

Lab: Getting Started with GeneXus for Smart Devices

GeneXus™ 16

September 2019

Copyright GeneXus S.A. 1988-2019.

All rights reserved. This document may not be reproduced by any means without the express permission of GeneXus S.A. The information contained herein is intended for personal use only.

Registered Trademarks:

GeneXus is a trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners.

CONTENTS

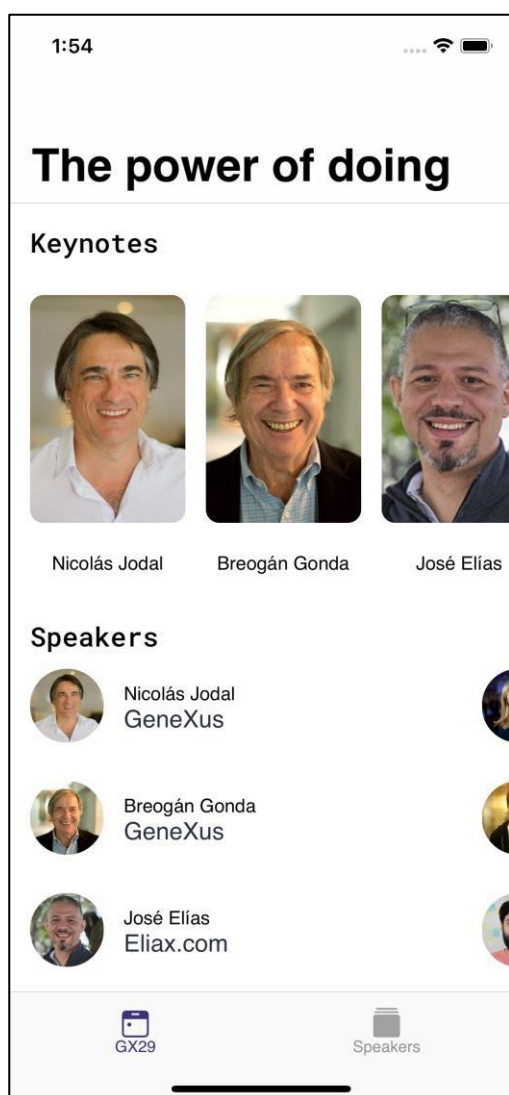
CONTENTS.....	2
OBJECTIVE	3
GETTING STARTED	3
INTRODUCTION – GENEXUS ENVIRONMENT	4
STEP 1 – EXECUTION AND INITIAL DATA LOADING	6
STEP 2 – WORKING WITH STENCILS	7
STEP 3 – CREATING SD PANELS AND MENU	10
CREATING THE APPLICATION MENU	16
STEP 4 - RUNNING THE APPLICATION ON ANDROID AND IOS	19
STEP 5 – ADDING DESIGN	21
NEW WAYS TO SHOW LISTS.....	24
STEP 6 – SPEAKER DETAIL USING CANVAS.	27

OBJECTIVE

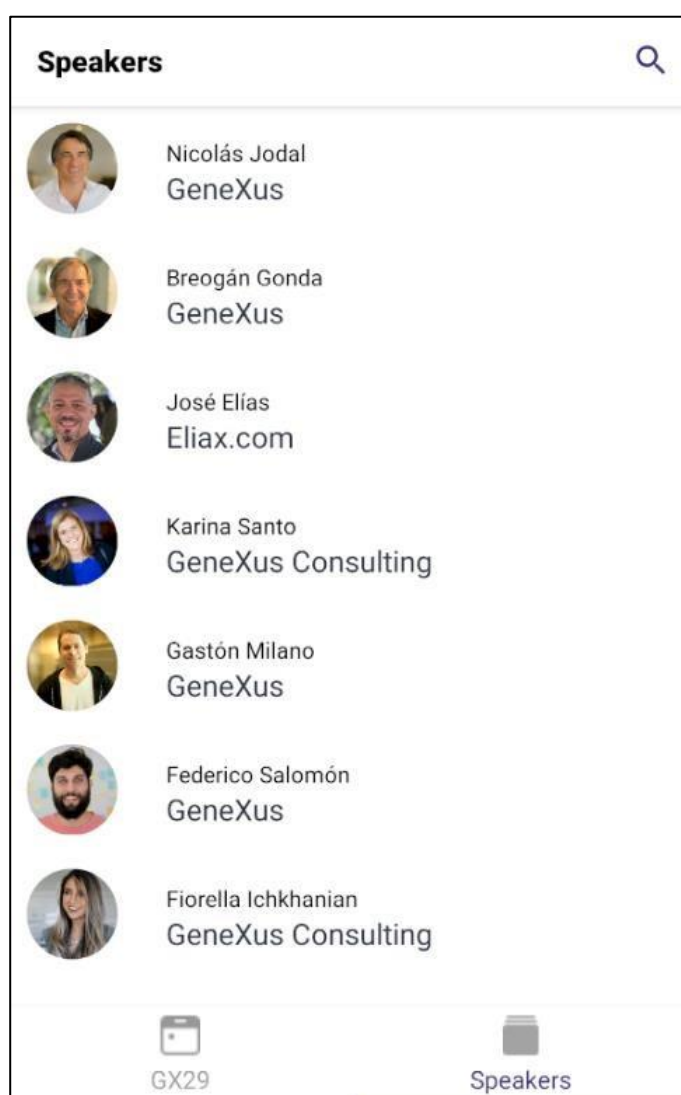
In this Lab you will have a step-by-step guide to build a simple application for **Smart Devices** using **GeneXus 16**. The objective of this Lab, however, is not training but getting familiar with the simplicity to develop applications for Smart Devices with GeneXus.

We will work on a fragment of the GX29 app. The starting point will be an initialized KB with its entities, and at the end of the Lab you should have an Android or iOS app, with design features and some functionalities.

Below are two sample screens that you will create in this Lab, among others.



iOS

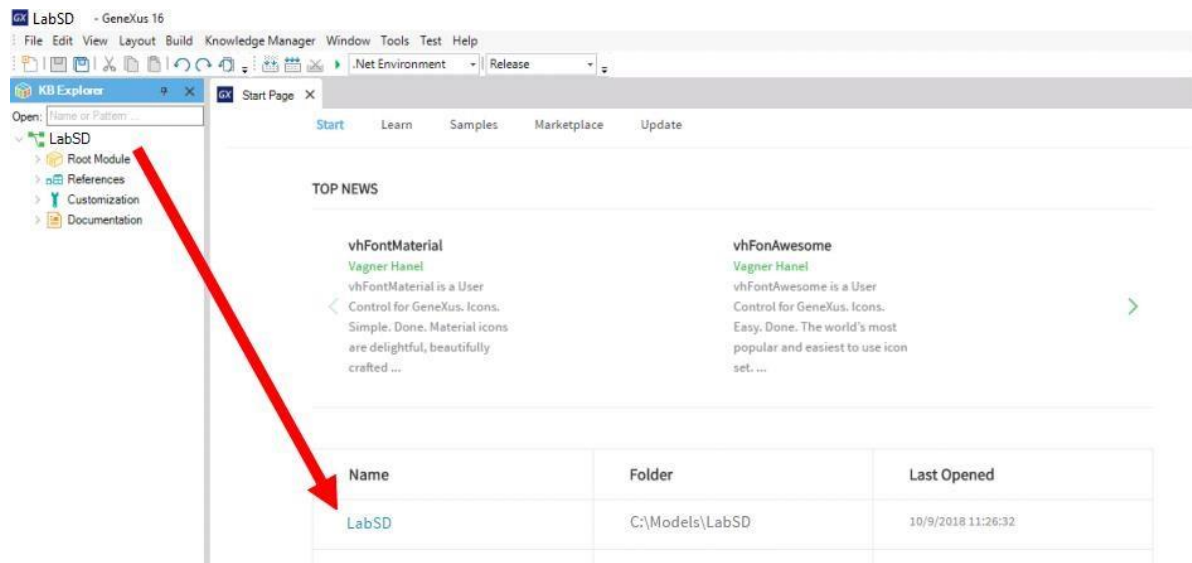


Android

GETTING STARTED

In this Lab, GeneXus 16 will be used.

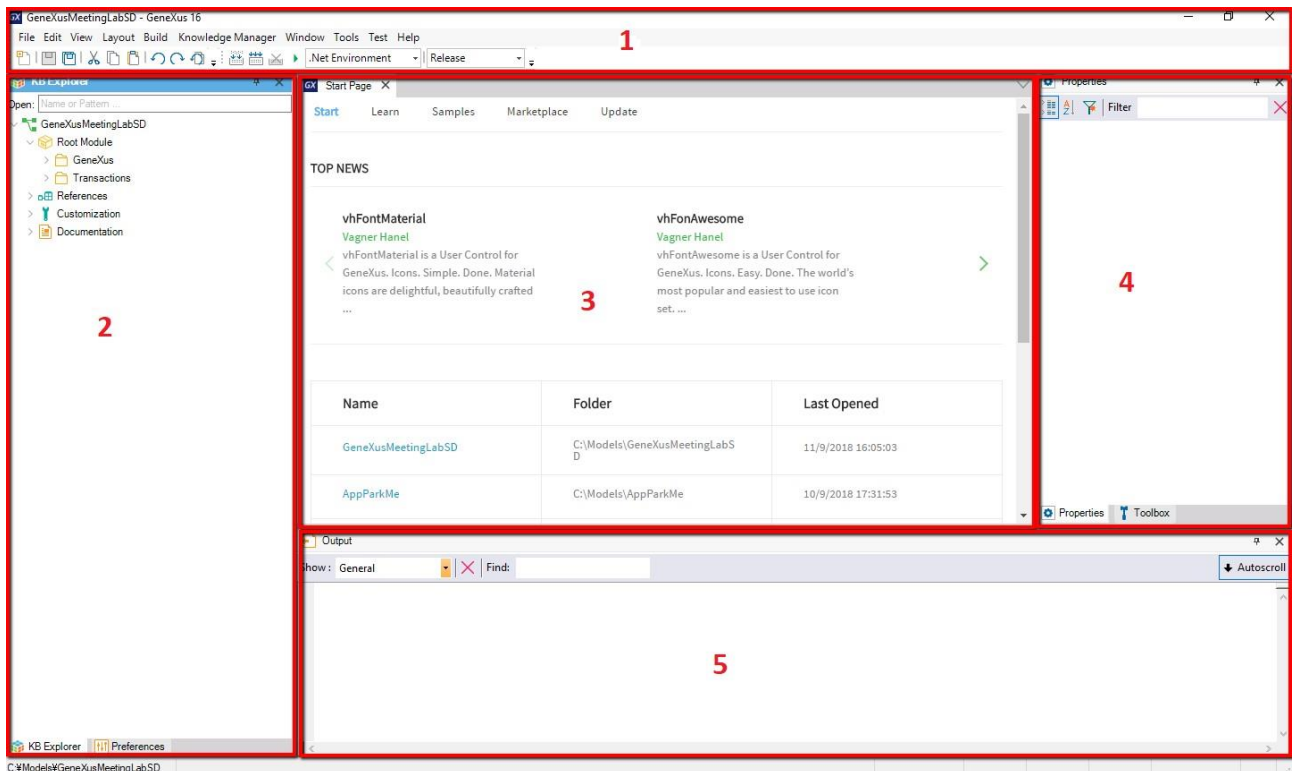
Start GeneXus, and open the Knowledge Base **LabSD** from Recent Knowledge Bases in the Start Page, as shown in the image:



This KB is limited to the specific purposes of the lab. It contains a web application and its entities, which we will take as a starting point to create a Smart Device version of this app.

INTRODUCTION – GENEXUS ENVIRONMENT

Next, you will see the welcome screen that is displayed when opening GeneXus. We'll go over its components before moving on (if you're already familiar with the GeneXus IDE, you may go to the first step):



1. **Toolbar:** It gives access to all the options of the GeneXus Knowledge Base.
2. **Knowledge Base Navigator:** It's a set of context menus to browse objects, whether using folders (Folder View) or categories (Category View), or to view the list of recent changes (Latest Changes View) and model properties (Preferences).
3. **Main:** It shows the Start Page and a tab for each open object.
4. **Properties and toolbox:** controls, objects and variables selected.
5. **Output:** of operations (specification, generation, compilation, etc.).

STEP 1 – EXECUTION AND INITIAL DATA LOADING

As you can see in the Folder View of the Knowledge Base Navigator, there is a pre-loaded application, which, as already mentioned, is only implemented for the web with its entities and it will be your job to create an **Android / iOS** version.

In this Lab, you will run your application **locally**, as can be seen in the following properties of the .NET generator (Preferences of the Knowledge Base Navigator), below the Execution node:

- Deploy to cloud = No
- Web Root = <http://localhost/LabSD.NetEnvironment/>

Change the Web Root property by changing “localhost” for the name of the virtual machine you're connected with, which can be seen in the upper part of the screen:

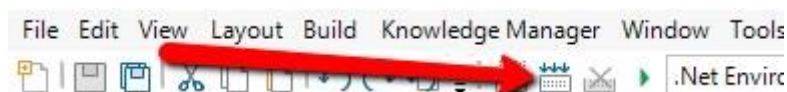
Example:



Now, set the Data Store properties:

- Database name = LABSDYourName
- Server Name = localhost\SQLEXPRESS

After doing this, select **Rebuild All**.



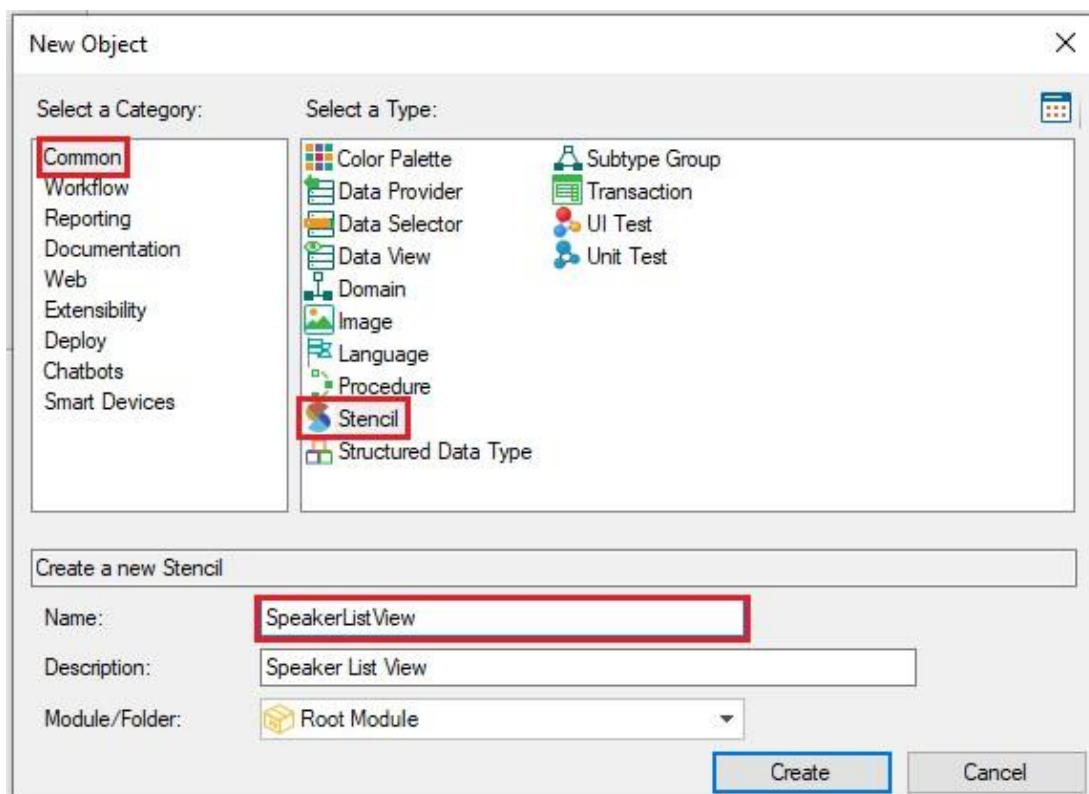
When the **Impact Analysis** window is displayed, click on **Create** to have the tables created in the database. In this way, when a success message is displayed in the Output, the entities will already be created with their preloaded data.

STEP 2 – WORKING WITH STENCILS

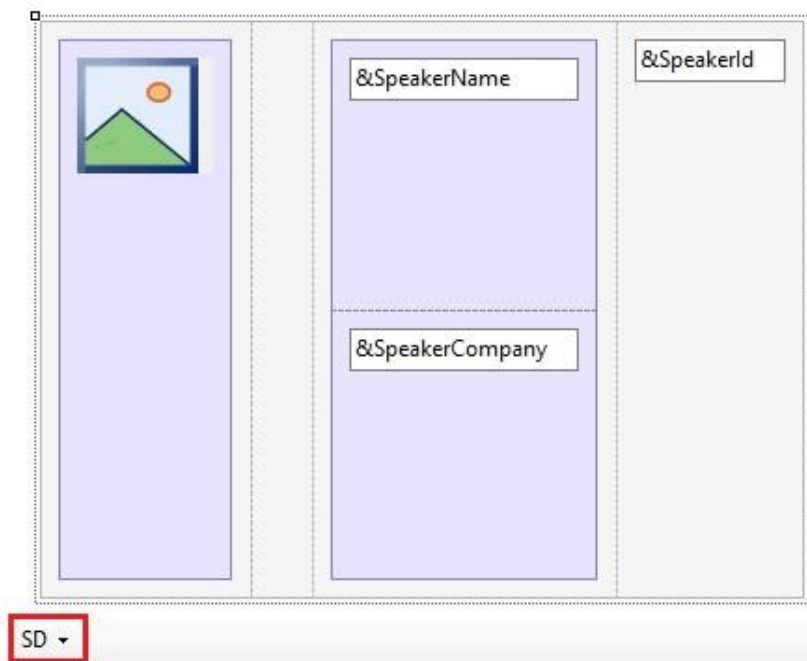
When developing applications, we usually reuse the same design in different screens. For example, in different parts of the GX29 app the speakers' information is displayed as follows:



In this Lab, we'll show the speakers' information in two different SDPanels, so we'll create a Stencil object called "**SpeakerListView**" to model this design.



Once the object is created, we'll create 4 variables: "**SpeakerId**," "**SpeakerName**," "**SpeakerImage**" and "**SpeakerCompany**" and drag them to the layout, arranged in this way:



Note that **SpeakerImage**, **SpeakerName** and **SpeakerCompany** are contained in their respective tables (don't forget to add them). Also, when adding the attributes, they are displayed with text on the left ("Label Caption"); this can be removed by changing the property "Label Position = none" for each attribute. Finally, we will set the property **Visible** of **SpeakerId** to **false**.

Now, set the properties of these objects.

The "Main Table" properties must be entered in this way:

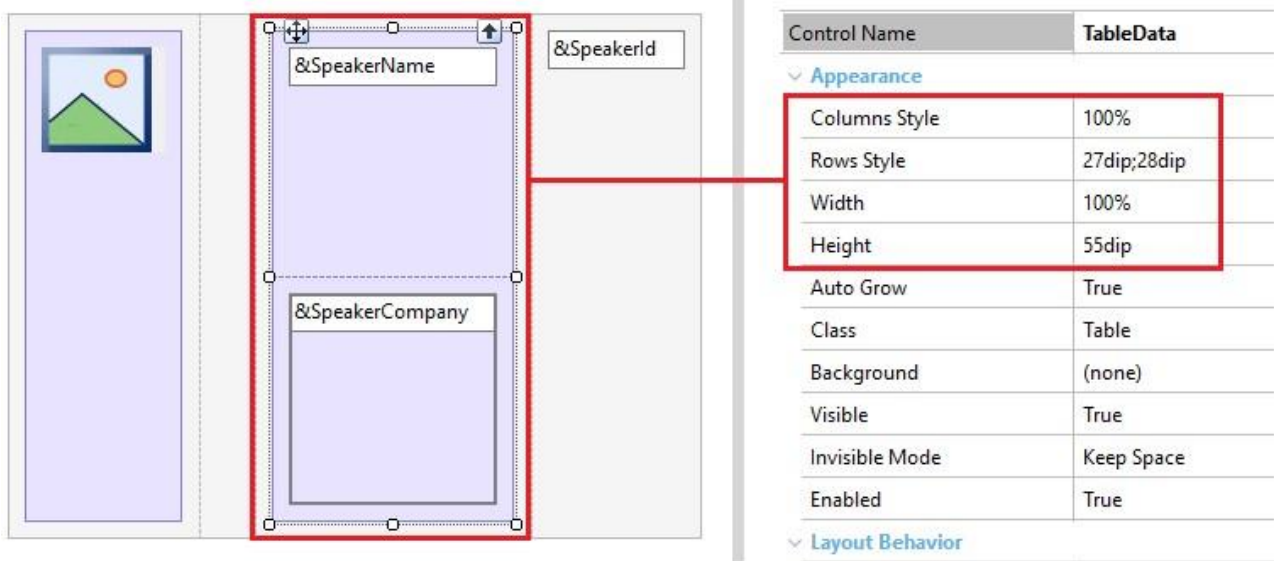
Table: MainTable	
Control Name	MainTable
Appearance	
Columns Style	55dip;15dip;100%;0dip
Rows Style	55dip
Width	100%
Height	100%
Auto Grow	True
Class	TableSpeakerListItem
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Layout Behavior	
Expand Bounds	Background Only
Expand Bounds Limit	Behind System Bars
Expand Bounds Directions	Top, Left, Bottom, Right

Next, set the properties of the table containing **SpeakerImage** and the table containing **SpeakerName** and **SpeakerCompany**.

SpeakerImage:

table: tableImage	
Control Name	TableImage
Appearance	
Columns Style	55dip
Rows Style	55dip
Width	55dip
Height	55dip
Auto Grow	True
Class	Table
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Layout Behavior	

SpeakerName and SpeakerCompany:

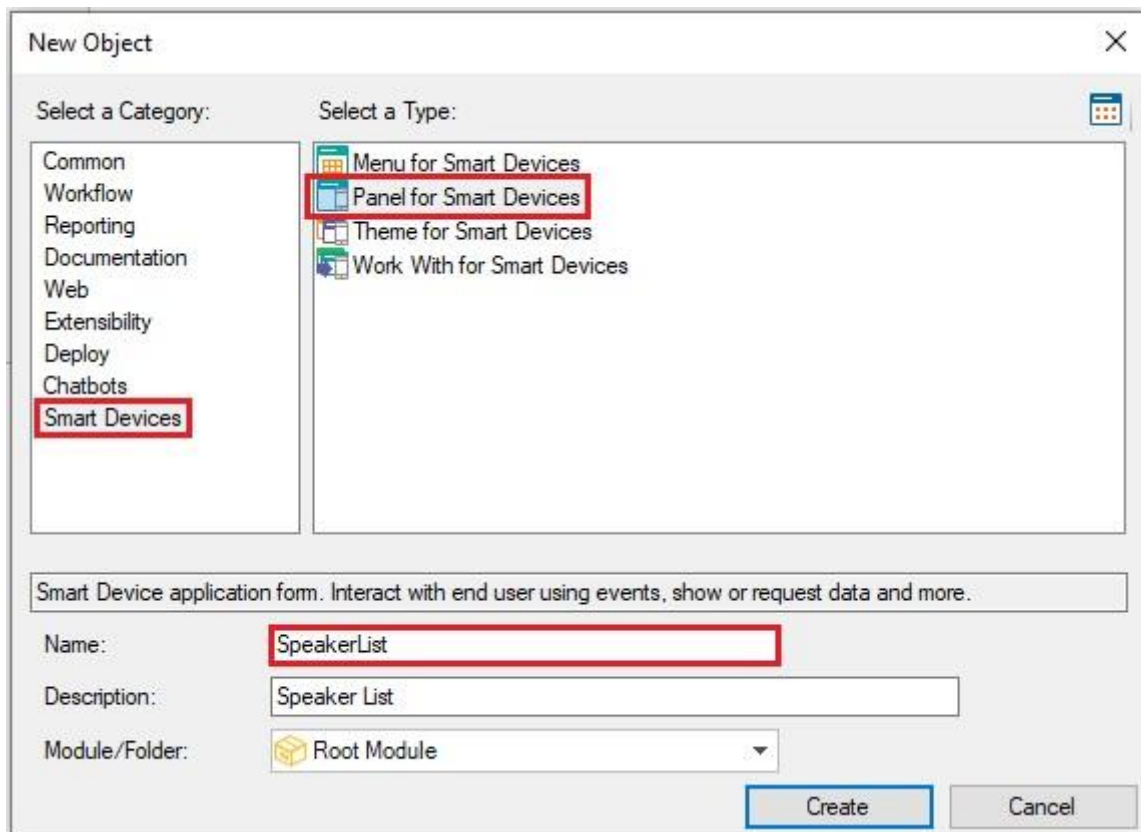


Lastly, in the “Class” property of **SpeakerImage** set the value “**ImageSpeakerList;**” in “**SpeakerName**” the value “**AttributeSpeakerName;**” and in “**SpeakerCompany**” the value “**AttributeSpeakerCompany.**” Save the changes made to the object.

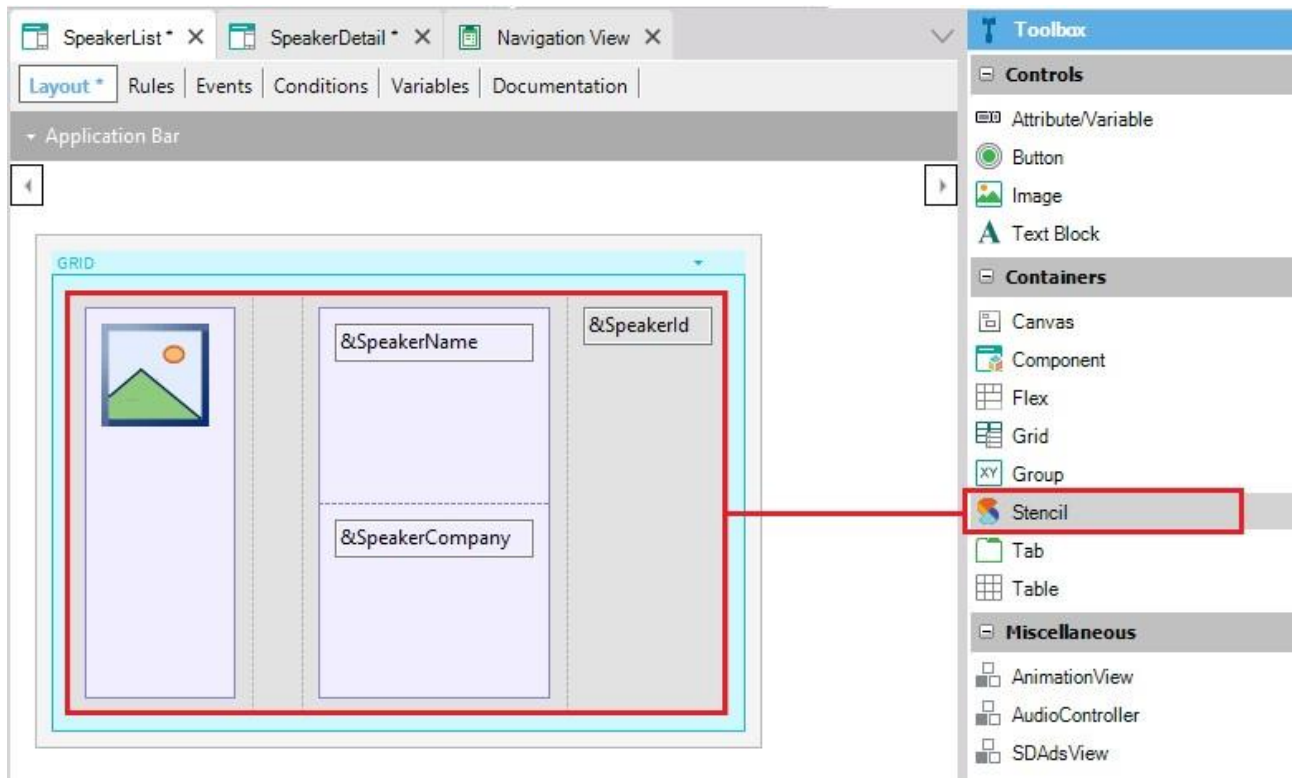
Your design is now ready to be used in all the screens necessary.

STEP 3 – CREATING SD PANELS AND MENU

Next, create the SD objects for your application. To do so, **right-click** on the folder **AppSD**, **New > Object**, and select Category: **Smart Device**, Type: **Panel for Smart Devices**. Call it **SpeakerList**.

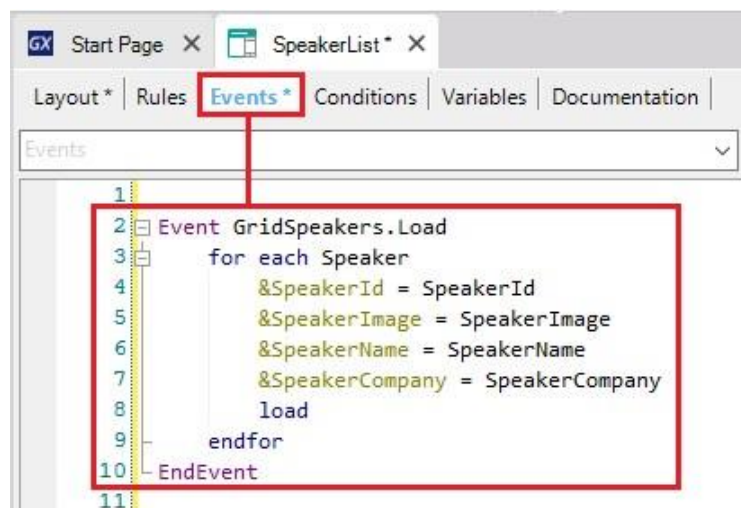



From the Toolbox, drag the **Grid** control to your layout. A popup will be displayed and just enter “OK.” A grid will then be created to show all the speakers of the event in the format that has already been created in the Stencil object. Drag the stencil from the Toolbox to the grid, as shown in the image below:



Once this format has been created, all you have to do is create the load function for this grid. Click on the Grid and change its name in the “**Control Name**” property, setting it to “**GridSpeakers.**”

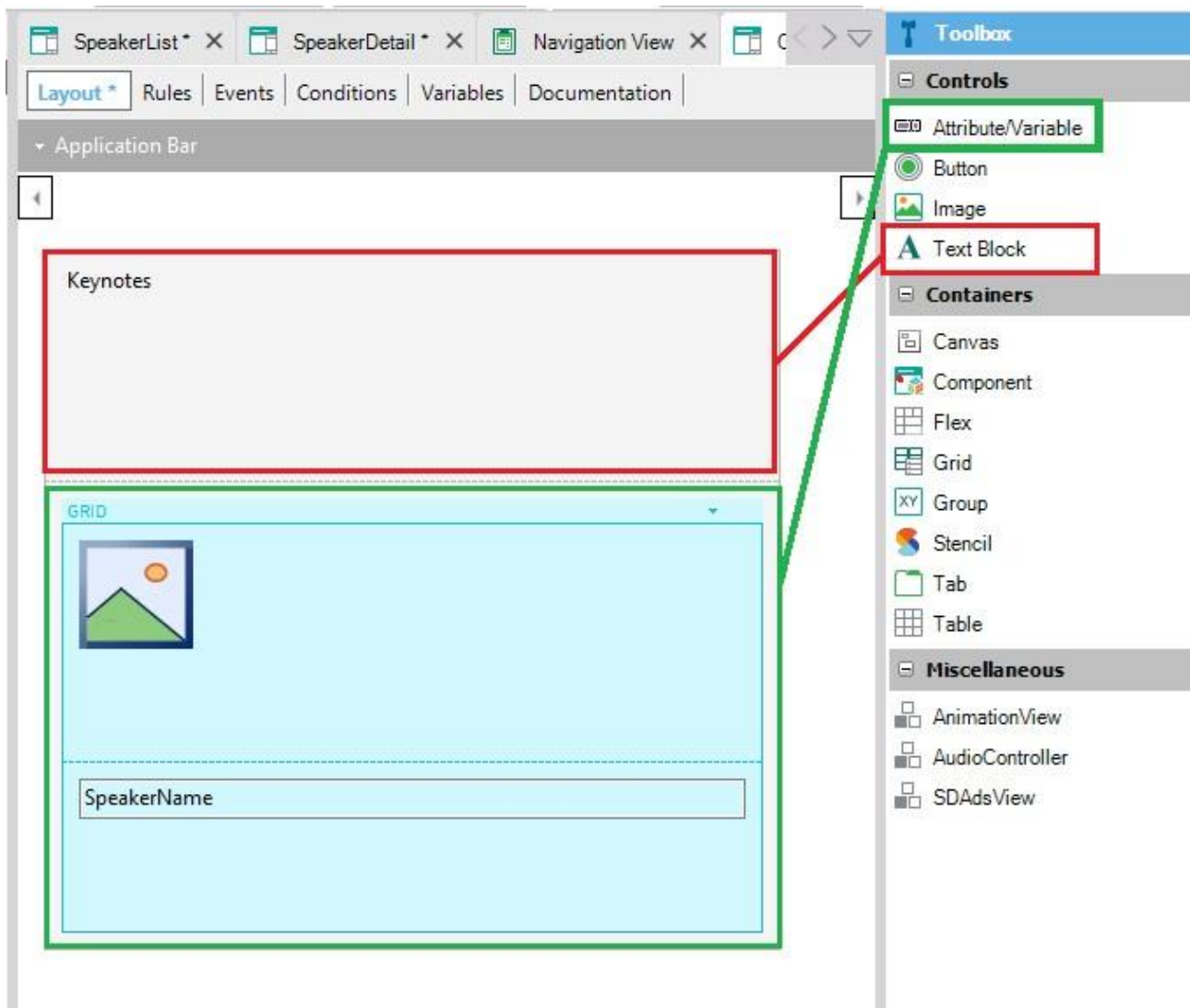
Next, select the “**Events**” tab in the upper part of the layout and create the following data load event:




Finally, click on the save button  and you will have successfully created a panel.



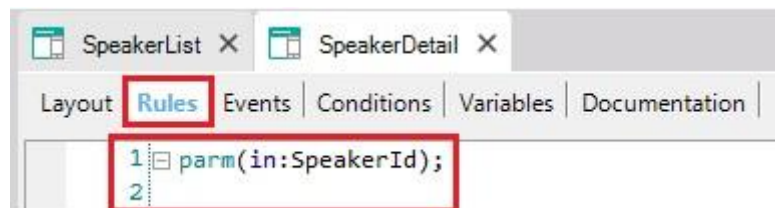
Repeat the process and create another **Panel for Smart Devices** called **GXHome**. Drag a **Grid** and this time select the attributes **SpeakerImage** and **SpeakerName**, and a Text Block for the title, in this way:



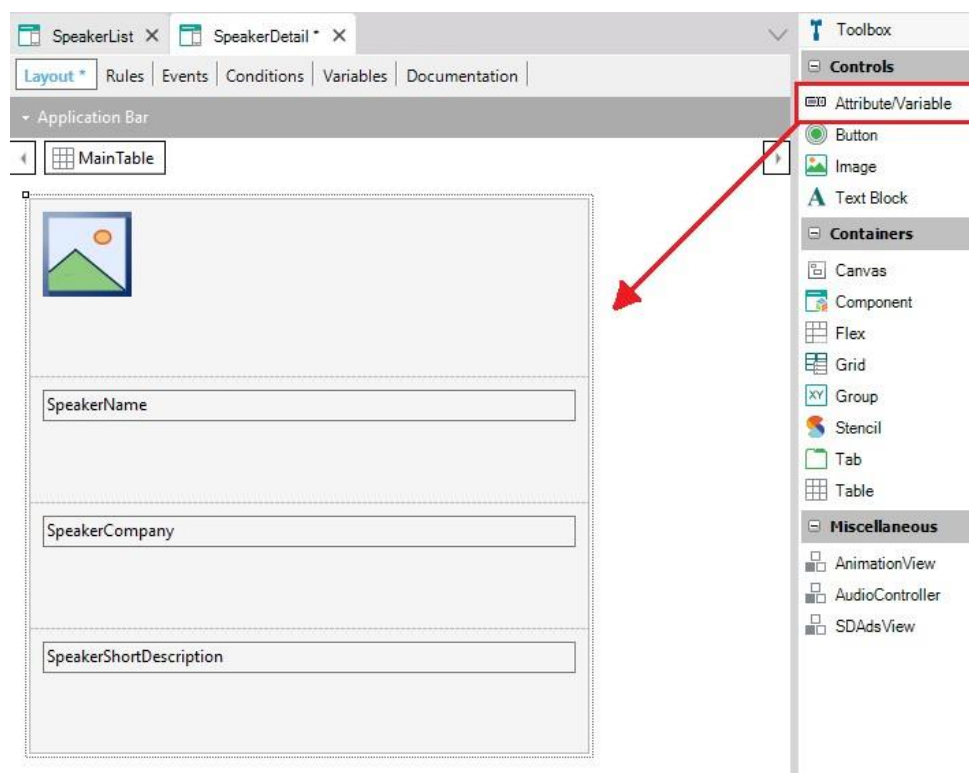
Since in this grid you only need to show the Keynote speakers, in the **“Conditions”** property of the grid, enter the following line: **“SpeakerIsKeynote = true;”**

Save  (**CTRL + S**) and you will have created your second Panel, which will show a grid with the Keynotes highlighted, among other data that we will see later.


When **tapping** on one of the speakers (Keynotes), we want to show detailed information about him/her. To do so, create another Panel called **SpeakerDetail**, adding to it in the **rules** that it receives as parameter the ID of the speaker tapped on in the previous screen, in the following way:



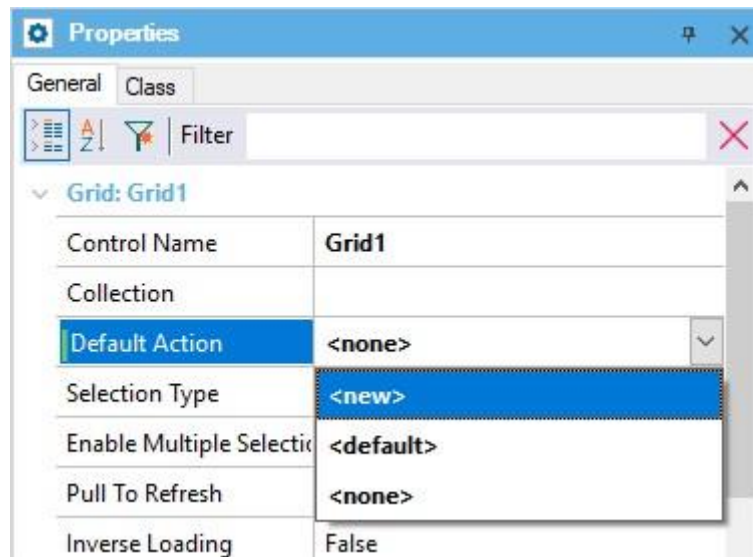
Back in the **Layout**, drag the attributes **SpeakerImage**, **SpeakerName**, **SpeakerCompany** and **SpeakerShortDescription**:



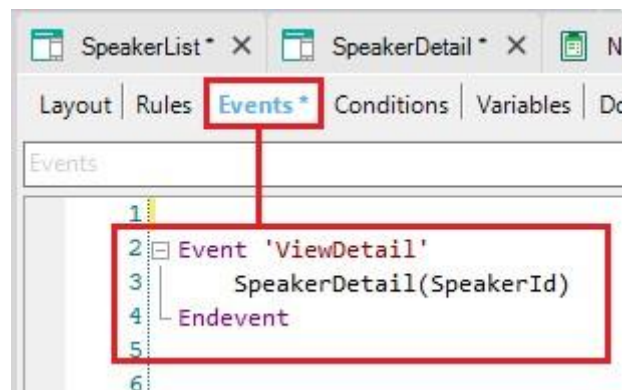
Do not forget to remove the attribute labels, and in **"SpeakerShortDescription"** set the property **"Auto Grow = true."**

Save  and return to the **GXHome** Panel, and in the **Grid** properties create a new action so that tapping on a Keynote opens the **SpeakerDetail** panel with the speaker's details.

To do so, click on the property **Default Action** > **new** of the Grid,



called "**ViewEventDetail**" and enter the following code:

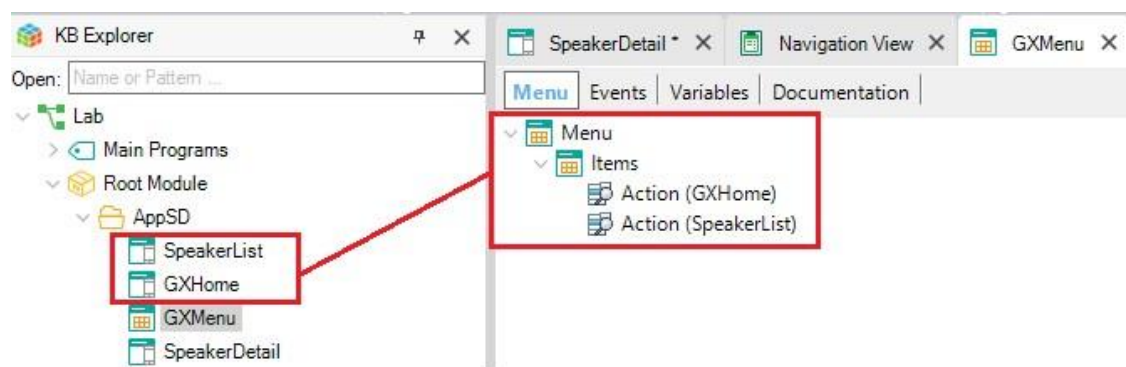


CREATING THE APPLICATION MENU

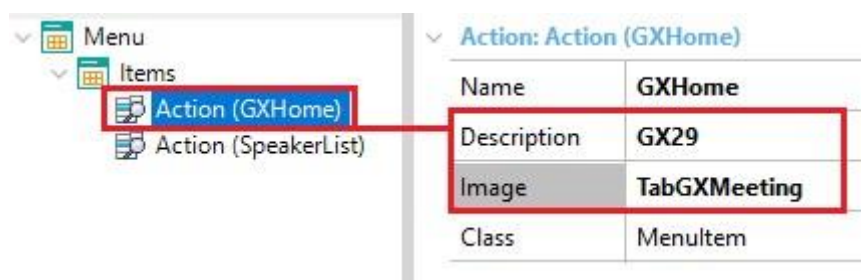
Now we'll create the **Menu for SD** to be able to navigate between the newly created panels and also be the entry point into the application.

To do so, right-click again on the folder **AppSD**, **New > Object**, and this time select the Type: **Menu for Smart Devices**; call it **GXMenu**.

Now you only need to add the panels you want to access from the menu. To do so, drag **GXHome** and **SpeakerList** from the KB Explorer to **Items**, obtaining the following image:



Once this is done, assign descriptive icon images to the two newly created menu options. To do so, click on **Action (GXHome)** and in the **Image** property, select the following value: **TabGXMeeting**. In the **Description** property, type the text **GX29**.



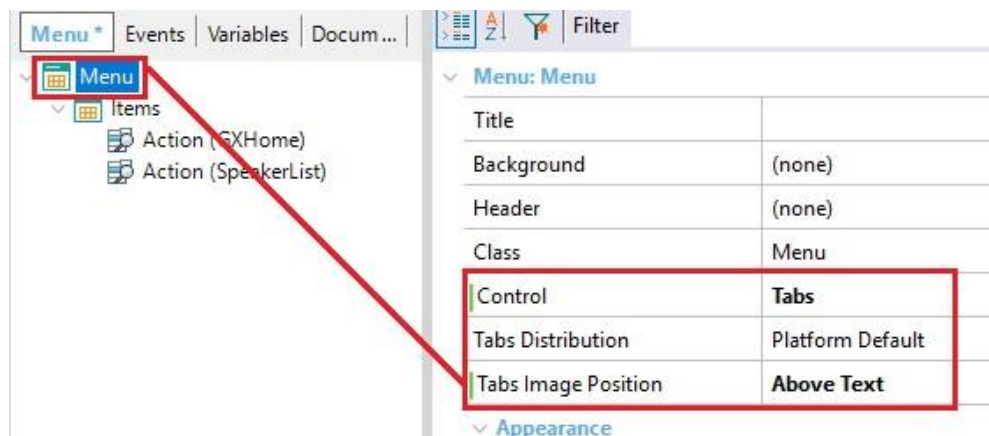
Likewise, for the **Action (SpeakerList)** item, select the image **TabSchedule** and in its description type **Speakers**.

Now, select a **name and icon** for your application.

To do so, click on **GXMenu** and look for the **Application Title** property. Set this property to **GX29**.

Next, to set an icon, search for the property “**Android Application Icon**” or “**Apple Application Icon**” (depending on which device you are going to perform the tests with) and set it to “**GXMeetingIconAndroid**” or “**GXMeetingIcon**,” respectively.

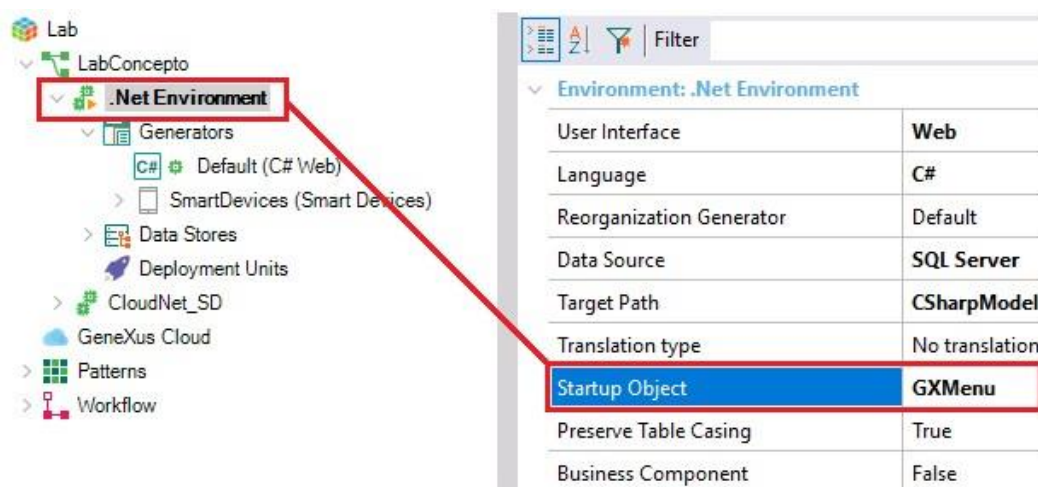
Lastly, set the “**Tabs**” property (as shown in the image) so that your app menu is displayed as tabs at the bottom.



Click on **Save All** to save all changes made so far.



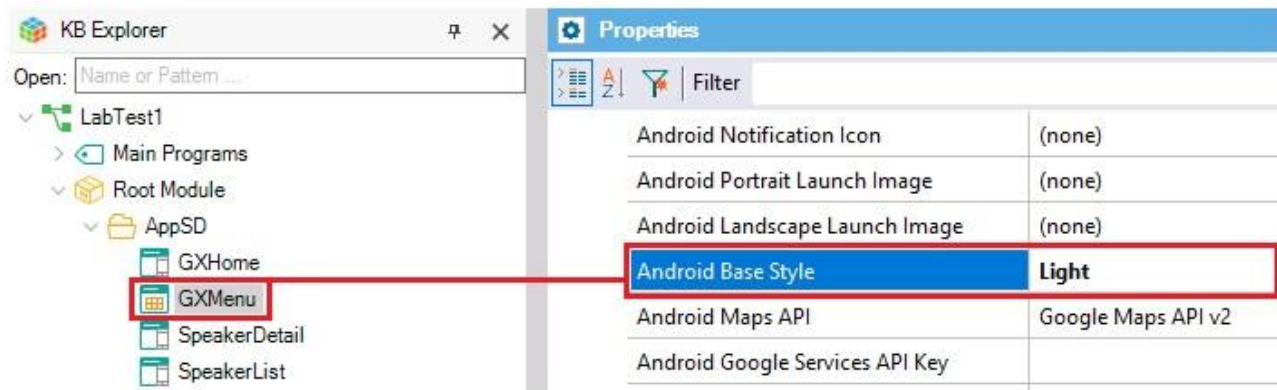
You’re now ready to **run** the first version of your application. To this end, you only need to indicate that the Menu you’ve just created will be the entry point of the application. This is done from the **Startup Object** property of the .Net Environment:



Note for Android users:

Participants who are going to run the application on an Android device must make one more configuration related to the “base style” that will be used. The base style is the base color that will be used in Android layouts; in this case, select the value "light", i.e. **white backgrounds, black fonts, and white application bars**.

To do so, click on the GXMenu object and change its property as shown in the image below:



STEP 4 - RUNNING THE APPLICATION ON ANDROID AND IOS

After doing this, you can run your application. Select “**Build all**” and “**View > Show QR codes.**” A web page will be opened with the QR codes that you will have to read from your Android or iOS device, respectively.

Install Android Apps



GXMenu

[Download](#)

Install iOS Apps



QRCode for iOS

[Services URL for KBN](#)

For **iOS** you will need to have previously installed “**GeneXus 16 KB Navigator**” which can be downloaded from the Apple Store.

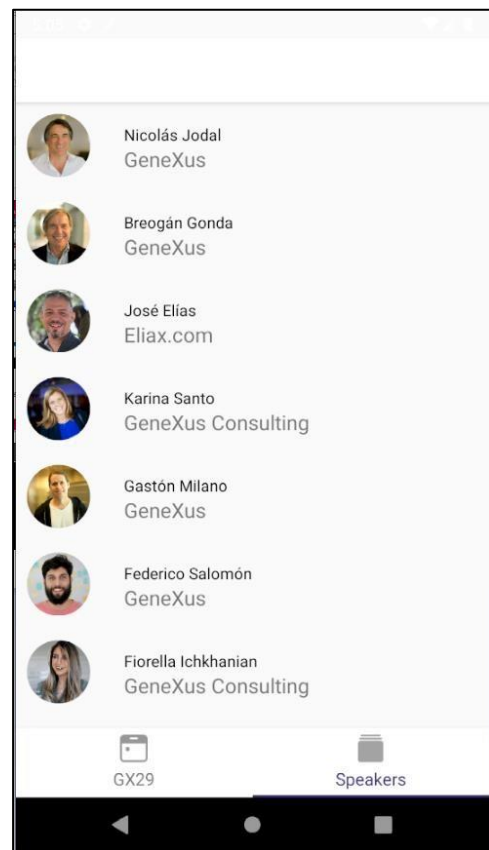
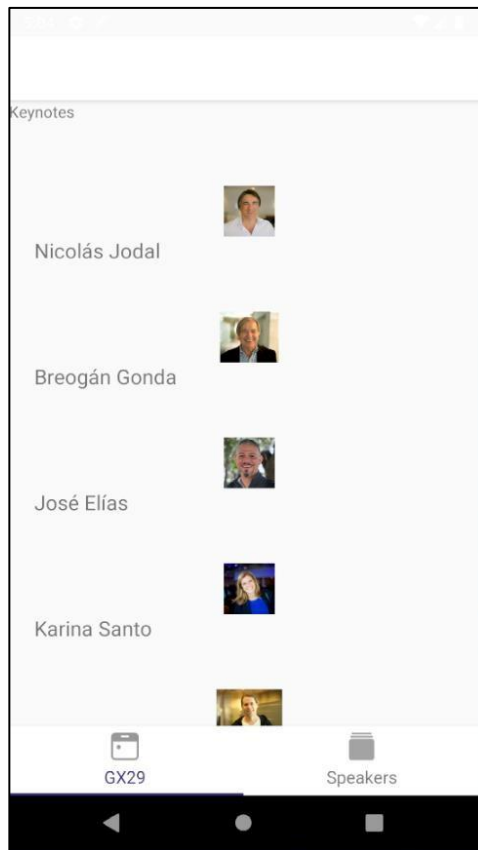


GeneXus 16 KB Navigator 17+

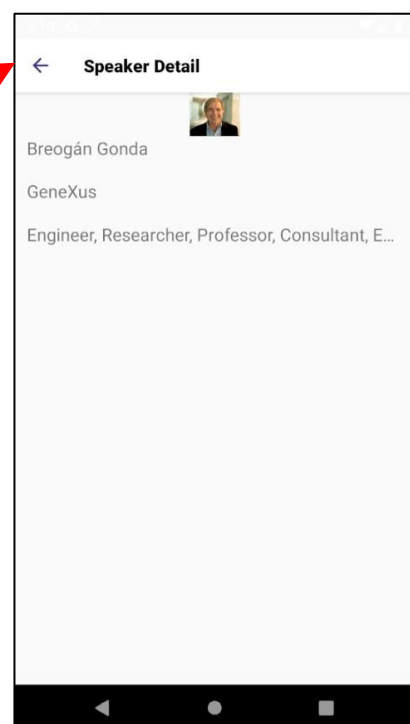
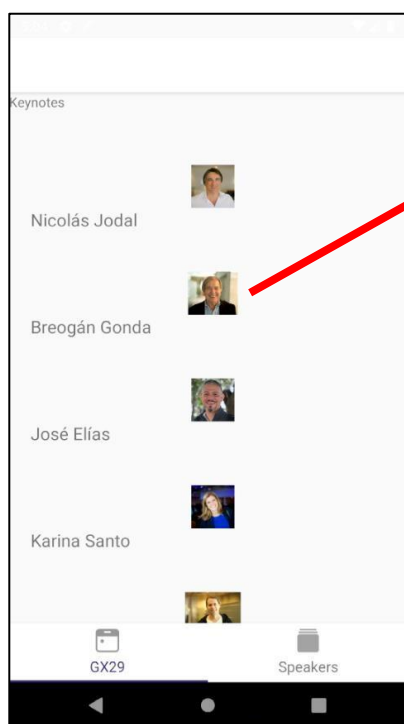
GeneXus S.A.

Free

When you run the app, the Startup Object you had previously configured is loaded:
The Menu for SD **GXMenu**, which will show at the bottom the menu options you have so far:
“**GX29**” where you can see the keynotes, and “**Speakers**” where you can see the list of speakers.



Also, tapping on a speaker will show his details, as programmed.



STEP 5 – ADDING DESIGN FEATURES

Now, add some design features to make your application look better.

To enhance the **GXHome** panel's design, change some of the grid properties. The objective is that the final result looks like the following image, with horizontal scroll to the right:

Keynotes



Nicolás Jodal



Breogán Gonda



José Elías

To this end, use the “SD Smart Grid” control, setting it as follows:

The screenshot shows the GeneXus IDE interface. On the left, the 'Application Bar' contains 'MainTable' and 'Grid1'. Below it, the 'Keynotes' panel is visible, containing a 'GRID' control and a 'SpeakerName' text box. A red box highlights the 'GRID' control. On the right, the 'Properties' window is open, showing the 'SD Smart Grid' control properties. The properties are as follows:

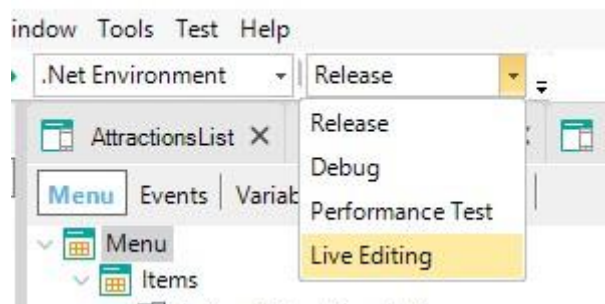
Control Info	
Control Type	SD Smart Grid
Auto Grow	False
Scroll Direction	Horizontal
Snap To Grid	False
Items Layout Mode	Multiple by Quantity
Items Per Column	1

Appearance	
Class	GridNoScrollBars
Visible	True

Next, in the **SpeakerImage** attribute located inside the grid, select the “**ImageKeynoteSpeaker**” class to change its appearance and make it more appropriate to the objective pursued. Do something similar with the **SpeakerName** attribute, and assign it the class “**AttributeSpeakerName**.”

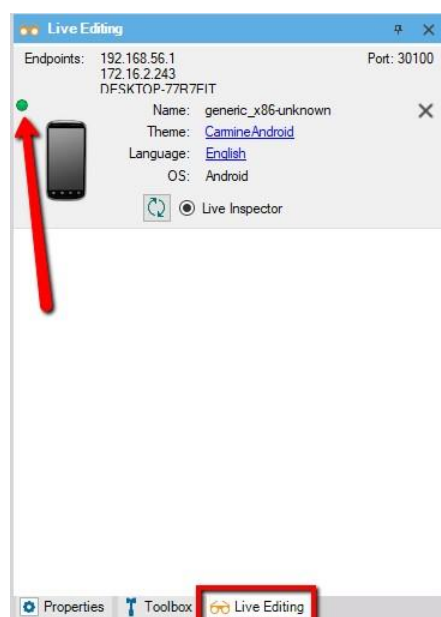
Before running your application to see how it looks with these changes, activate the **Live Editing** functionality that will show, in real time, the design changes made with GeneXus, directly on the device.

To this end, change the value of the combo located in the second row of the toolbar, setting it to **“Live Editing”** instead of the **“Release”** value that is currently selected.

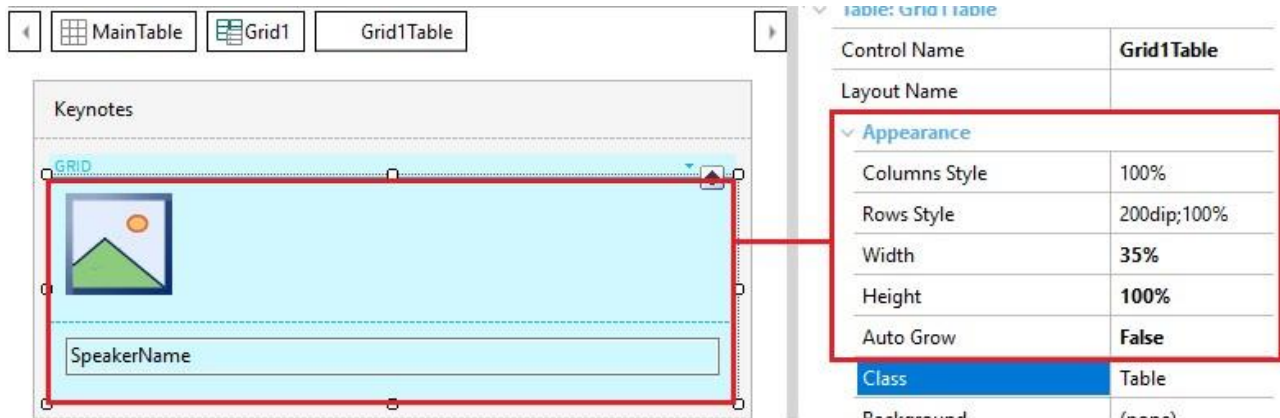


After doing this, select **“Build All”** and read the QR codes again in **“View > Show QR codes.”** In this way, **GeneXus will be connected to your application**. We recommend having the mobile phone with the application in sight when modifying the design in GeneXus.

Confirm that the Live Editing feature has been enabled by clicking on the **“Live Editing”** option in the toolbox. A green dot should be displayed.



With the Live Editing feature enabled and running on your device, resize the table inside the grid to make it look better, as shown below:

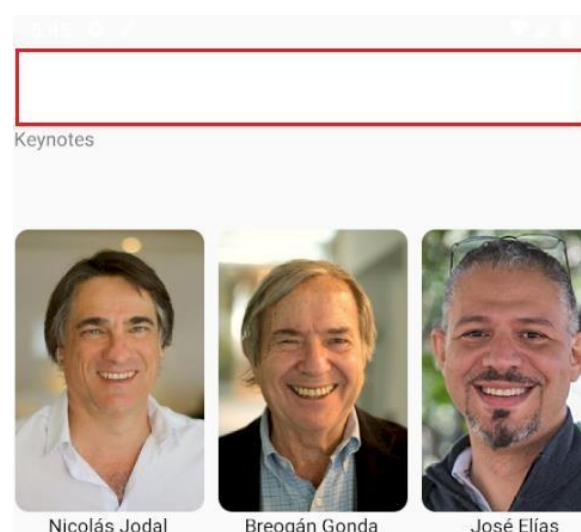


Both the **image** and **SpeakerName** must be aligned horizontally to the center, with the corresponding property assigned:

Horizontal Alignment	Center
----------------------	--------

Note for Android users:

When running the application, you will notice that the application bar is empty.



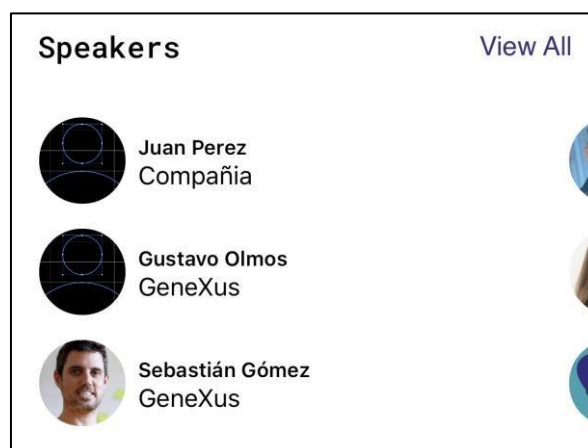
Add text to it to improve the panel's appearance. To do so, open the **GXHome** object and in the **"Events"** tab enter the following code:

```
Event Refresh
    Form.Caption = "The power of doing"
EndEvent
```

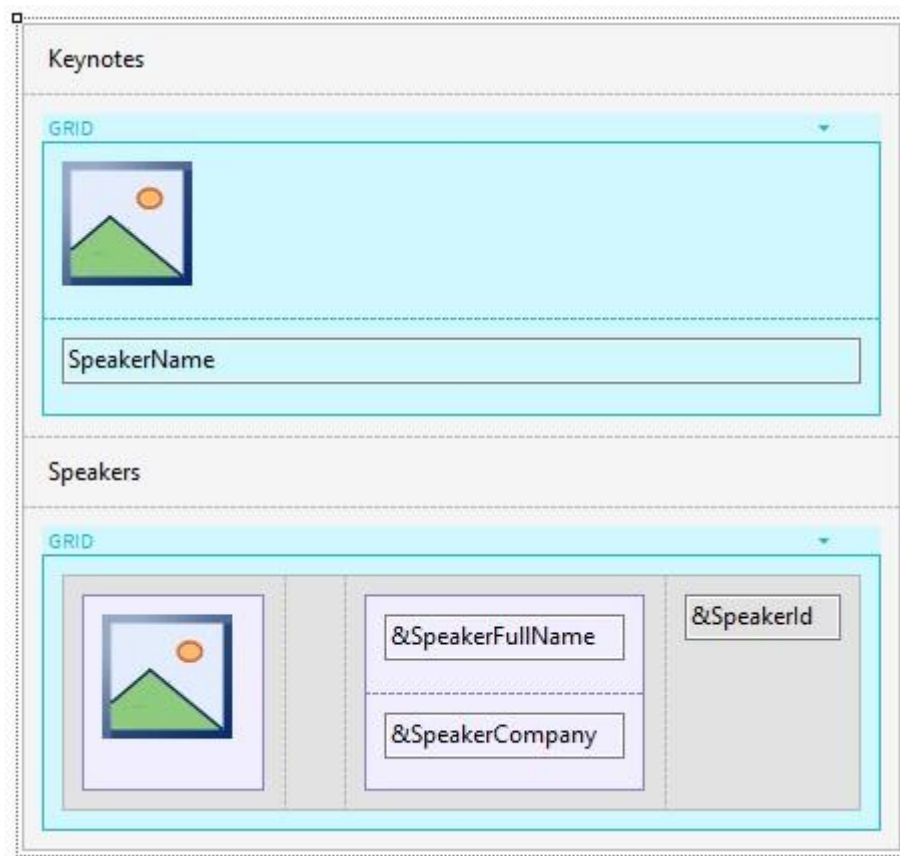
Do the same for the **SpeakerList** panel, but with the **"Speakers"** text.

NEW WAYS TO SHOW LISTS

Add another list style to your **GXHome** screen that will show speakers. The objective is to achieve a result similar to this one:



To do so, add a new Text Block with the text **"Speakers,"** and another grid containing the **Stencil** that has been created. Note that as you will show speakers with the same style as in the other **"SpeakerList"** panel, you can reuse the Stencil that you have already created. The final result should look similar to this:



To give the Speakers grid the desired design, set it in this way:

▼ Control Info	
Control Type	SD Smart Grid
Auto Grow	True
Scroll Direction	Horizontal
Snap To Grid	True
Items Layout Mode	Staggered by Quantity
Items Per Column	3
▼ Appearance	
Class	GridNoScrollBars

Now you need to load the grid because, since it is made up with variables, it doesn't have a base table and you need to tell GeneXus how it will be populated. To do so, in the "Events" tab of **GXHome**, type the following code:

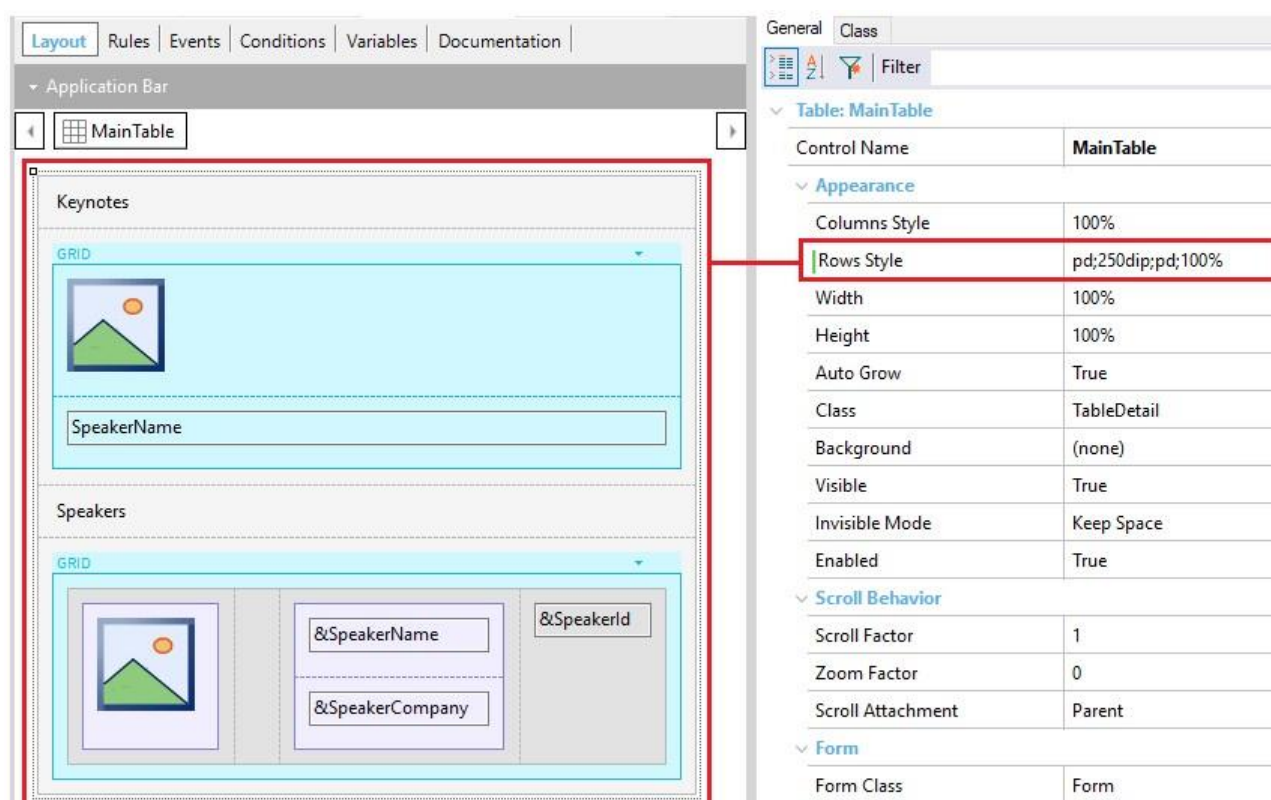
```

6 | Event GridSpeakers.Load
7 |   for each Speaker
8 |     &SpeakerId = SpeakerId
9 |     &SpeakerPhoto = SpeakerImage
10 |    &SpeakerName = SpeakerName
11 |    &SpeakerCompany = SpeakerCompany
12 |    load
13 |   endfor
14 | EndEvent

```

Note that the name of the event “**GridSpeakers.load**” refers to the name of the grid in the layout. If you haven’t changed it, by default it will have the name “**Grid2.**”

Finally, before running it again to see all the changes made, set the size of the MainTable in the layout to correctly show the grids you make.



The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' tab is active, displaying a form with two sections: 'Keynotes' and 'Speakers'. The 'Keynotes' section contains a 'GRID' control with a placeholder image and a text field labeled 'SpeakerName'. The 'Speakers' section contains another 'GRID' control with a placeholder image and two text fields labeled '&SpeakerName' and '&SpeakerCompany'. On the right, the 'Properties' panel is open, showing the 'General' tab for the 'MainTable' control. The 'Rows Style' property is highlighted with a red box and set to 'pd;250dip;pd;100%'. Other properties like 'Columns Style', 'Width', 'Height', 'Auto Grow', 'Class', 'Background', 'Visible', 'Invisible Mode', 'Enabled', 'Scroll Factor', 'Zoom Factor', 'Scroll Attachment', and 'Form Class' are also visible.

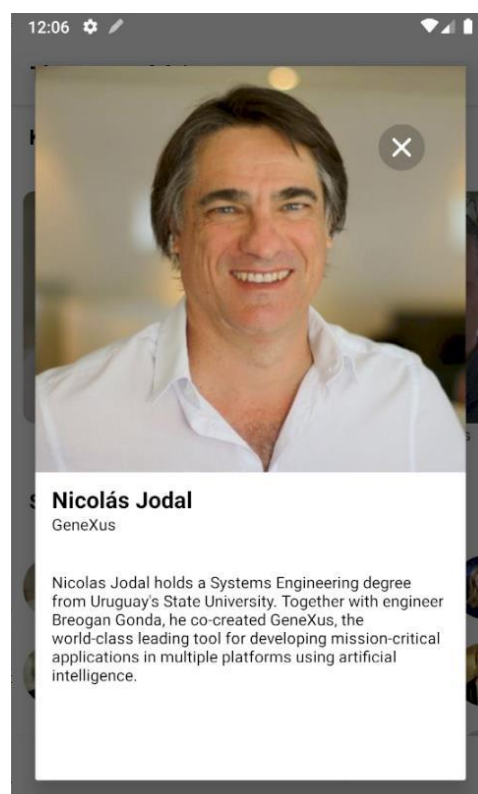
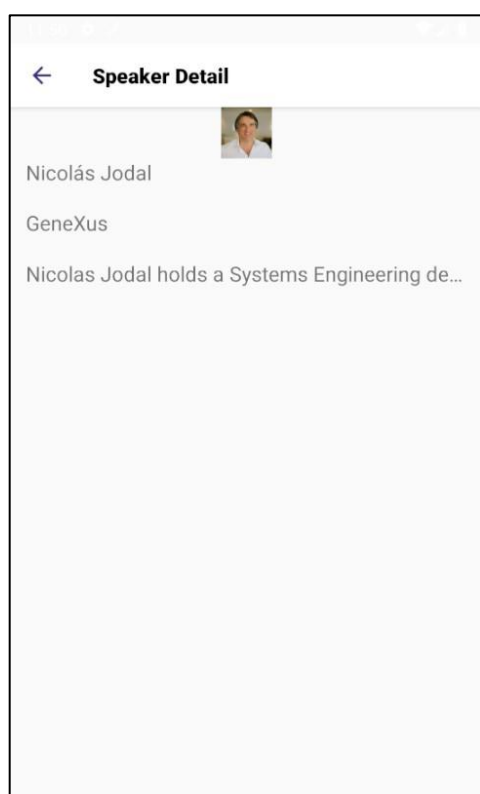
Control Name	MainTable
Appearance	
Columns Style	100%
Rows Style	pd;250dip;pd;100%
Width	100%
Height	100%
Auto Grow	True
Class	TableDetail
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
Form	
Form Class	Form

As a last detail, apply the class “**TxtKeynotesTitle**” to the two text blocks you have in the layout: **Keynotes** and **Speakers**.

Now you can run your application again by selecting “**Build all**” and then “**View > ShowQR codes**” to read the QR code again.

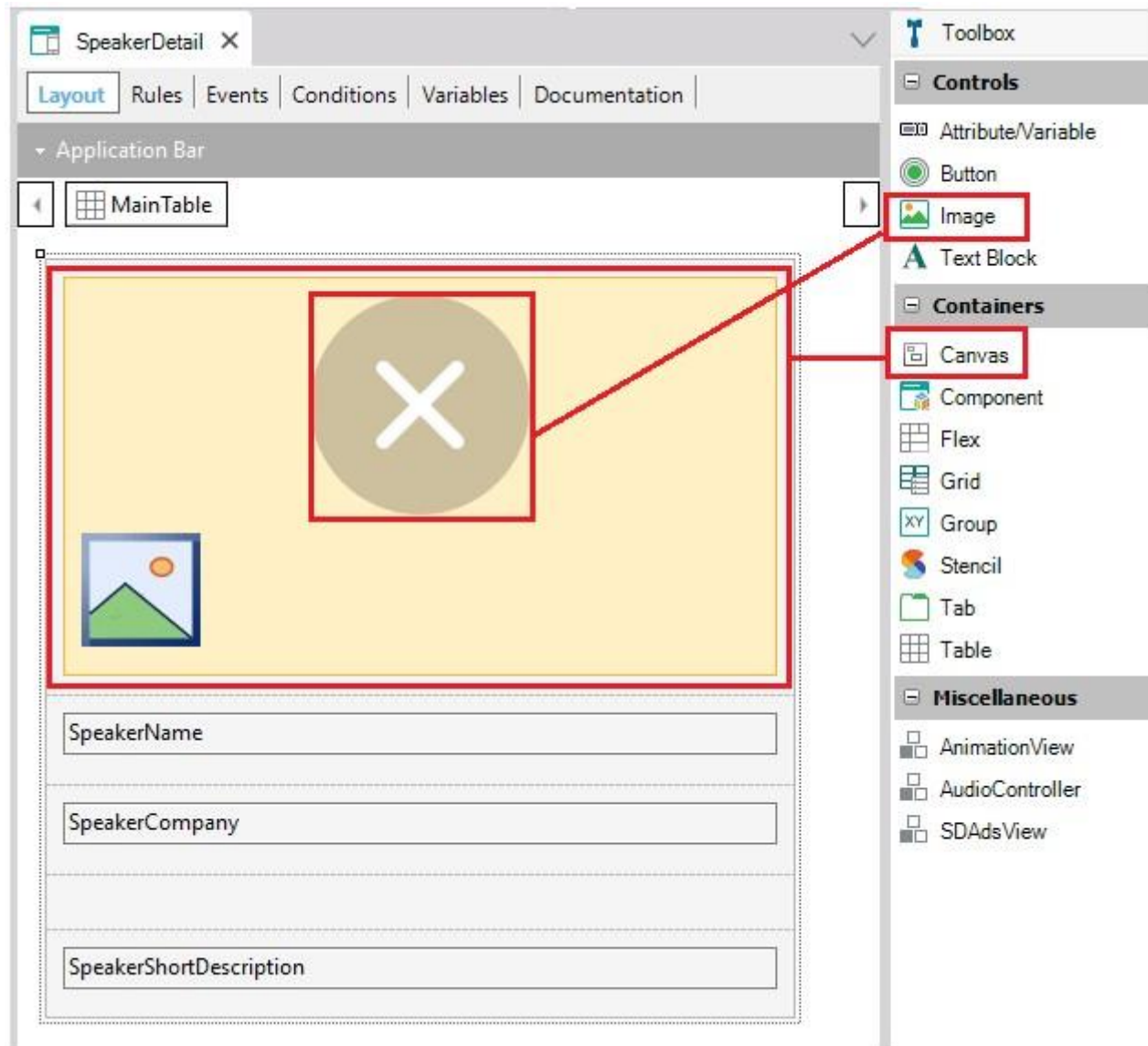
STEP 6 – SPEAKER DETAIL USING CANVAS.

Now, design a speaker's detail panel. We will see an image of how it is so far and another with the result we want to achieve.



To achieve this design, the first thing you must do is, in the object **SpeakerDetail**, create a **canvas** in your layout that will allow you to show a photo (in this case, of Nicolás) with an overlapping image (the cross to close the panel).

Drag a **canvas** to your layout, in which you will also include an **image**. In this case, look for the image called "**Close**." Lastly, drag to the canvas the **SpeakerImage** attribute that you already had in the MainTable and the result should be as follows:



Next, in the canvas set which object goes on top of the other; that is to say, you need to set the “**Z-Order**” property of each element within the canvas. The higher the **Z-Order**, the higher the element will be positioned.

Start by changing the properties of the “close” image. They must look as shown below:

The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' tab is active, displaying a form with three input fields: 'SpeakerName', 'SpeakerCompany', and 'SpeakerShortDescription'. Above these fields is a yellow rectangular area containing a large circular button with a white 'X' (close button) and a small image icon (SpeakerImage). A red box highlights the 'ImageCloseDetail' control on the layout. On the right, the 'Properties' panel is open, showing the 'Class' property set to 'ImageCloseDetail'. The 'Absolute position' section is also highlighted, showing the following values:

Top	35dip
Left	100%
Bottom	100%
Right	35dip
Width	40dip
Height	70dip
Z- Order	2

The **SpeakerImage** properties should look as follows:

The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' tab is active, displaying the same form as the previous image. A red box highlights the 'SpeakerImage' control on the layout. On the right, the 'Properties' panel is open, showing the 'Class' property set to 'ImageSpeakerDetail'. The 'Absolute position' section is also highlighted, showing the following values:

Top	0dip
Left	0dip
Bottom	0dip
Right	0dip
Width	100%
Height	100%
Z- Order	1

Lastly, change the size of the rows in the “MainTable” to give the desired space to each element.

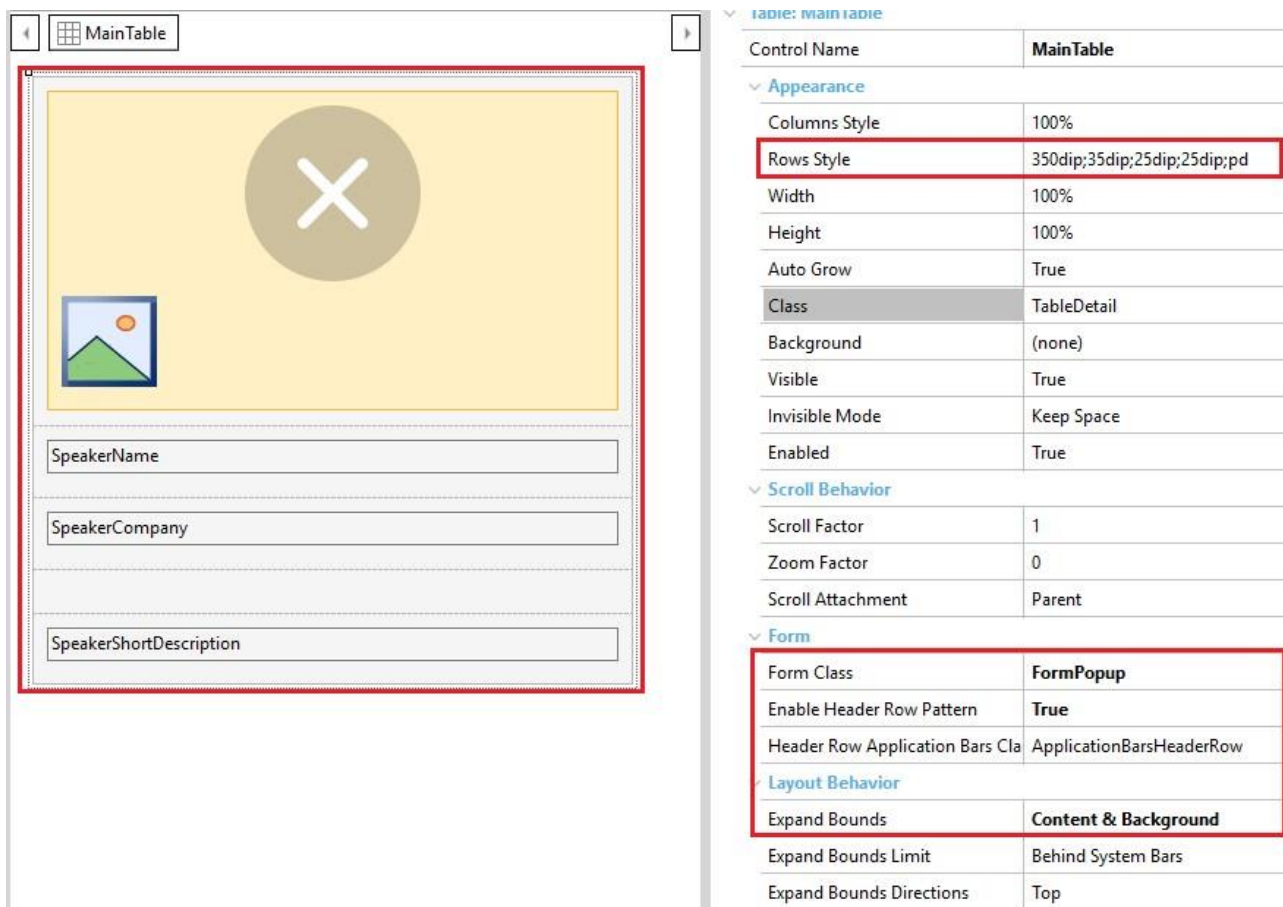
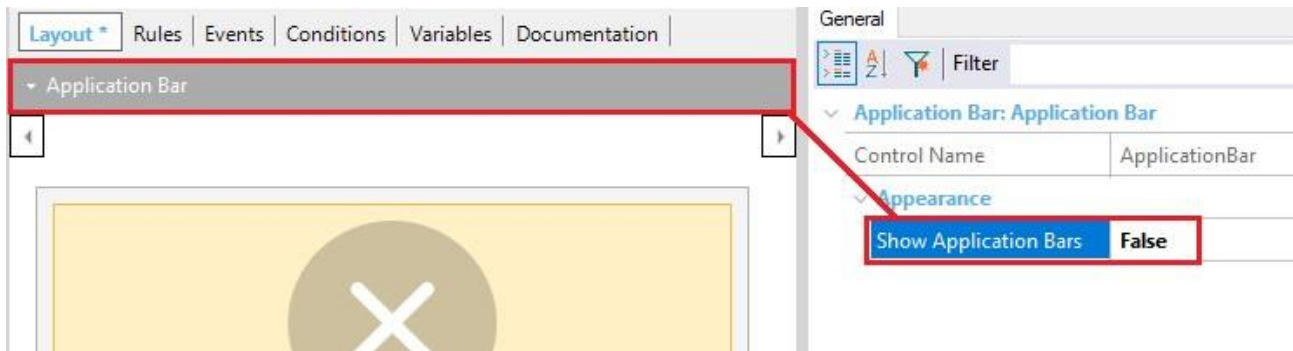


table: main table	
Control Name	MainTable
Appearance	
Columns Style	100%
Rows Style	350dip;35dip;25dip;25dip;pd
Width	100%
Height	100%
Auto Grow	True
Class	TableDetail
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
Form	
Form Class	FormPopup
Enable Header Row Pattern	True
Header Row Application Bars Cla	ApplicationBarsHeaderRow
Layout Behavior	
Expand Bounds	Content & Background
Expand Bounds Limit	Behind System Bars
Expand Bounds Directions	Top

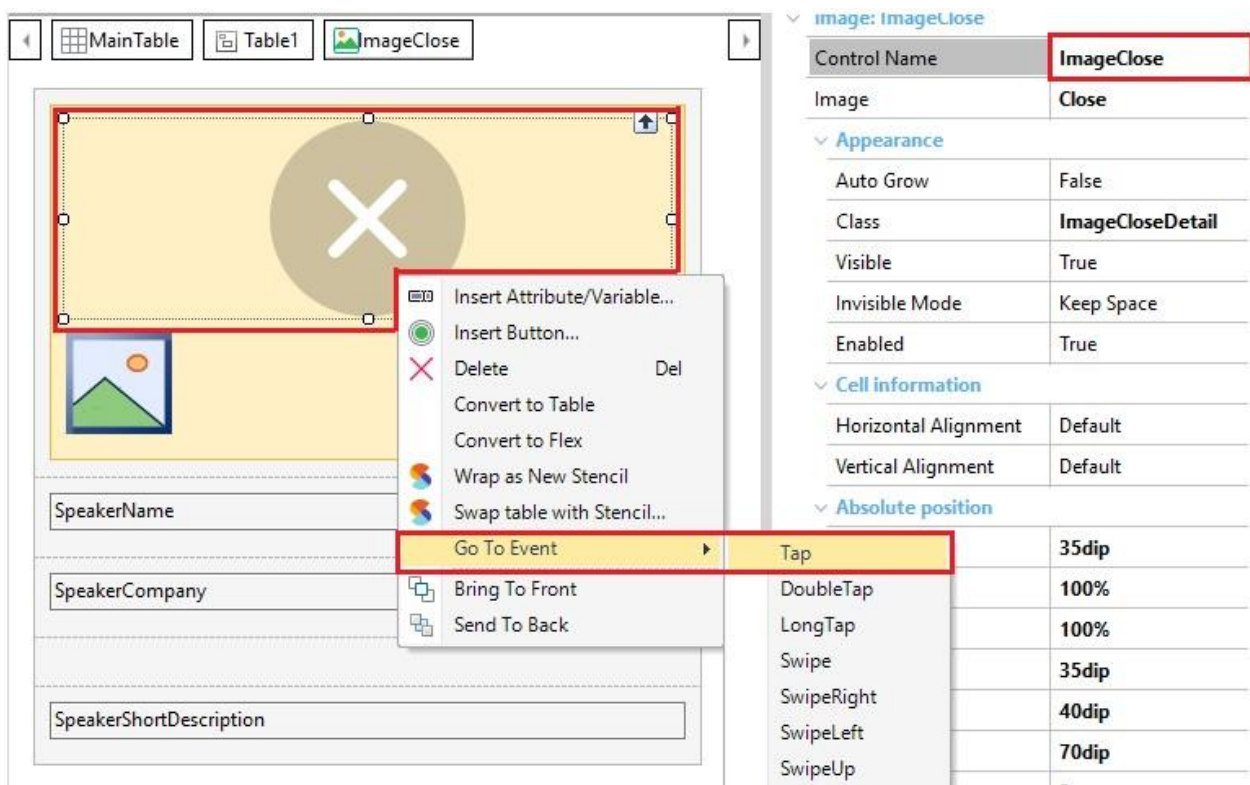
Add the desired sizes for the rows and also assign the “**FormPopup**” class so that the panel when called is displayed as a pop up without an application bar.

Next, in the **canvas** properties, set the “**Height**” property to “350dip” so that it has the space required to display the image.

Click on the application bar of the panel, and set the property “**Show Application Bars = false,**” as shown in the image below:



You have to assign a behavior to the “Close” button that you’ve just placed in the layout. To do so, change its name to “**ImageClose**” and create an event that makes the return to the caller panel, as follows:



Program the desired behavior in the event:

```
Event ImageClose.Tap
    return
Endevent
```

Once finished, set the last “class” properties of the “SpeakerDetail” attributes.

The **“SpeakerName”** attribute will need to have this property assigned:

“Class = AttributeSpeakerDetailName”

The attribute **“SpeakerCompany”** should have the property:

“Class = AttributeSpeakerDetailCompany”

And the attribute **“SpeakerShortDescription,”** the property:

“Class = AttributeSpeakerDetailDescription”.

Finally, you can run the application again to see all the changes at runtime, by selecting **“Build”** and reading the QR code again.

THANK YOU FOR PARTICIPATING!

Glossary

Live Editing

<https://wiki.genexus.com/commwiki/servlet/wiki?27806,Live+Editing+in+Smart+Devices+applications>,

Menu for Smart Devices

<https://wiki.genexus.com/commwiki/servlet/wiki?16321,Category%3AMenu+for+Smart+Devices+object>

Canvas control

<https://wiki.genexus.com/commwiki/servlet/wiki?22452,Canvas%20control>

Stencils

<https://wiki.genexus.com/commwiki/servlet/wiki?38418,Category%3AStencil+object>,