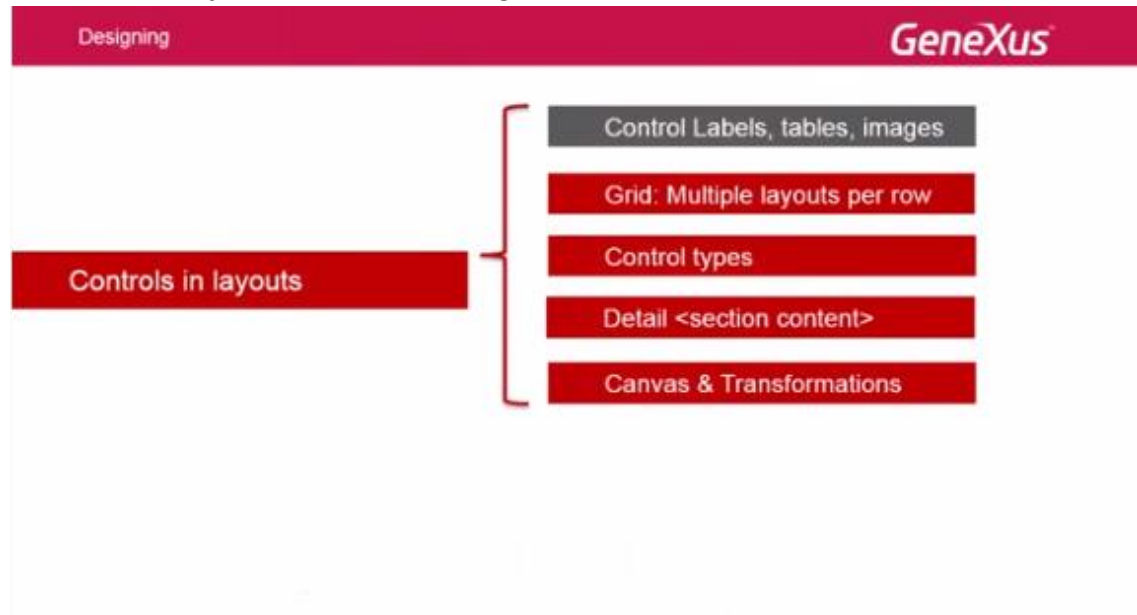


**Controles: múltiplos layouts por linha em um grid.**

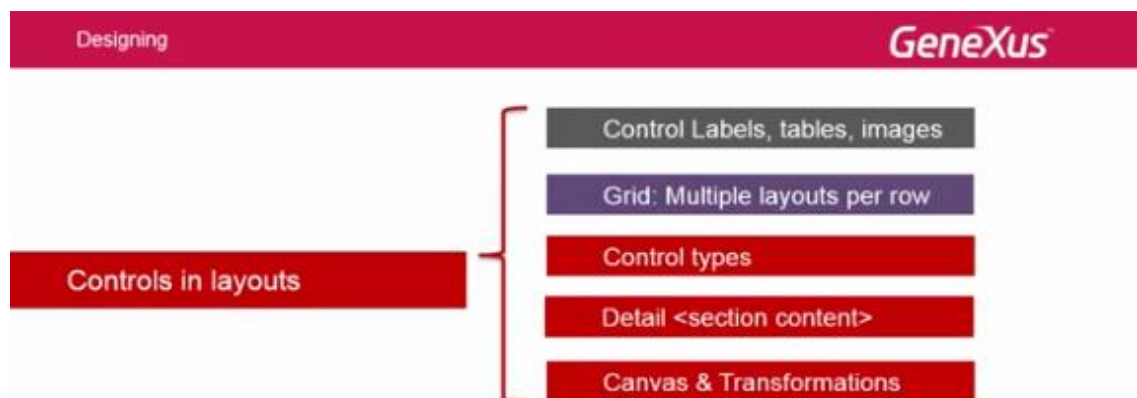


No vídeo anterior havíamos começado a trabalhar as particularidades dos controles nos layouts dos objetos Smart Devices em relação ao uso conhecido. Tínhamos visto o uso das

labels e a utilização de tabelas e das imagens.

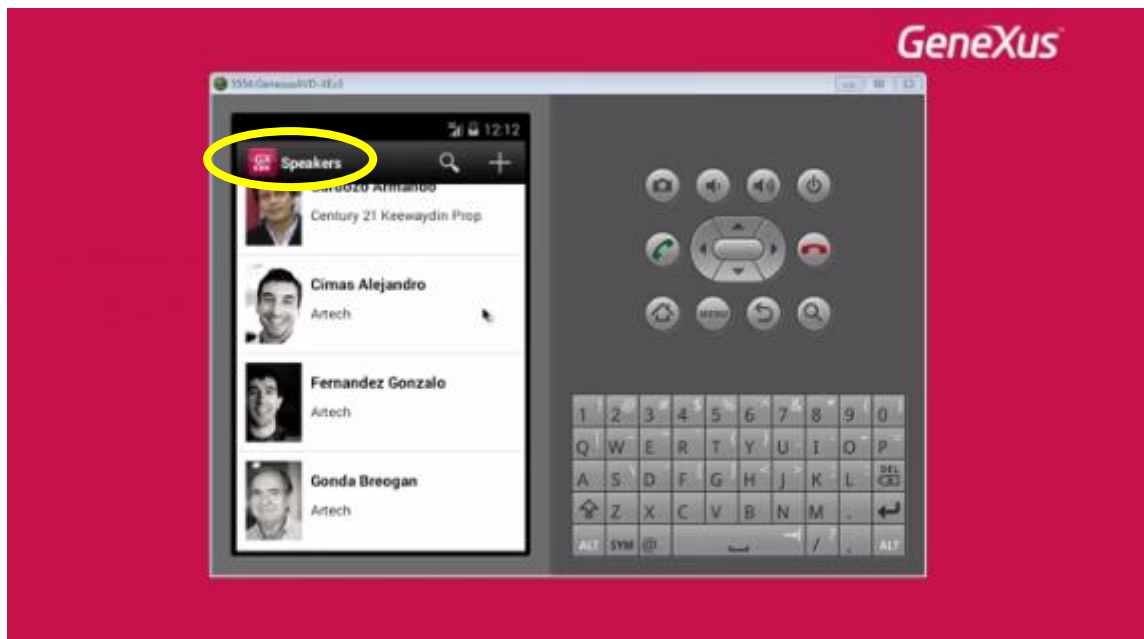


Agora estudaremos o controle do grid em um aspecto: a possibilidade de variar o desenho de cada linha, para que dada oportunidade possa assumir um desenho diferente entre vários possíveis.



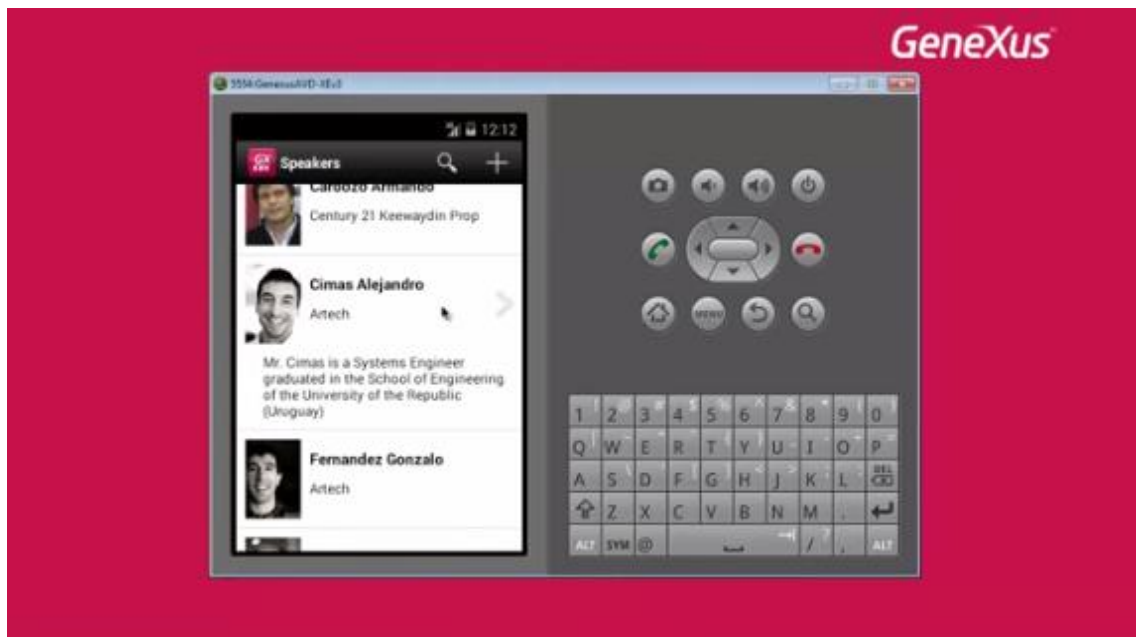
---

Veremos com um exemplo.

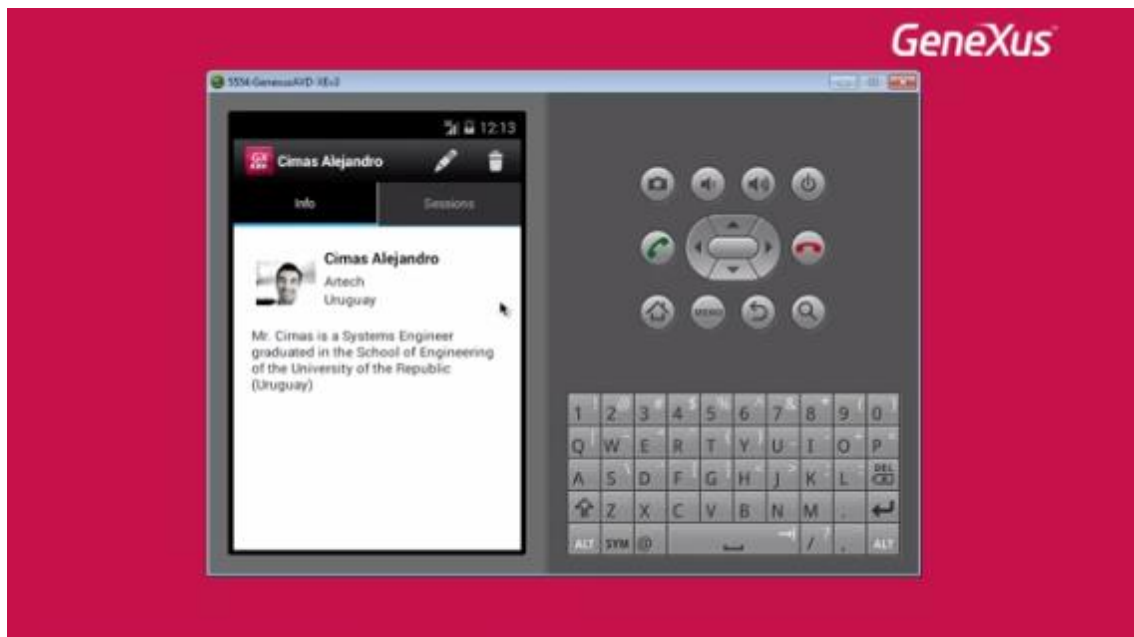


Aqui temos uma aplicação final que está no GXserver. Aquela qual nos aproximaremos.

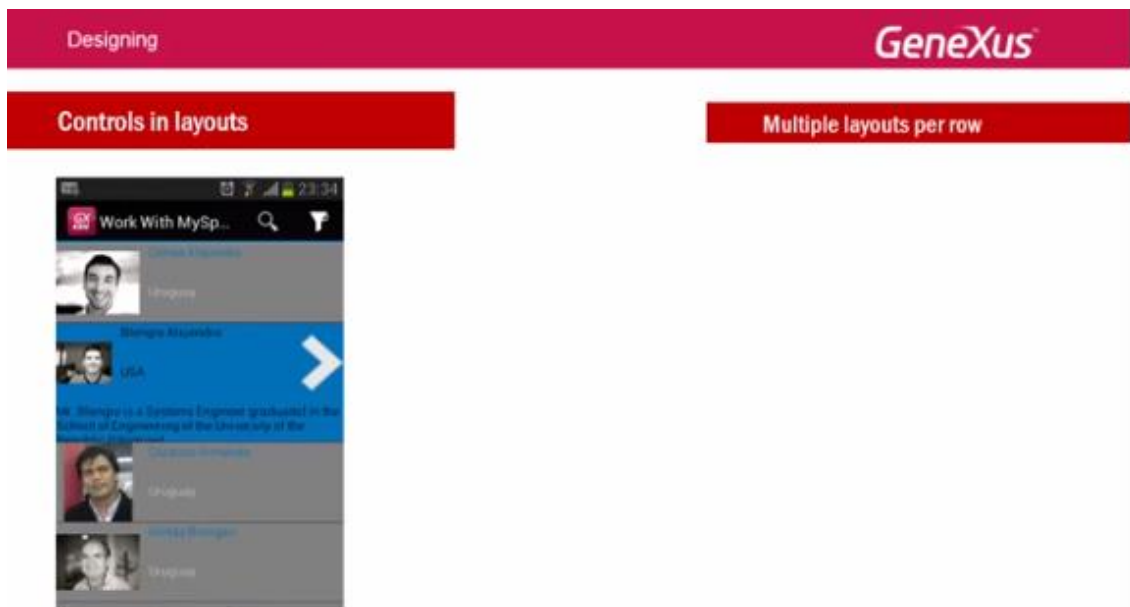
Temos o List de speakers, com seu grid. Queremos que ao selecionar 1 speaker do grid fazendo um TAP



No lugar chamar a Detail, mostre uma parte do curriculum vitae e uma seta para poder escolher ver dando um TAP sobre a mesma.



agora sim a informação detalhada de um conferencista.



Portanto, necessitamos para cada item ou linha do grid 2 layouts.

Um que valha quando a linha não está selecionada:



Designing

GeneXus

Controls in layouts

Multiple layouts per row

E outro quando essa está:

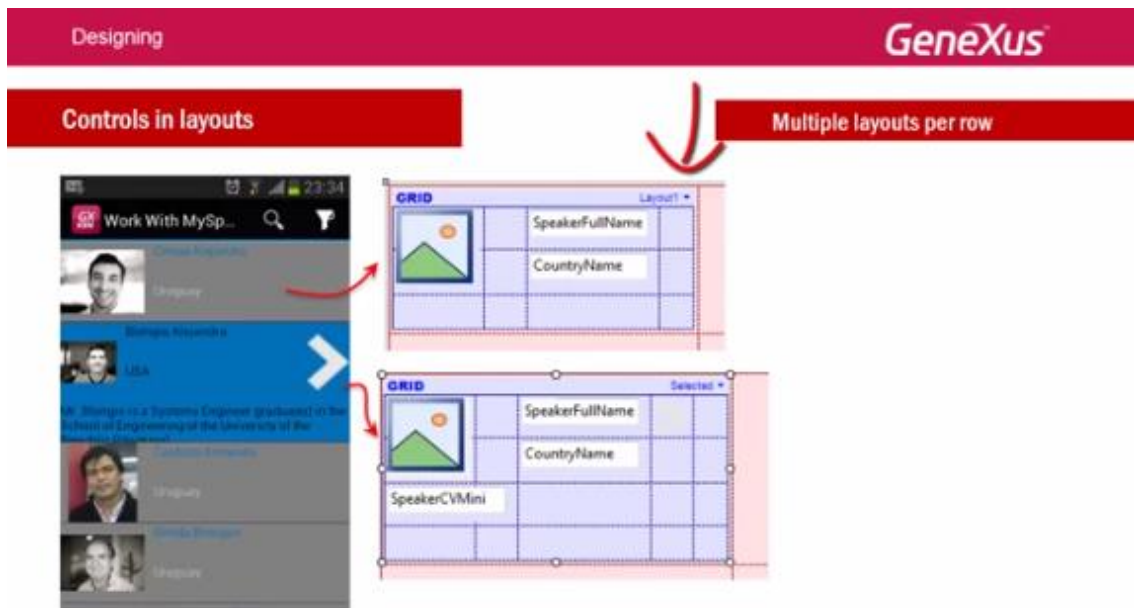
Designing

GeneXus

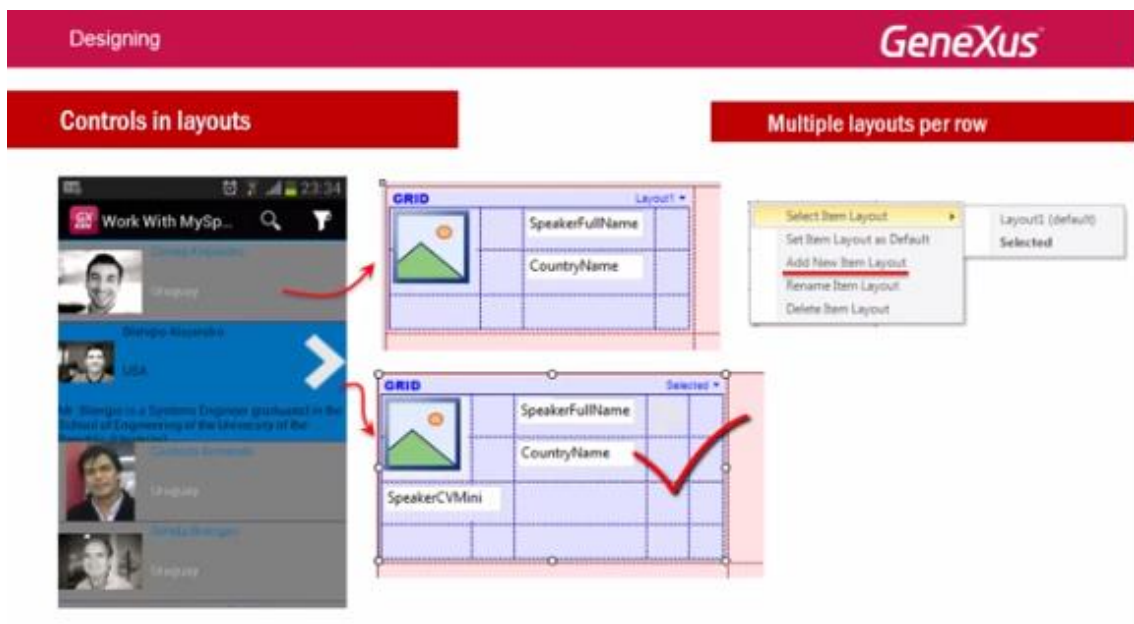
Controls in layouts

Multiple layouts per row

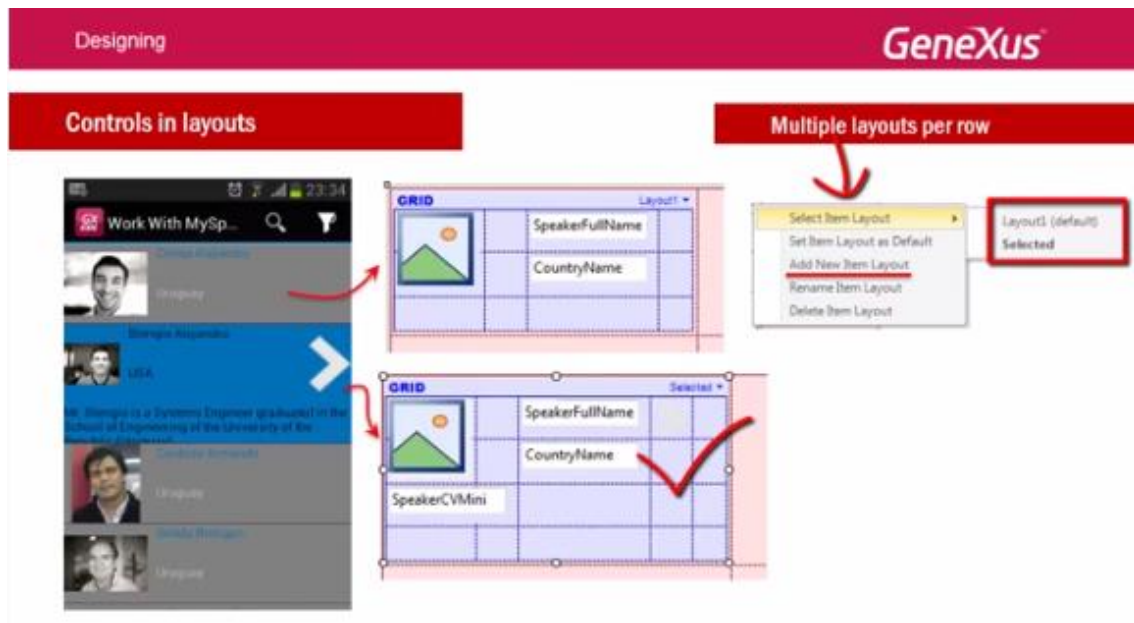
Para isso, no canto superior direito do grid



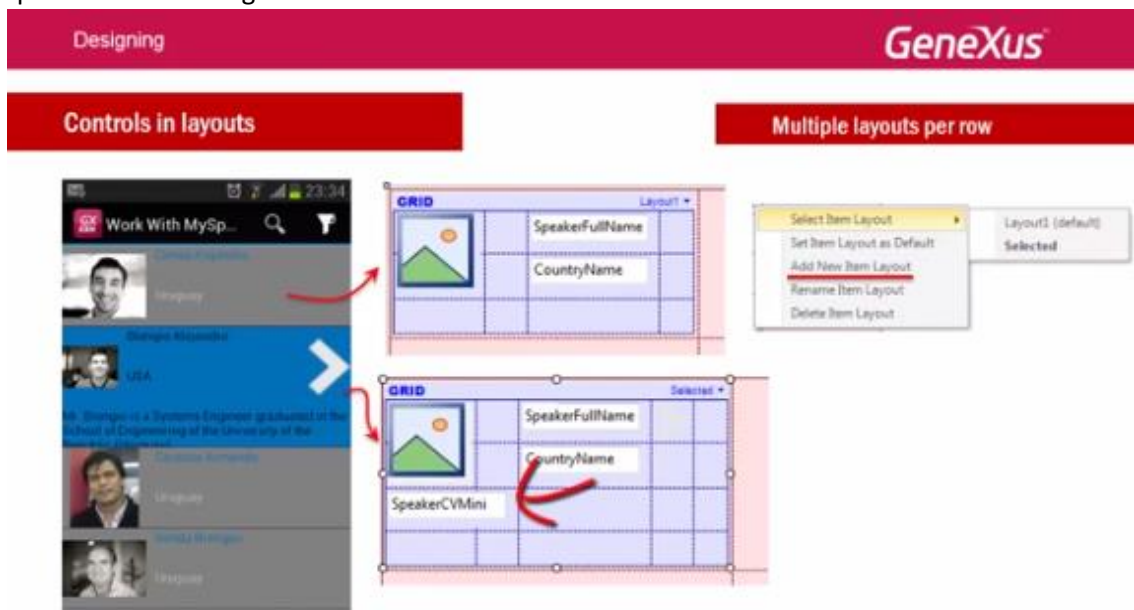
Se desprega um menu contextual que nos permite entre outras coisas, **agregar um novo layout para o ítem.**



Fazendo-o, a primeira opção do menú nos permite seleccionar o layout que queremos editar de cada vez.

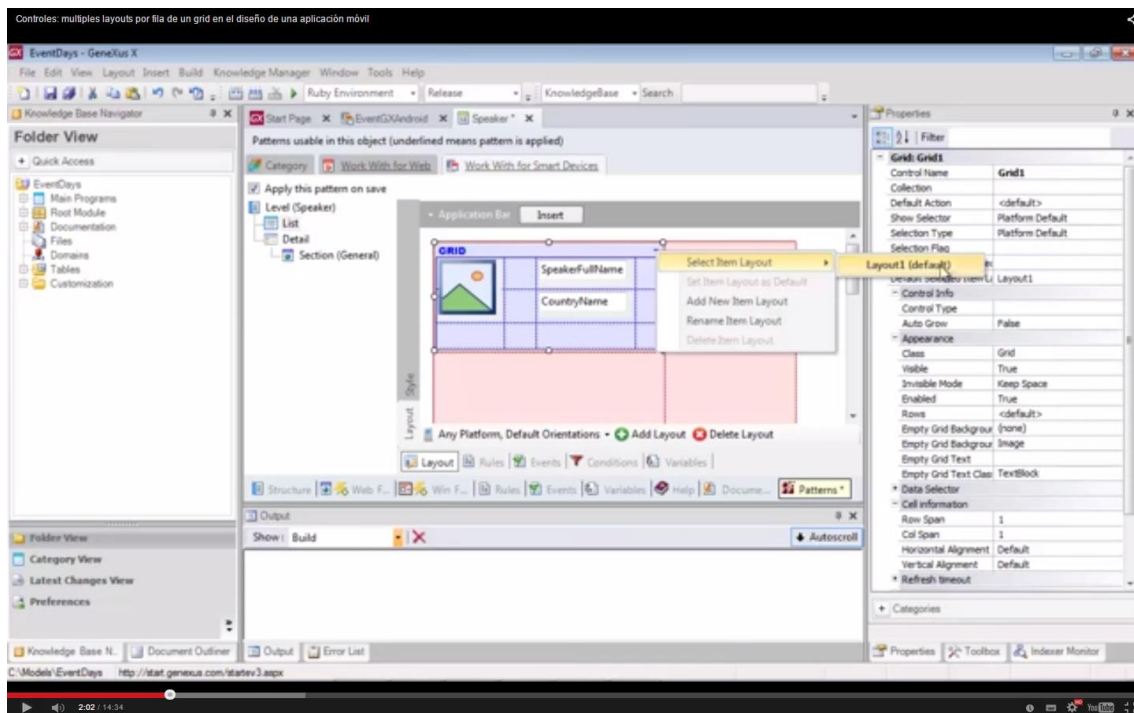


Aqui, no layout default, nós adicionamos um segundo layout ao qual damos o nome: **Selected**, que inclui uma imagem e o currículo vitae do orador.

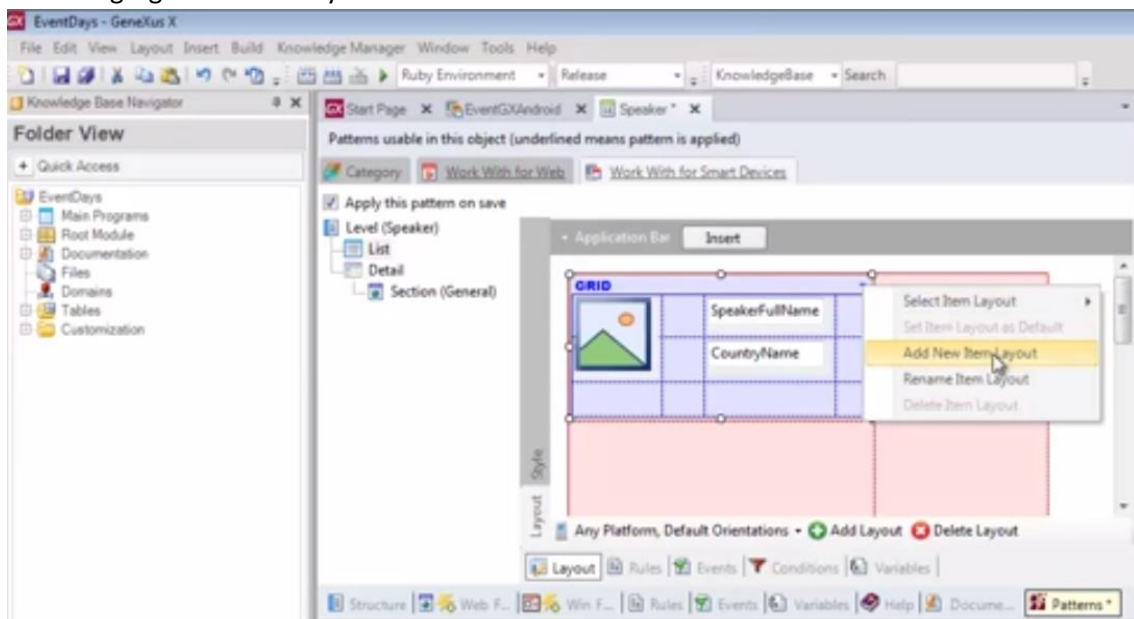


Façamos no Genexus.

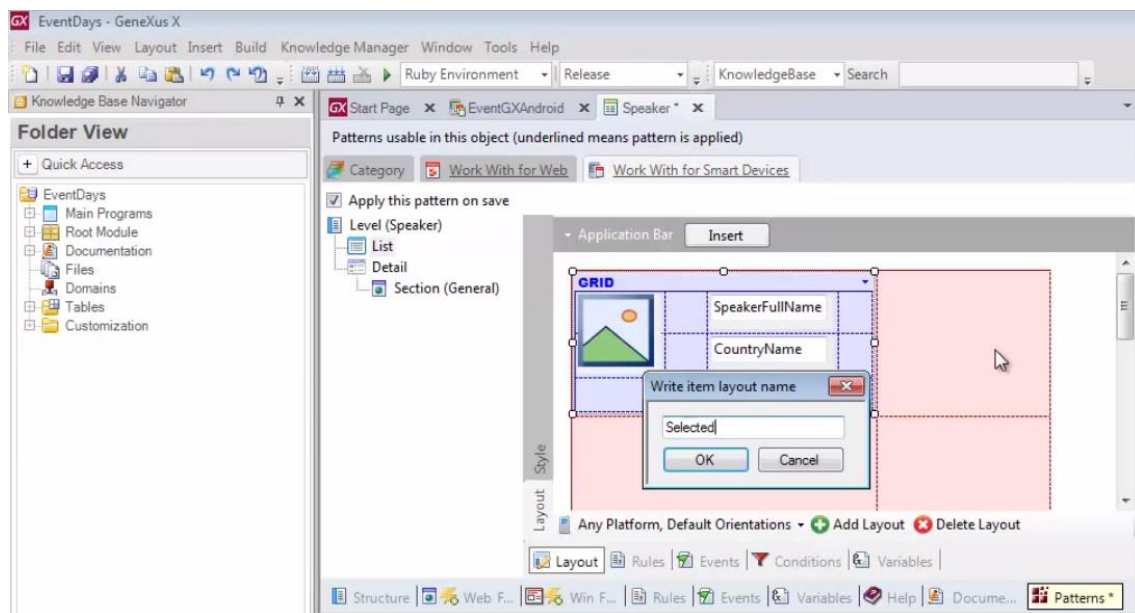
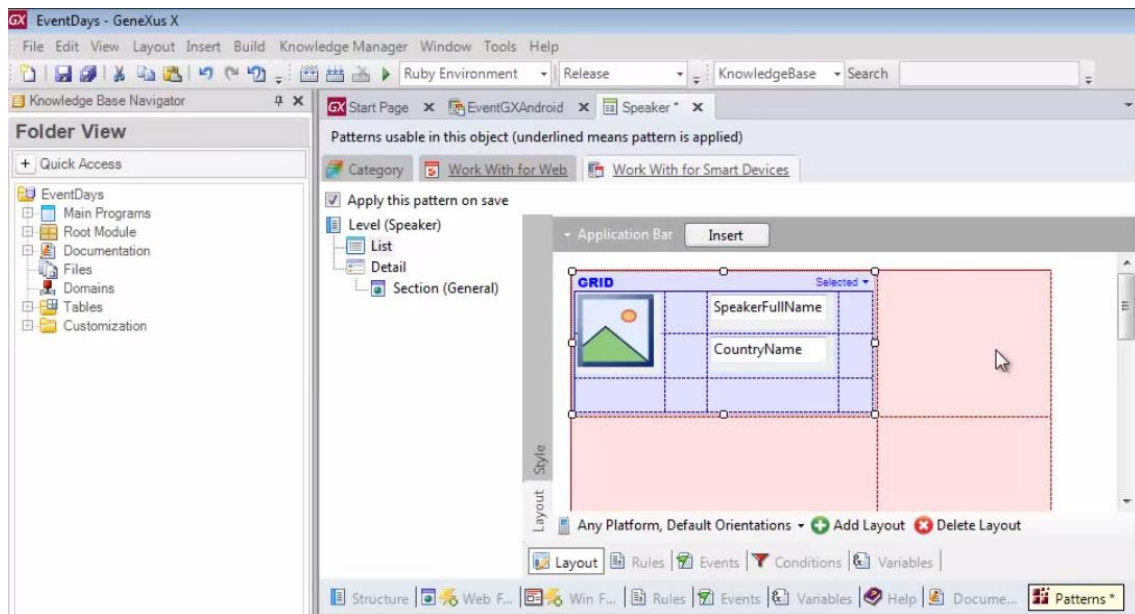
Clicamos e vemos que somente temos o Layout default.



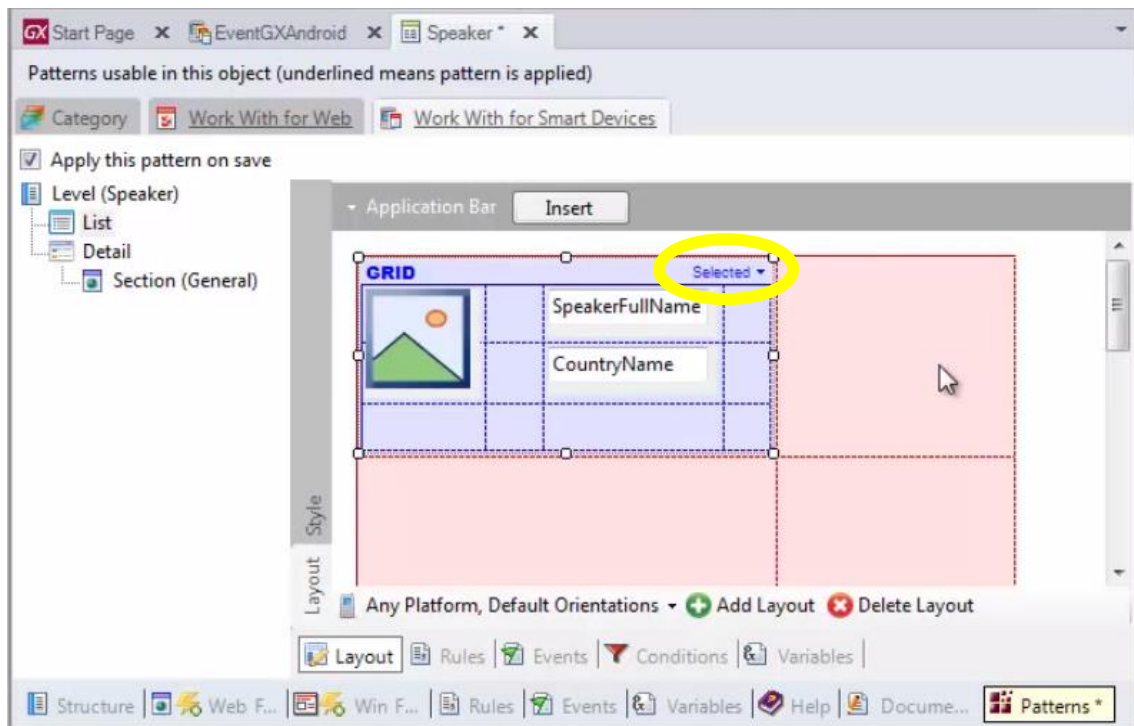
Vamos agregar um novo layout



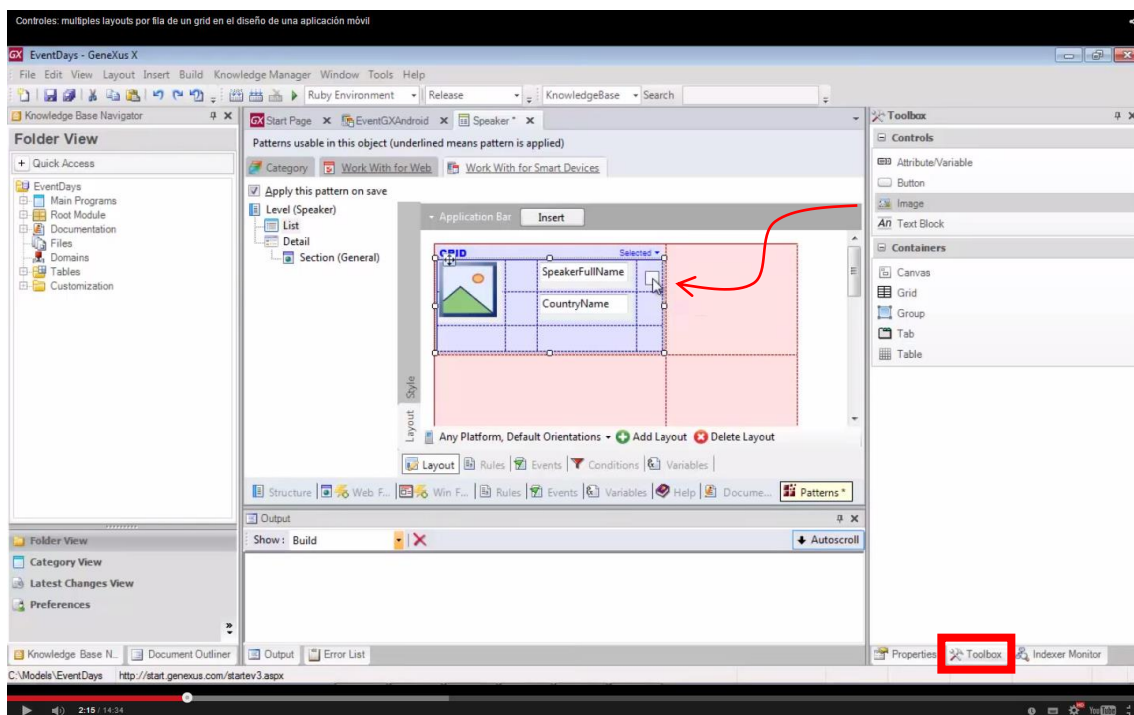
A que chamaremos: Selected

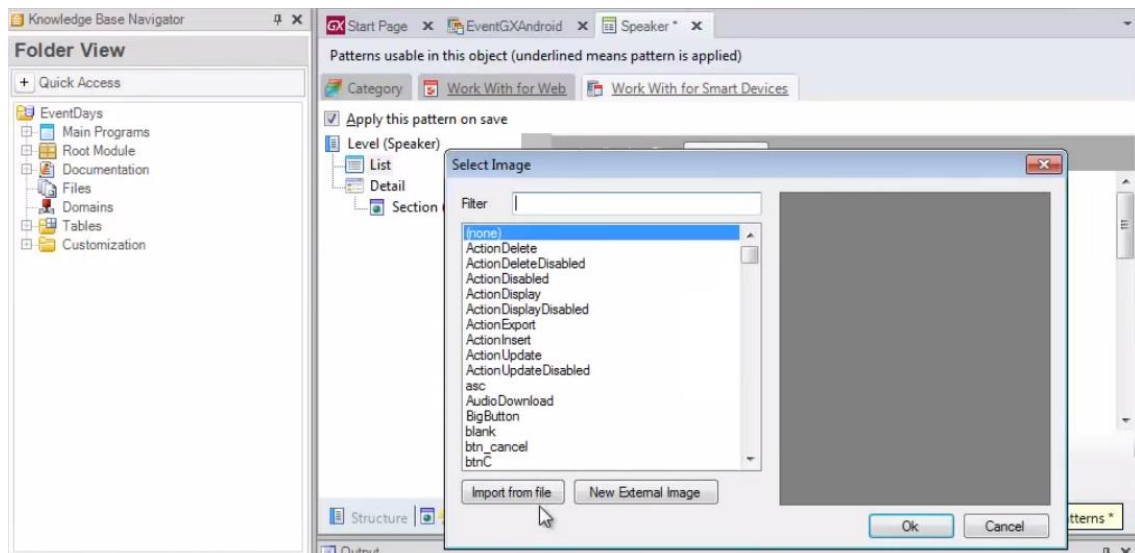




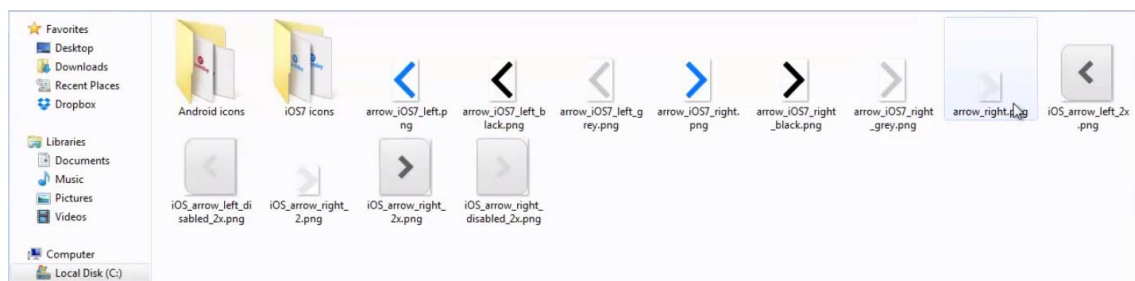


Adicionamos a imagem

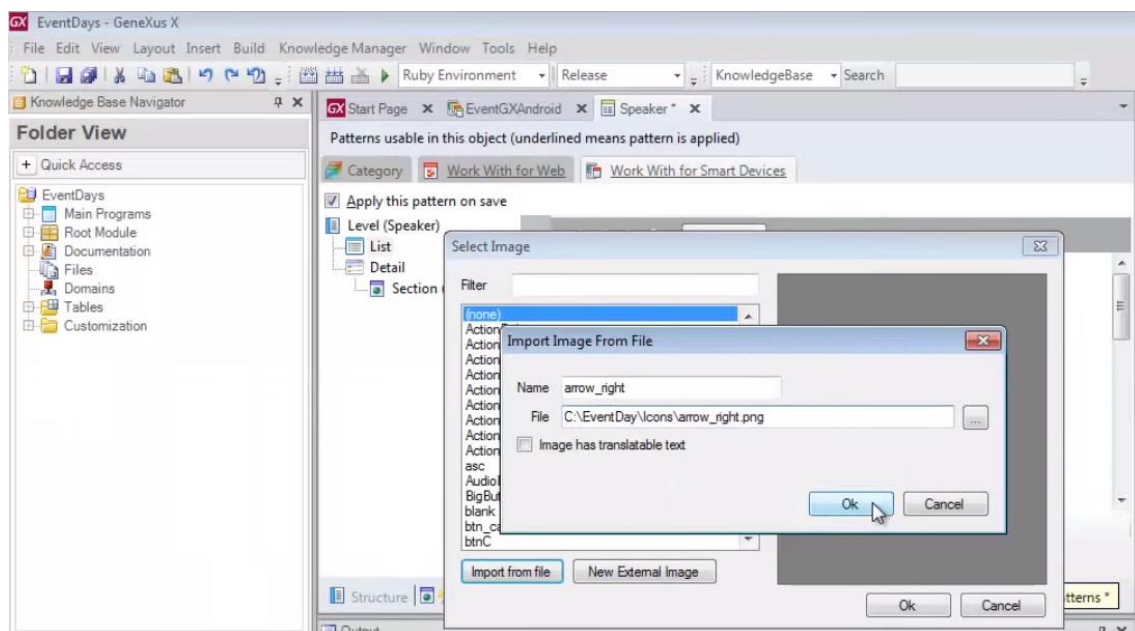


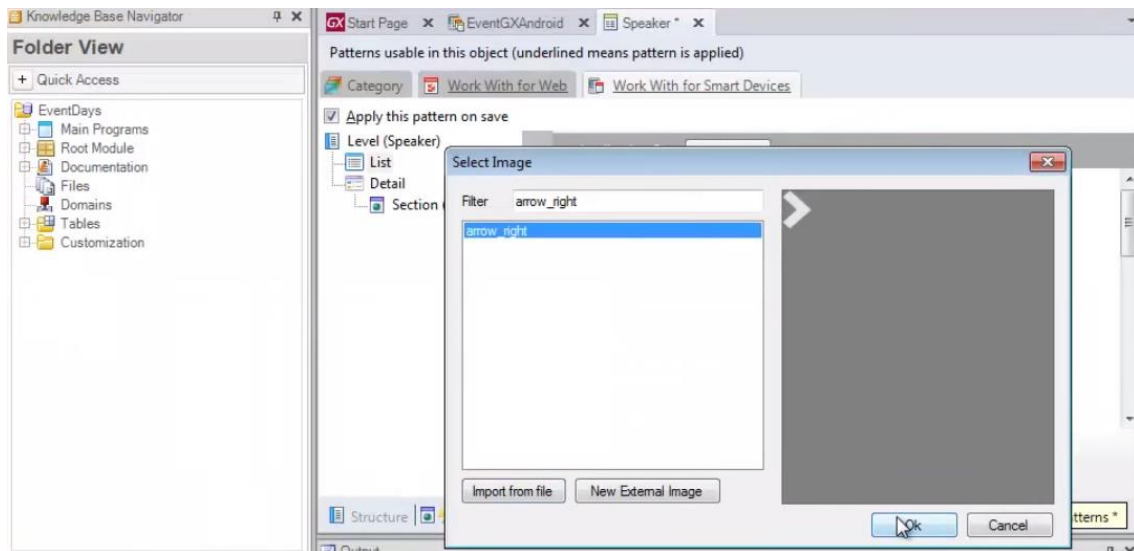


Importando-a de um arquivo.



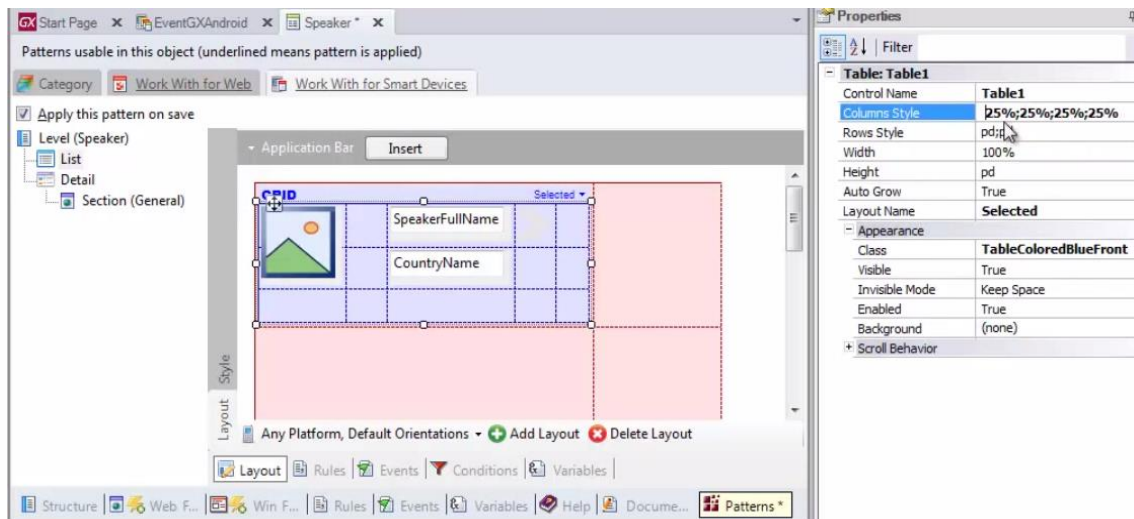
Arrow\_right





Ok

Veamos o espaço que queremos que ocupe a coluna.



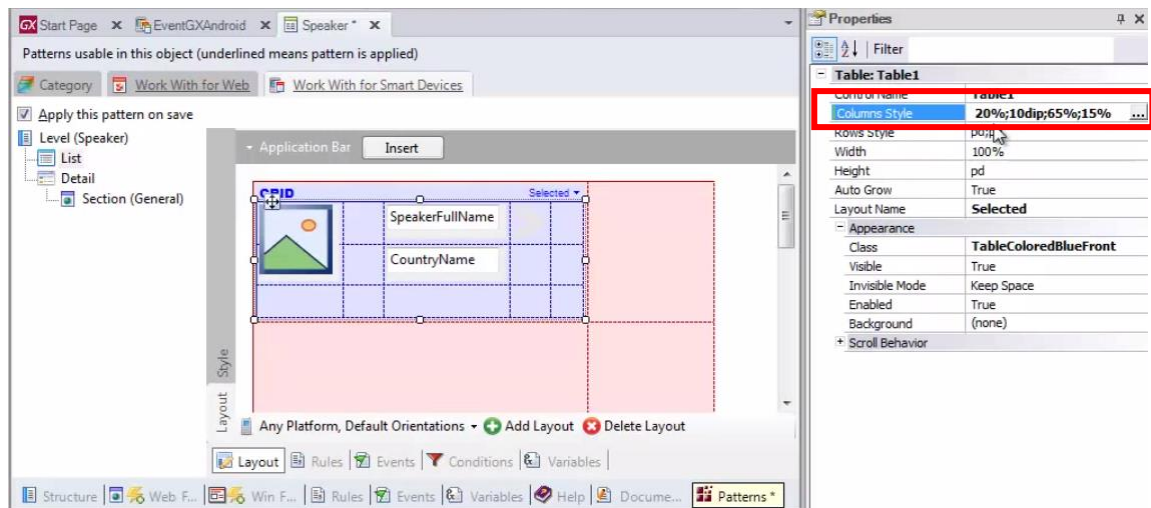
A primeira: 20%

A segunda que é simplesmente para deixar espaço: 10 DIPS

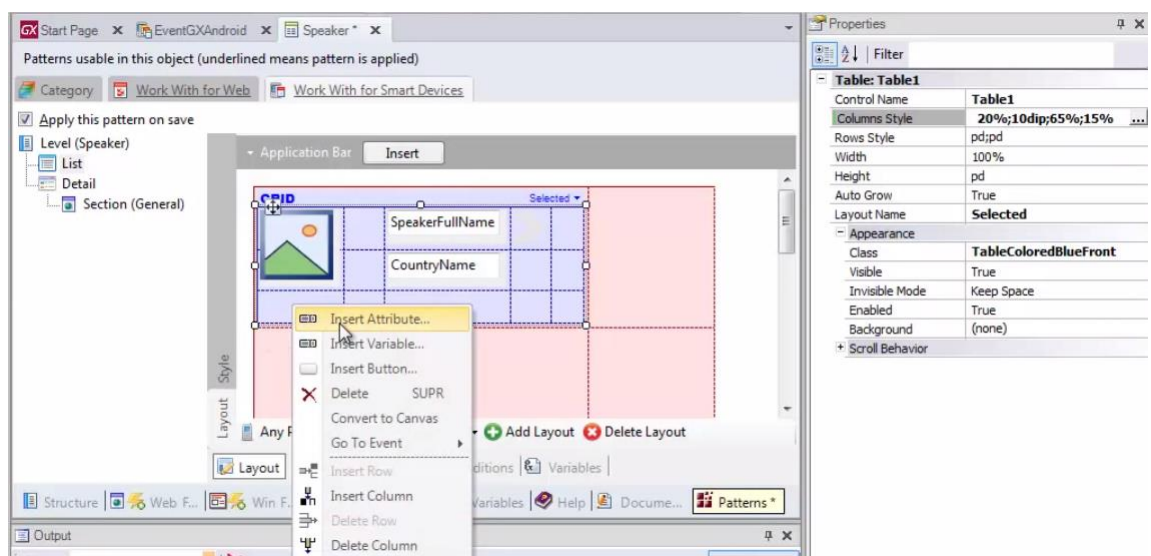
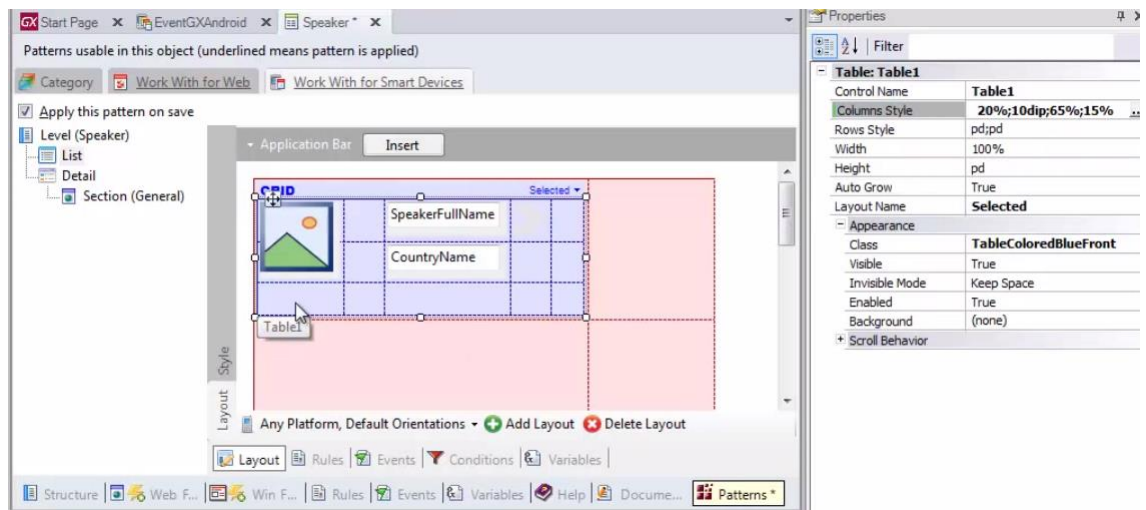
A terceira: 65%

E a última: somente 15% restante (a da imagem)

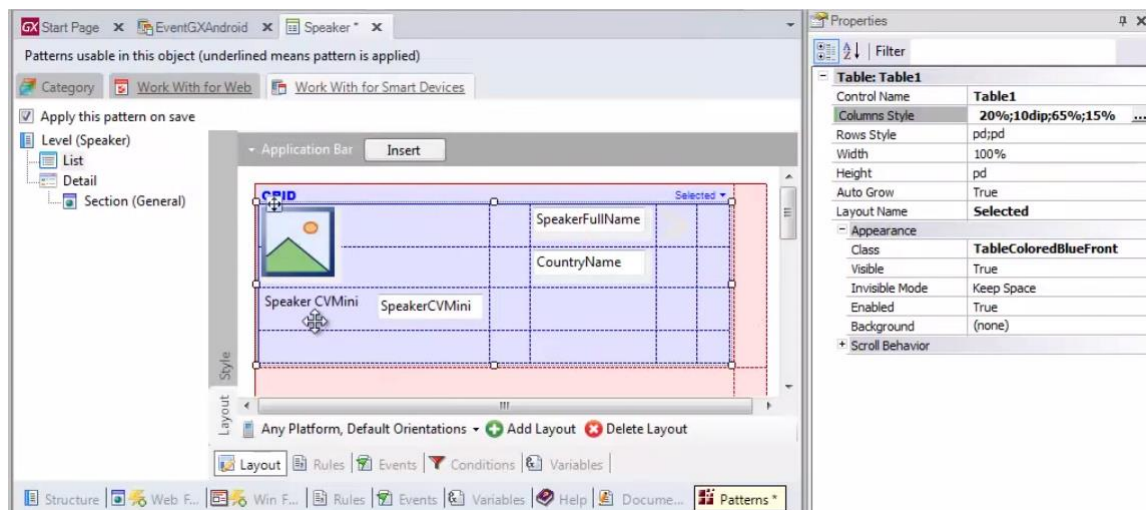
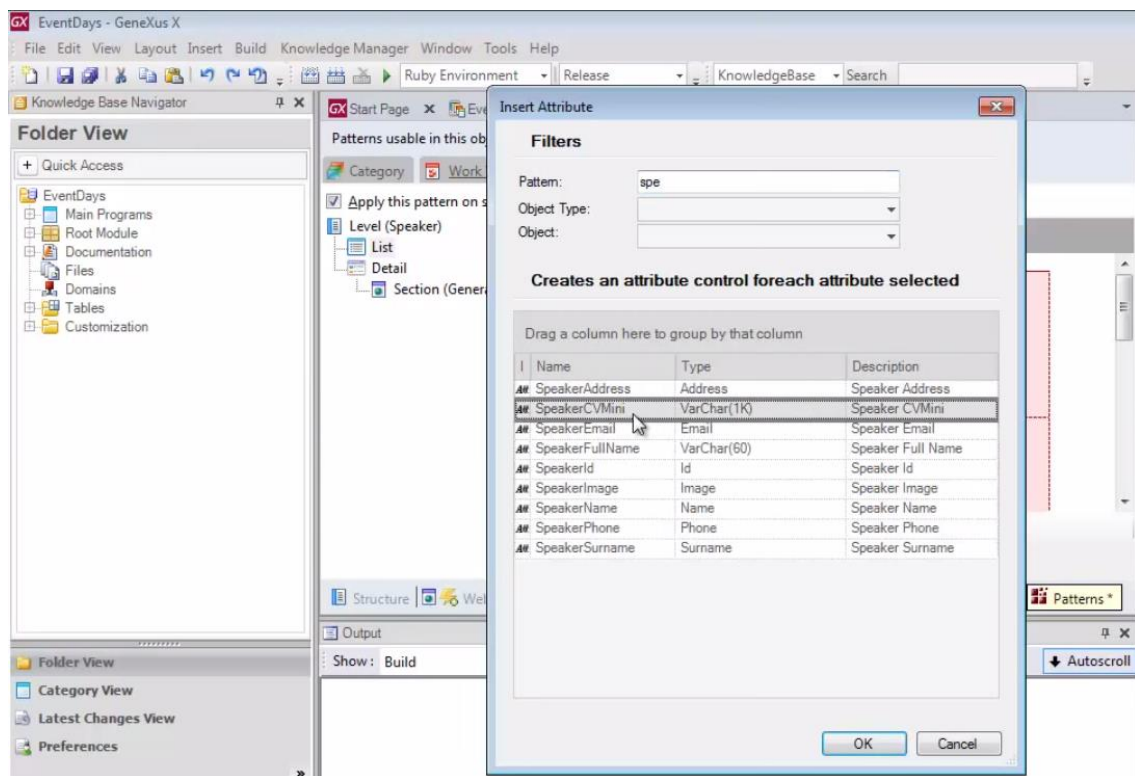




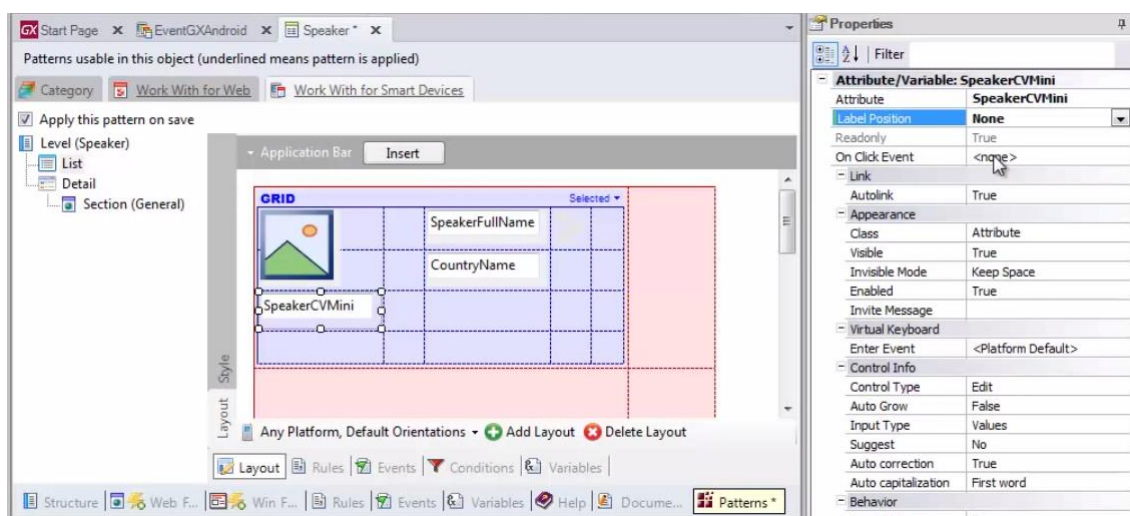
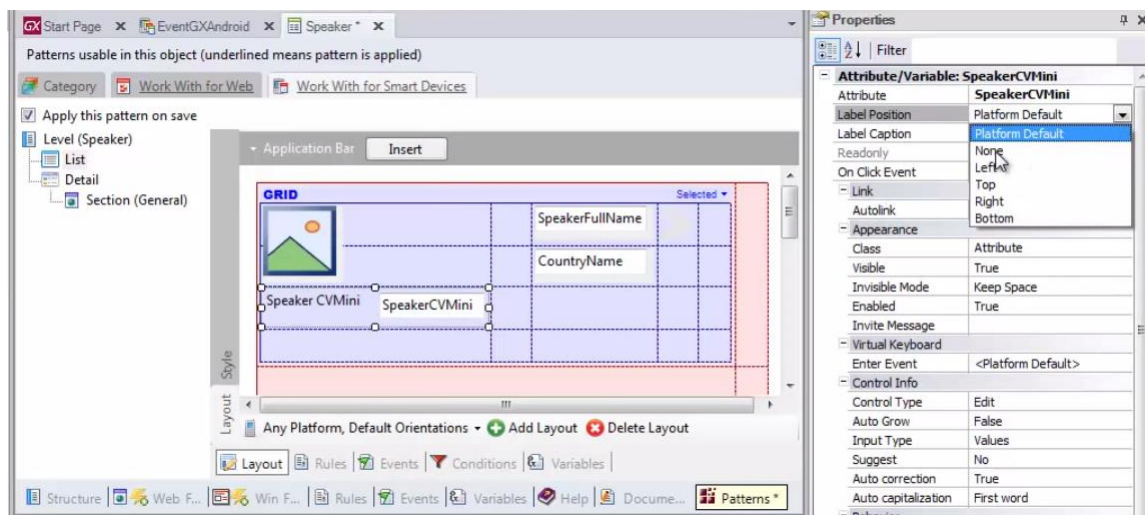
Agregamos agora o curriculum vitae



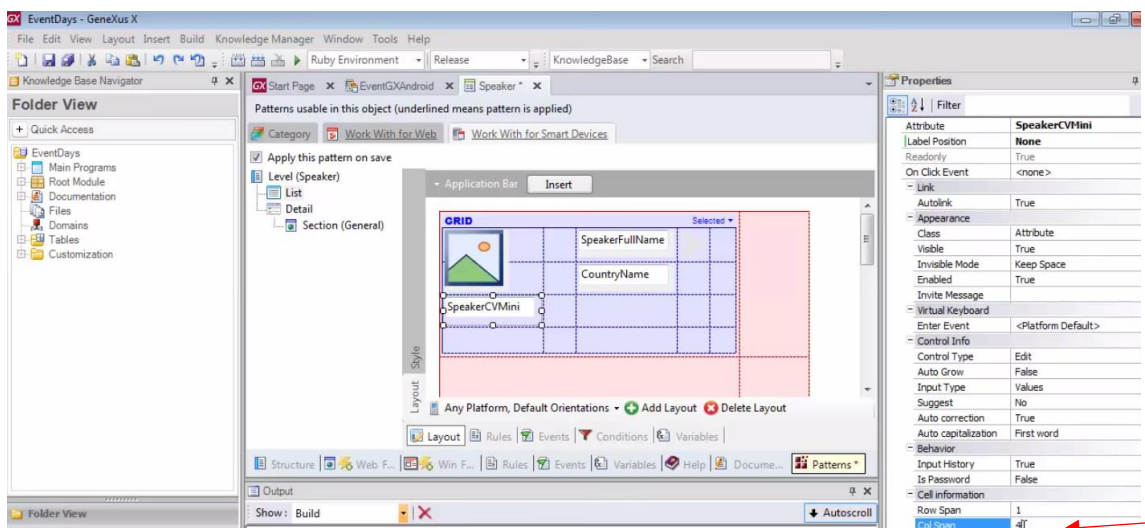
Inserindo o atributo..



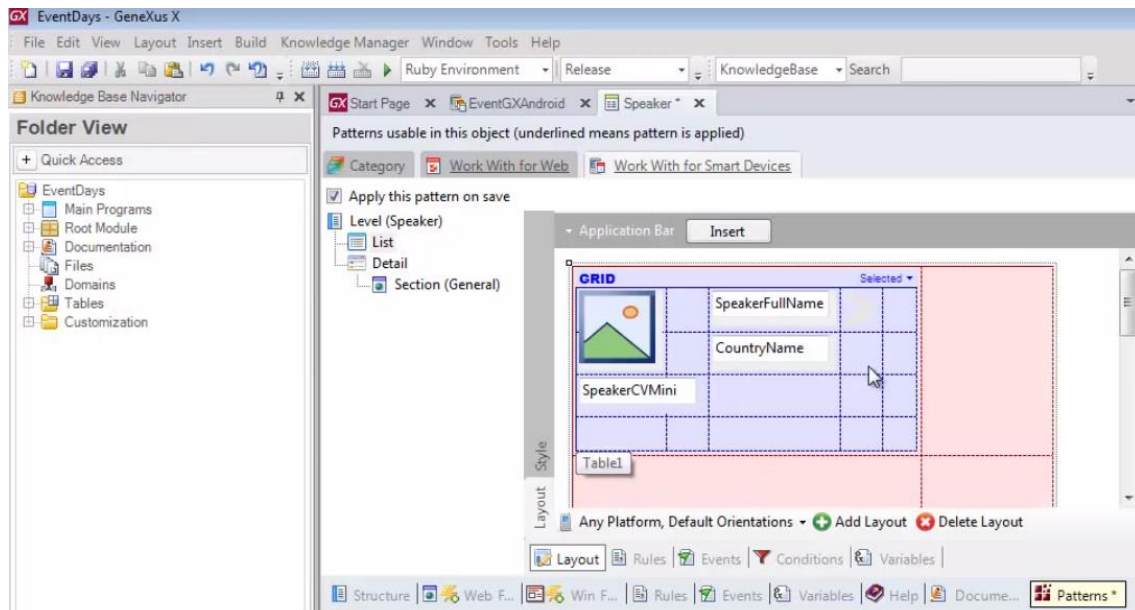
Vamos remover a etiqueta



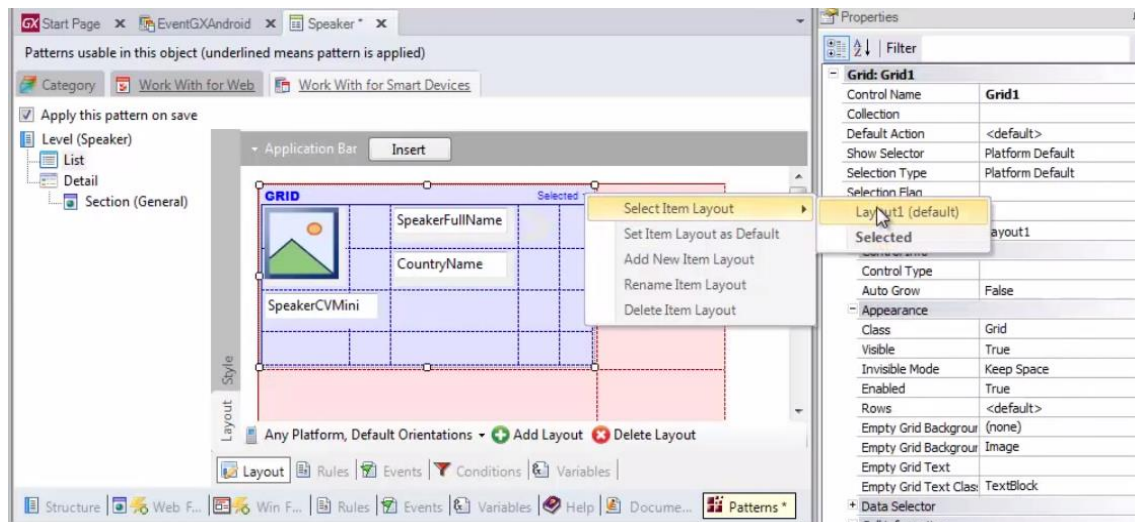
E vamos permitir-lhe expandir-se ao longo das 4 colunas.







Se agora queremos editar um outro layout, o default, vamos aqui.

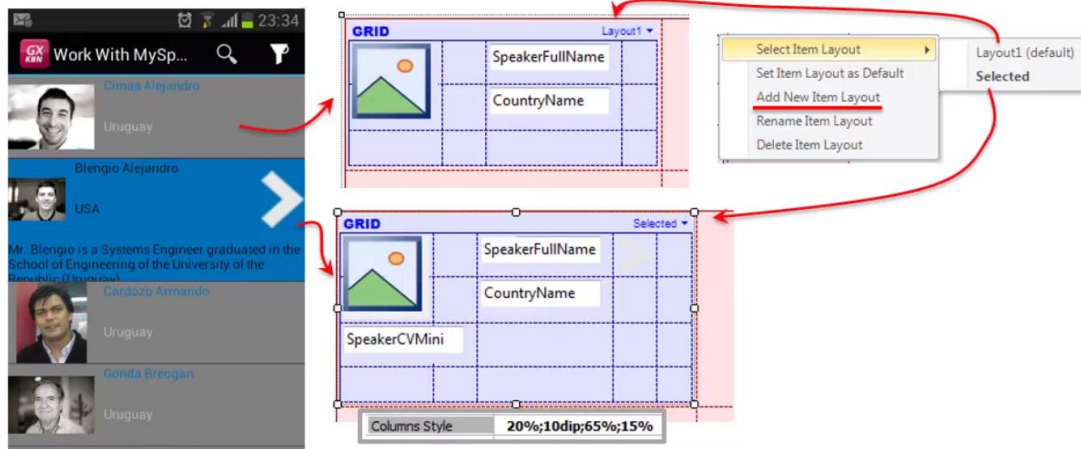


E vemos como teremos então as 2 opções. Em **negrito (bold)** a que estamos editando no momento.. e com isto mudamos para o outro layout

Até aqui, definimos os 2 layouts

## Controls in layouts

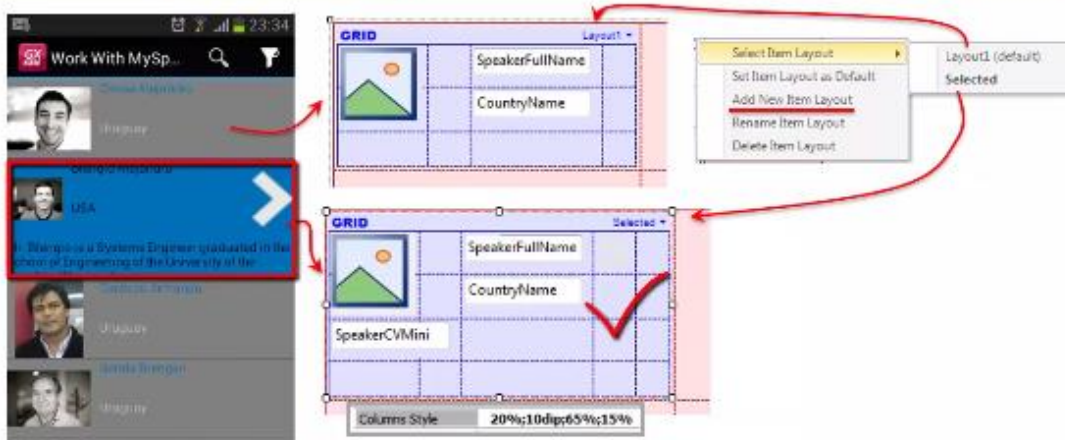
## Multiple layouts per row



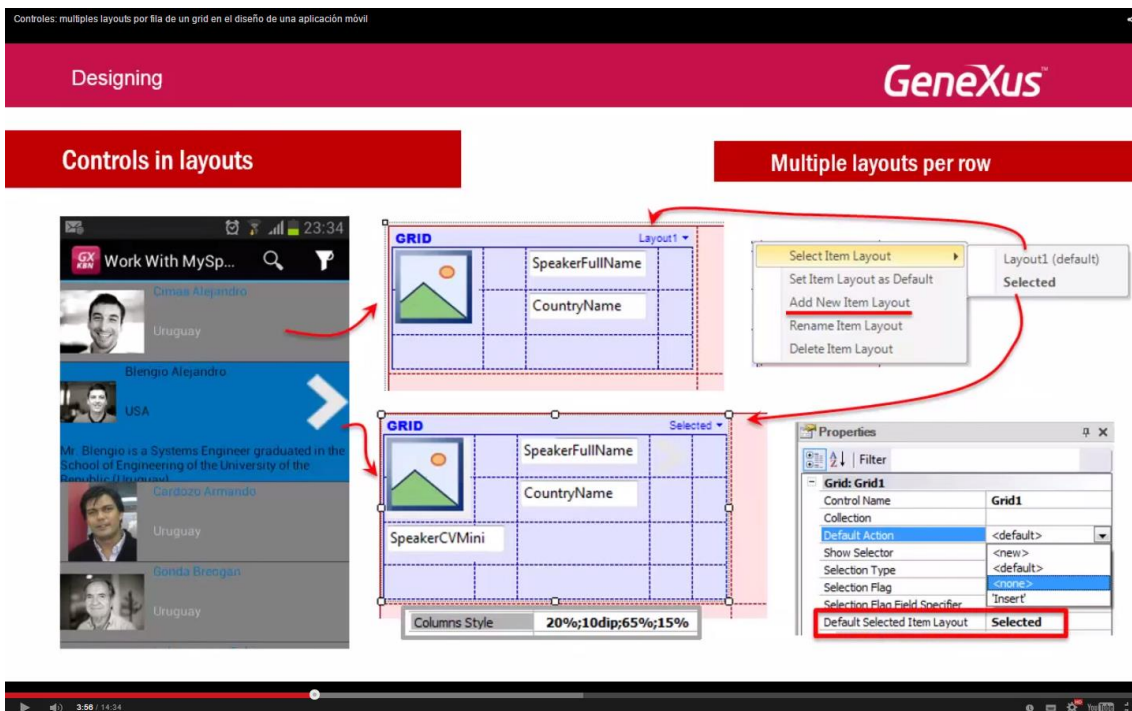
Agora temos que fazer que quando a linha esteja selecionada, use este segundo layout (Selected) e não o primeiro:

## Controls in layouts

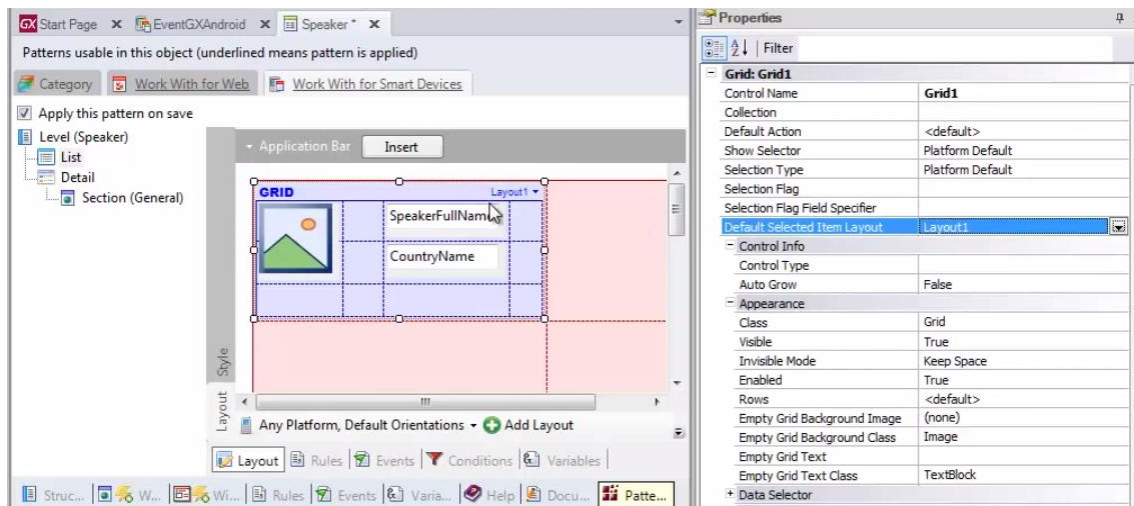
## Multiple layouts per row



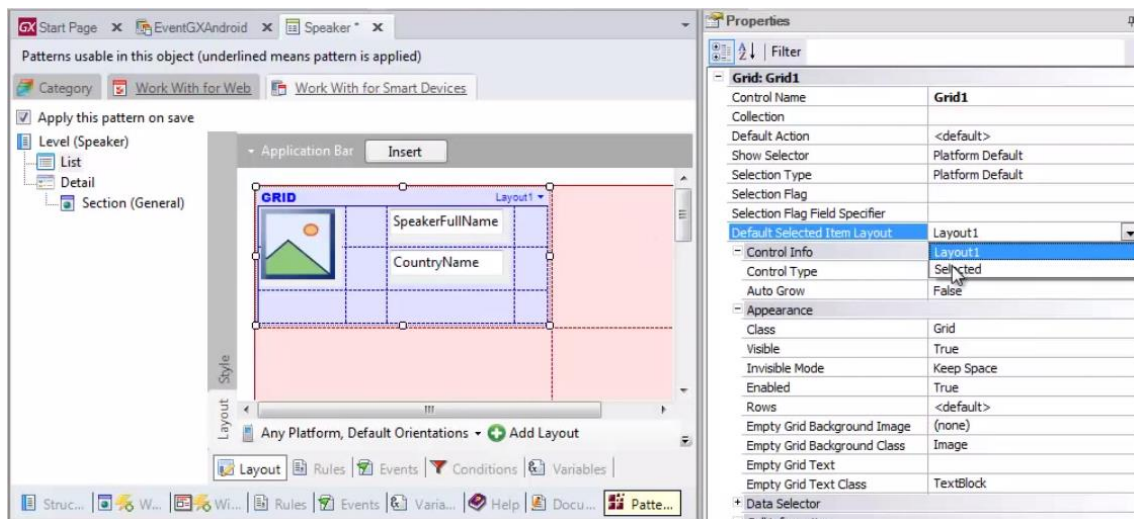
Isto é indicado com a propriedade do grid **Default Selected Item Layout...**



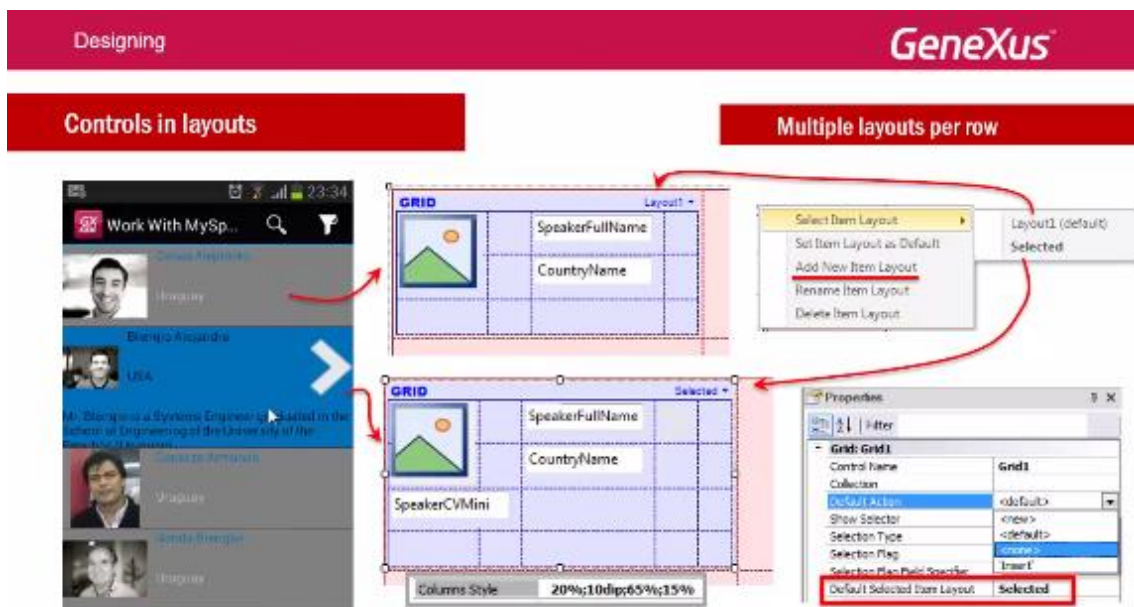
como vemos, por padr o est  con el valor: Layout1



E vamos mudar o novo layout

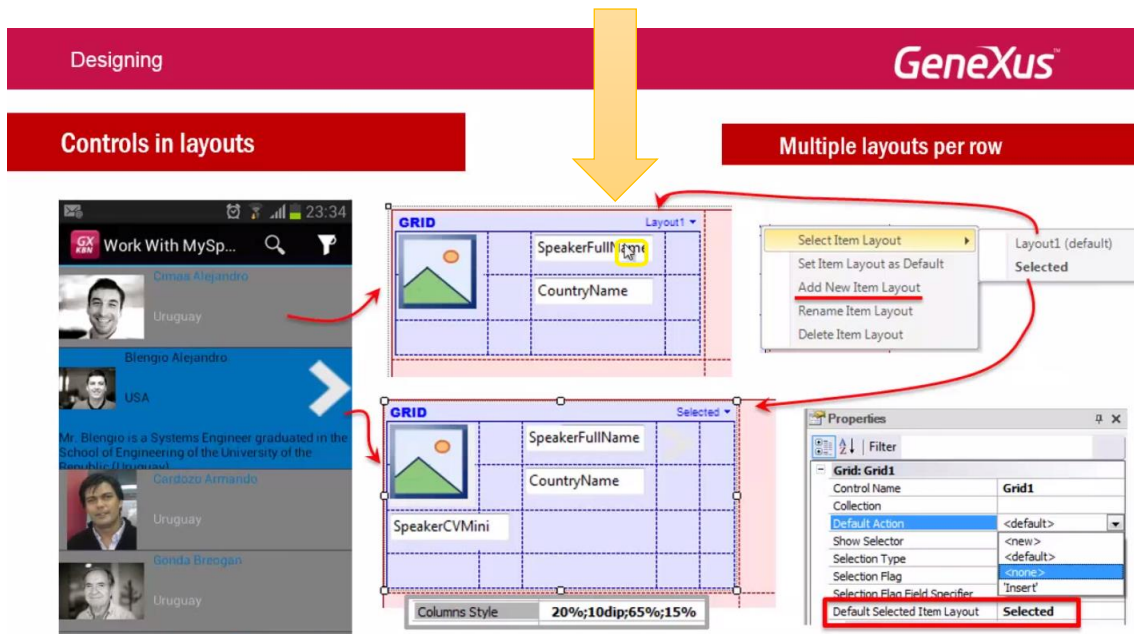


Desta forma então estamos indicando que cada vez que selecionamos uma linha do grid

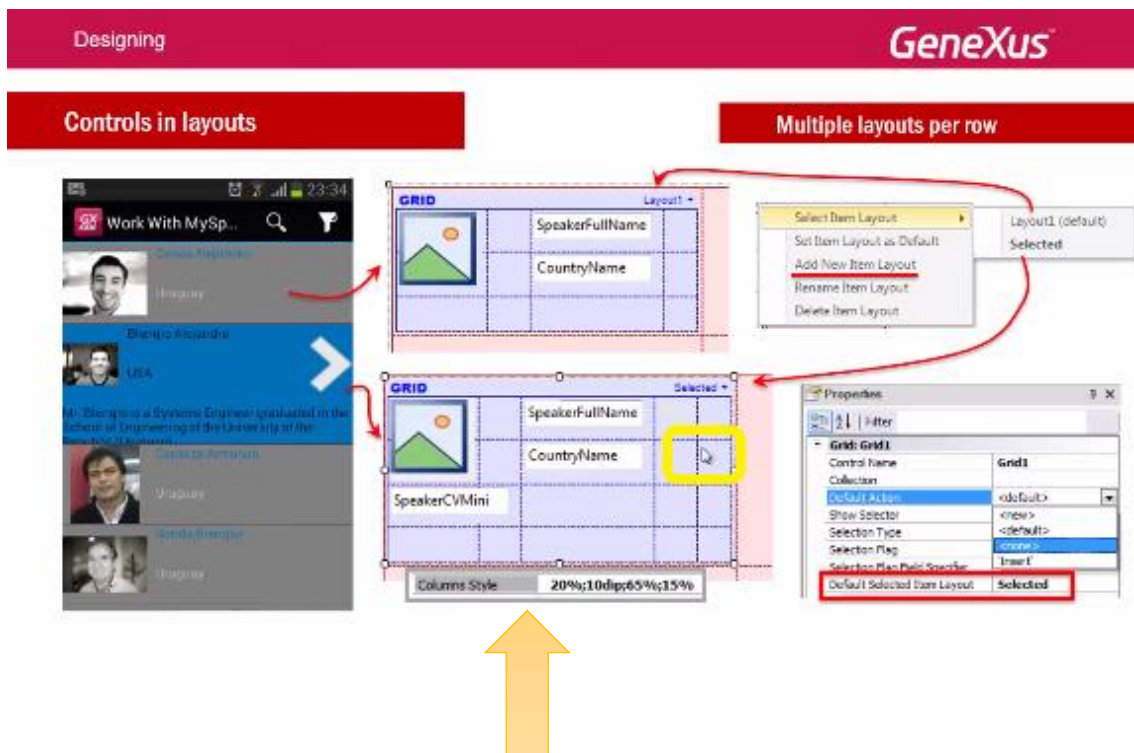


No lugar de utilizar este layout:





Se utilize este outro:

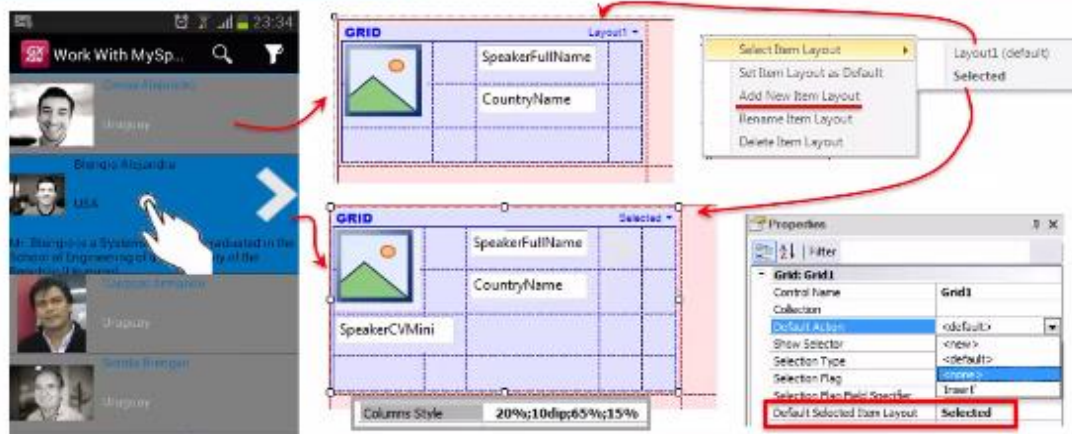


Mas... quando damos um TAP sobre a linha para selecioná-la:



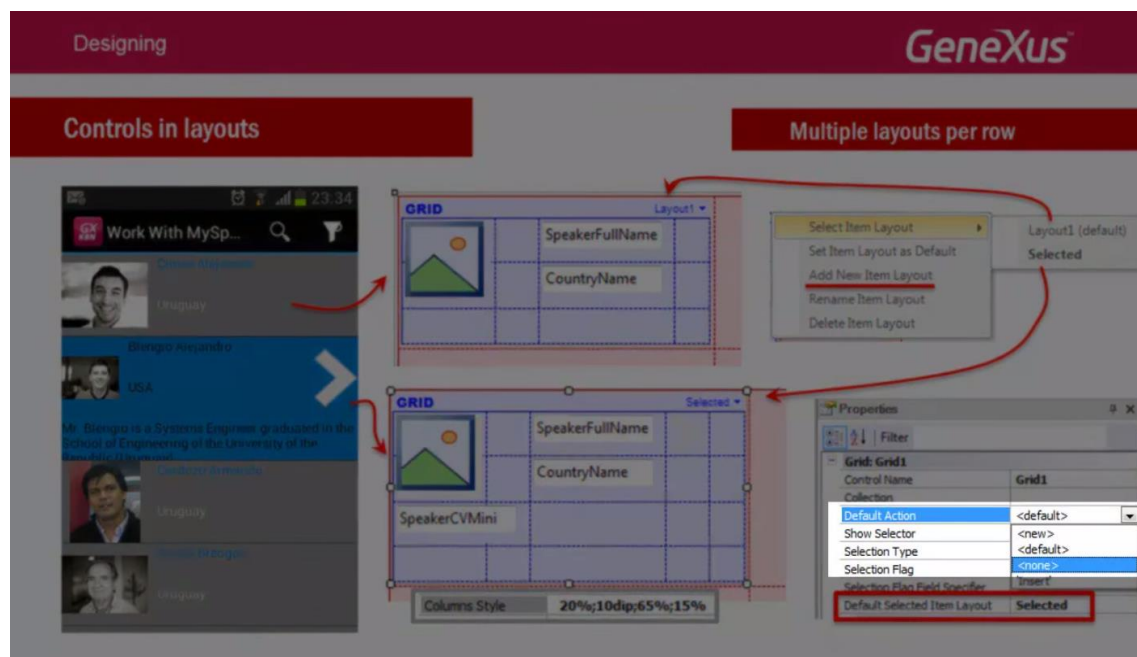
## Controls in layouts

## Multiple layouts per row



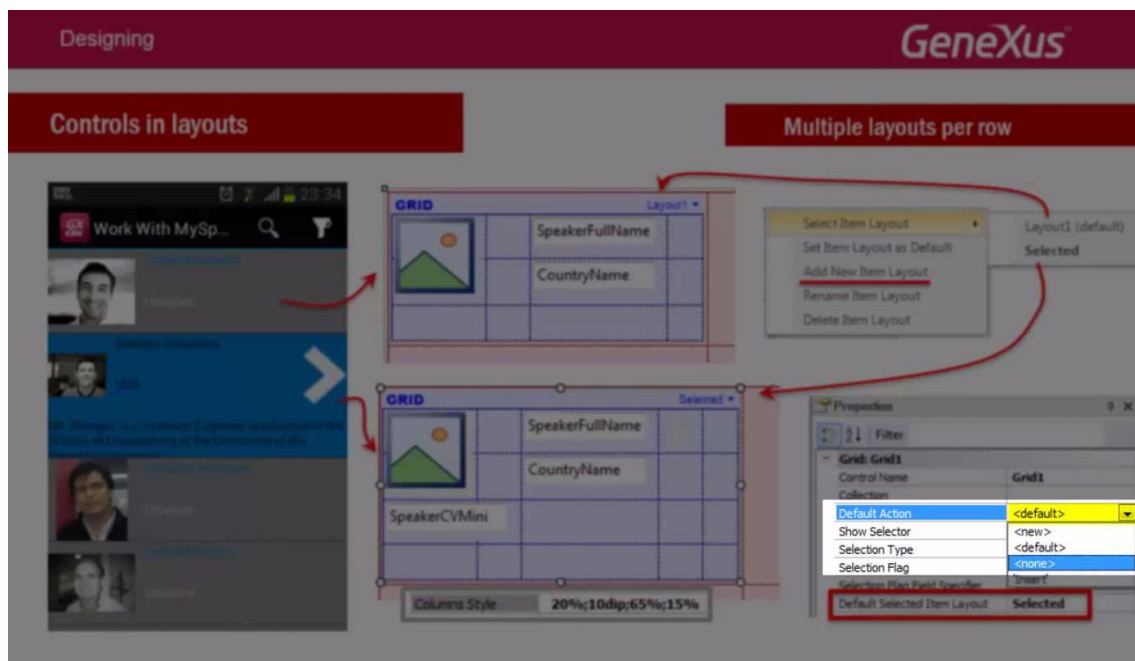
Não havíamos visto que o comportamento por padrão era levar-nos ao Detail desse orador?

Necessitamos desabilitar esse comportamento. Onde isto encontra-se configurado?

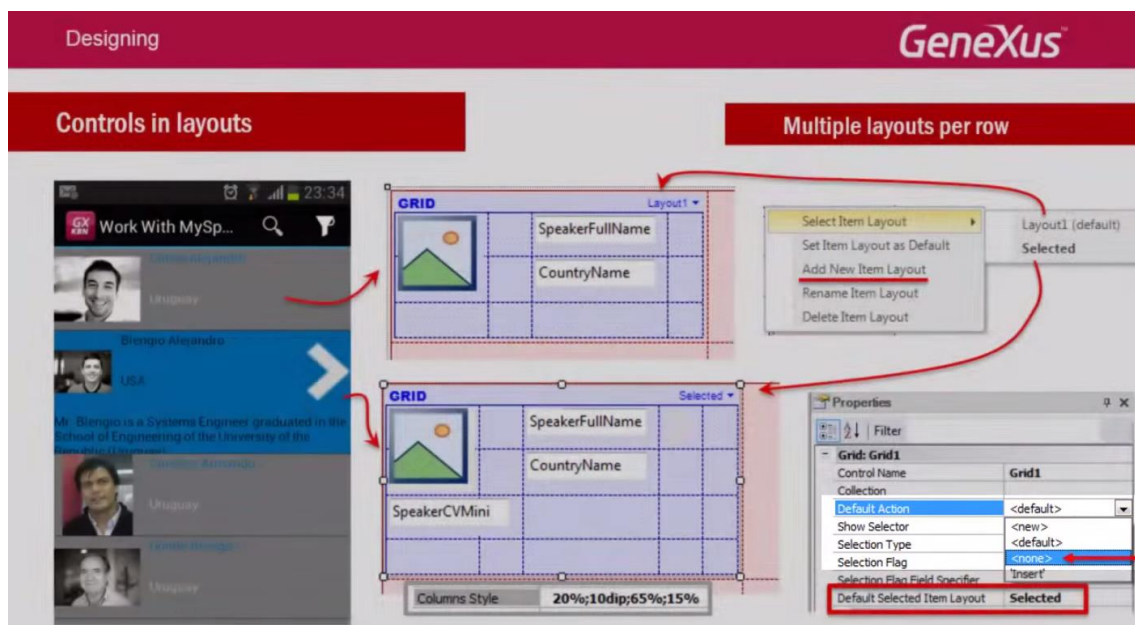


Na propriedade **Default Action** do grid.

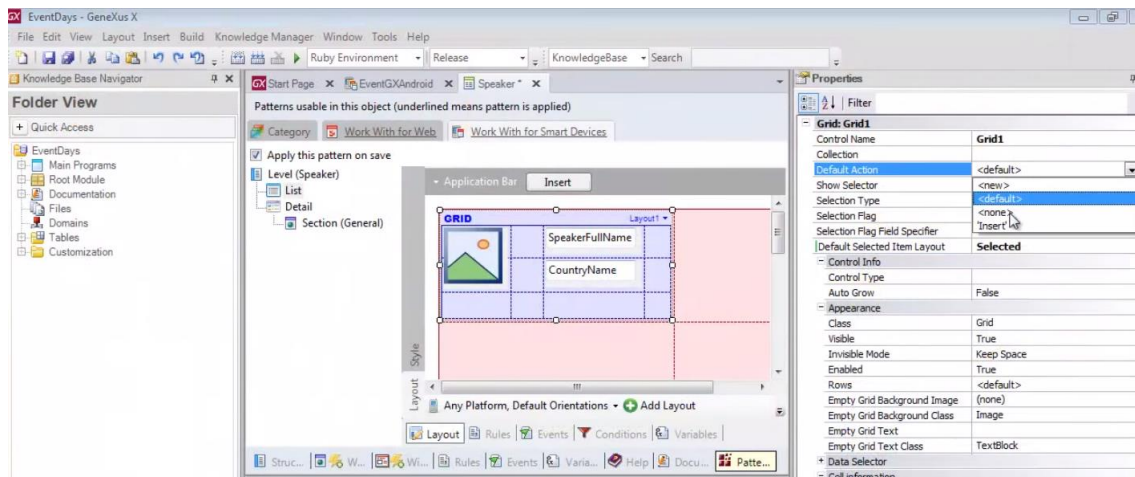
O valor “Default” que estamos vendo, consiste em chamar a Detail:



Mudaremos o: None

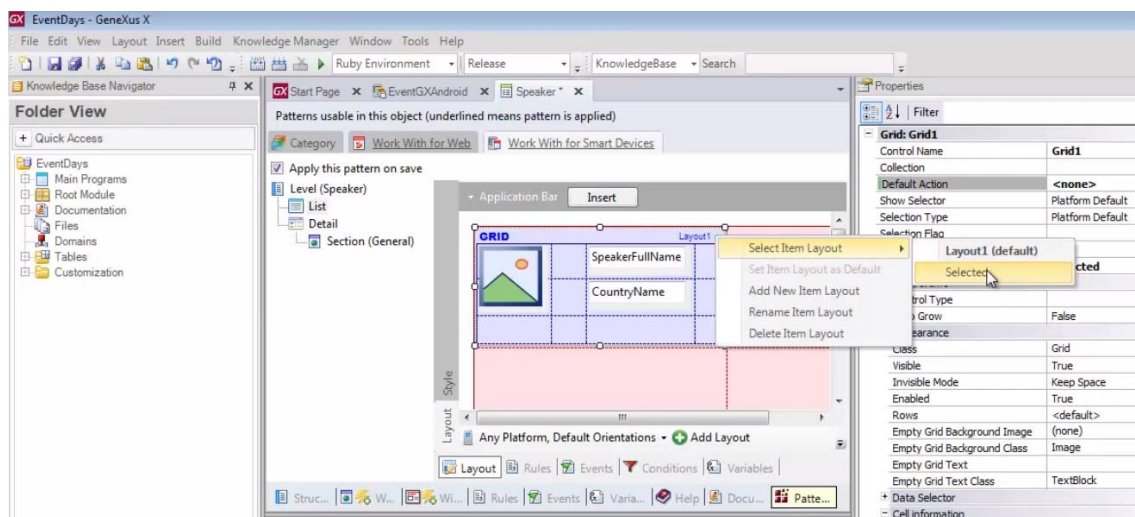


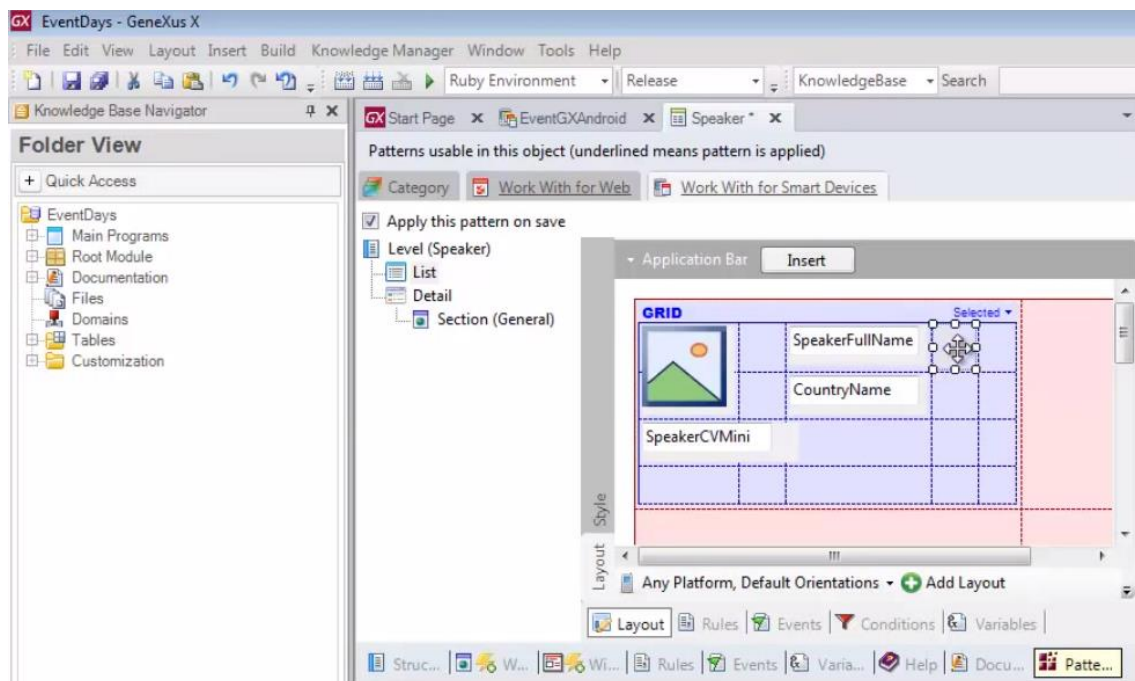
Deste modo não é invocado nenhum evento como consequência ao escolher uma linha.



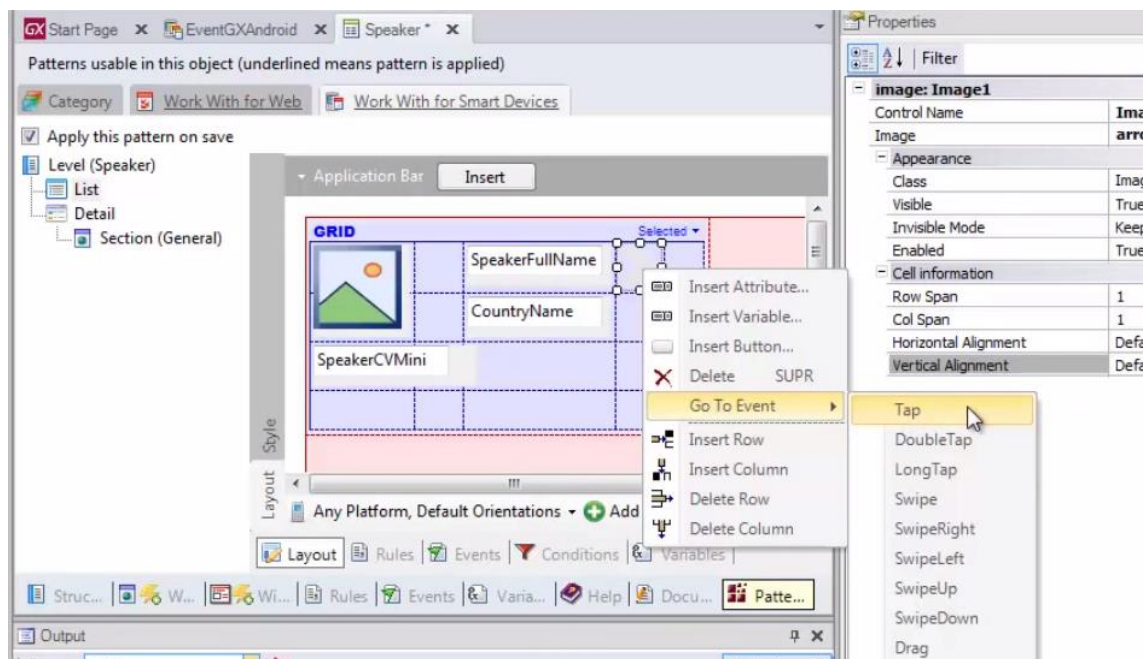
Mas... queremos que o usuário possa acessar os detalhes do orador se o desejar.. para isso, o programamos explicitamente.

Escolhemos o layout “Selected”:



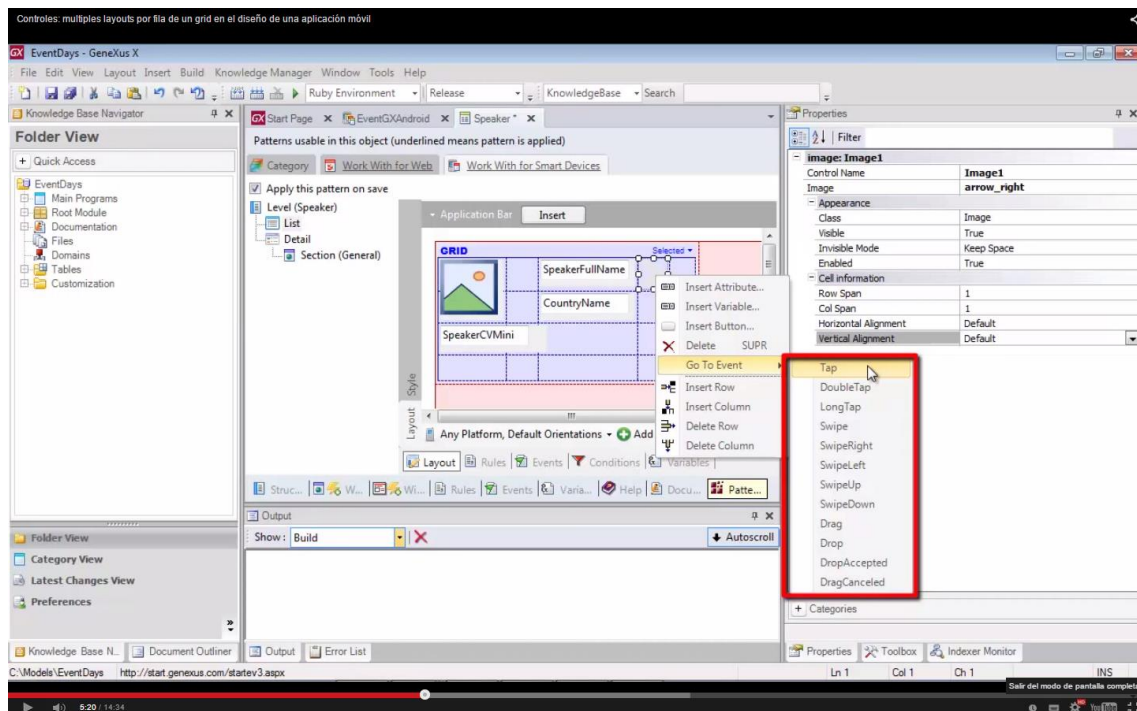


E sobre a imagem com a seta, escolhemos o evento: Tap

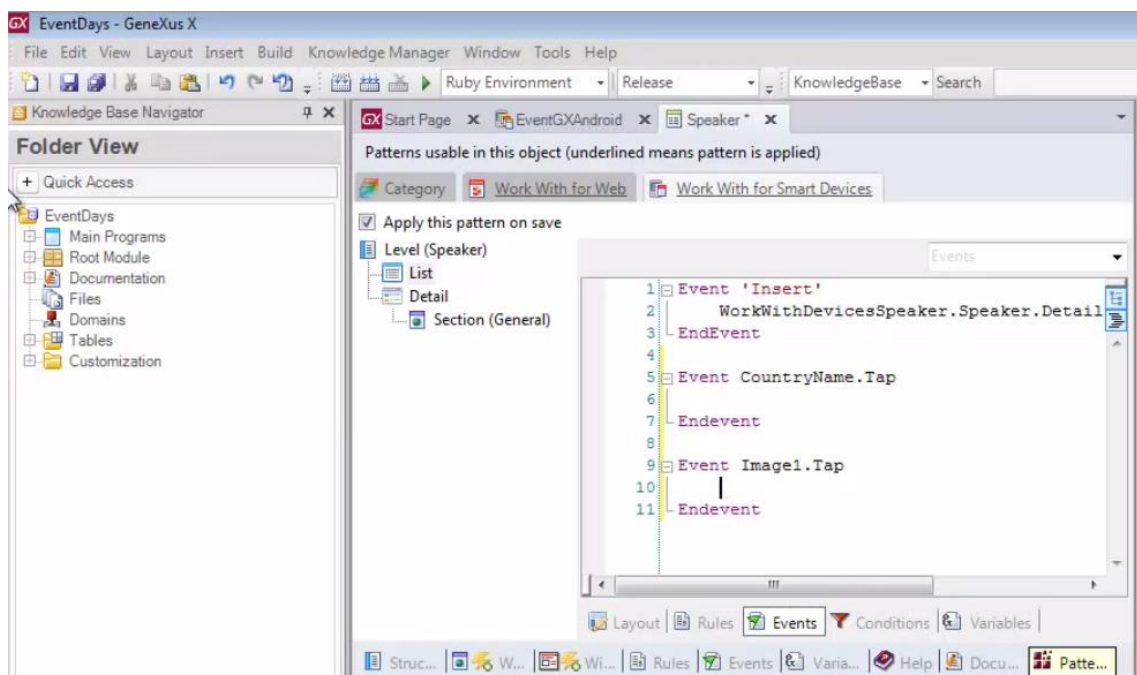


Vemos aqui como os controles tem a possibilidade de programar os eventos touch aos que são sensíveis:

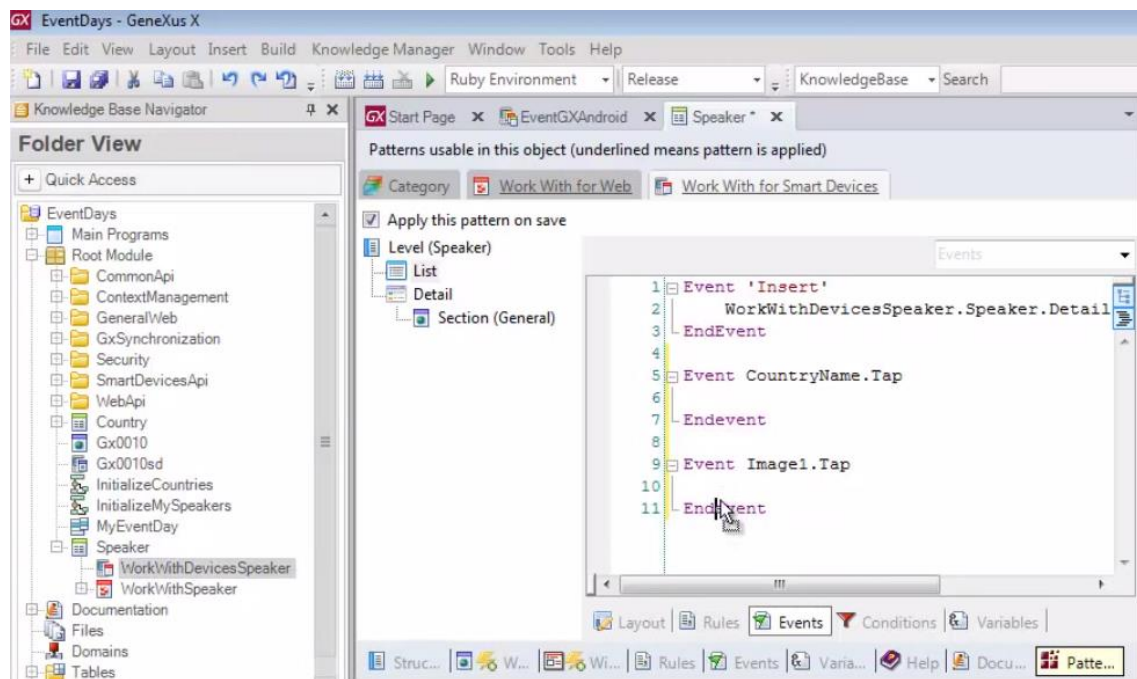




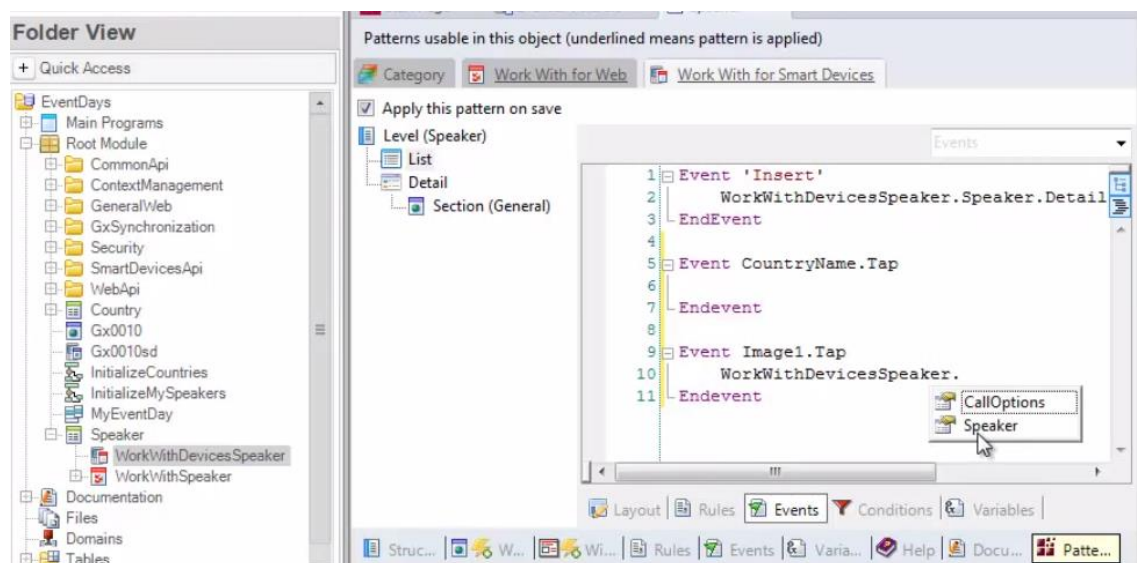
Voltaremos ao tema eventos mais adiante.



Quem queremos chamar então como resultado do Tap?

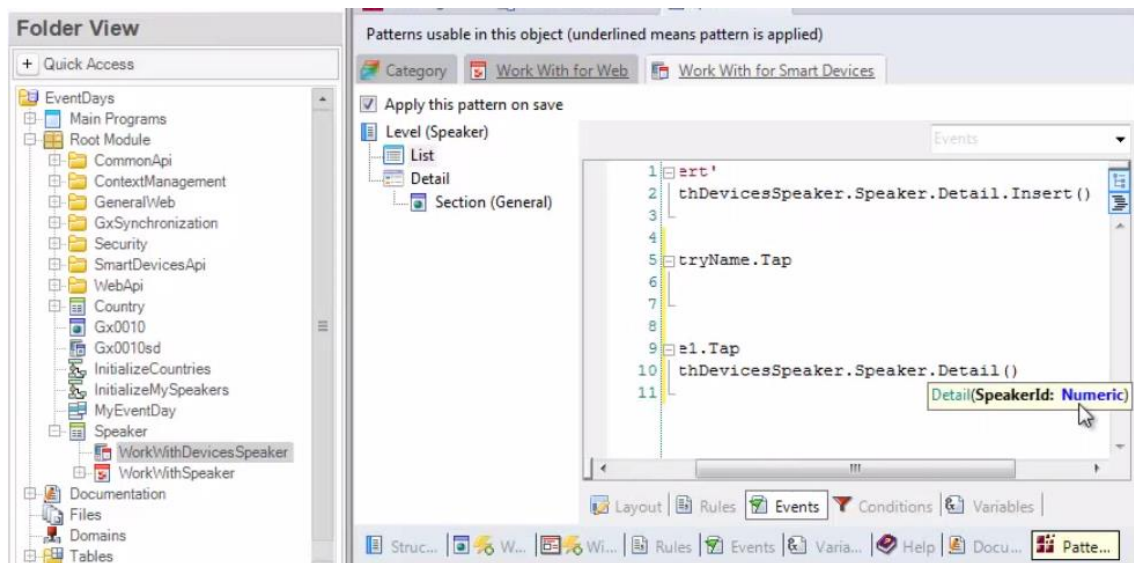
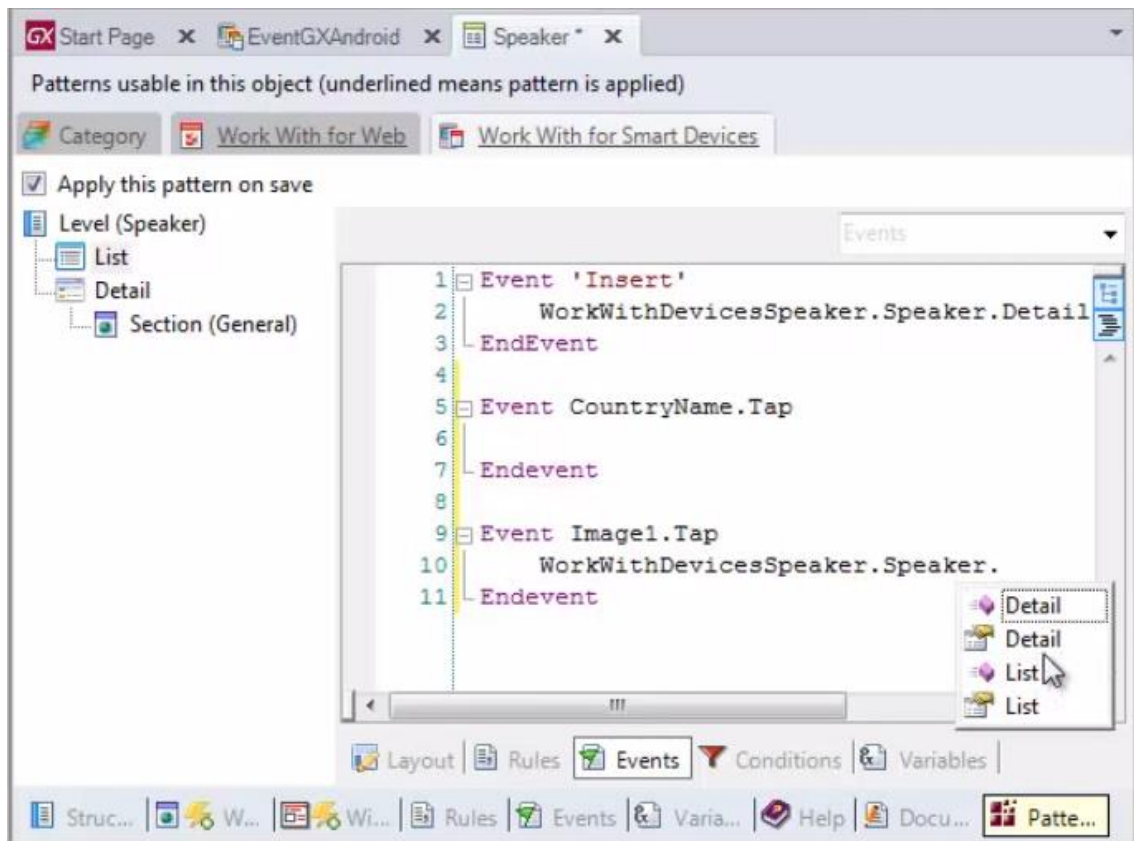


Ao WorkWith do Smart Devices de Speakers

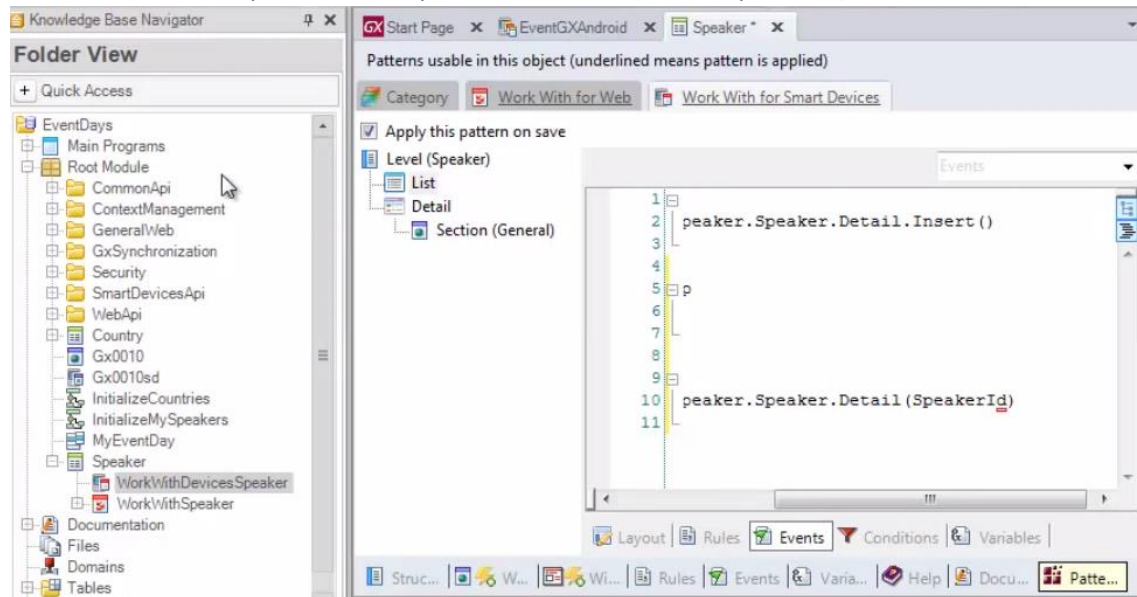


Recordamos a sintaxe:

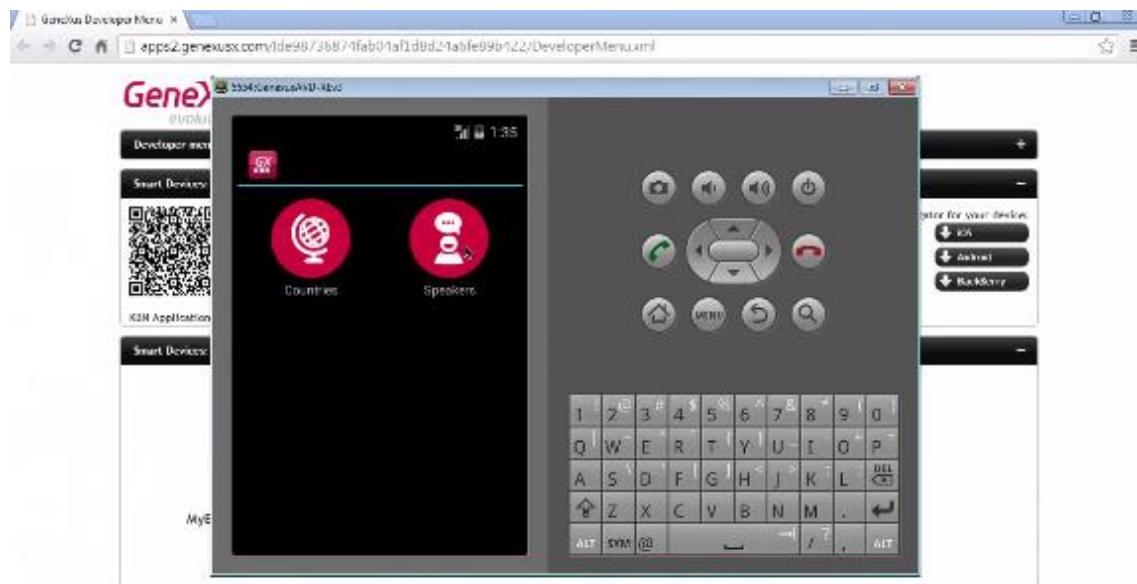
Work With – nível – e a quem iremos chamar (no nosso caso a Detail)



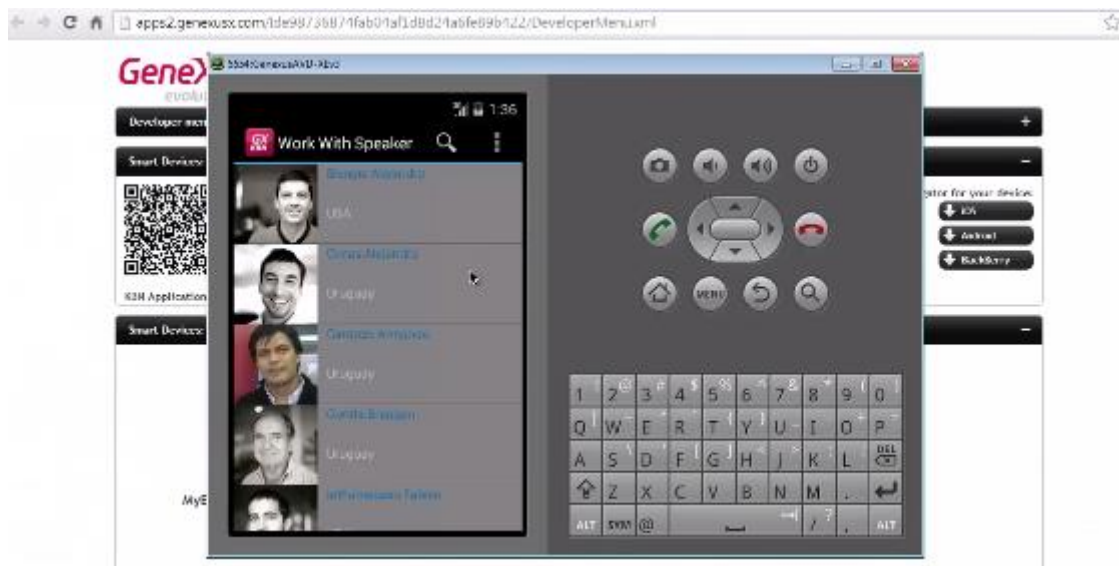
Passando-lhe como parâmetro o SpeakerId da linha (a chave primária)



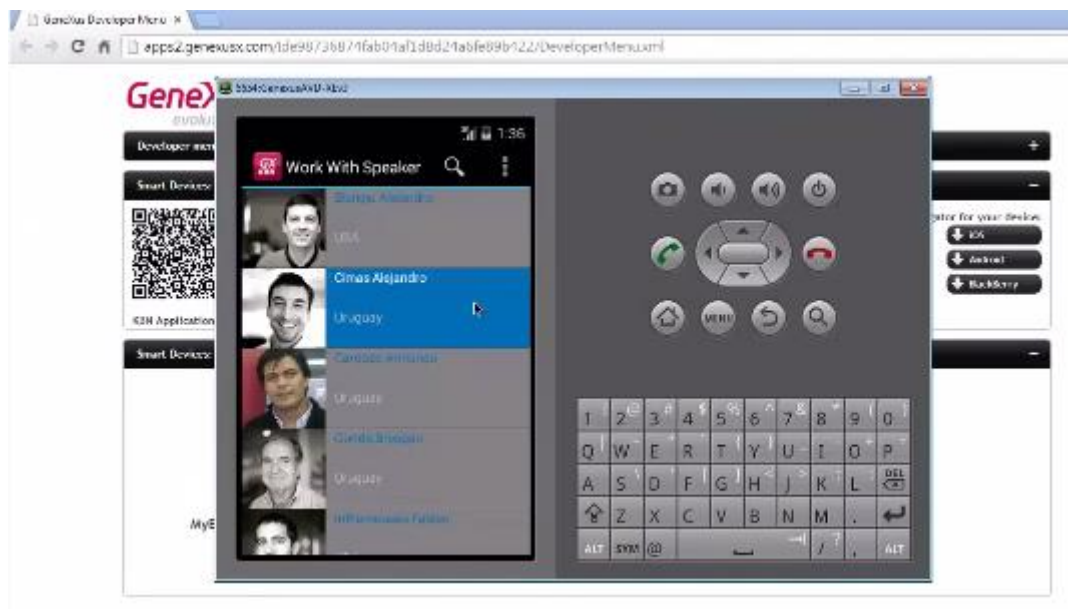
Gravamos... vamos tentar executá-lo. F5.

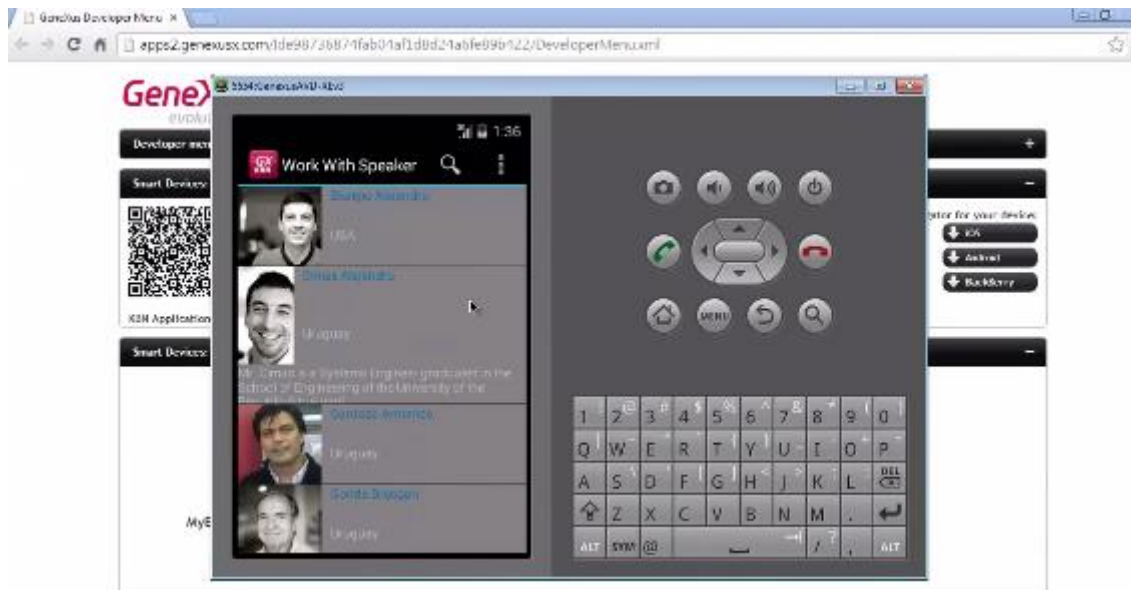






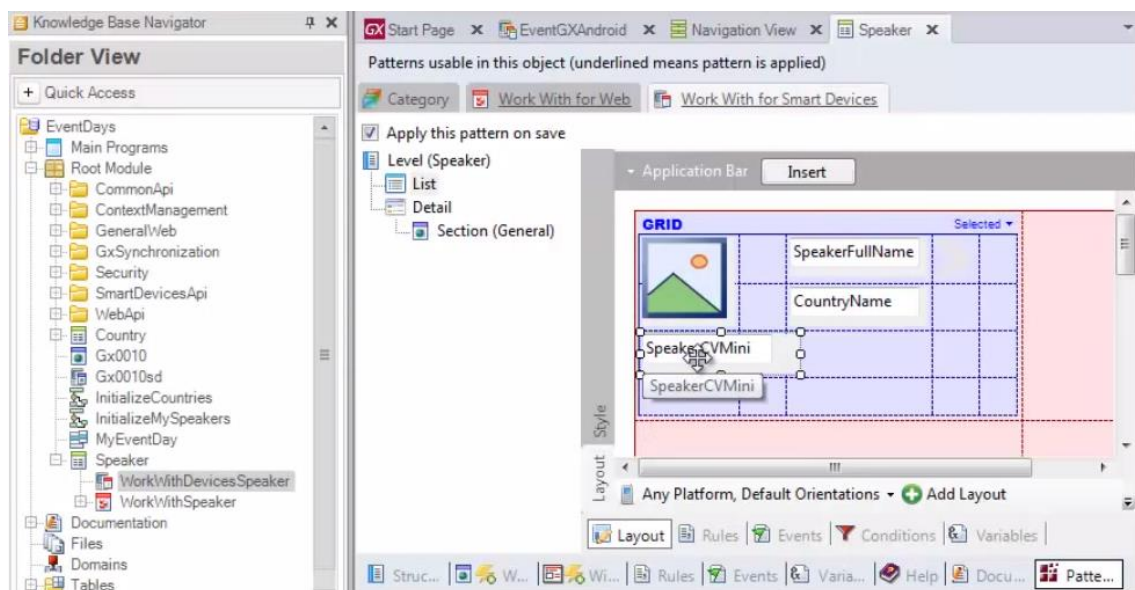
Damos um Tap



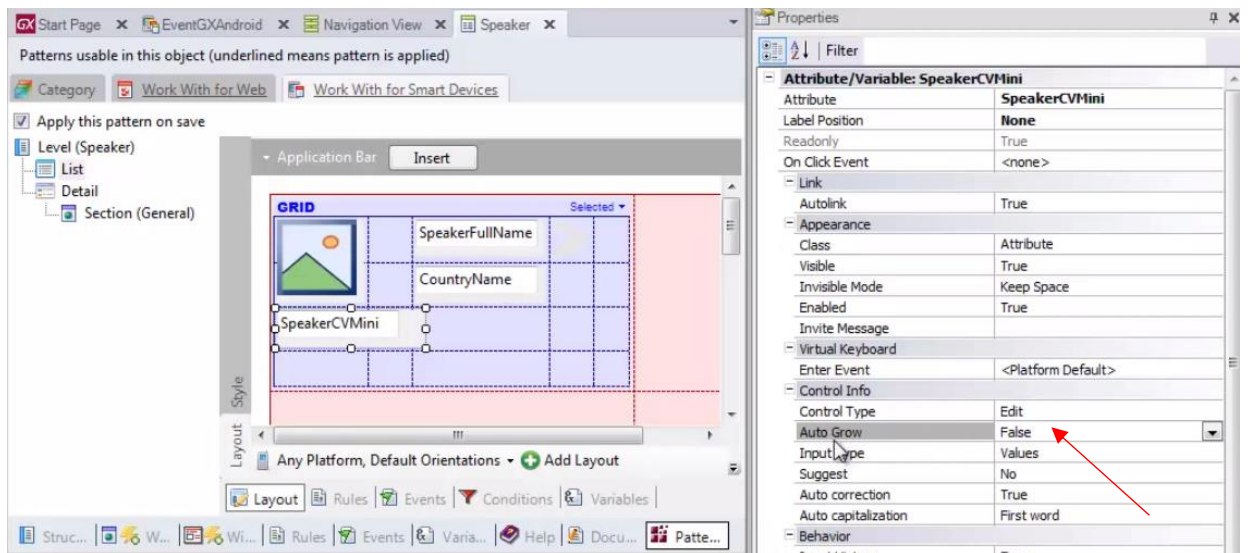


Porque está aparecendo cortado o curriculum vitae?

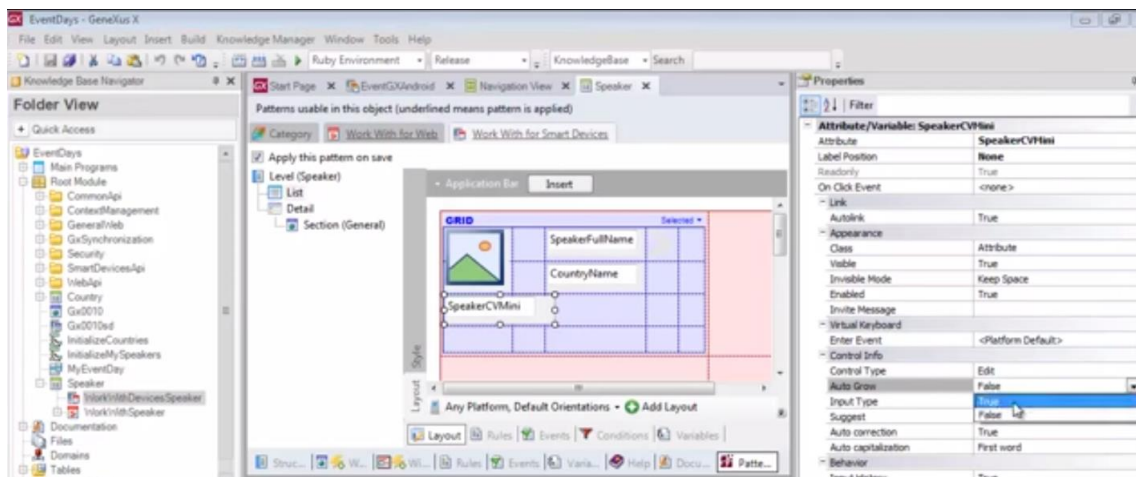
Porque ao atributo



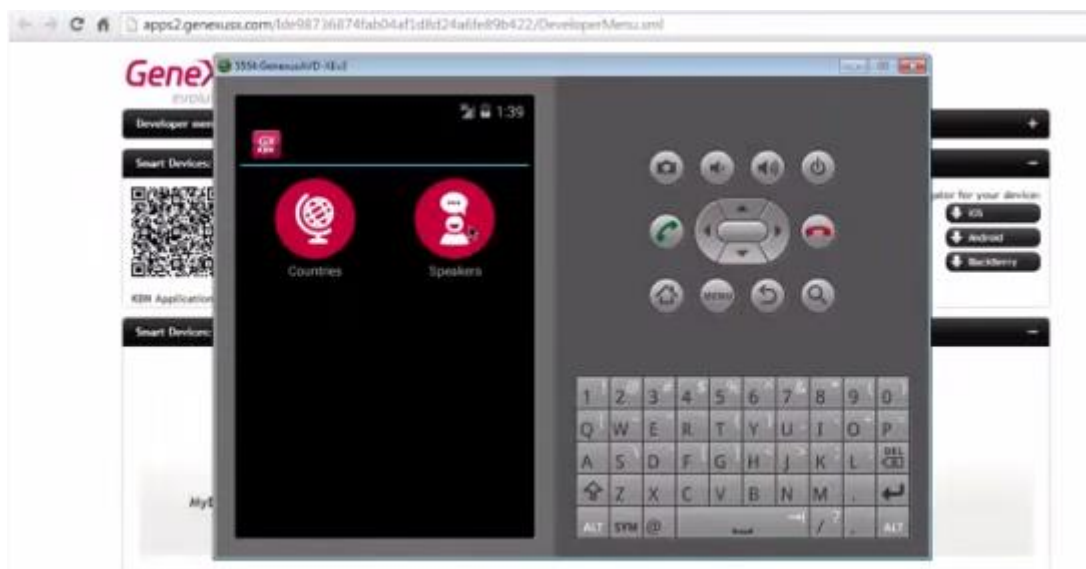
Não lhe permitimos expandir de acordo com seu conteúdo.



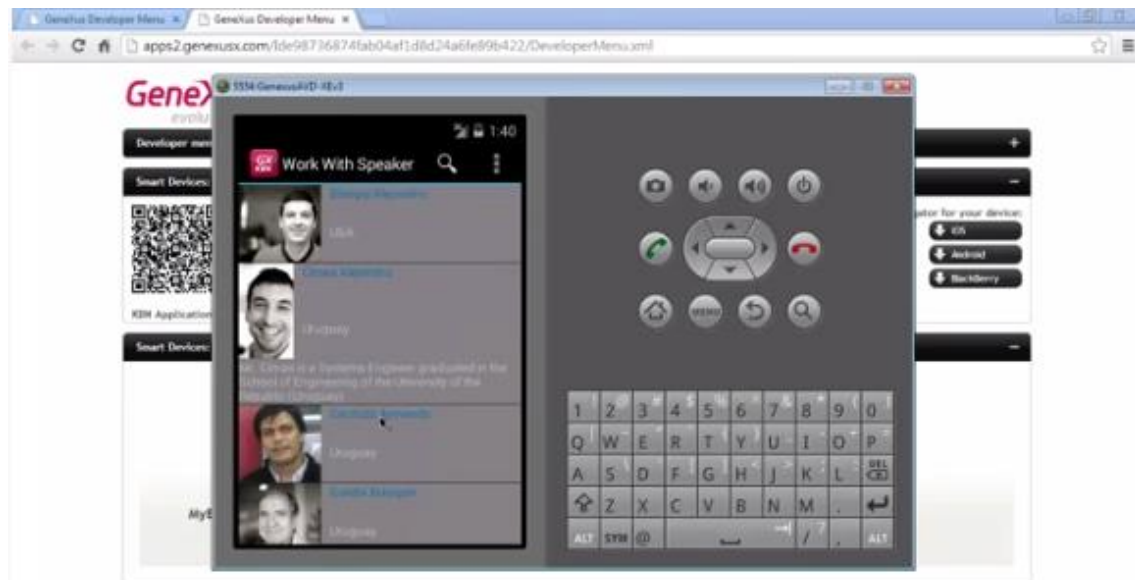
Veamos que a propriedade Auto Grow está como False. Pasemos-a para True:



E outro F5.



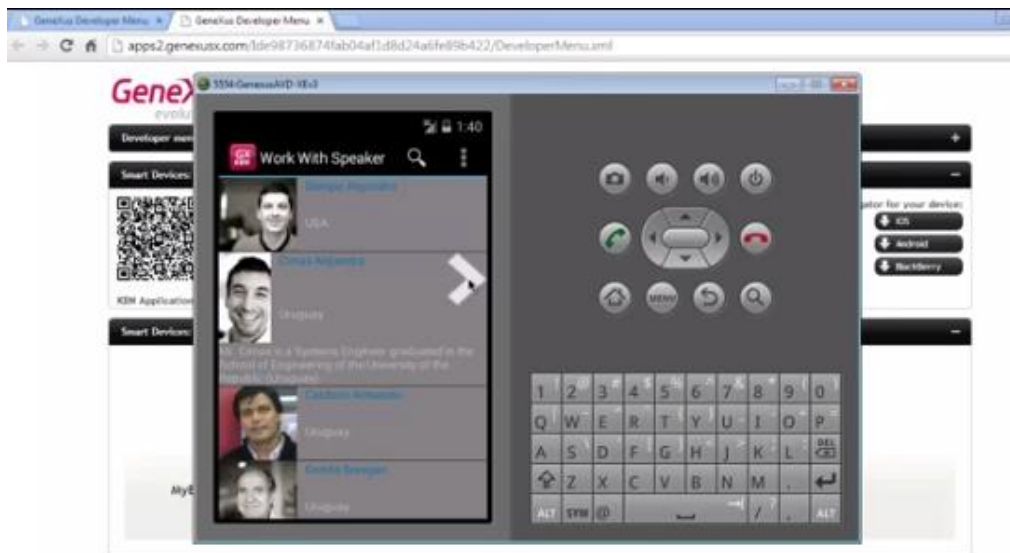
E agora TAP:



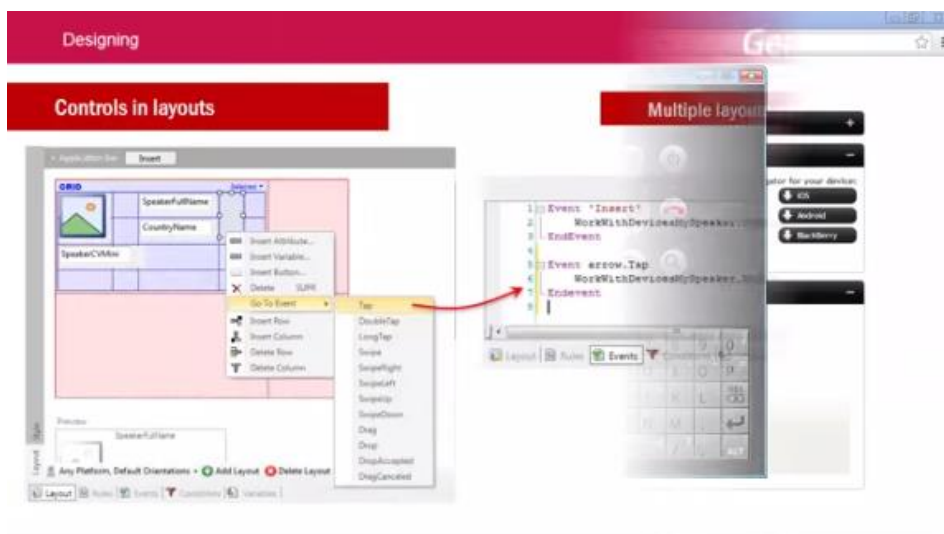
Agora sim funcionou ..

E ao dar um TAP sobre a seta



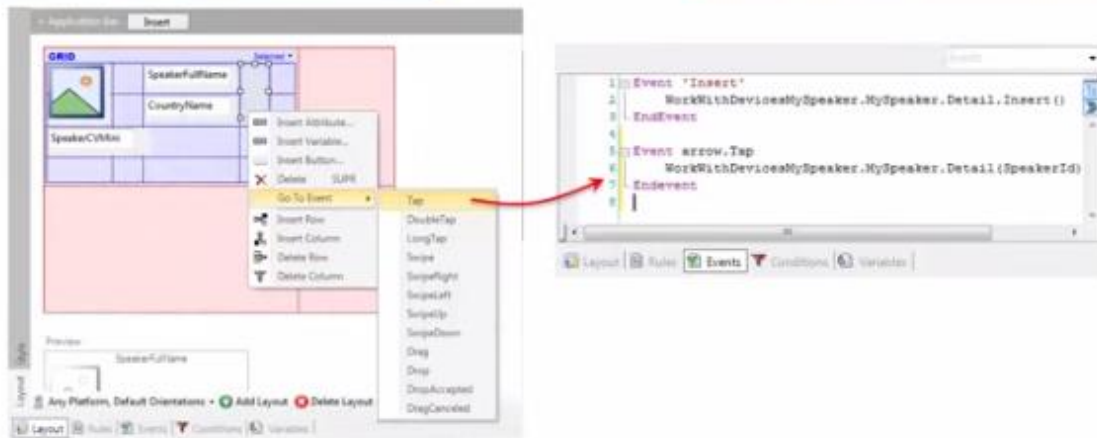


Vemos que chamou efetivamente a Detail.



## Controls in layouts

## Multiple layouts per row



O layout a ser carregado para cada linha do grid também pode ser especificado em Runtime:

## Controls in layouts

## Multiple layouts per row



e não somente fixar-se no desenho.

Isto é possível com a propriedade **ItemLayout** do grid.

E veremos no seguinte exemplo.

Aqui temos a transação que registra conferências, com seus Speakers e suas Tracks:



E a seu lado, a lista de conferências “Trabalhar com” sessions



agrupados por data e hora.

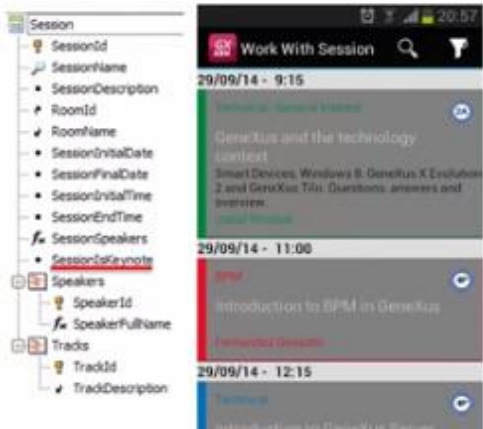
Se queremos que a session seja uma keynote:

Designing

GeneXus

Controls in layouts

Multiple layouts per row



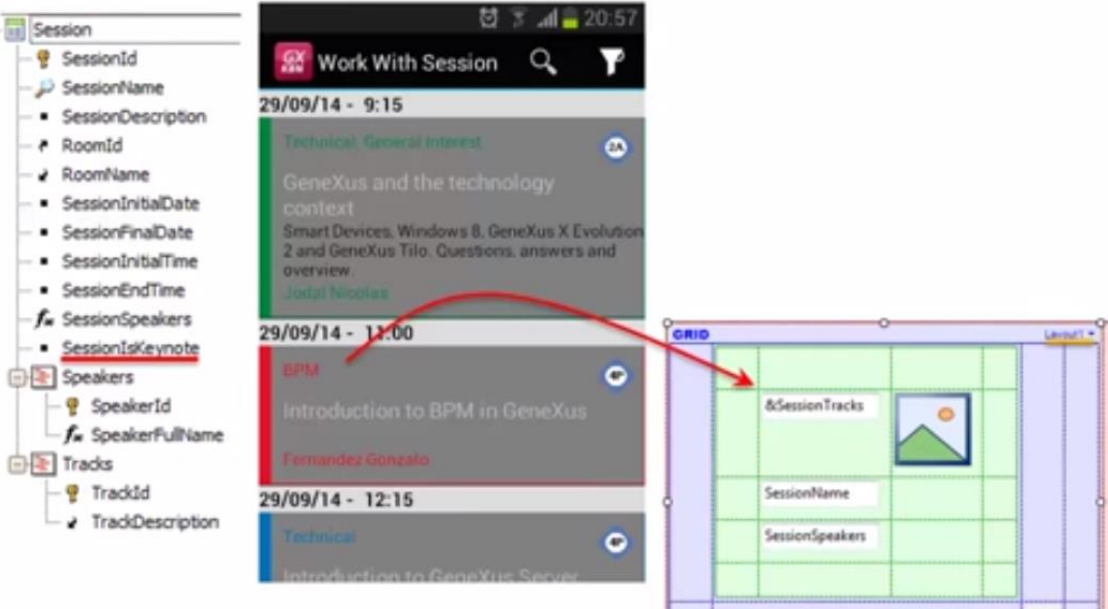
The screenshot shows the GeneXus Work With Session app. On the left, the data model tree includes Session, Speakers, and Tracks. The Session entity has a highlighted attribute SessionIsKeynote. The app interface on the right displays a list of sessions with columns for time slots (e.g., 29/09/14 - 9:15) and session details like title and speaker.

além das informações dos tracks onde a conferência pode ser classificada:

Designing

Controls in layouts

Multiple



The screenshot shows the GeneXus Work With Session app. On the left, the data model tree includes Session, Speakers, and Tracks. The Session entity has a highlighted attribute SessionIsKeynote. The app interface on the right displays a list of sessions with columns for time slots (e.g., 29/09/14 - 9:15) and session details like title and speaker. A red arrow points from the SessionIsKeynote attribute to a control in the layout, which is a grid containing SessionName and SessionSpeakers.

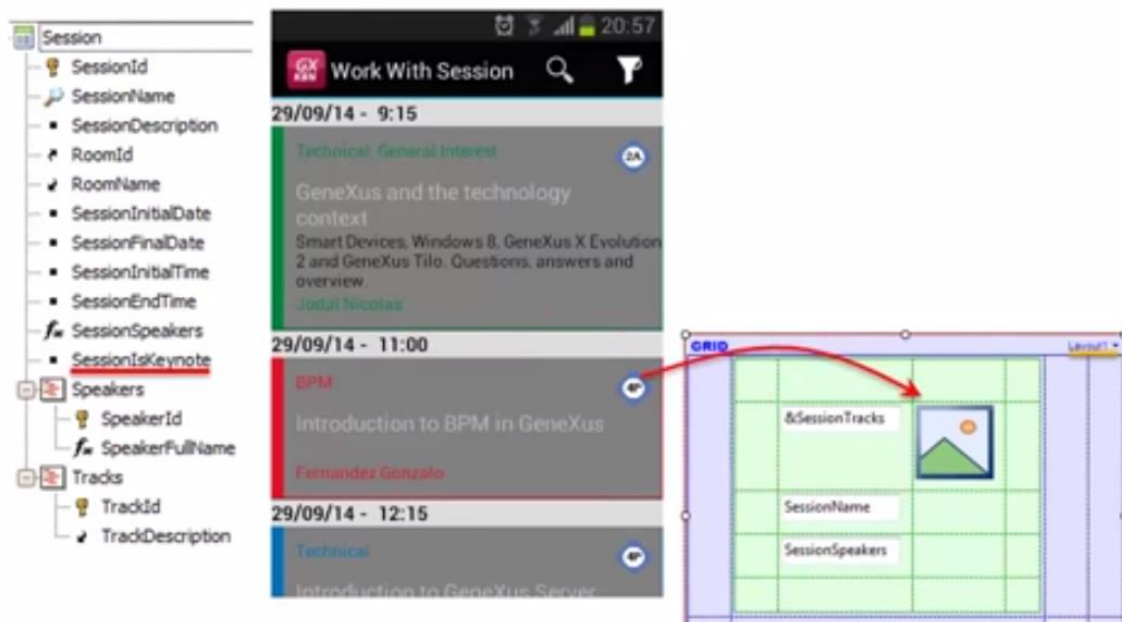
a imagem da sala:



## Designing

### Controls in layouts

Multip

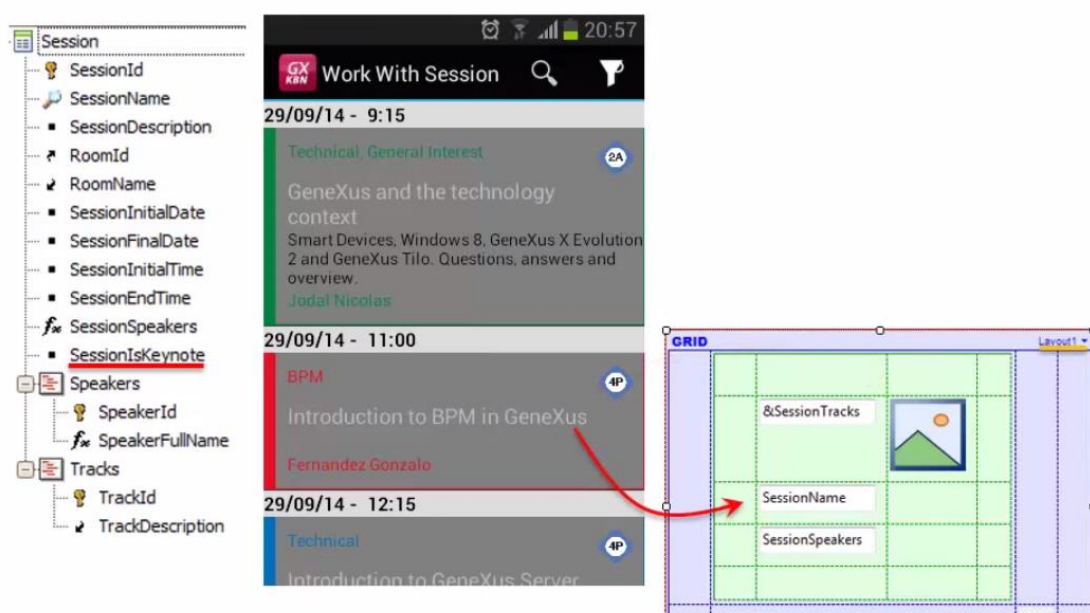


o nome da conferência:

## Designing

### Controls in layouts

Multiple

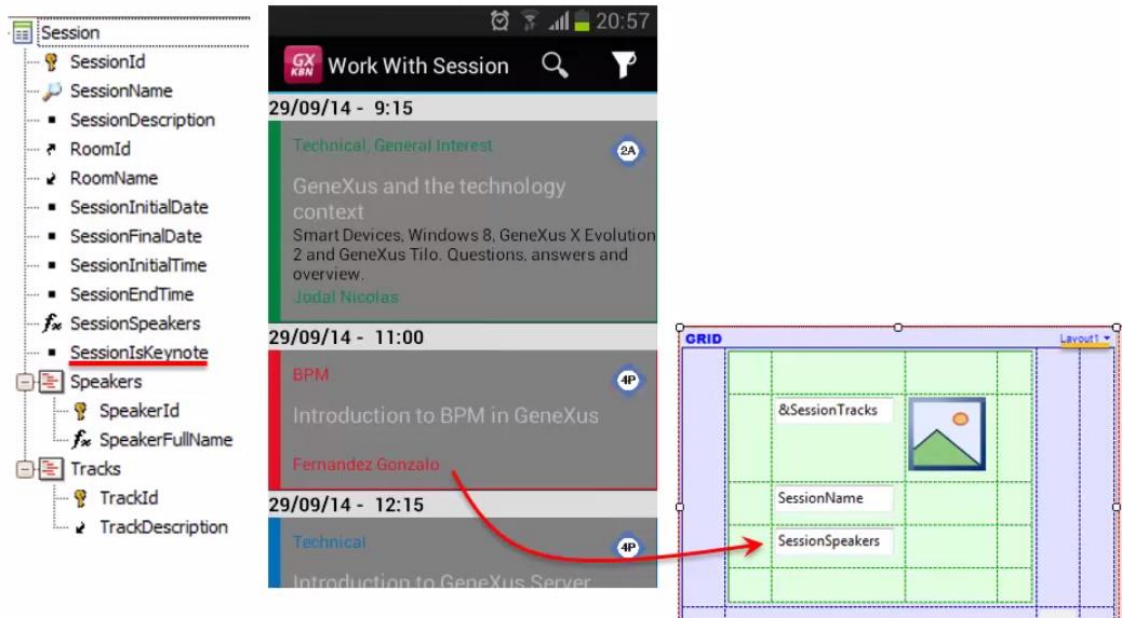


e seus oradores:

Designing

Controls in layouts

Multipl

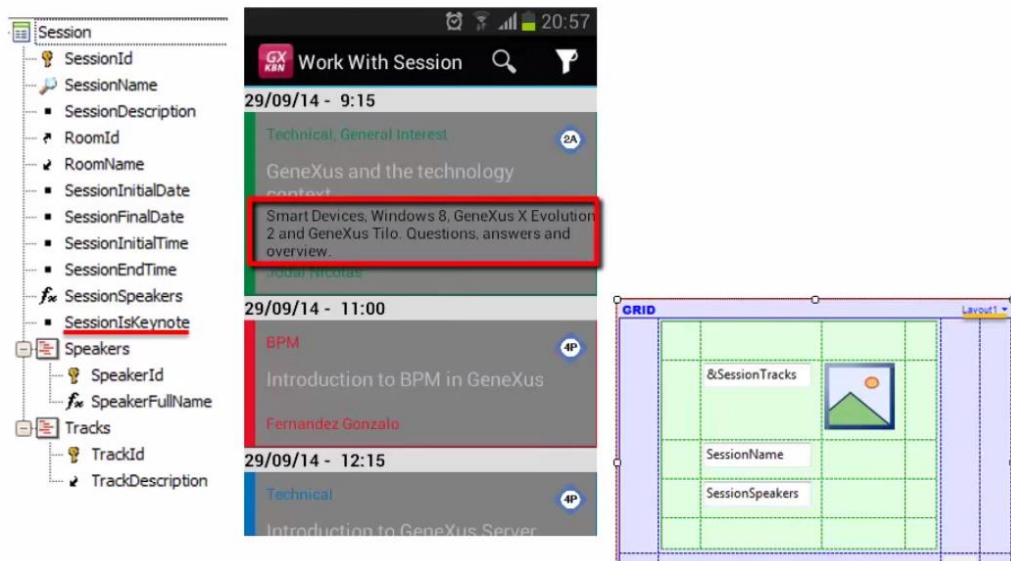


aparecerá também um resumo de sua descrição como podemos ver a primeira sessão listada( diferente da segunda),

Designing

Controls in layouts

Multipl



necessitamos definir além do default, outro layout para os itens do grid.

Designing

GeneXus

Controls in layouts

Multiple layouts per row

Layout em que colocaremos a descrição.

Aqui definimos um ao qual chamamos de: Keynote

Logo, em execução,

Designing

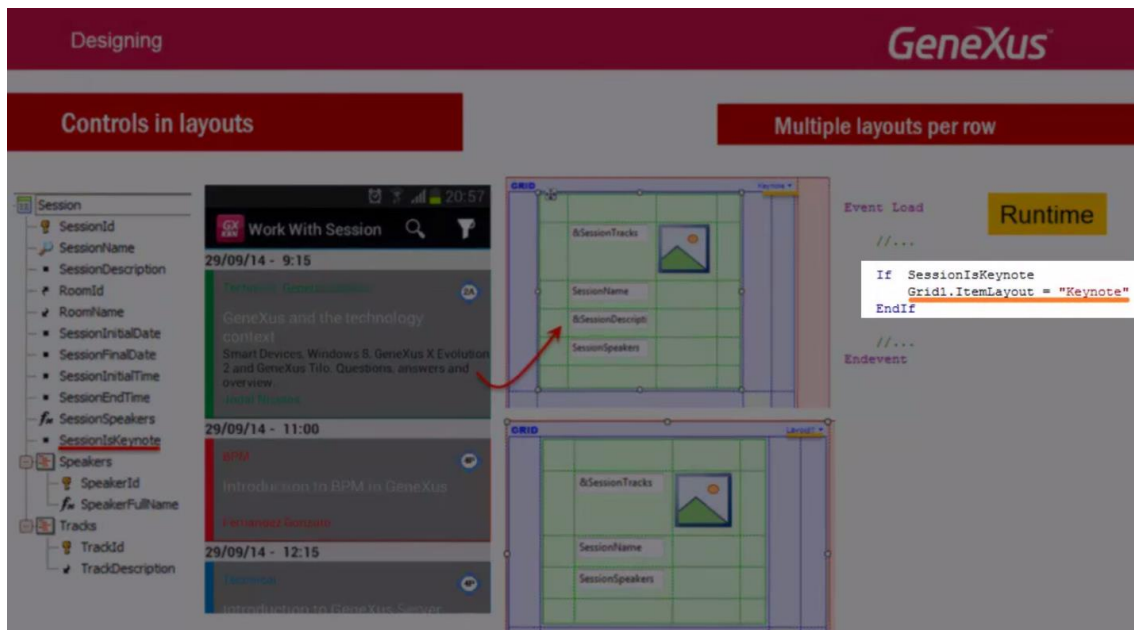
GeneXus

Controls in layouts

Multiple layouts per row

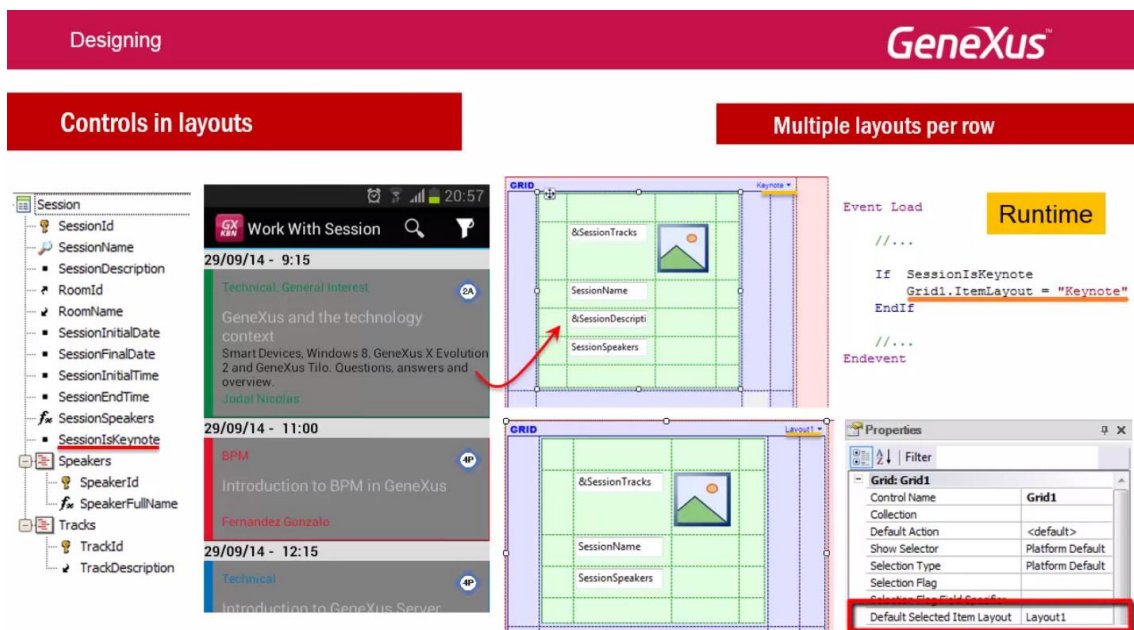
Runtime

cada linha deverá ser carregada com um outro layout, dependendo do valor do atributo:  
**SessionIsKeynote**



Fazemos isto no evento Load do grid.

Observemos que a propriedade **Default Selected Item Layout**



fica com o valor por padrão:



## Controls in layouts

## Multiple layouts per row

The screenshot displays the GeneXus design environment. On the left, a tree view shows the 'Session' object with properties like SessionId, SessionName, RoomId, SessionInitialDate, SessionFinalDate, SessionInitialTime, SessionEndTime, SessionSpeakers, SessionIsKeynote, Speakers, SpeakerId, SpeakerFullName, Tracks, TrackId, and TrackDescription. In the center, a mobile application preview shows a session list with details for 'GeneXus and the technology context' and 'Introduction to BPM in GeneXus'. To the right, a grid layout is shown with controls for SessionTracks, SessionName, SessionDescription, and SessionSpeakers. Further right, a code editor shows the following runtime code:

```

Event Load
//...
If SessionIsKeynote
  Grid1.ItemLayout = "Keynote"
EndIf
//...
Endevent

```

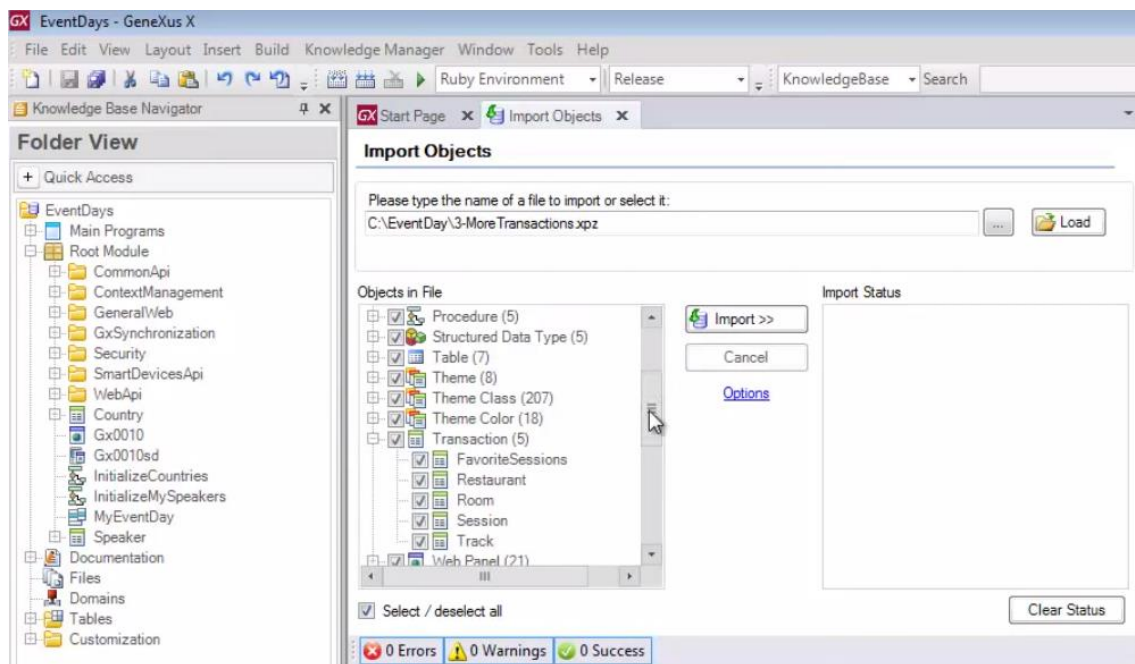
Below the code, the Properties window for 'Grid: Grid1' is visible, showing the 'Default Selected Item Layout' property set to 'Layout1'.

Vamos fazê-lo no GX.

Vamos começar importando um xpz.

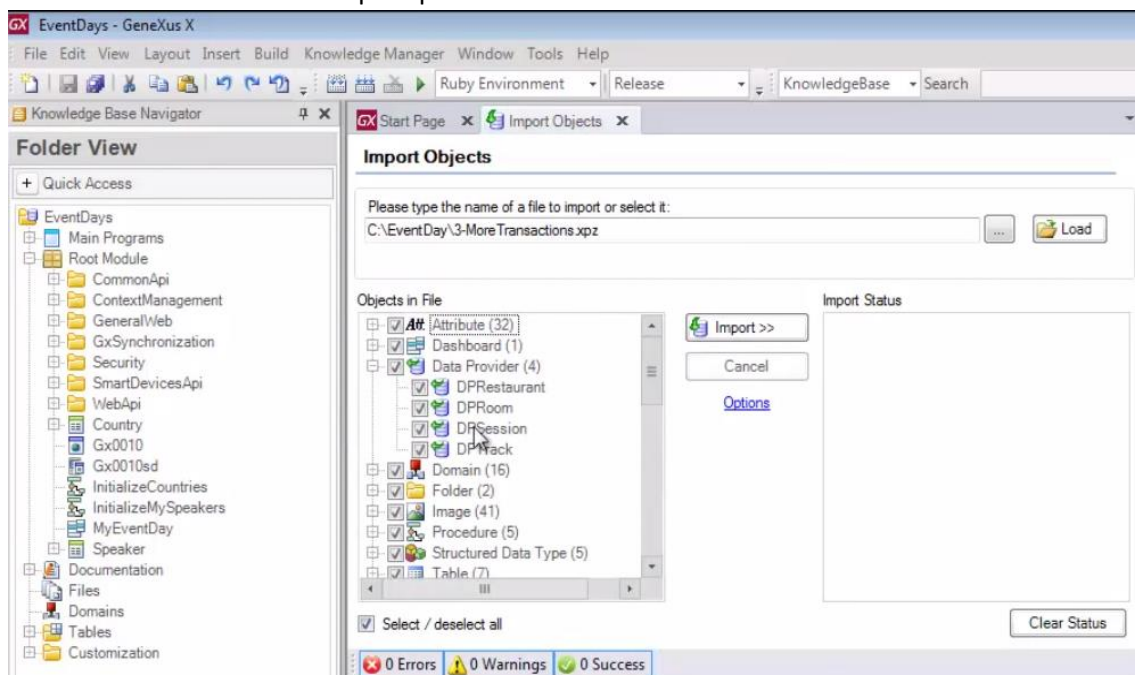
The screenshot shows the GeneXus EventDays application interface. The 'Import Objects' dialog is open, prompting the user to type the name of a file to import or select it. The file path 'C:\EventDay\3-More Transactions.xpz' is entered. The 'Objects in File' list shows various objects including Attribute (32), Dashboard (1), Data Provider (4), Domain (16), Folder (2), Image (41), Procedure (5), Structured Data Type (5), Table (7), Theme (8), Theme Class (207), Theme Color (18), and Transaction (5). The 'Import Status' section shows '0 Errors', '0 Warnings', and '0 Success'. The 'Output' window is also visible at the bottom.

que contêm entre outras coisas, as transações para completar nosso modelo de dados.

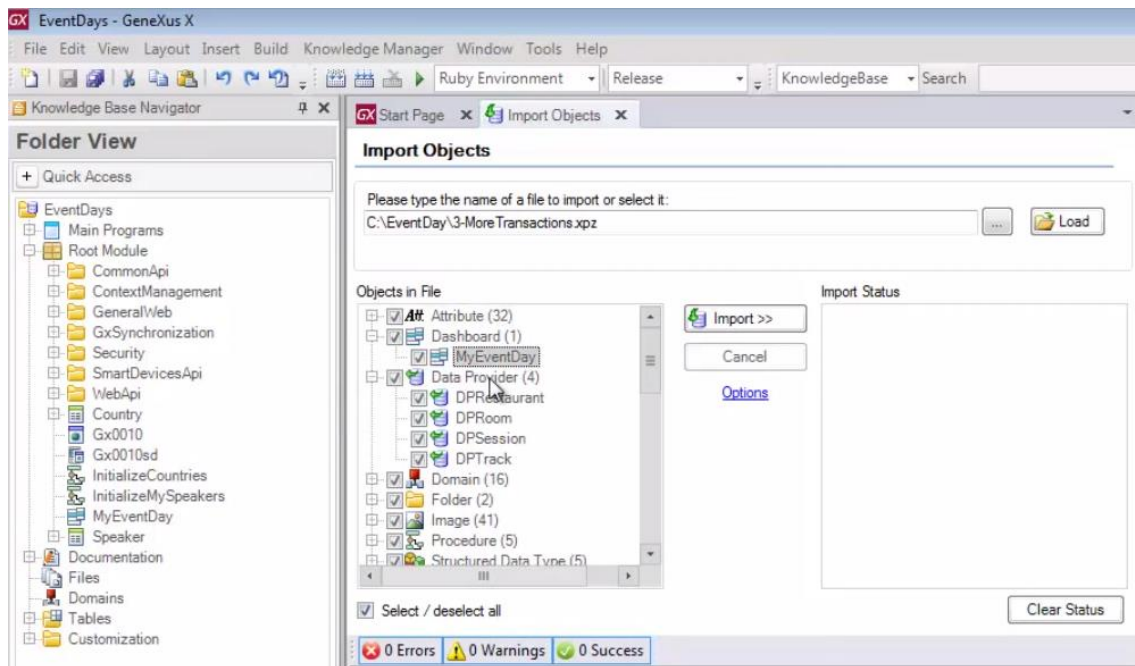


Track, Session, Room, Restaurant e a transação para marcar uma sessão como favorita.

Também terá Data Providers para poder inicializar com dados essas entidades.

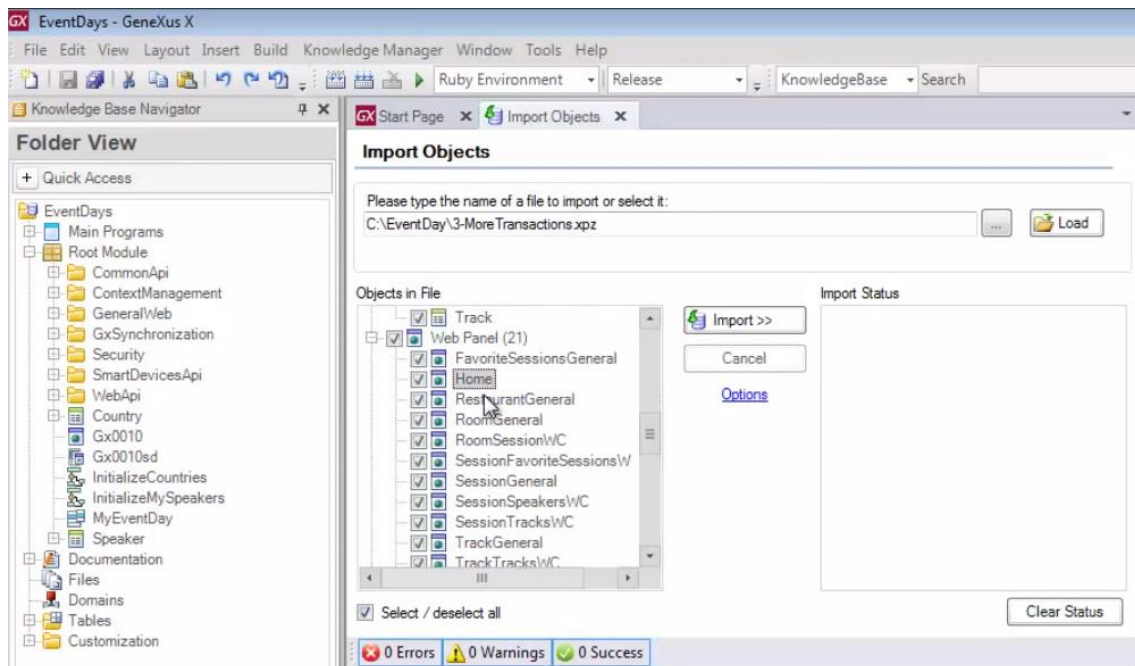


E um Dashboard: MyEventDay

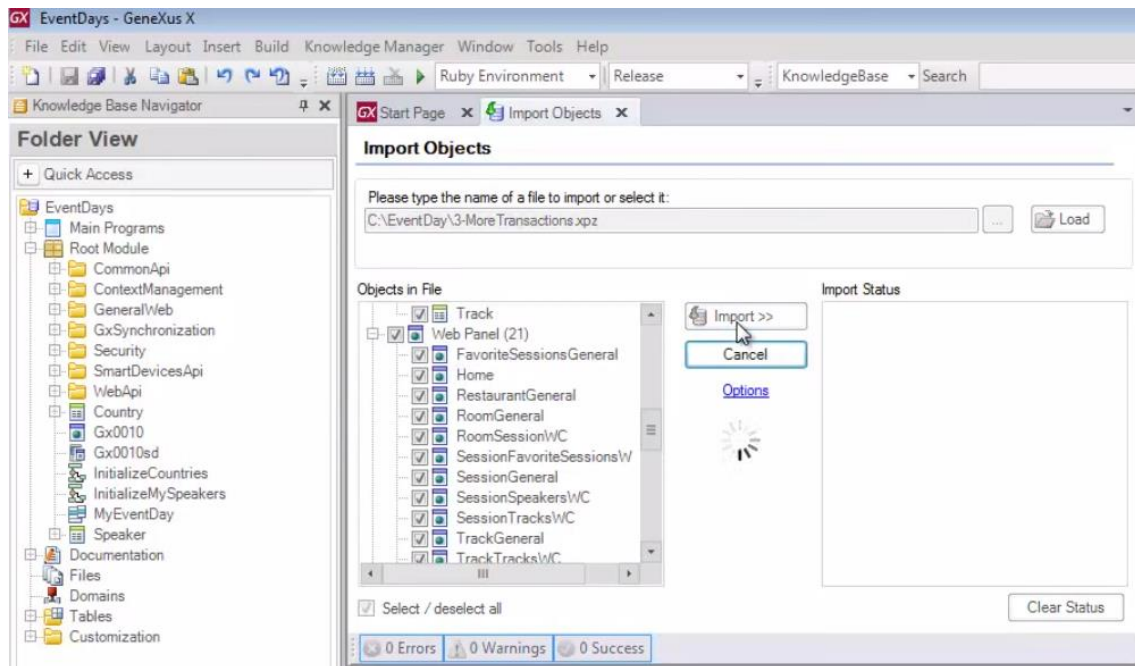


que irá subscrever o que tínhamos, posto que agora deverá poder chamar o Work With para Smart Devices das novas entidades.

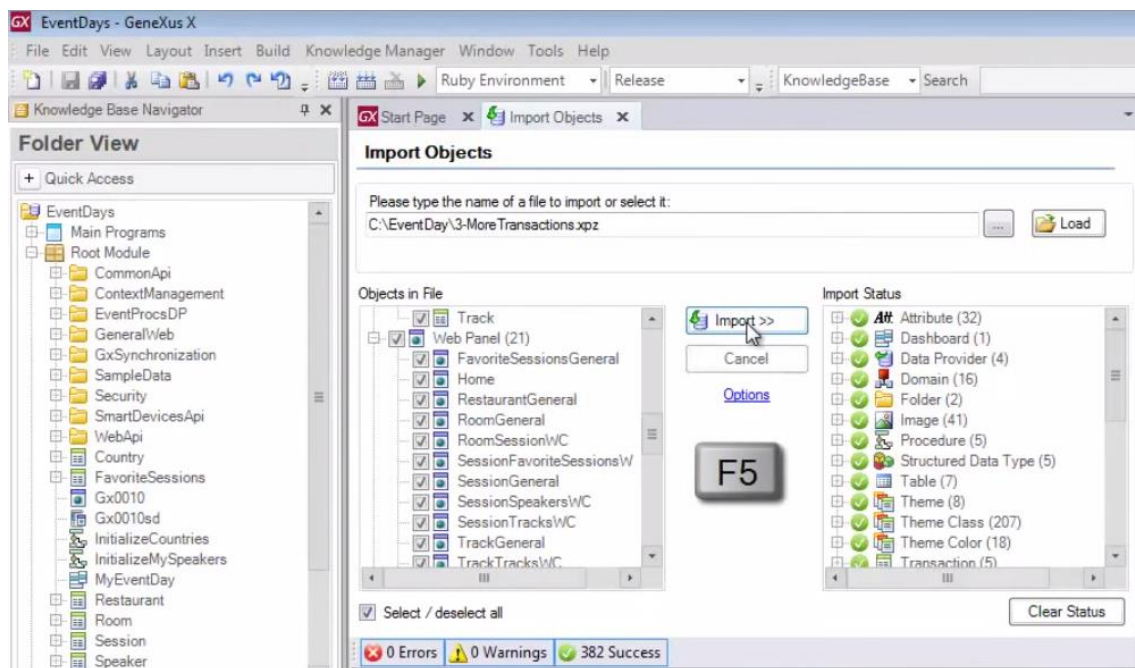
Também subscreve o objeto Home que tínhamos (web) para poder justamente inicializar todos os dados.



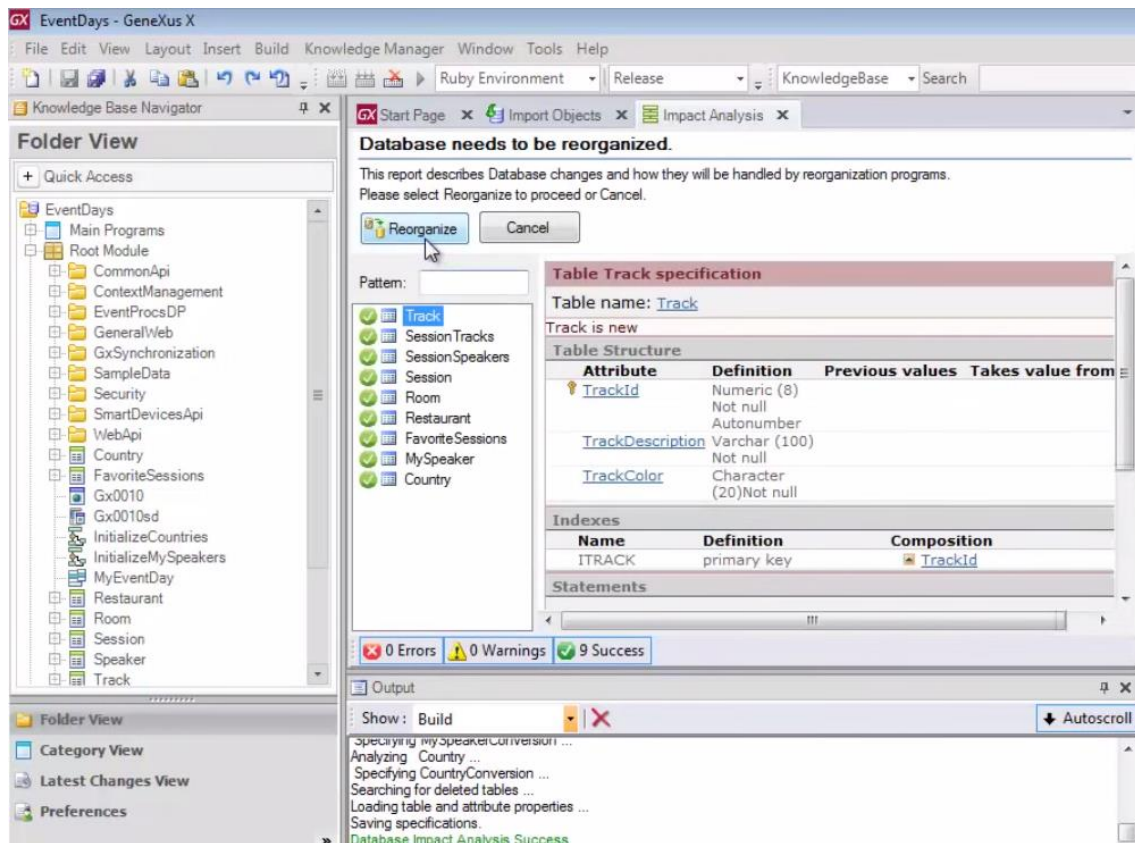
Importemos:



Agora reorganizamos e executamos: F5

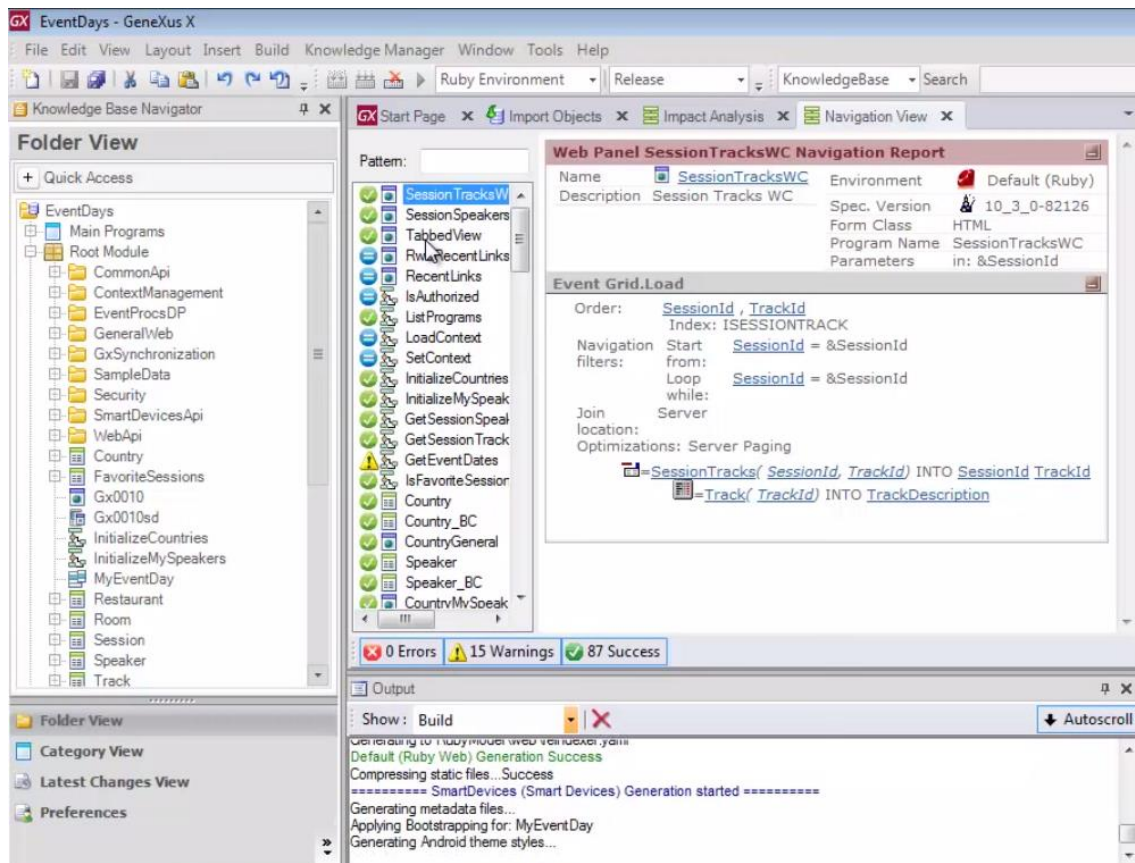




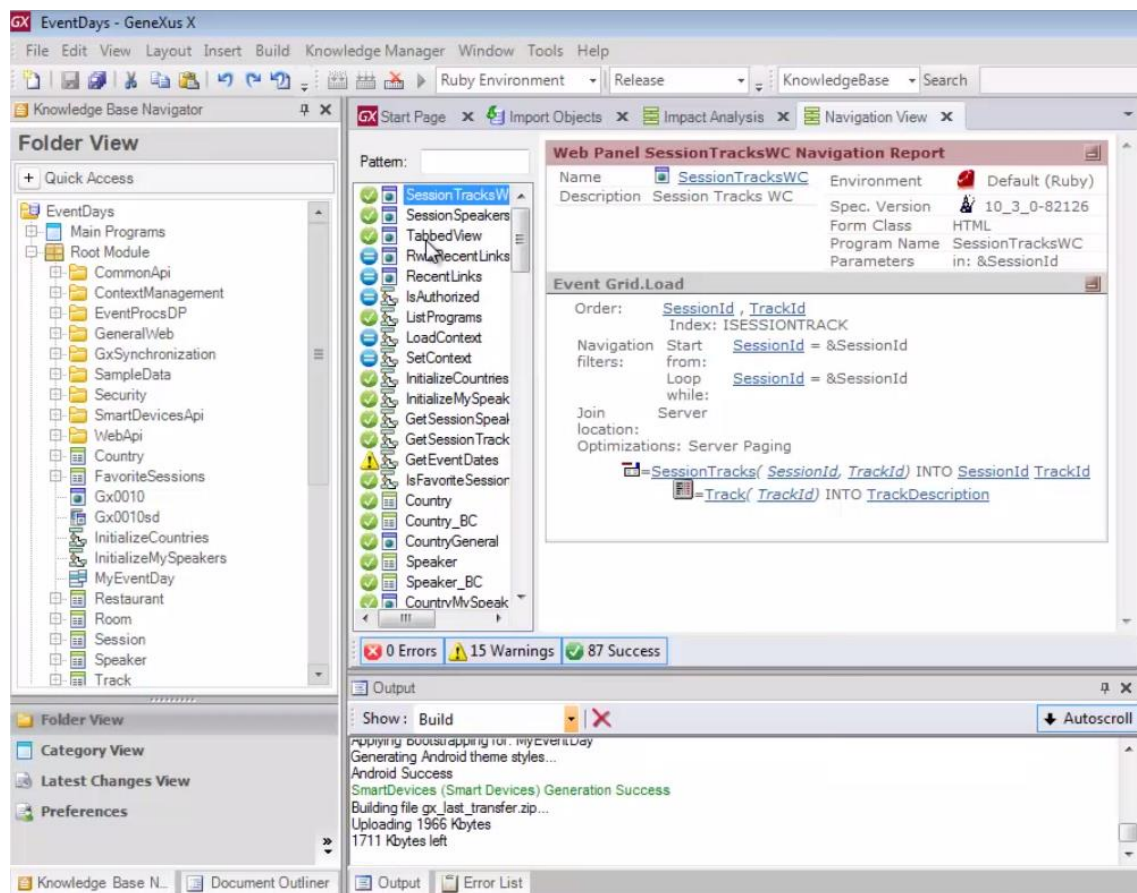


Reorganizamos..

Vemos que esta gerando os arquivos de metadata



E começa a subir na nuvem:



Depois disso é aberto o Developer Menu



Vamos executar a web panel: Home

apps2.genexus.com/Id98736874fab04af1d8d24a6fe89b422/DeveloperMenu.xml

**GeneXus™**  
evolution 3

Developer menu

- Home
- Work With Speaker
- Work With Country
- Work With Track
- Work With Session
- Work With Room
- Work With Restaurant
- Work With Favorite Sessions

Smart Devices: Application QR code for Knowledge Base Navigator (All devices)



Get Knowledge Base Navigator for your device:




- ↓ iOS
- ↓ Android
- ↓ BlackBerry

KBN Application Url: <http://apps2.genexus.com/Id98736874fab04af1d8d24a6fe89b422>

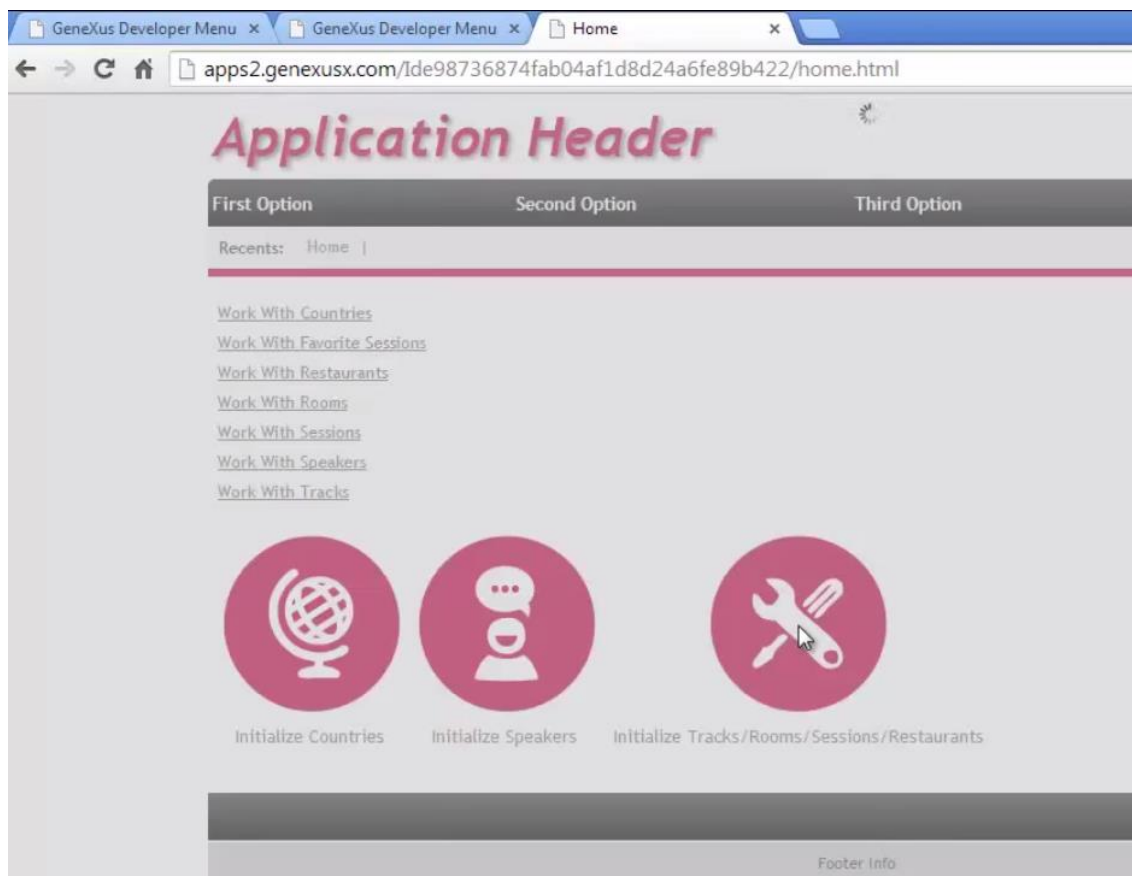
E clicar aqui para inicializar as entidades novas com dados:

apps2.genexus.com/Id98736874fab04af1d8d24a6fe89b422/home.html

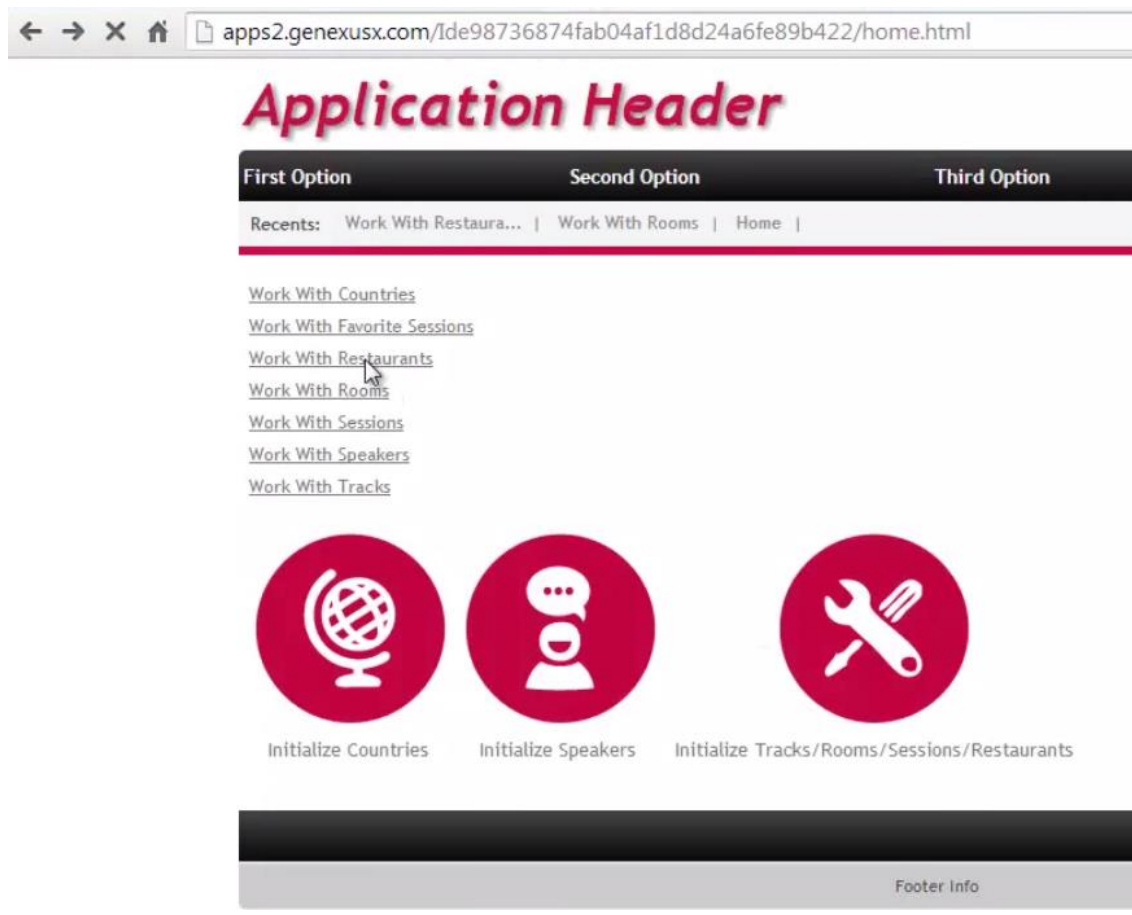
## Application Header

First Option	Second Option	Third Option
Recents: Home		
<a href="#">Work With Countries</a>		
<a href="#">Work With Favorite Sessions</a>		
<a href="#">Work With Restaurants</a>		
<a href="#">Work With Rooms</a>		
<a href="#">Work With Sessions</a>		
<a href="#">Work With Speakers</a>		
<a href="#">Work With Tracks</a>		
		
Initialize Countries	Initialize Speakers	Initialize Tracks/Rooms/Sessions/Restaurants

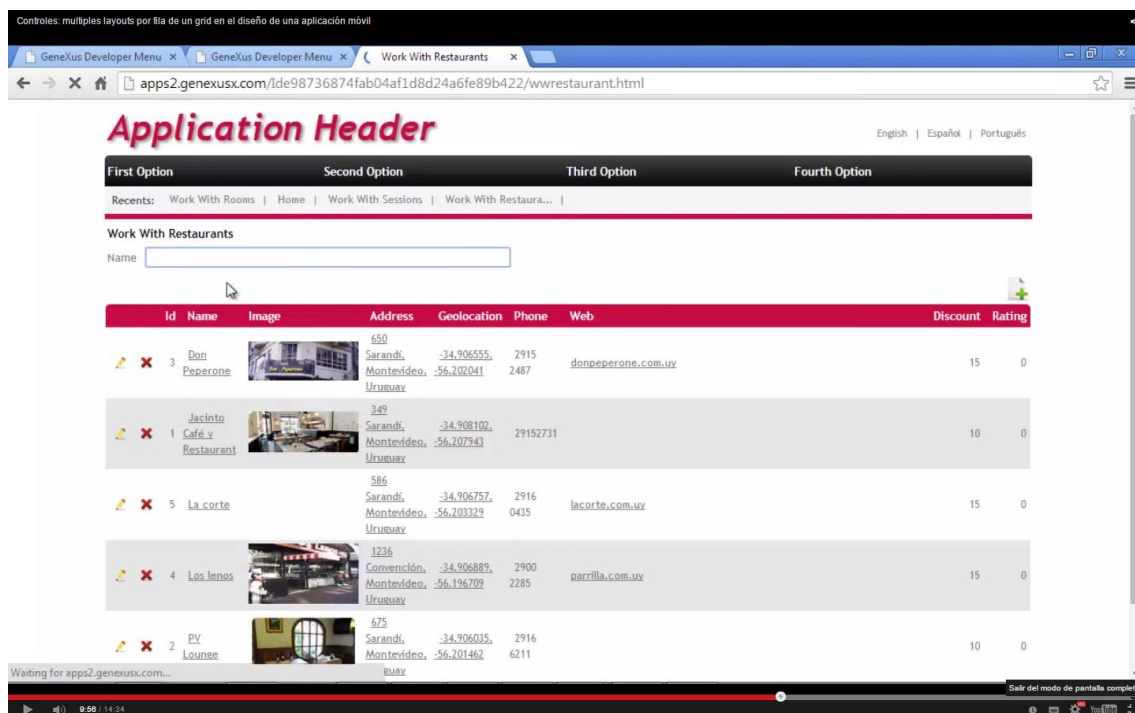
Footer Info



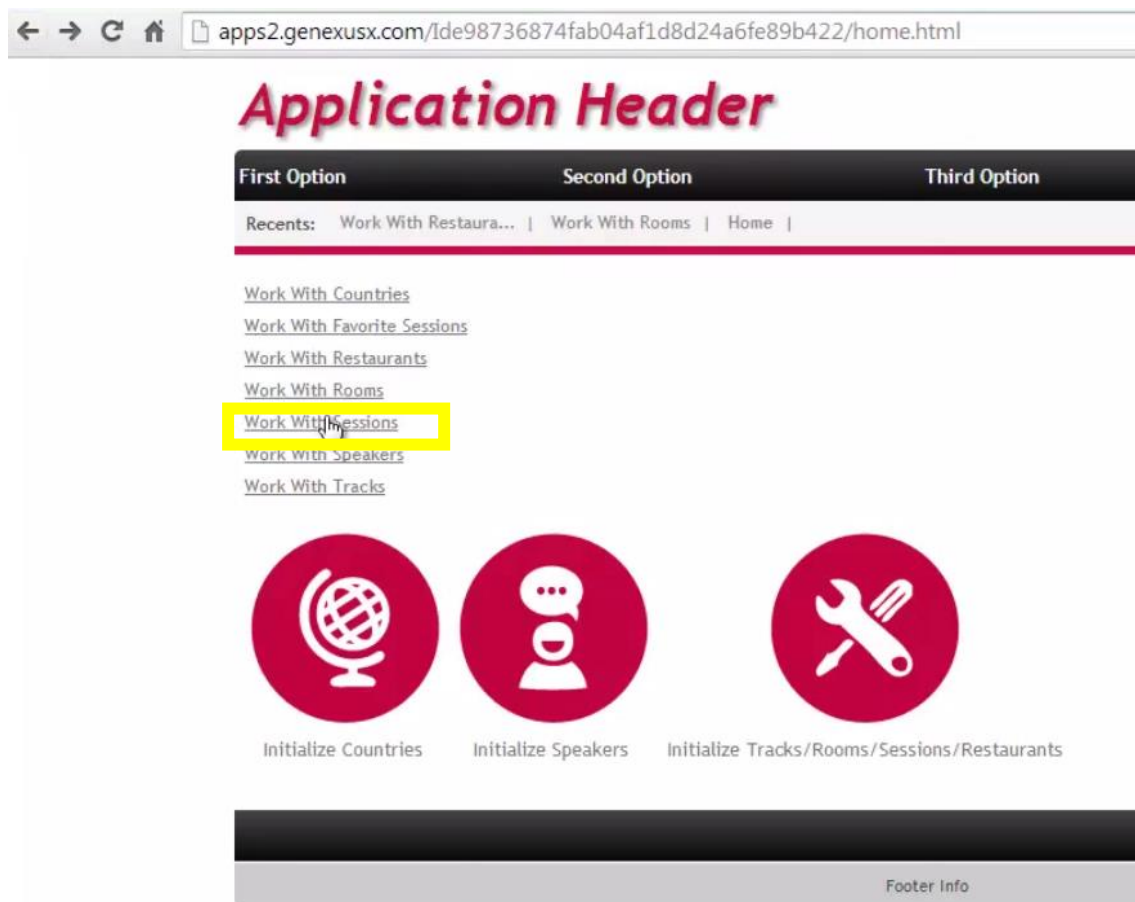
Para então quando você invocar os distintos work with







Vejamos a informação inicializada



apps2.genexus.com/Idc98736874fab04af1d8d24a6fe89b422/wwsession.html

## Application Header

English | Español | Português

First Option      Second Option      Third Option      Fourth Option

Recents: Work With Rooms | Home | Work With Restaurants | Work With Sessions |

Work With Sessions

Name

Id	Name	Description	Room Id	Room Name	Initial Date	Final Date	Initial Time	End Time	Speakers	Is Keynote
9	<a href="#">Android and the GeneXus Smart Device Generator</a>	Summary of the Android platform, its features and current state. Find out the details of the GeneXus Smart Device Generator for Android. The Android operating System currently leads the global S&T Device market with more than a 59% share and 1.3 million new Android devices activated on a daily basis. With the GeneXus Smart Device Generator you can create applications for this popular platform. Find out the details of this platform and the GeneXus generator.	4	BallRoom B	10/01/14	10/01/14	10:30 AM	11:00 AM	Cardozo Armando	<input type="checkbox"/>
8	<a href="#">Coffee and SD Projects</a>	We invite you over a cup of coffee to talk about your projects for Smart Devices and our experience with them from the technical aspects to the strategies we follow for their development.	4	BallRoom B	10/01/14	10/01/14	09:30 AM	10:00 AM	Gonzalez Guillermo, Jodal Sebastian,	<input type="checkbox"/>

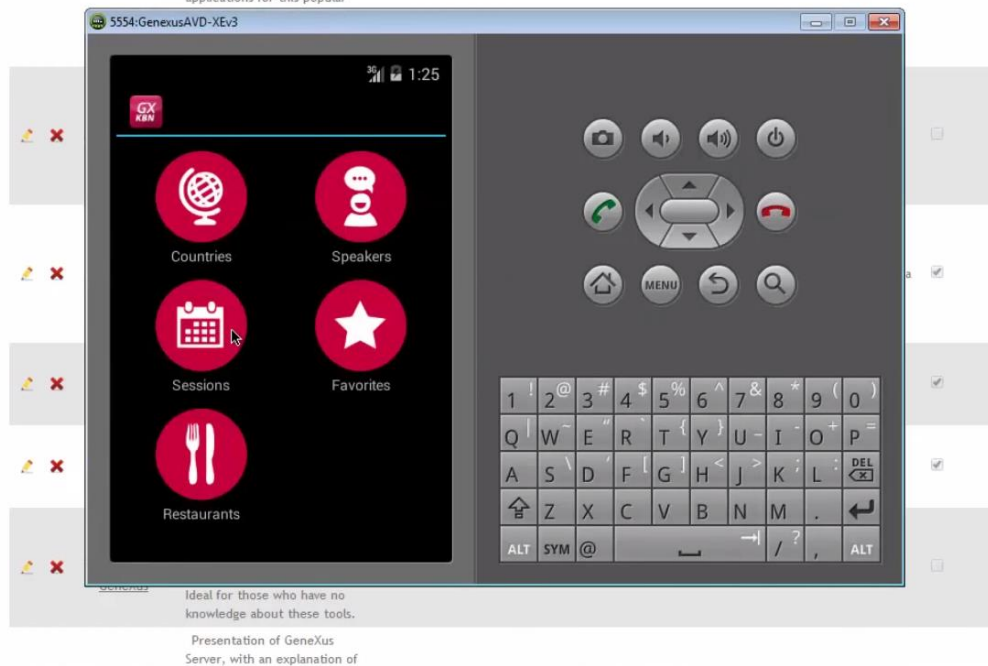
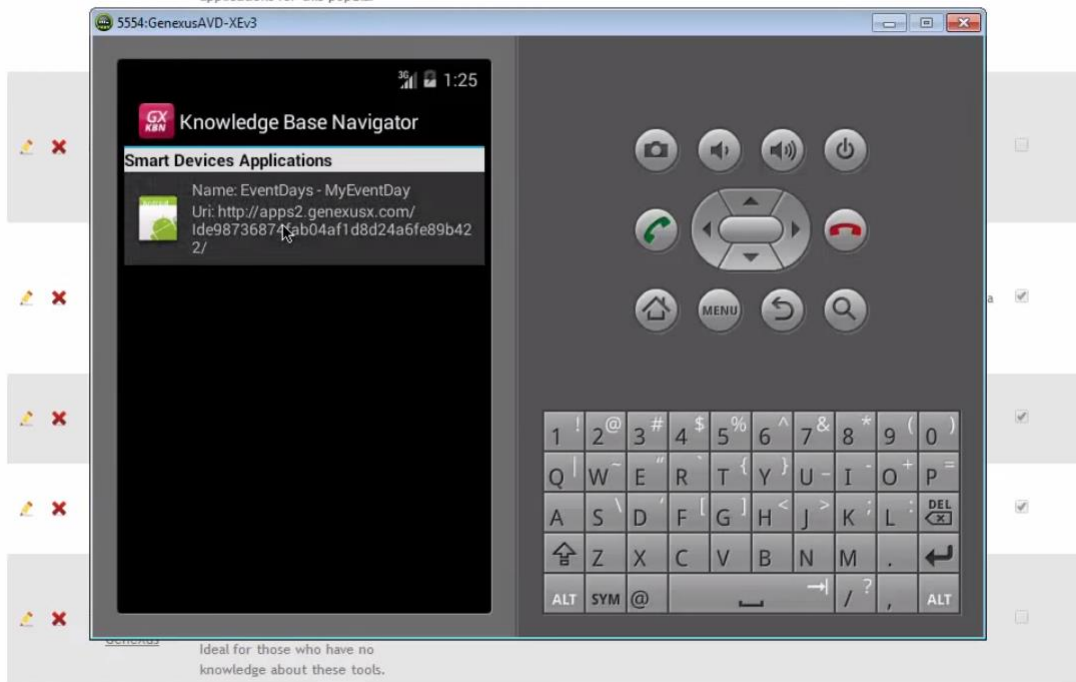
Notemos em alguma conferência que seja Keynote. Por exemplo esta : a 1, “GeneXus and the technology context” de Nicolás Jodal, que tem esta descrição aqui:

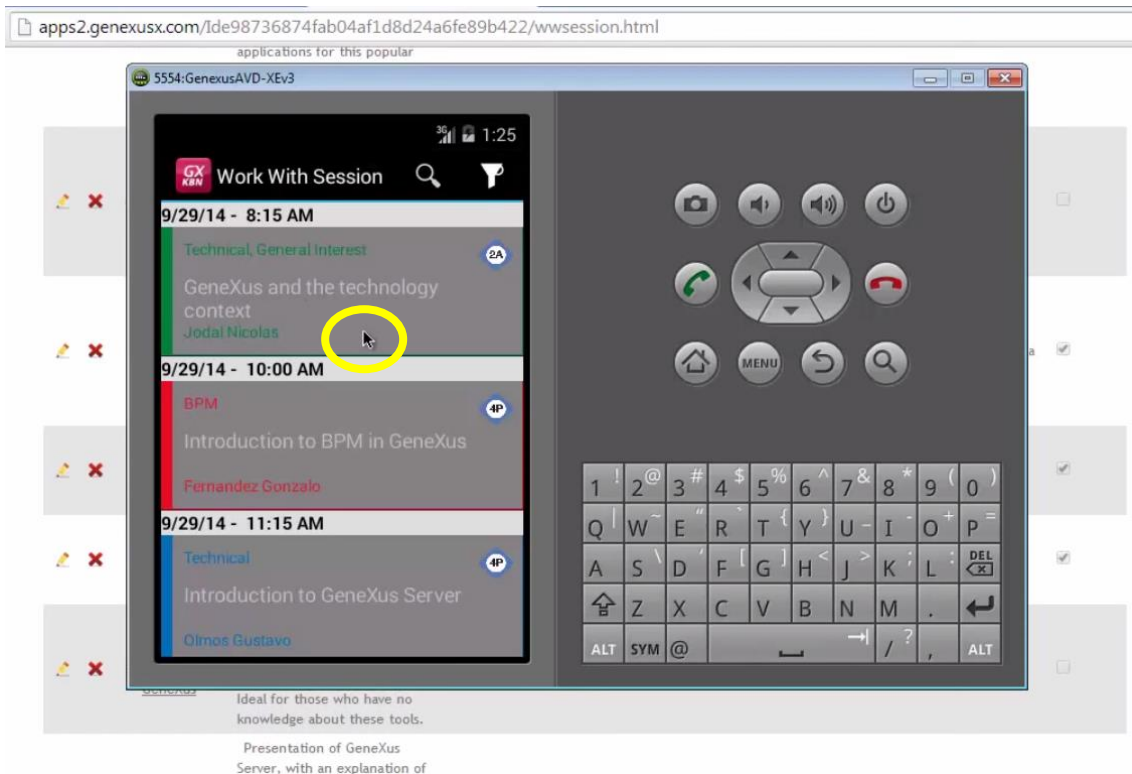
apps2.genexus.com/Idc98736874fab04af1d8d24a6fe89b422/wwsession.html

8	<a href="#">Coffee and SD Projects</a>	We invite you over a cup of coffee to talk about your projects for Smart Devices and our experience with them from the technical aspects to the strategies we follow for their development.	4	BallRoom B	10/01/14	10/01/14	09:30 AM	10:00 AM	Gonzalez Guillermo, Jodal Sebastian, Cardozo Armando	<input type="checkbox"/>
4	<a href="#">Design and Development: searching for integration</a>	What is the process we currently apply to develop SD applications? We will be sharing the experience of the past year in design and development for SD applications.	3	BallRoom A	09/30/14	09/30/14	09:00 AM	09:30 AM	Cimas Alejandro, Guedes Lucia, Bonilla Fabian, Blengio Alejandro	<input checked="" type="checkbox"/>
1	<a href="#">GeneXus and the technology context</a>	Smart Devices, Windows 8, GeneXus X Evolution 2 and GeneXus Tilo. Questions, answers and overview.	3	BallRoom A	09/29/14	09/29/14	09:15 AM	10:00 AM	Jodal Nicolas	<input checked="" type="checkbox"/>
5	<a href="#">GeneXus: reviewing its theoretical foundations</a>	Theoretical foundations 23 years later	2	Renoir	09/30/14	09/30/14	11:45 AM	12:30 PM	Gonda Breogan	<input checked="" type="checkbox"/>
2	<a href="#">Introduction to BPM in GeneXus</a>	Presentation of the version GXflow X Evolution 2 that allows developing applications based on Workflow & BPM. Ideal for those who have no knowledge about these tools.	1	Picasso	09/29/14	09/29/14	11:00 AM	11:30 AM	Fernandez Gonzalo	<input type="checkbox"/>

Presentation of GeneXus Server, with an explanation of

Agora testemos o mesmo “work with” mas para a aplicação móvel

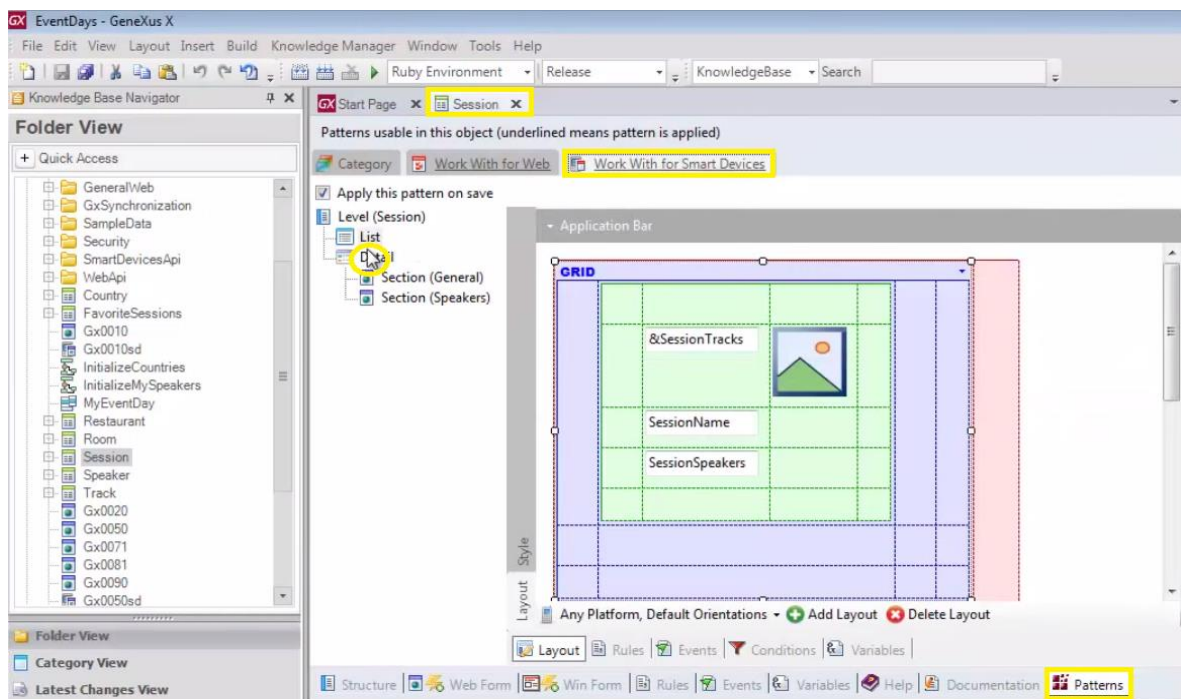




Vemos que para essa conferência, não está mostrando a descrição, apesar de ser Keynote.

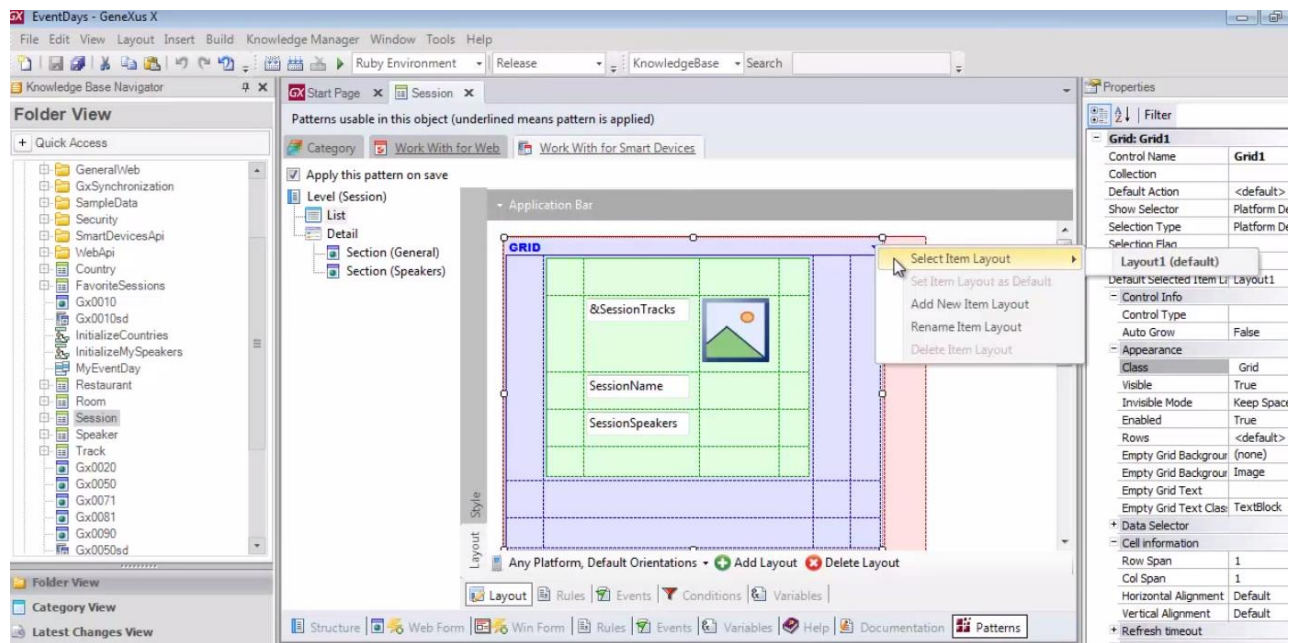
Queremos que ao carregar cada linha do grid, utilize um layout ou outro, dependendo do valor desse atributo.

Vamos ao work with, no list:

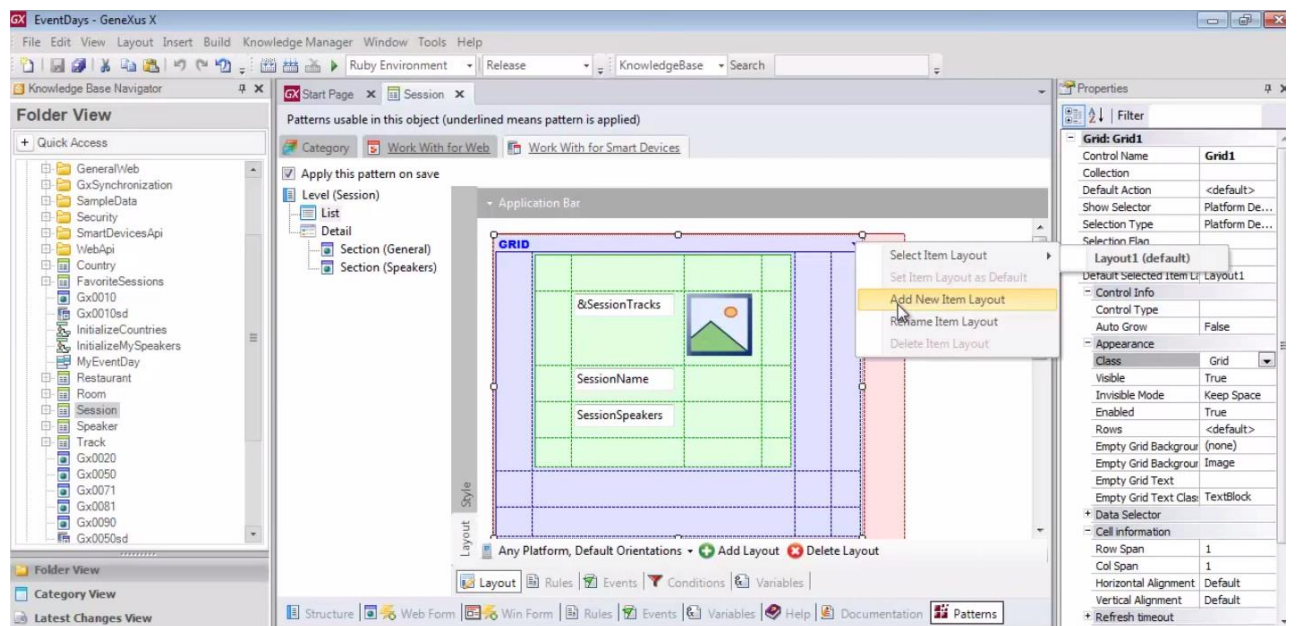


E aqui vemos que só temos unicamente um layout no momento.



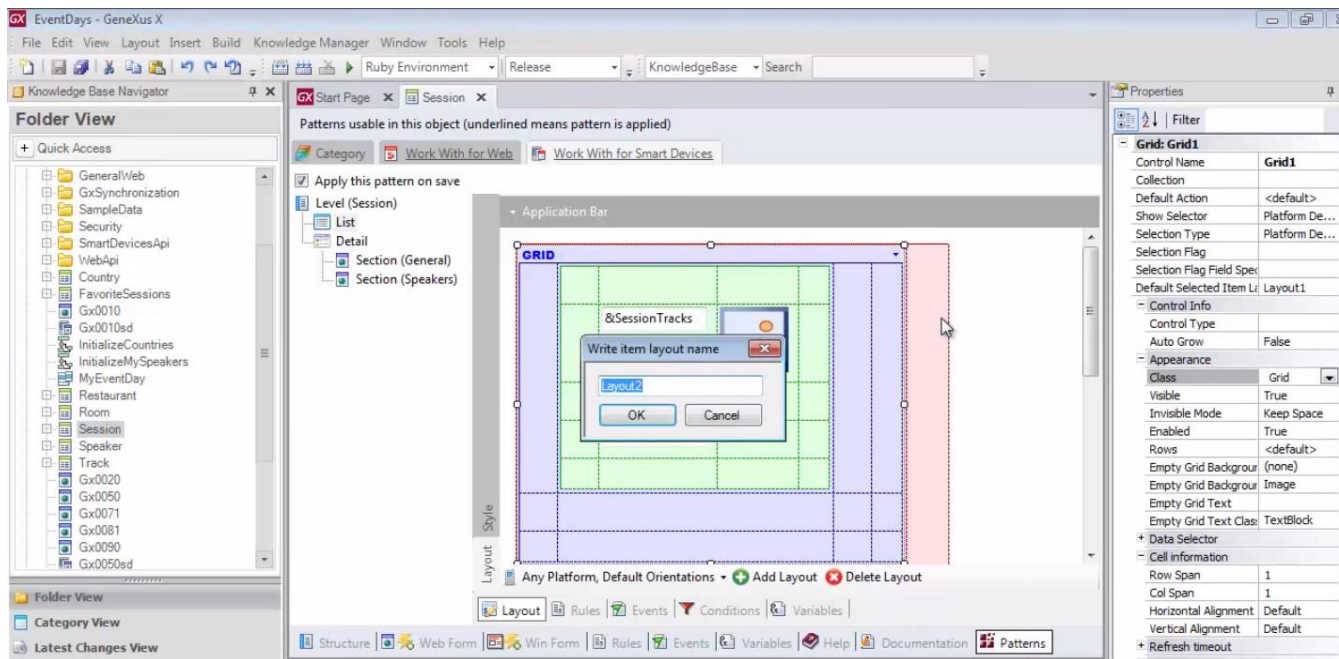


Vamos adicionar um novo layout

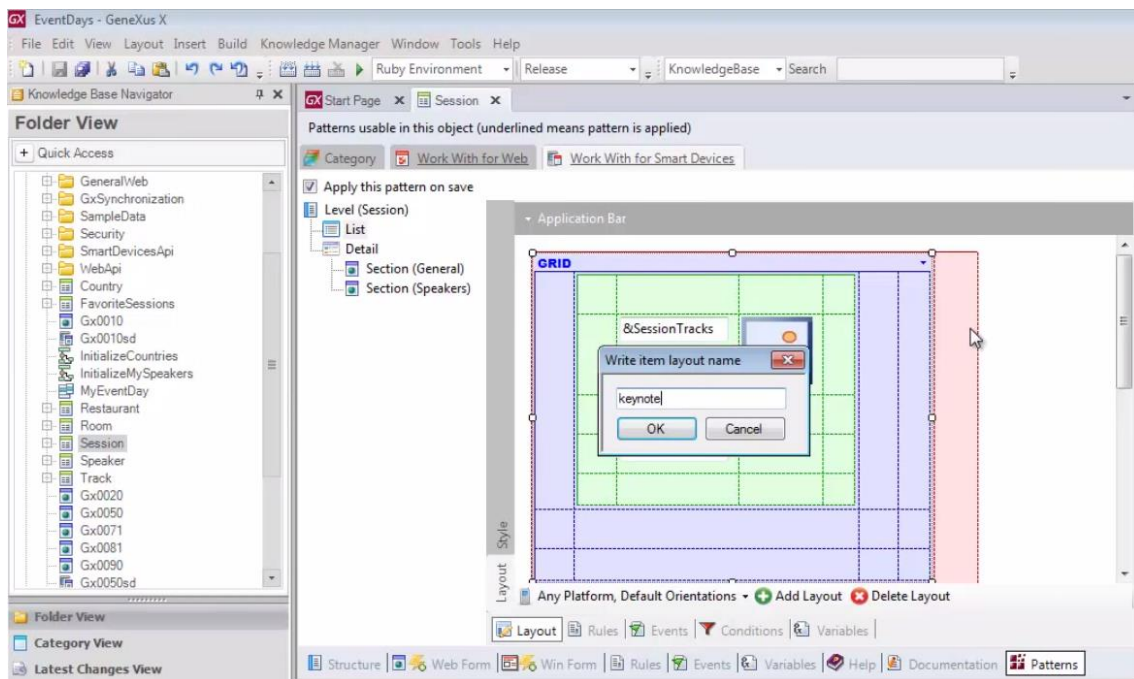


Vamos adicionar um novo layout

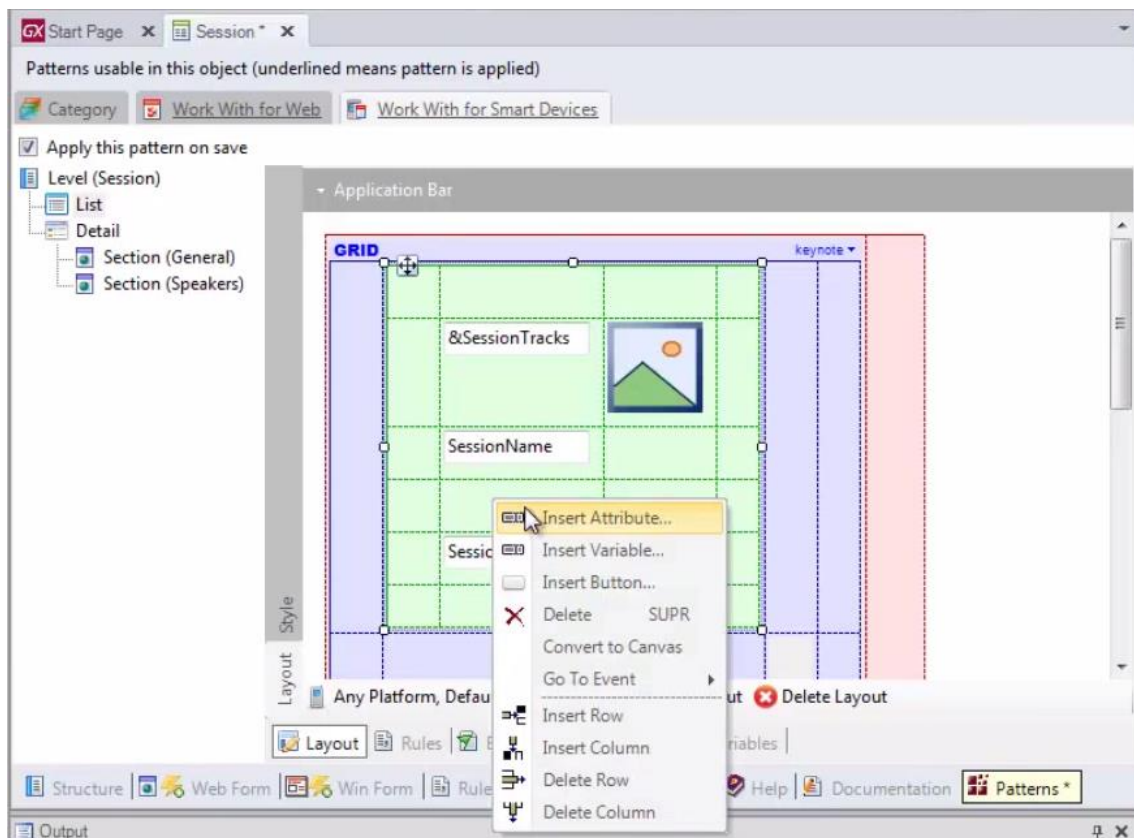
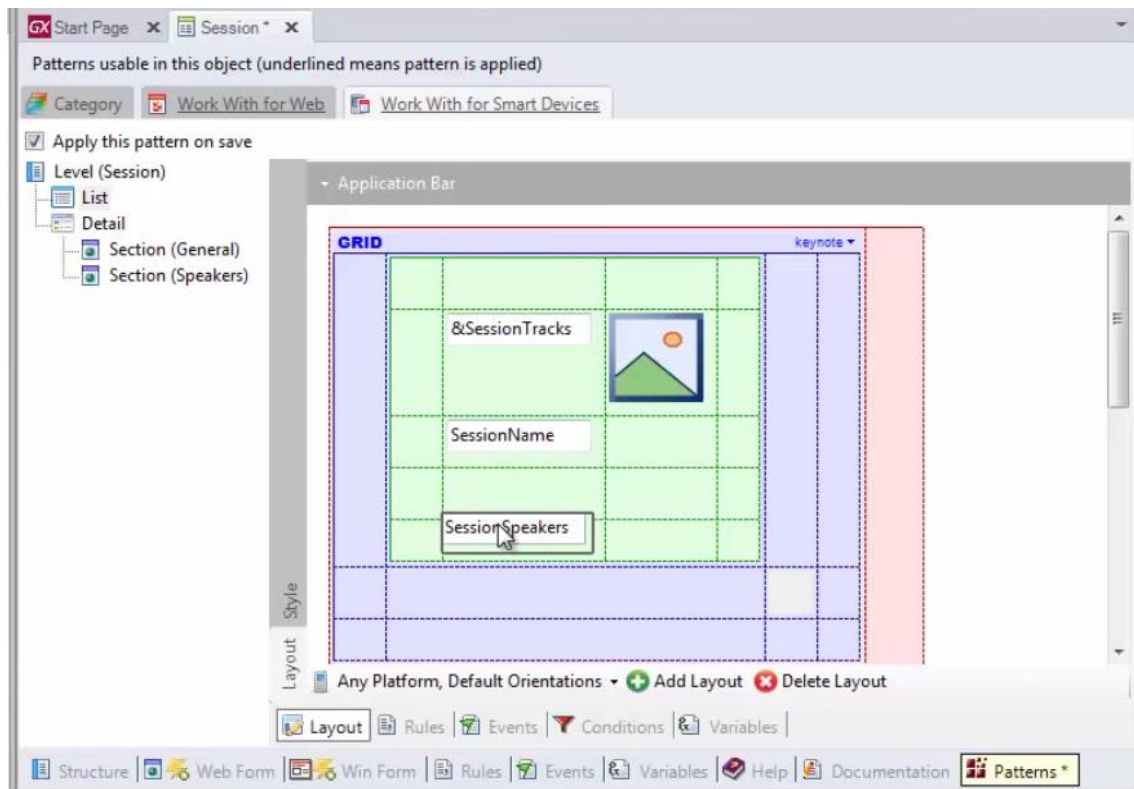


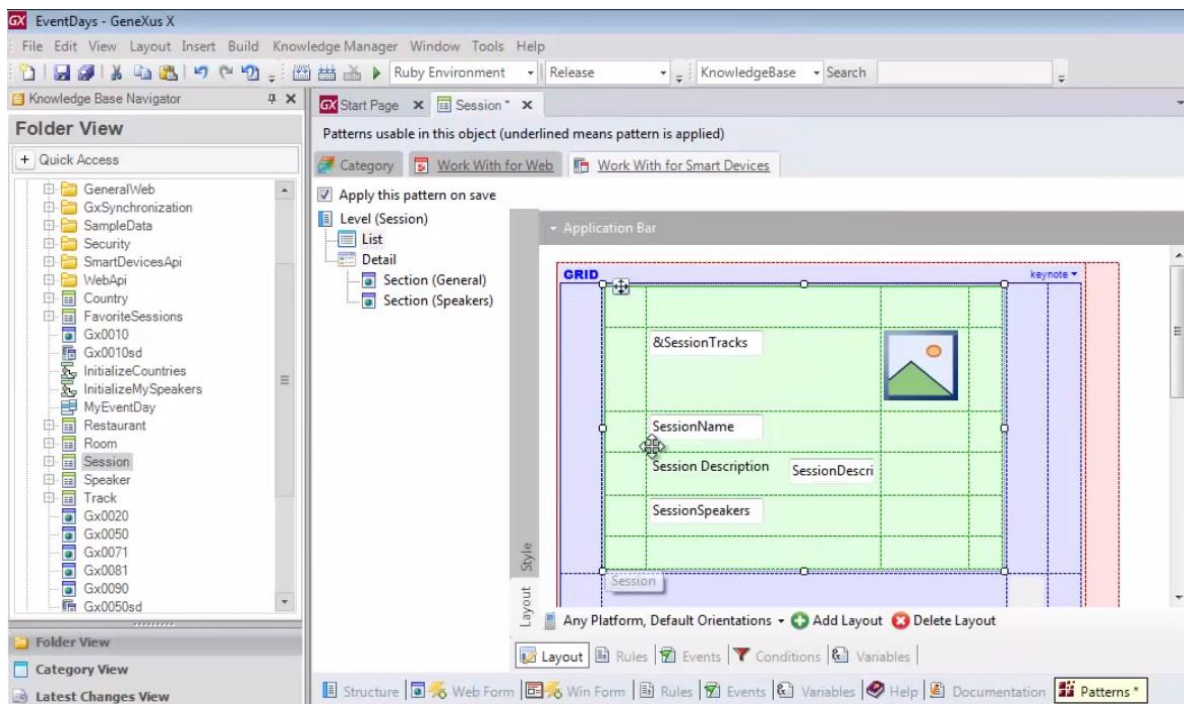
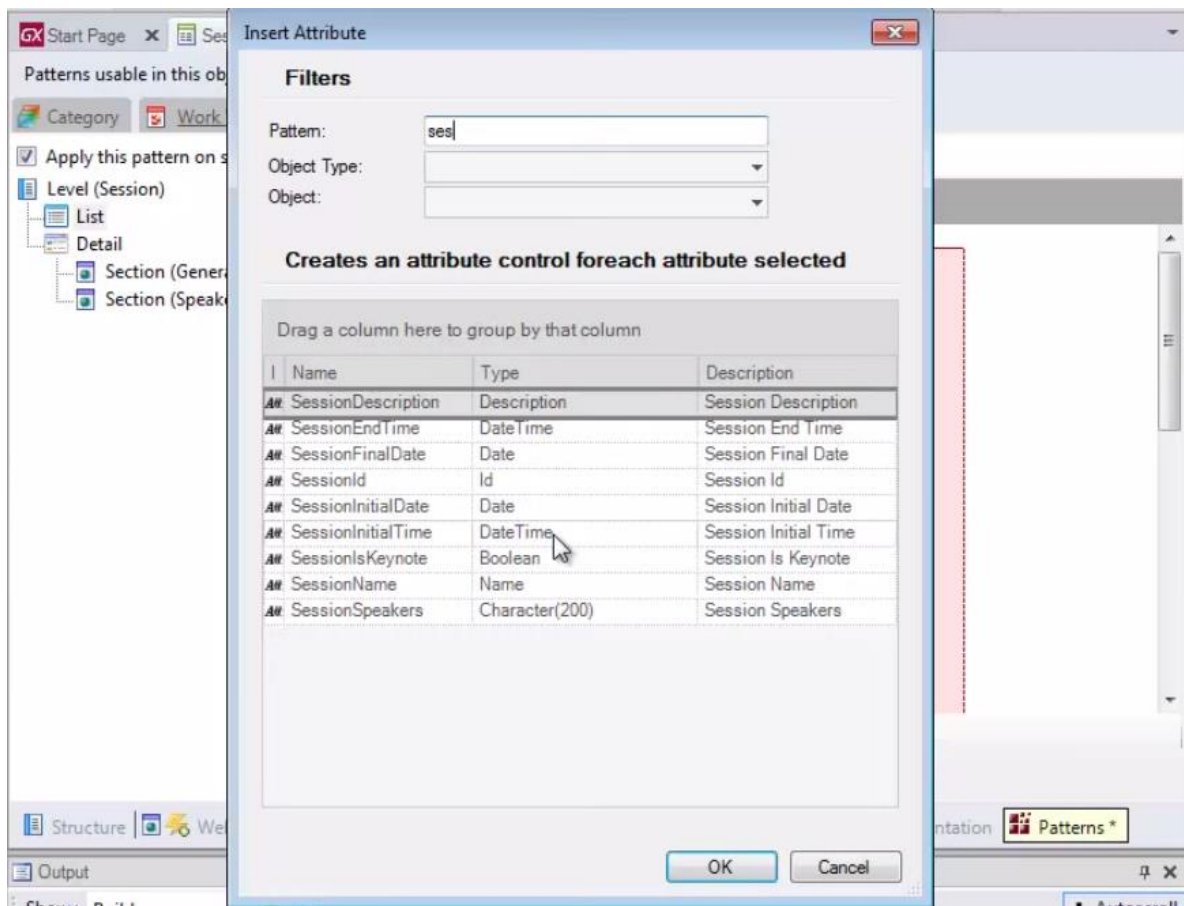


Que iremos chamar de: keynote

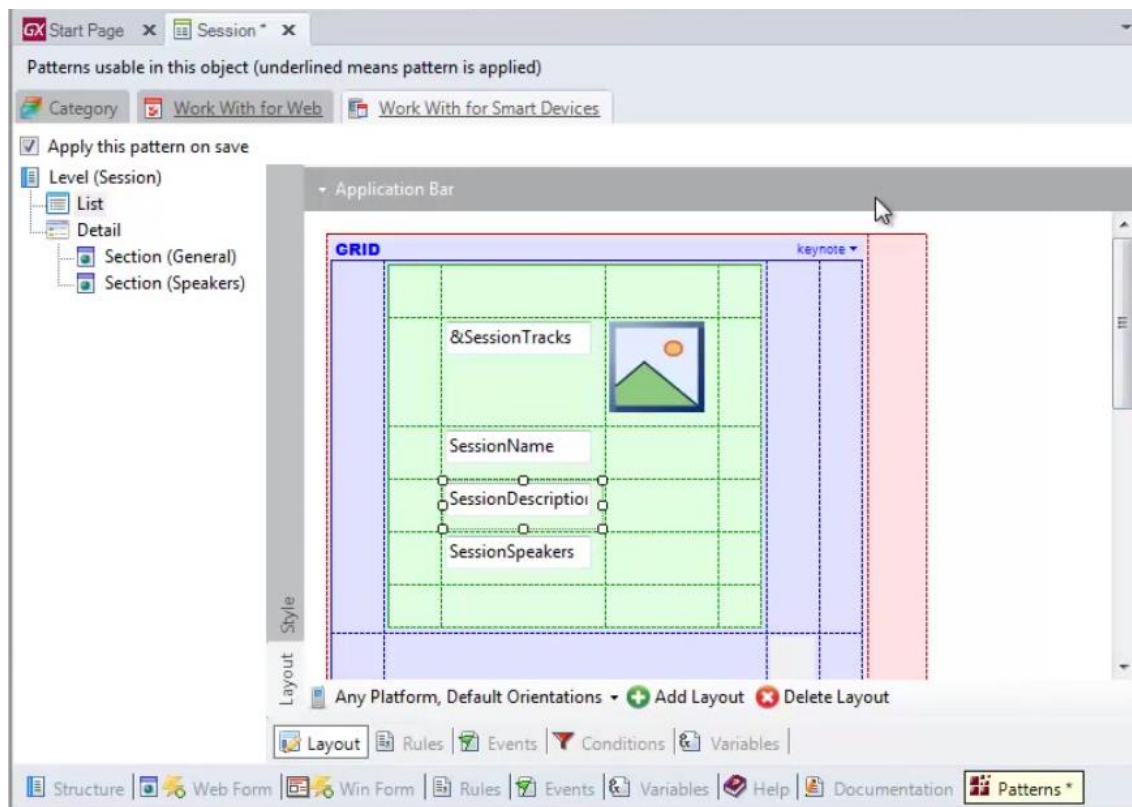
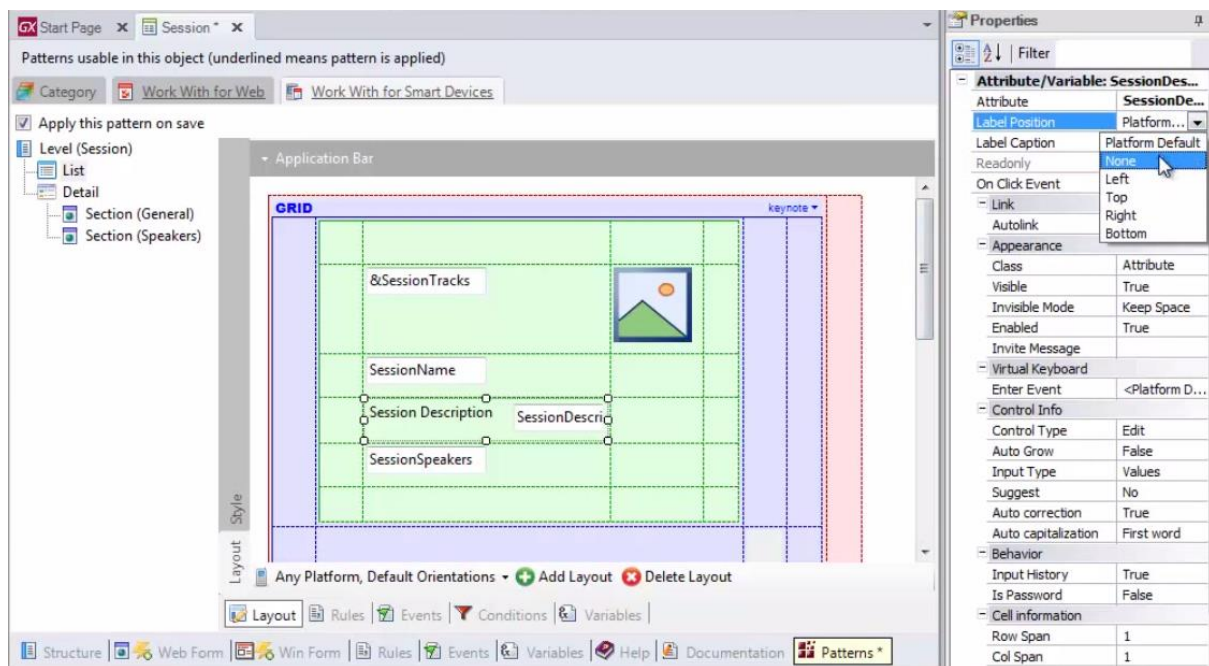


E vamos simplesmente adicionar a descrição: SessionDescription



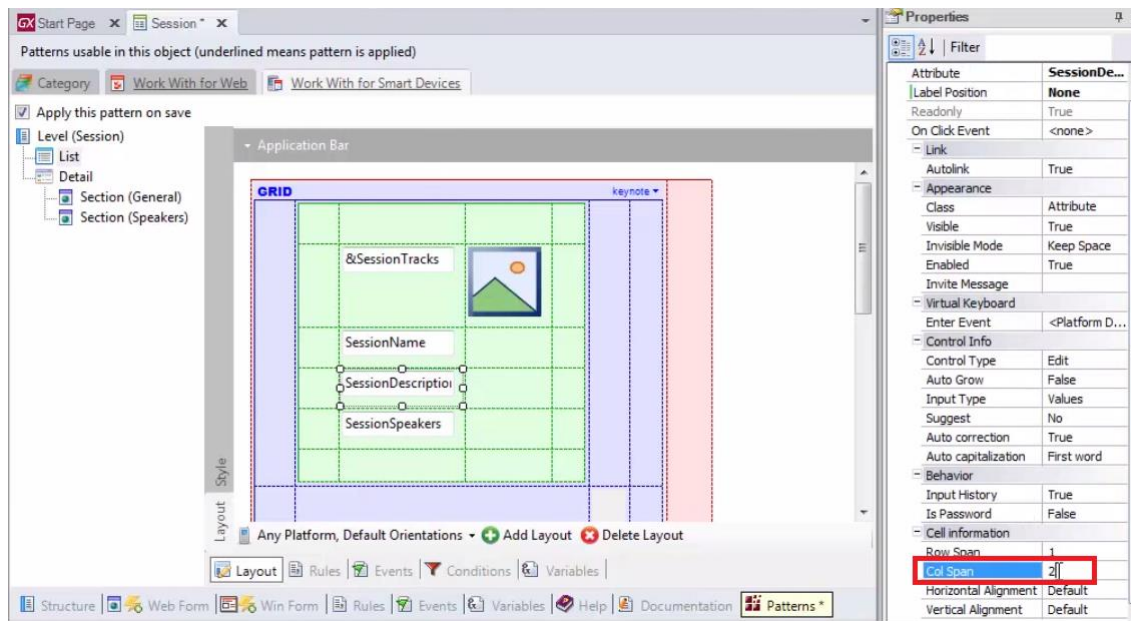




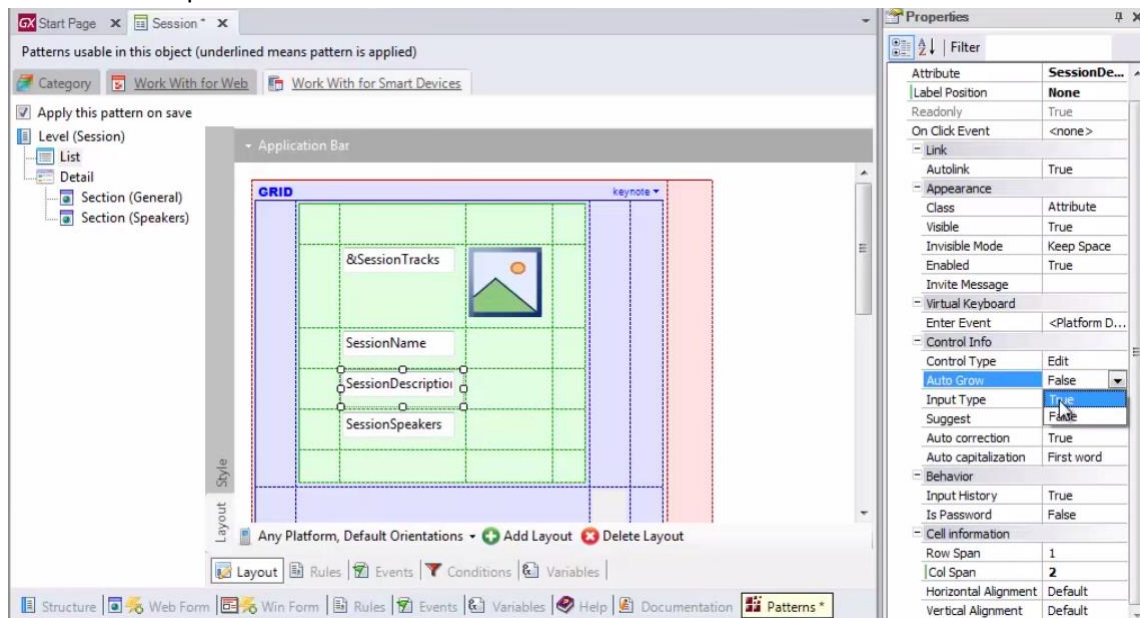


No qual vamos permitir expandir pelas 2 colunas.



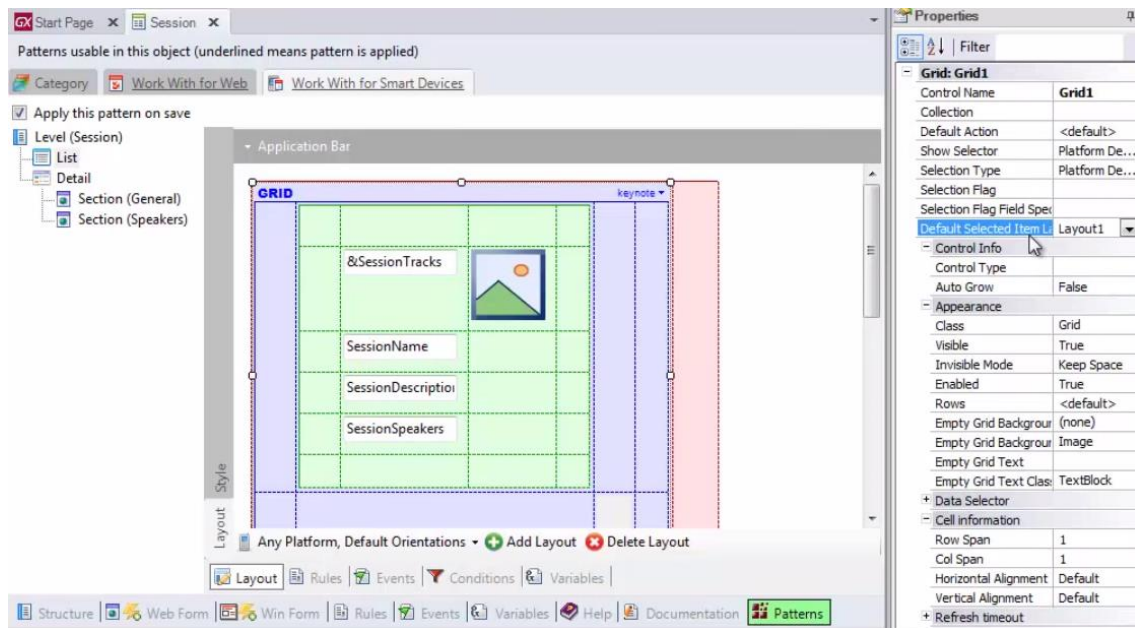


Assim como expandir de acordo com sua necessidade.



Agora então temos 2 layouts.

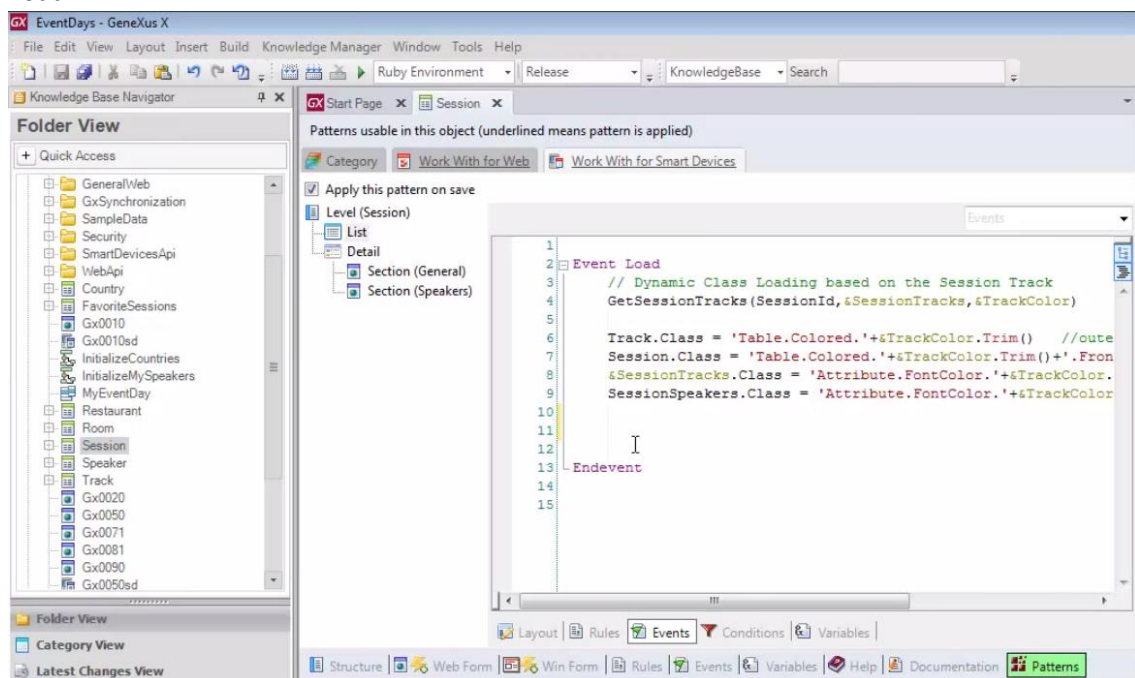
Se observarmos a propriedade do grid que indica qual irá ser o layout default que irá carregar quando o item estiver selecionado.



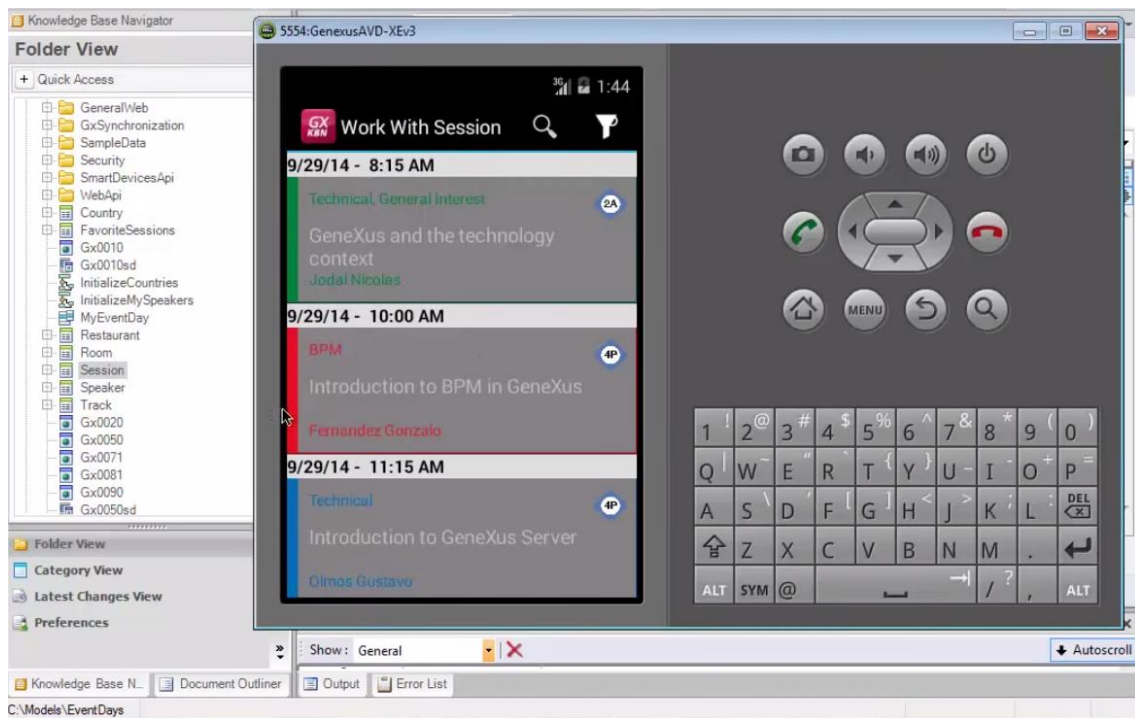
Vemos que é : Layout1, e não vamos modificar, porque não estamos interessados neste momento em alterar o layout de acordo com a seleção.

Estamos interessados em mudar o layout de acordo com o conteúdo do elemento da informação que está sendo carregada em cada oportunidade.

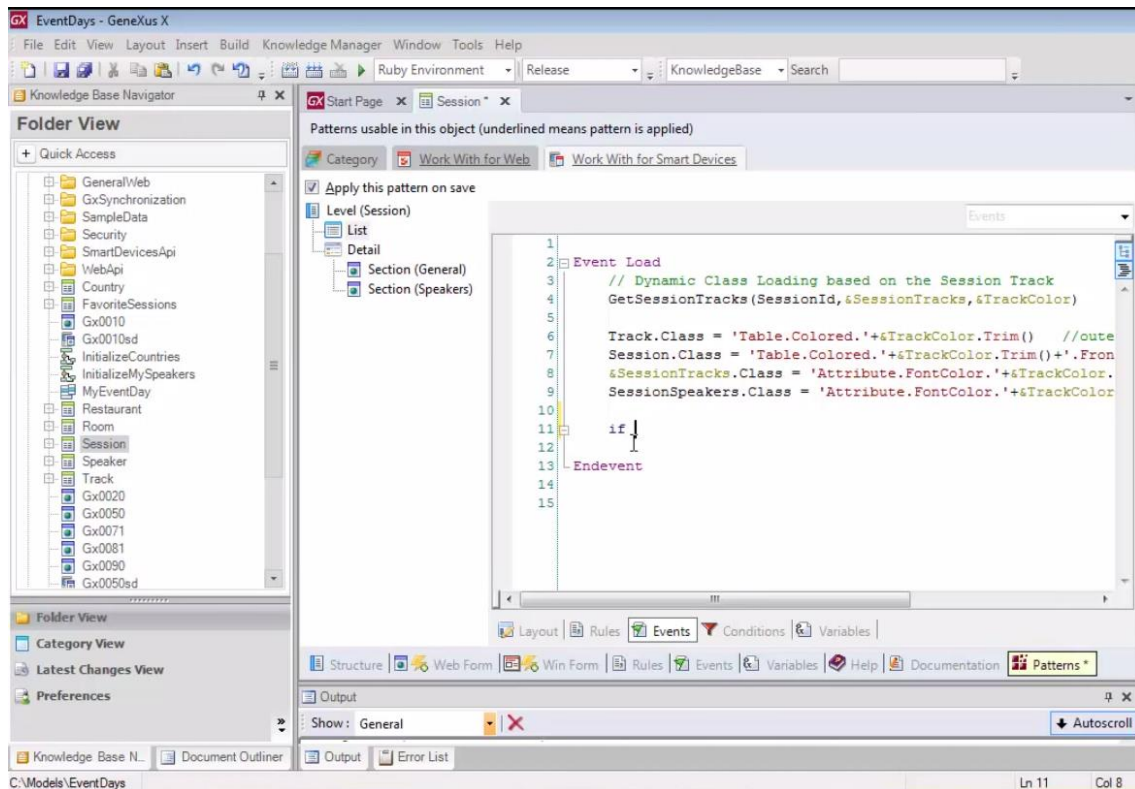
Portanto, isto tem que ser feito dinamicamente no lugar apropriado, isto é: dentro do evento Load.



Este evento já estava programando as classes que irão assumir os diferentes controles que estão no layout relacionados com o efeito colorido das tabelas.

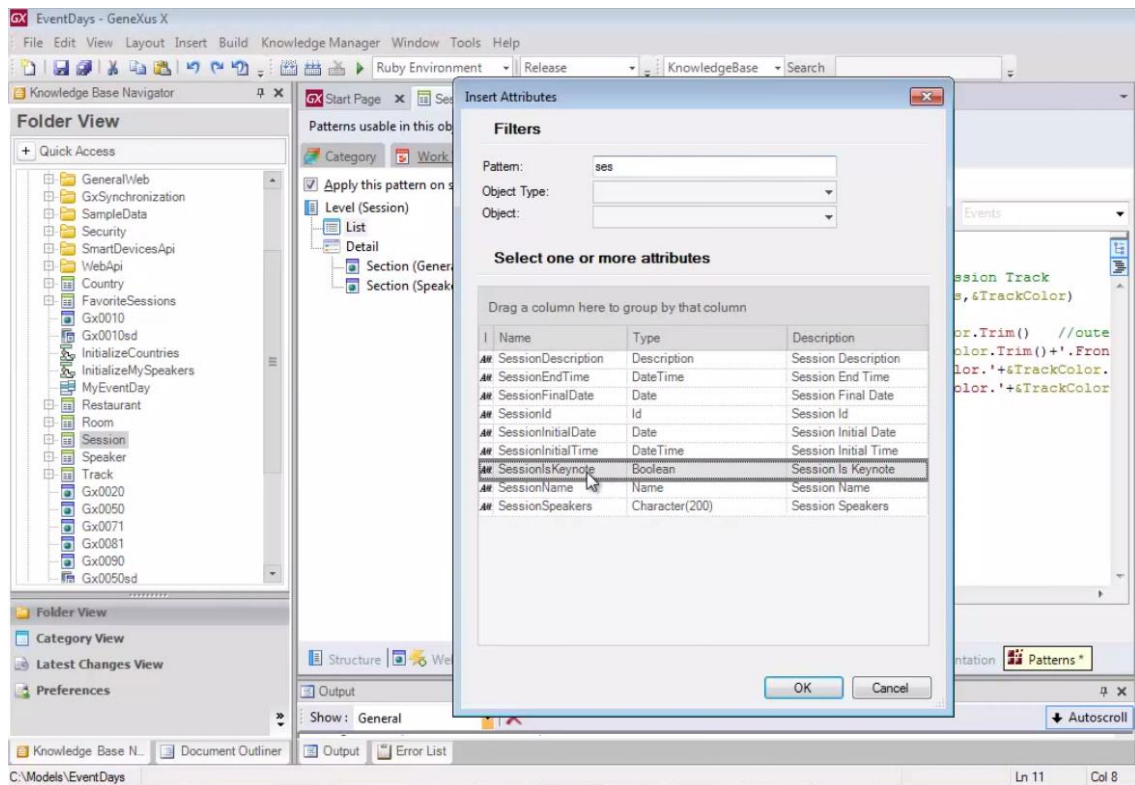


Agora então, vamos especificar.

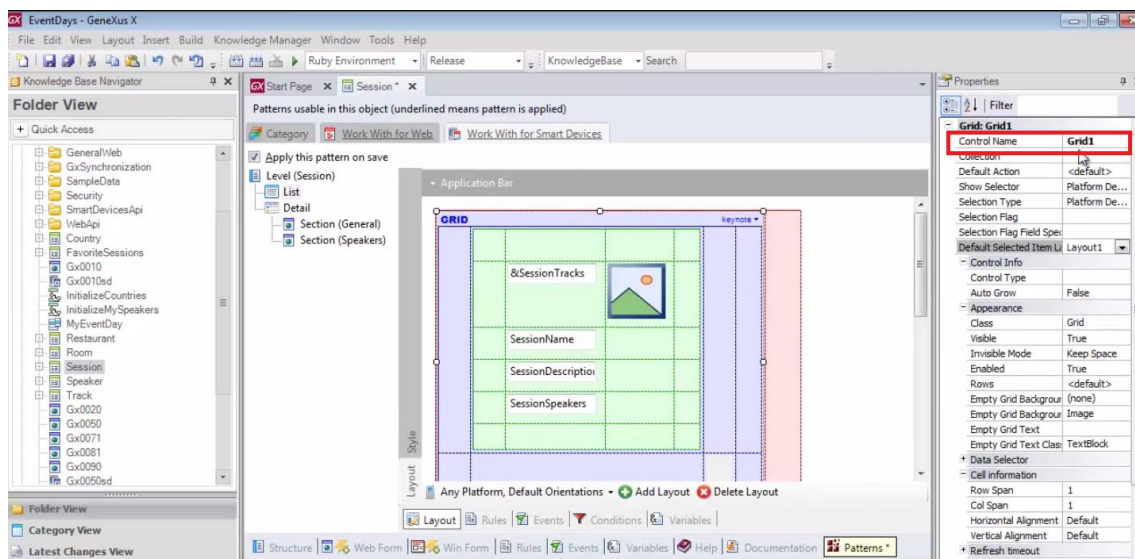


que se o atributo SessionIsKeynote

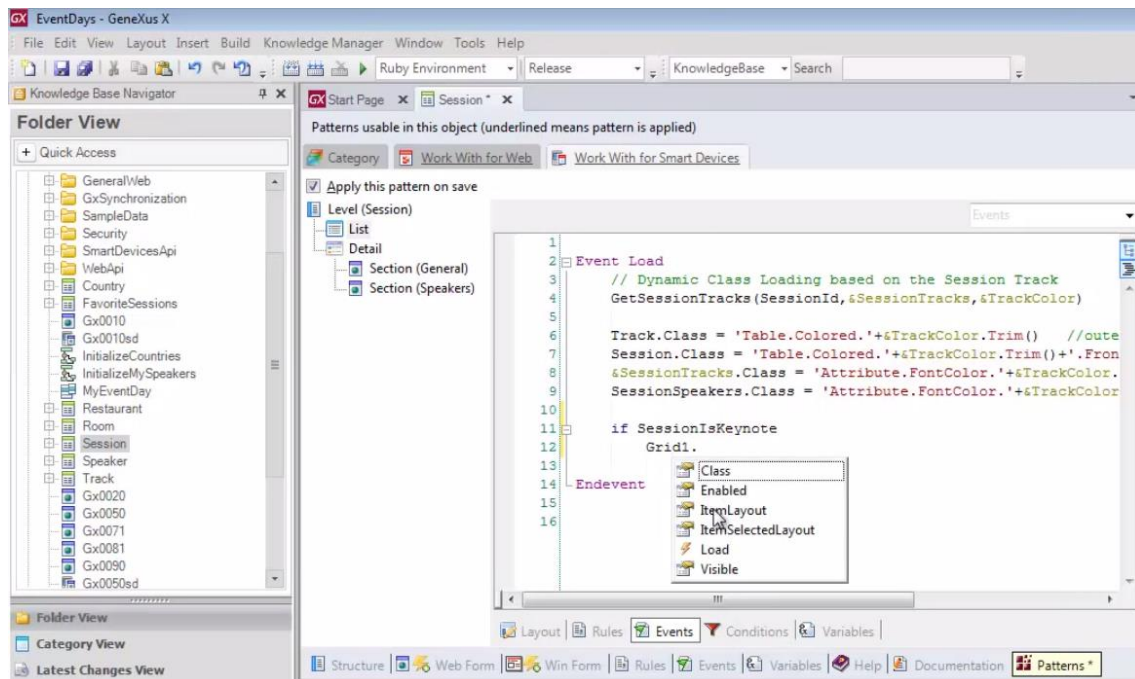




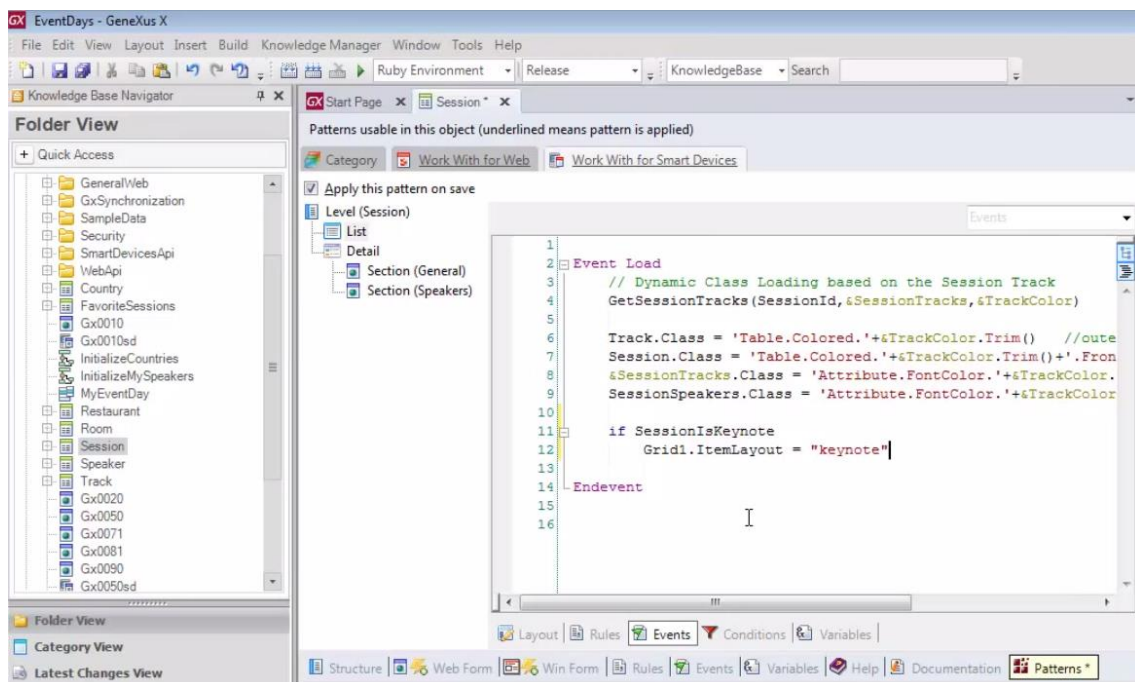
for True, então o grid de nome **Grid1**





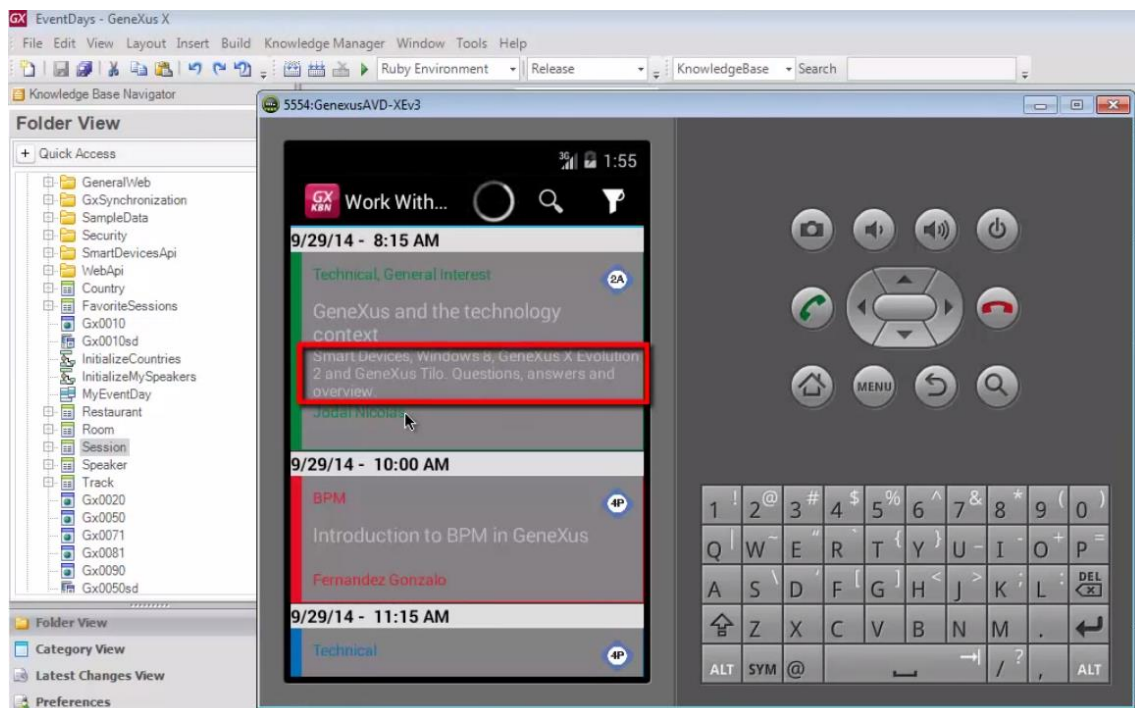
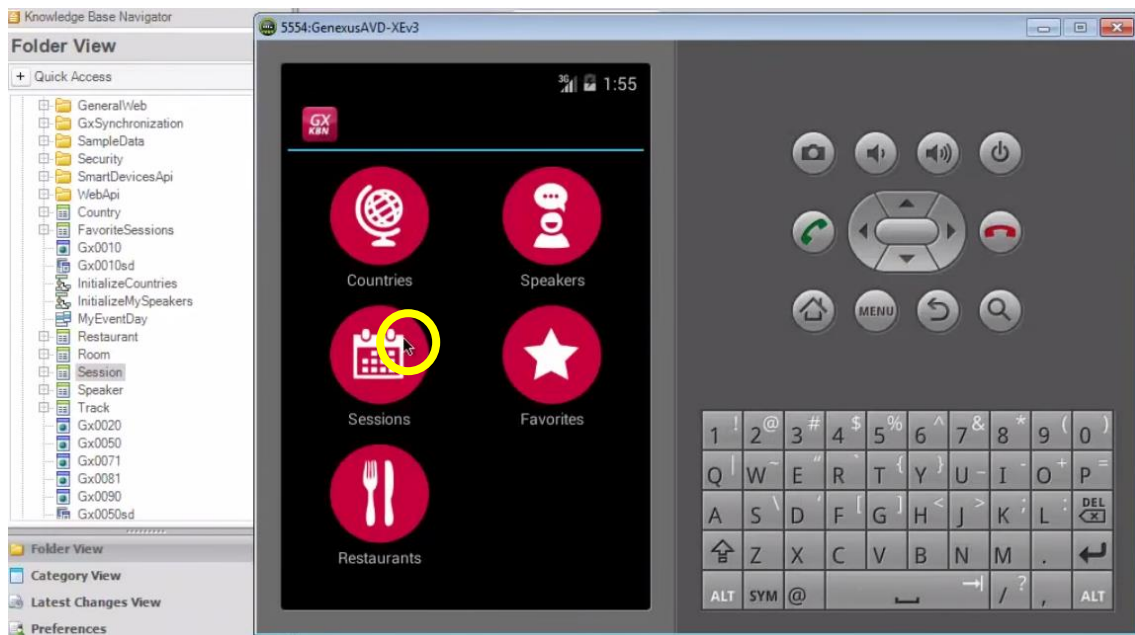


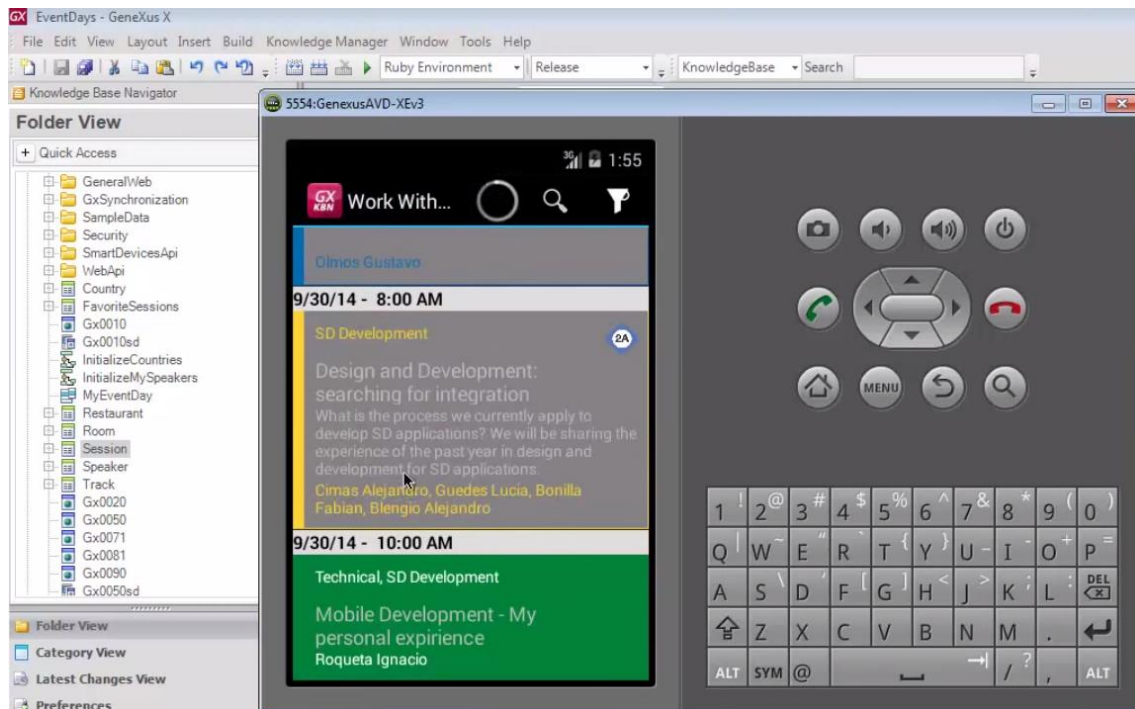
para o item que está carregando nessa oportunidade, vamos atribuir o layout de nome “keynote”



E caso contrário, não indicamos nada, o que deverá significar que será carregado com o layout padrão.





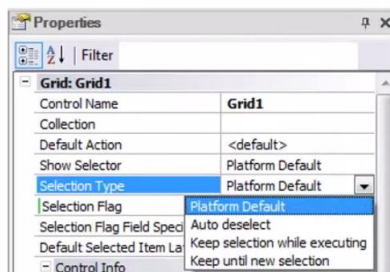


Designing

GeneXus™

Controls in layouts

Multiple layouts per row



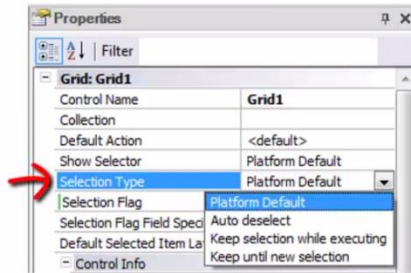
- Platform Default:**
- Uses the platform default behavior.
  - This is the default value.
- 
- Auto Deselect:**
- The grid's Highlighted Background color is shown while the user interacts with the control.
  - The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.
- 
- Keep Selection While Executing:**
- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
  - The selection ends when the Default Action ends.
  - This value only makes sense when the Default Action of the Grid is defined.
  - If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.
- 
- Keep Until New Selection:**
- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
  - The selection ends when another selection is done.
  - This value only makes sense when the [Default Selected Item Layout](#) is defined.
  - If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again

Por último mencionemos que entre as propriedades do grid, encontra-se **Selection Type**



## Controls in layouts

## Multiple layouts per row



## Platform Default:

- Uses the platform default behavior.
- This is the default value.

## Auto Deselect:

- The grid's Highlighted Background color is shown while the user interacts with the control.
- The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.

## Keep Selection While Executing:

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when the Default Action ends.
- This value only makes sense when the Default Action of the Grid is defined.
- If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.

## Keep Until New Selection:

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when another selection is done.
- This value only makes sense when the [Default Selected Item Layout](#) is defined.
- If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again.

que permite indicar o comportamento quando seleciona uma linha (um item) do grid.

Observe os valores

Designing

GeneXus™

Controls in layouts

Multiple layouts per row

Platform Default:

- Uses the platform default behavior.
- This is the default value.

Auto Deselect:

- The grid's Highlighted Background color is shown while the user interacts with the control.
- The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.

Keep Selection While Executing:

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when the Default Action ends.
- This value only makes sense when the Default Action of the Grid is defined.
- If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.

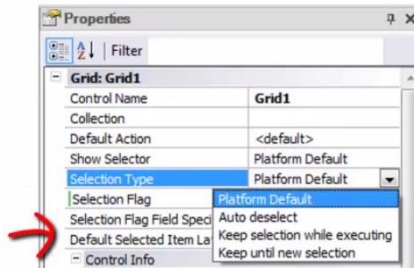
Keep Until New Selection:

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when another selection is done.
- This value only makes sense when the [Default Selected Item Layout](#) is defined.
- If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again.

Dependendo de cada um e de se foi definido um layout diferente do default na propriedade **Default Selected Item Layout** que havíamos visto

## Controls in layouts

## Multiple layouts per row

**Platform Default:**

- Uses the platform default behavior.
- This is the default value.

**Auto Deselect:**

- The grid's Highlighted Background color is shown while the user interacts with the control.
- The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.

**Keep Selection While Executing:**

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when the Default Action ends.
- This value only makes sense when the Default Action of the Grid is defined.
- If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.

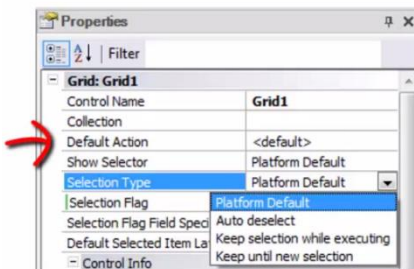
**Keep Until New Selection:**

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when another selection is done.
- This value only makes sense when the [Default Selected Item Layout](#) is defined.
- If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again

E também dependendo do valor da propriedade **Default Action**

## Controls in layouts

## Multiple layouts per row

**Platform Default:**

- Uses the platform default behavior.
- This is the default value.

**Auto Deselect:**

- The grid's Highlighted Background color is shown while the user interacts with the control.
- The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.

**Keep Selection While Executing:**

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when the Default Action ends.
- This value only makes sense when the Default Action of the Grid is defined.
- If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.

**Keep Until New Selection:**

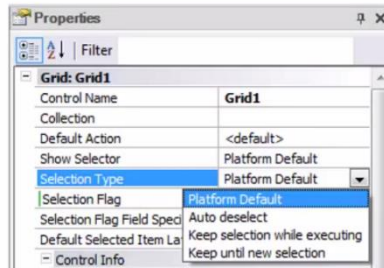
- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when another selection is done.
- This value only makes sense when the [Default Selected Item Layout](#) is defined.
- If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again

será o comportamento.

Se você quiser se aprofundar neste tema

## Controls in layouts

## Multiple layouts per row

**Platform Default:**

- Uses the platform default behavior.
- This is the default value.

**Auto Deselect:**

- The grid's Highlighted Background color is shown while the user interacts with the control.
- The [Default Selected Item Layout](#) is not going to be shown even if it is defined or not.

**Keep Selection While Executing:**

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when the Default Action ends.
- This value only makes sense when the Default Action of the Grid is defined.
- If an item is selected, and the user selects it again, the item is still going to be selected and the Default Action of the Grid is executed again.

**Keep Until New Selection:**

- If the [Default Selected Item Layout](#) is defined it is shown, if not, the grid's Highlighted Background color is shown.
- The selection ends when another selection is done.
- This value only makes sense when the [Default Selected Item Layout](#) is defined.
- If an item is selected, and the user selects it again, the item is going to be deselected and the Default Action is executed again

<http://wiki.genexus.com/commwiki/servlet/hwikipageid?24120>

Sugerimos que visite nossa wiki.

## Controls in layouts

Control Labels, tables, images

Grid: Multiple layouts per row

Control types

Detail &lt;section content&gt;

Canvas &amp; Transformations

Estudamos até aqui a possibilidade de definir vários layouts para as linhas de um grid e escolher qual será carregado em cada oportunidade. Paralelamente vimos como configurar as características da seleção.

No próximo vídeo, veremos a possibilidade de modificar para um controle, seu tipo, para que pareça e se comporte de uma maneira diferente da predefinida.

Controls in layouts

Control Labels, tables, images

Grid: Multiple layouts per row

Control types

Detail <section content>

Canvas & Transformations

**GeneXus™**  
TRAINING PROGRAM