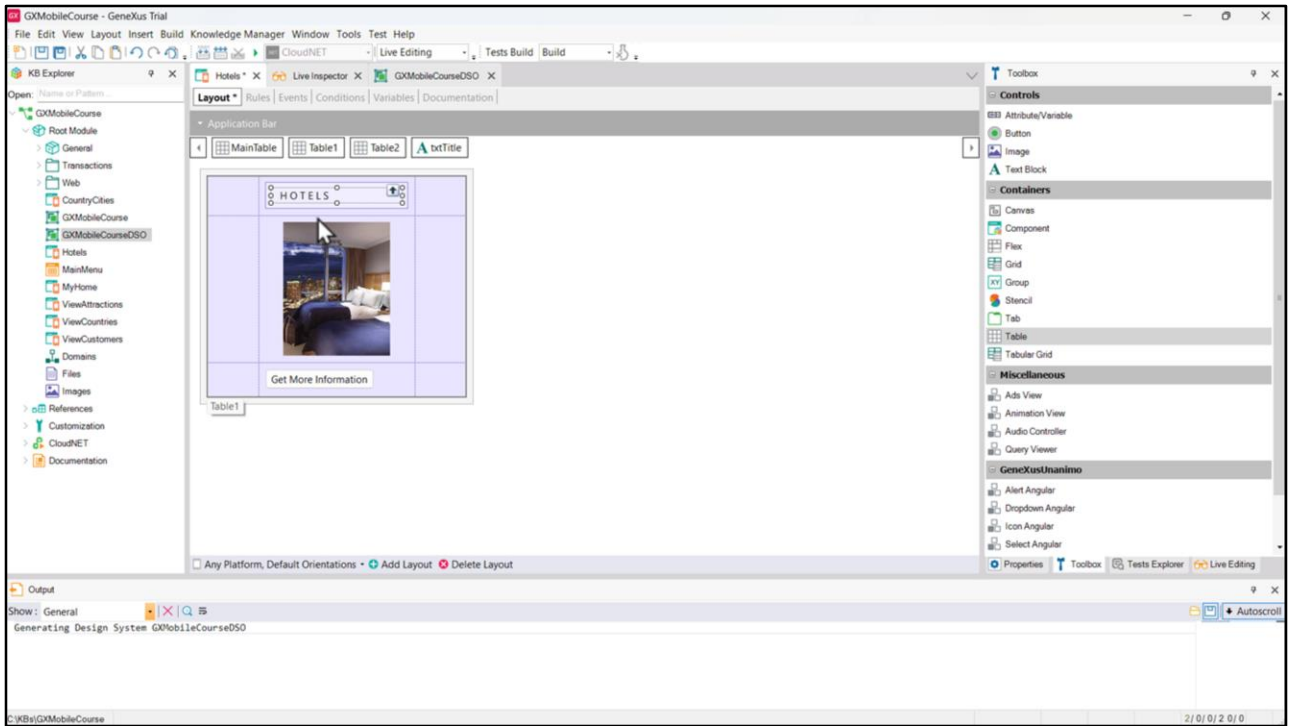
We can see that the Panel is now getting its shape.



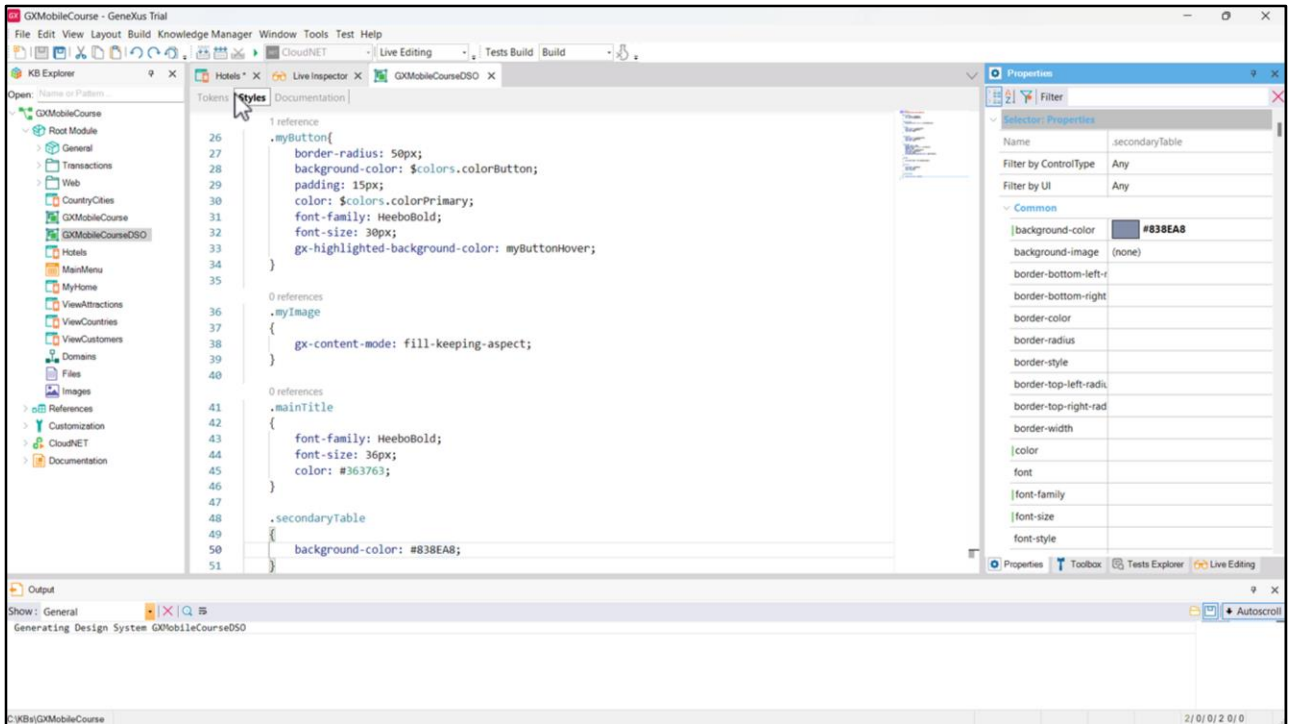
Let's also create a class so that the title has a format: we call it *mainTitle*, and it will have the HeeboBold font, size 36px, and this dark blue color (#363763).



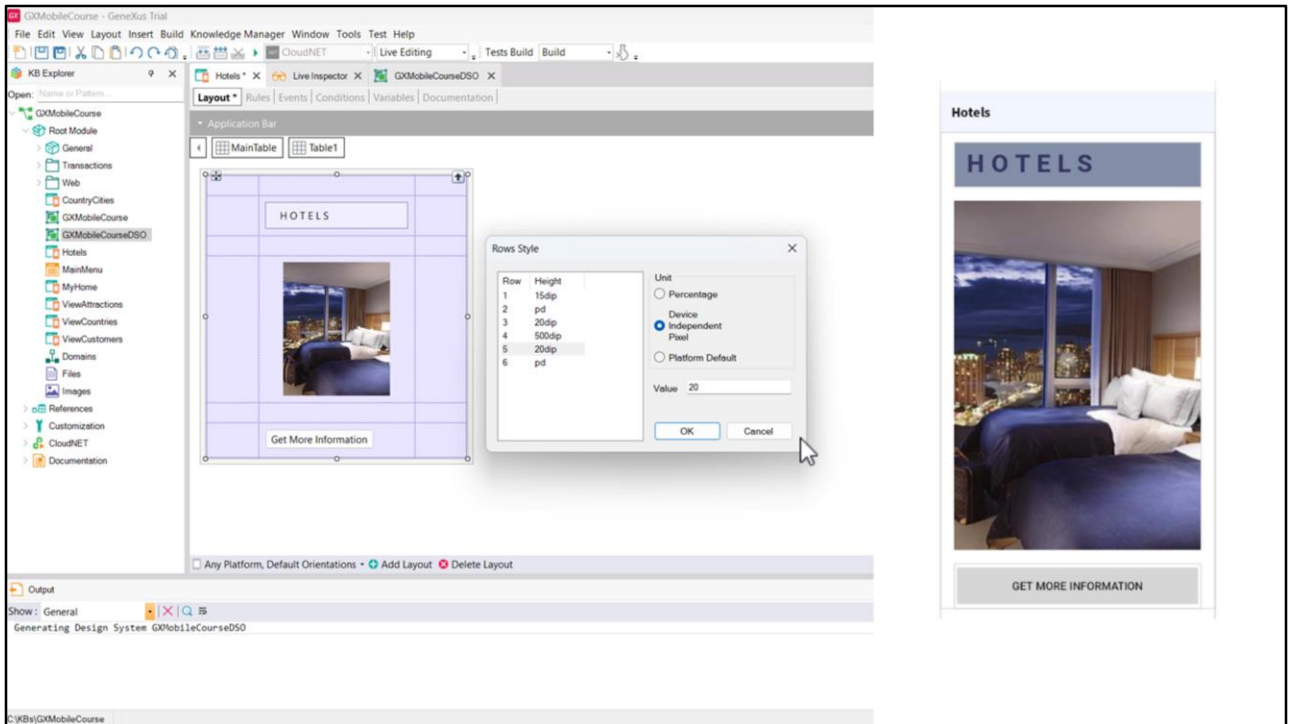
Let's change the textblock's Vertical Alignment property to Middle.



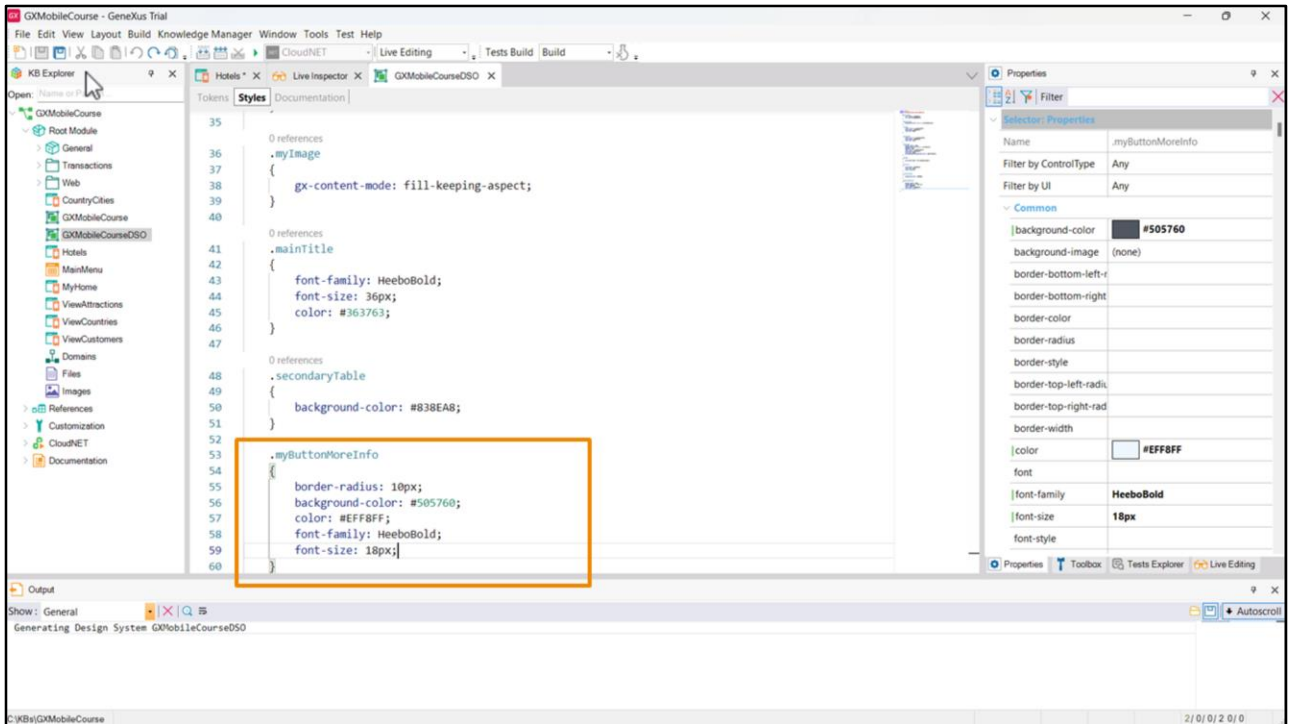
We now add another table where we will be moving our Textblock control ...



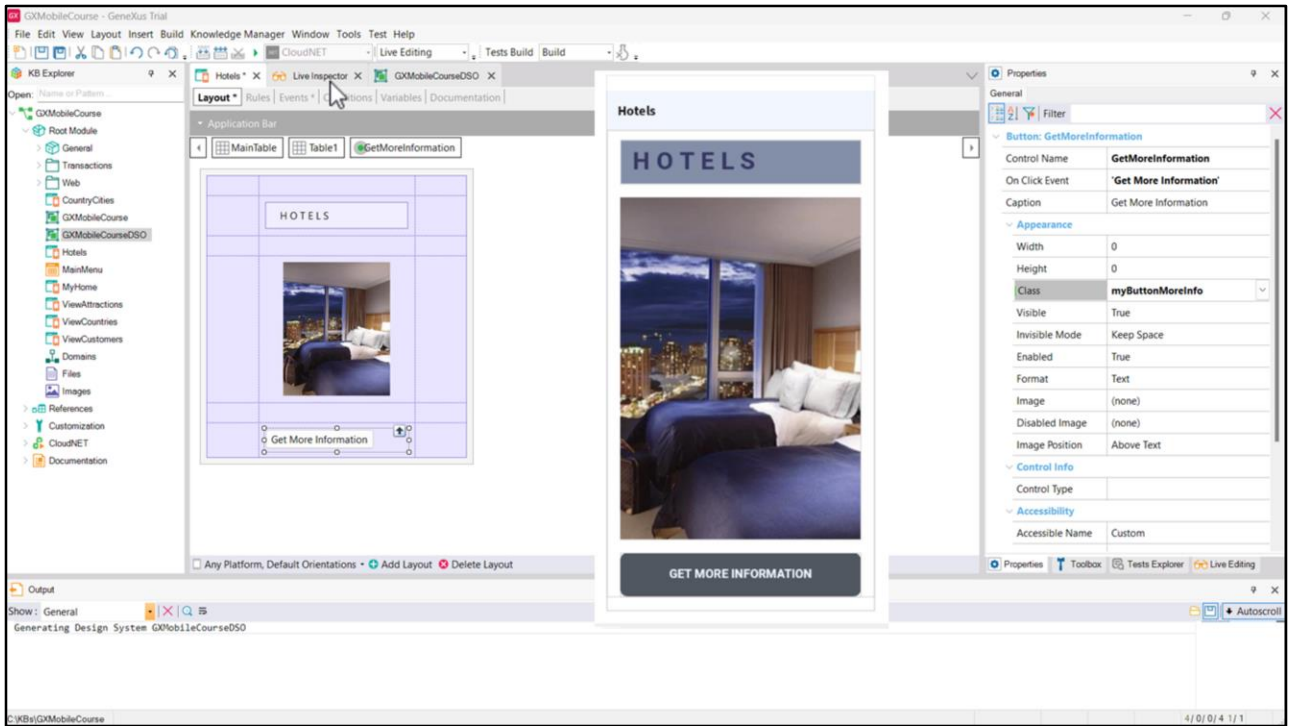
... and creating a class to color the cell's background. Let's call it *secondaryTable* and set the background-color property with the hexadecimal value that corresponds to this strong shade of sky blue (#838EA8).

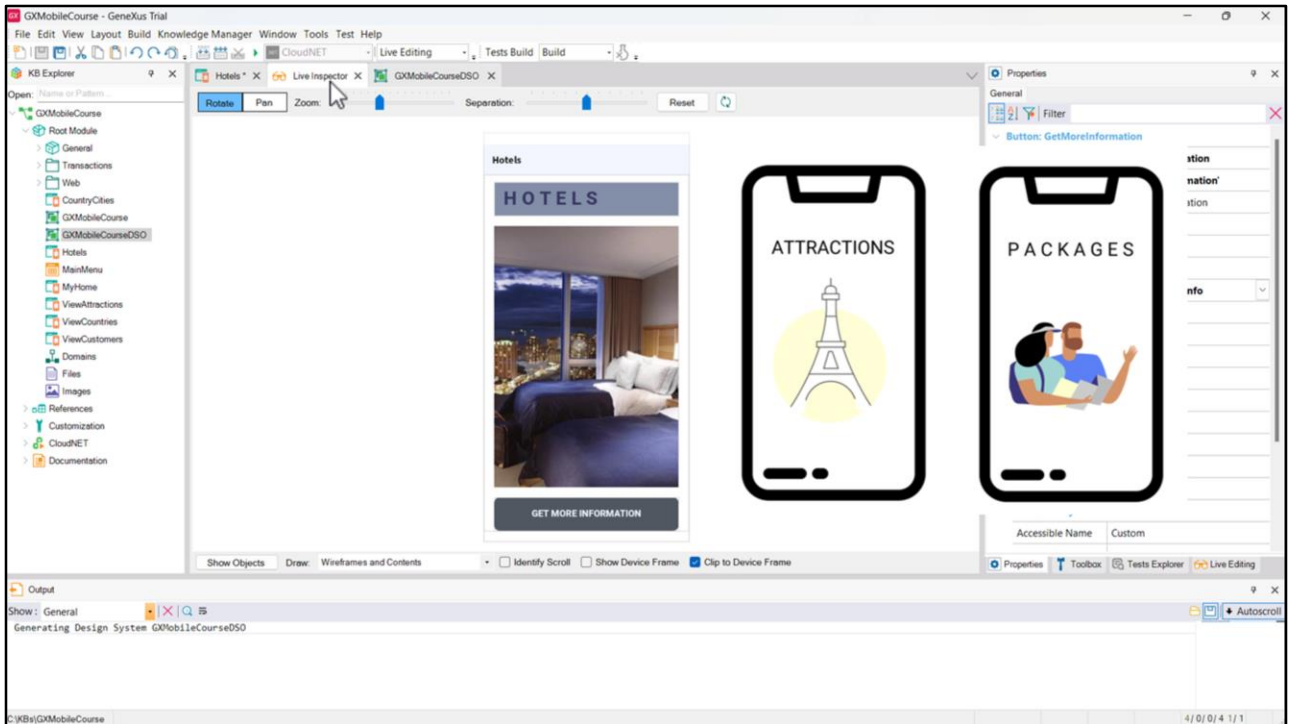


Now let's add 3 more rows in the table to separate our elements: one before the title, another one before the image, and the other one before the button. Now we adapt the Rows Style property so that the first row occupies only 15 dips, while the third row occupies somewhat more -20 dips-, as well as the fifth row with 20 dips.



Let's now work with the button. We first create the class `myButtonMoreInfo` and then add the properties to define its style: `border-radius` with 10px to slightly round its borders, the `background-color` with this dark grey (#505760), and the color for the font (#EFF8FF) with this very light sky blue -almost white-, and `font-family` to set its font as HeeboBold, and to end, the `font-size` so that the font occupies 18px.





Let's suppose that, so far, we are satisfied with the design.

A reminder that the travel agency also requested a screen to view the attractions, and another screen to view the packages.

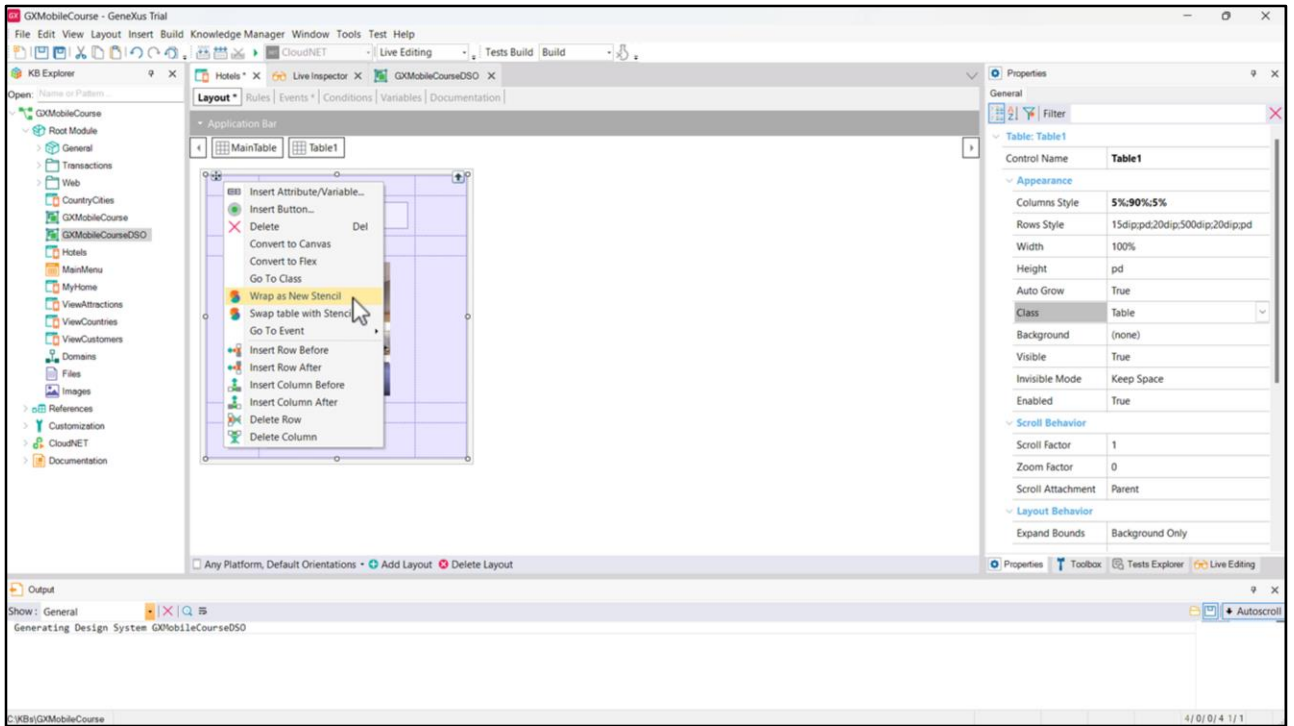
Is it necessary to create each Panel and then repeat all the setups for each of them? That would not make any sense.

We could think of doing a Save as on this Panel to then change whatever is required.

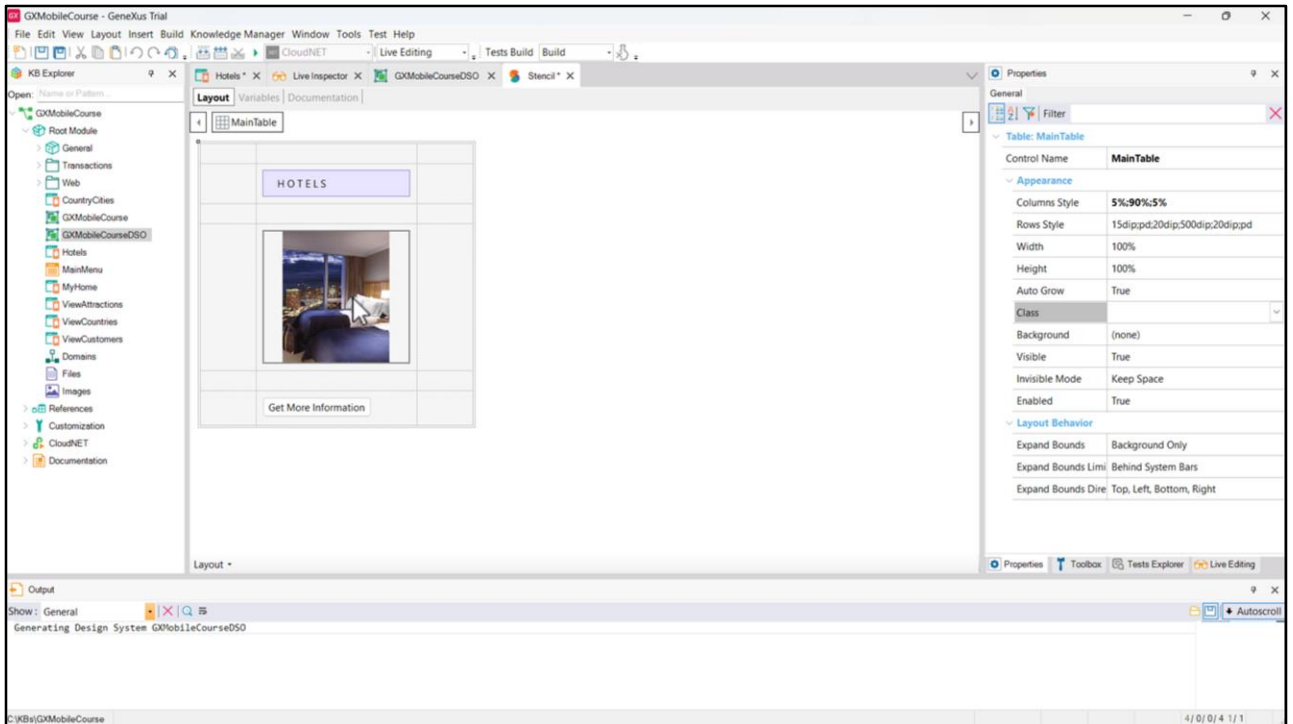
But, what would happen if a change in the design was necessary after they have been implemented? Those changes would have to be done on every Panel.

Using Stencils prevents us from doing all that because we can simply "wrap" the main table that we created as a Stencil that we can later use for each Panel.

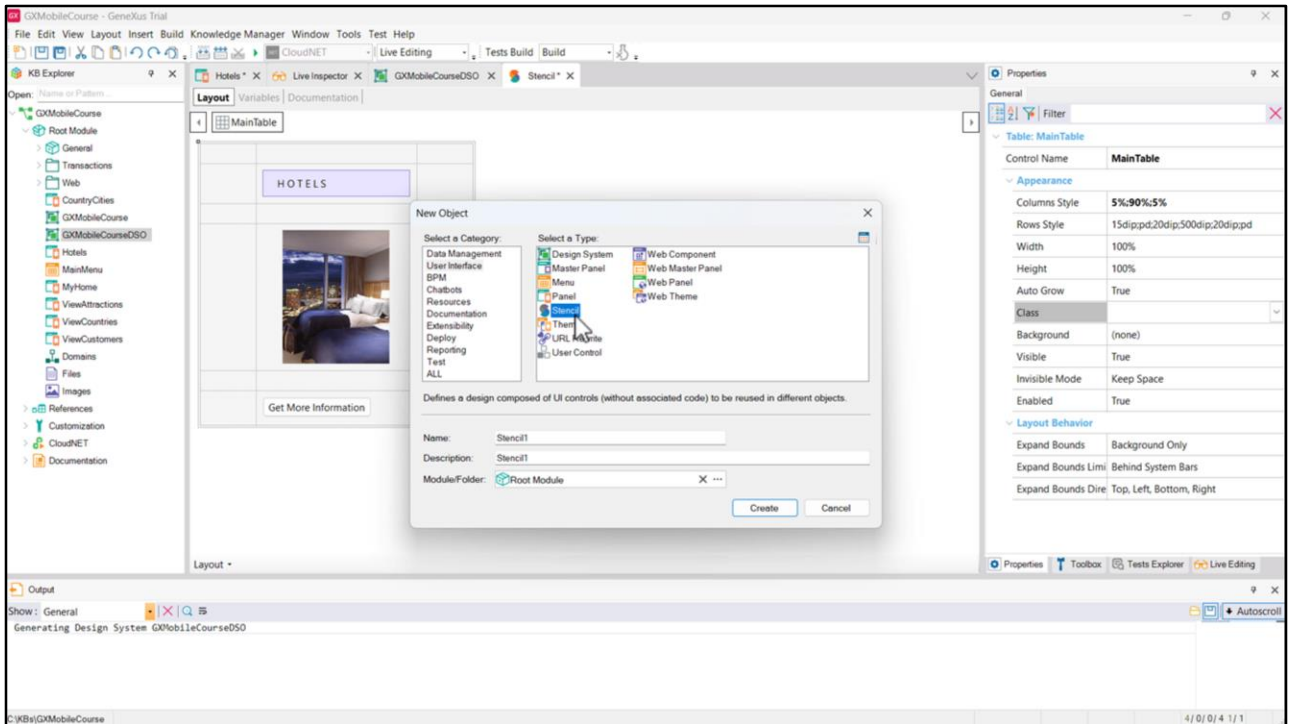
Let's get to that.



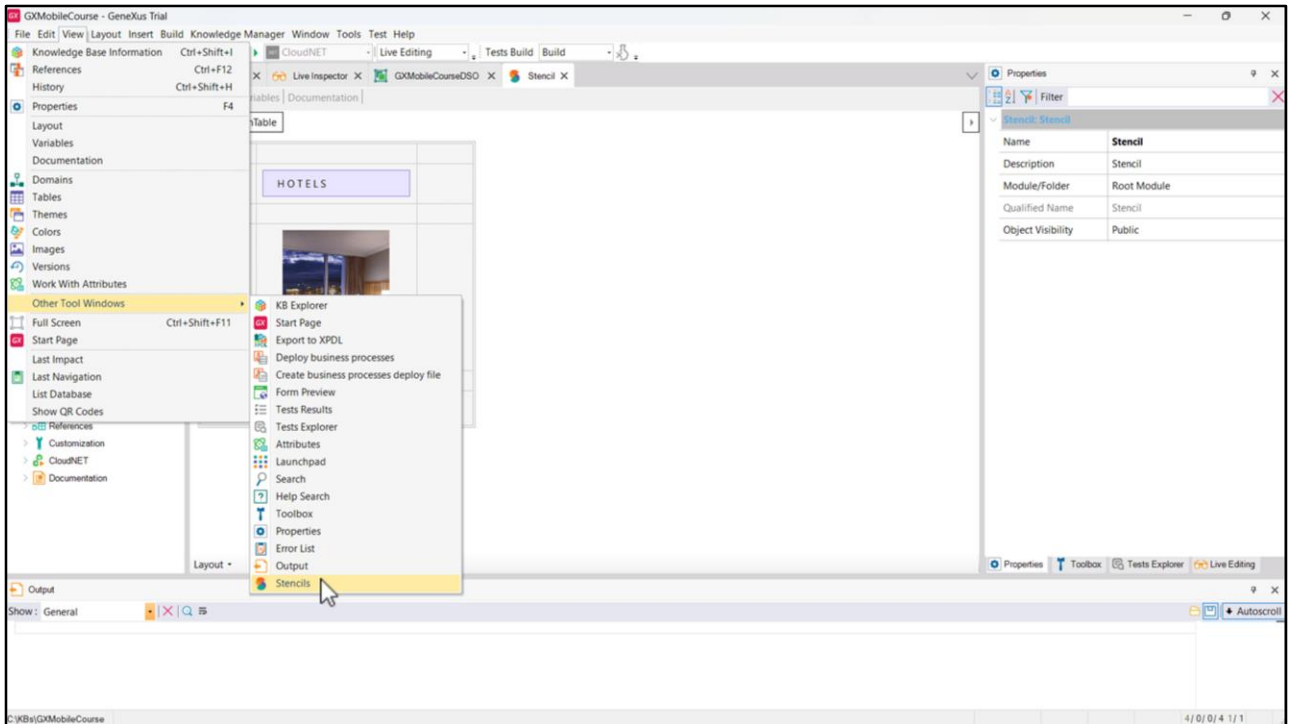
Upon right-clicking on the table, we will see an option called *Wrap as New Stencil*.



A dialog box is opened to enter a name, and when the Create button is pressed, the Stencil will be created as a new object. From here we will be able to do any changes required in the properties. Once saved, those changes will be reflected on the object that contains the Stencil (up to this point, only the Hotels Panel).

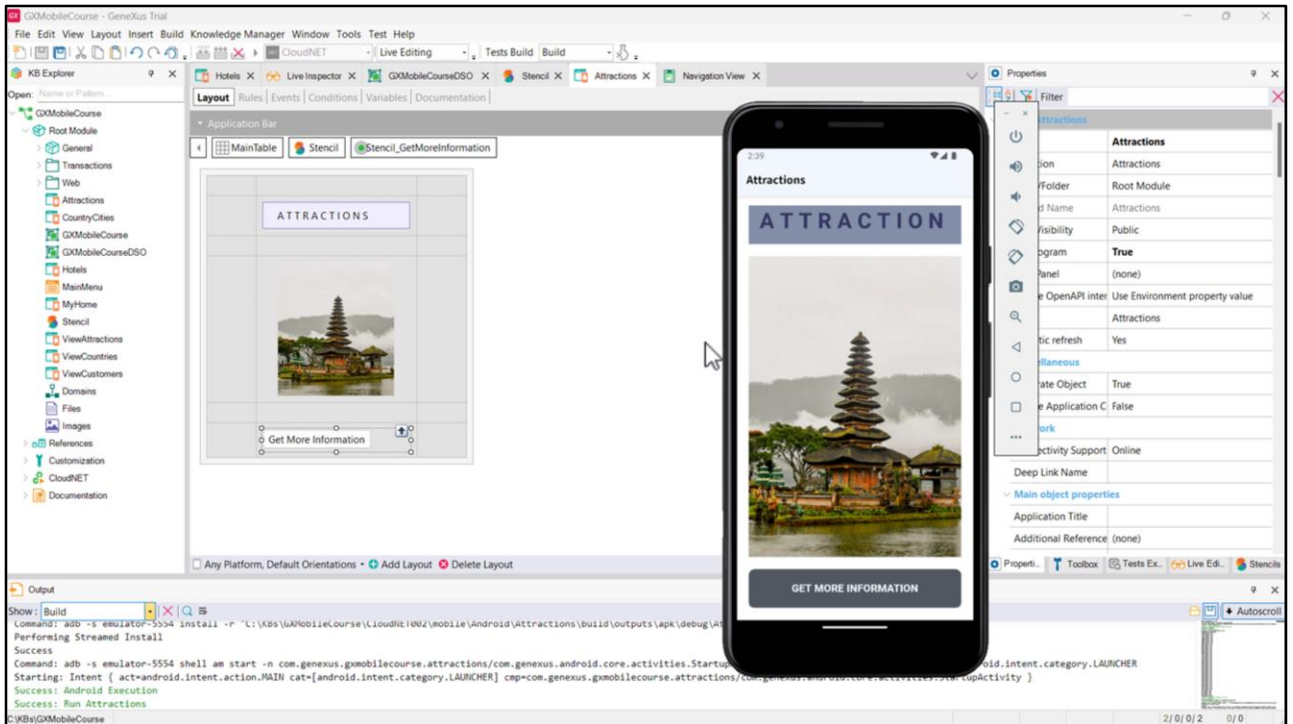


Another possibility to create a Stencil object is to go to the Genexus main menu and directly select: File > New > Object.



After creating it, to include it in an object we must drag it from the KB Explorer or from the Stencils window, which may be accessed from the menu View > Other Tool Windows > Stencils.

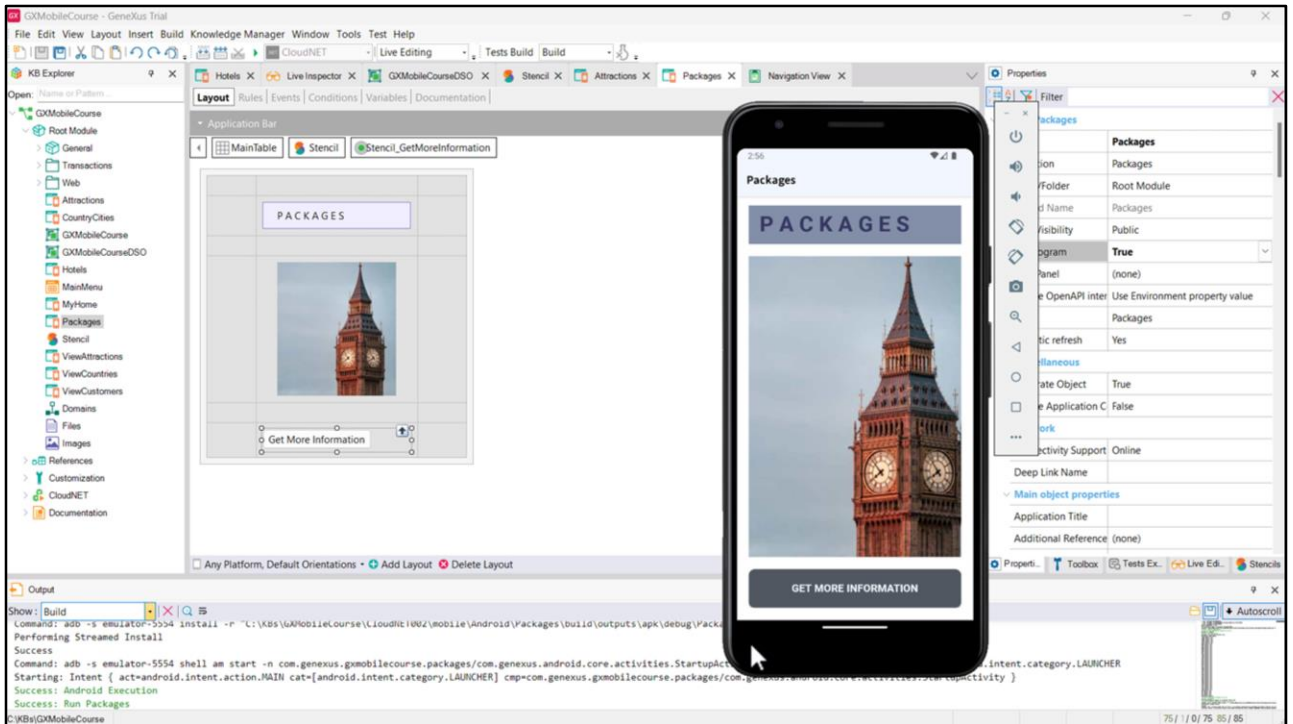
If a Stencil is created from a Panel that contains attributes, when the Stencil is saved, the attributes turn into variables. This is because Stencils have no defined logic or navigation.



Now we indeed create the Attractions Panel using the Stencil by dragging it from the KB Explorer.

When we click on the main table, we will find that we no longer have the setup properties we had before. This is because, now, all properties referred to its design are only accessible from the Stencil object.

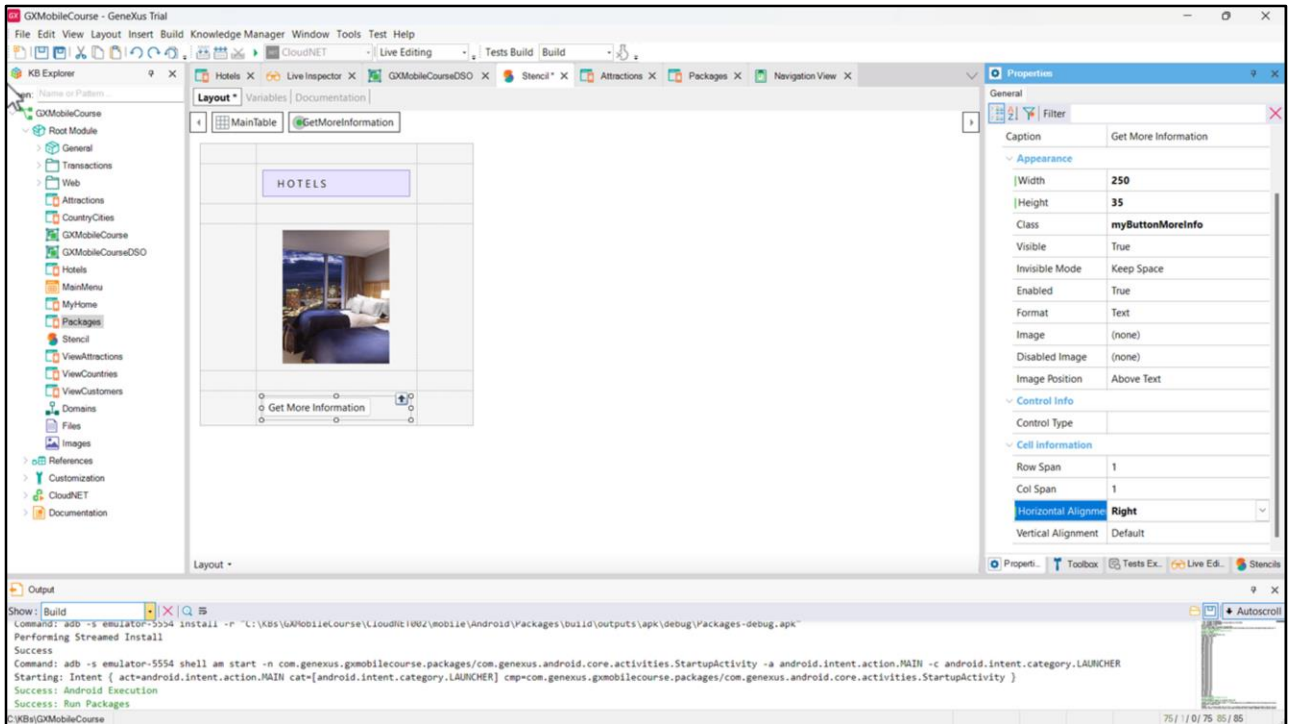
Let's change the title label to *A T T R A C T I O N S*, and replace the image with the one called *imgAttractions*. Since the button must always have an associated event, let's double click on it so that it is created.



Now let's create the Packages Panel and add the Stencil from the Stencils window using the *INSERT* option. The *OPEN* option opens the Stencil object implied and the *SWAP* option is used to replace a table with the same set of controls as the Stencil, with the Stencil.

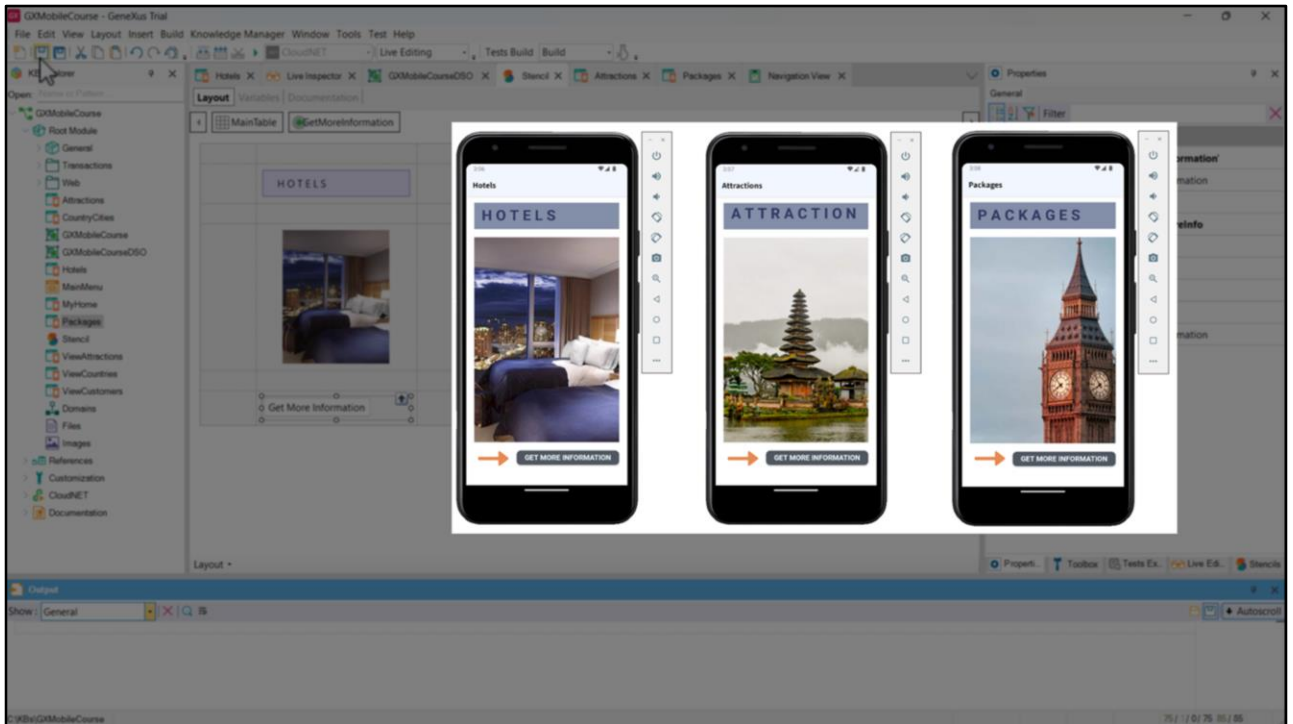
We may stop using it, at anytime, with the *Unwrap Stencil* option.

Let's change the title label to *P A C K A G E S*, and replace the image with the one called *imgPackages*.



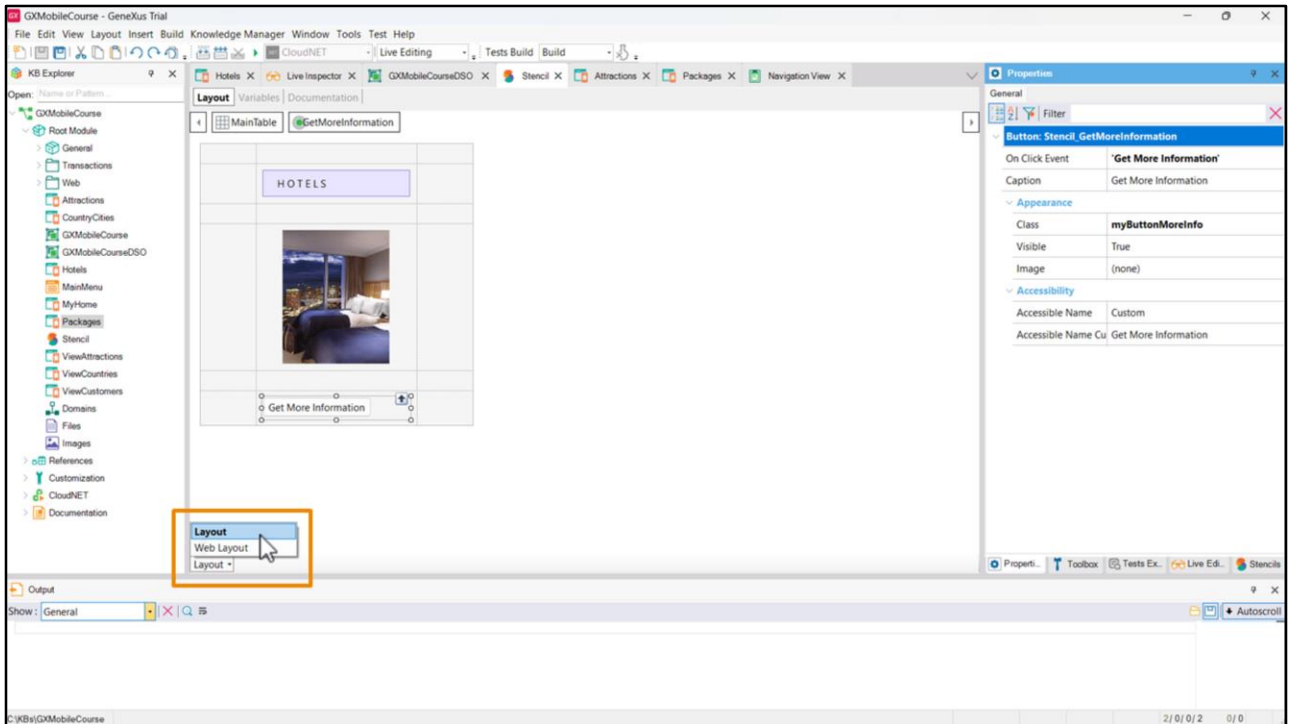
Now let's suppose that the client did not like our button and asks for it to be smaller and aligned on the right.

As explained before, the changes in appearance made on a Stencil object are applied to objects containing it, so, we go back to the Stencil and change the button's Width property to 250, and its Height property to 35. Additionally, we will change the horizontal alignment to place it on the right.



We can see that all the Panels adopted the change made to the button.

In sum, controls may be devised as minimum units that make up the UI, and the classes allow a centralized design. Stencils create bigger design units that consist of controls, in order to define their design (with no associated code), in order to be reused in different objects.

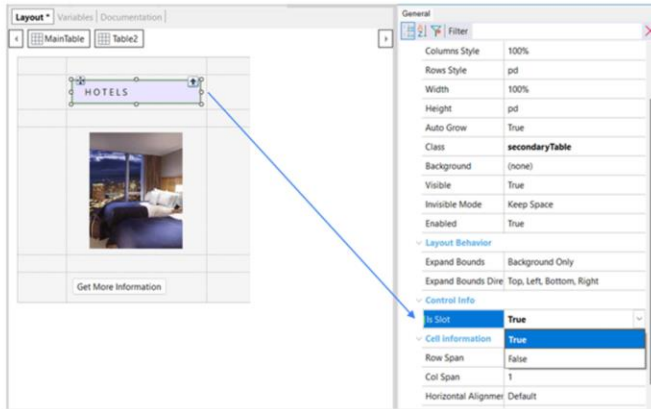


In designing an application, either for Web or for mobile devices, some visual elements in both platforms are conceptually the same, even when they must be designed differently.

To bear this in mind, Stencils contain two designs:

- the Web Layout: to be used in objects containing a Web Layout, such as Web Panels, Web Components, etc.
- and the Layout: to be used in objects containing a Layout, such as Panels and Components.

Slots in Stencils



<https://wiki.genexus.com/commwiki/wiki?51385>

<https://wiki.genexus.com/commwiki/wiki?38418>

As from Upgrade 11 of GeneXus 17, a container may be set up as Slot.

This means that it is possible to edit and customize that part of the Stencil, in each object where it is included, setting its Is Slot property as True.

For further information on this subject, refer to the WIKI article on Slots in Stencils:

<https://wiki.genexus.com/commwiki/wiki?51385>

Other examples and scenarios for using Stencils are included in this article:

<https://wiki.genexus.com/commwiki/wiki?38418>



GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com