

Components e objeto Master Panel

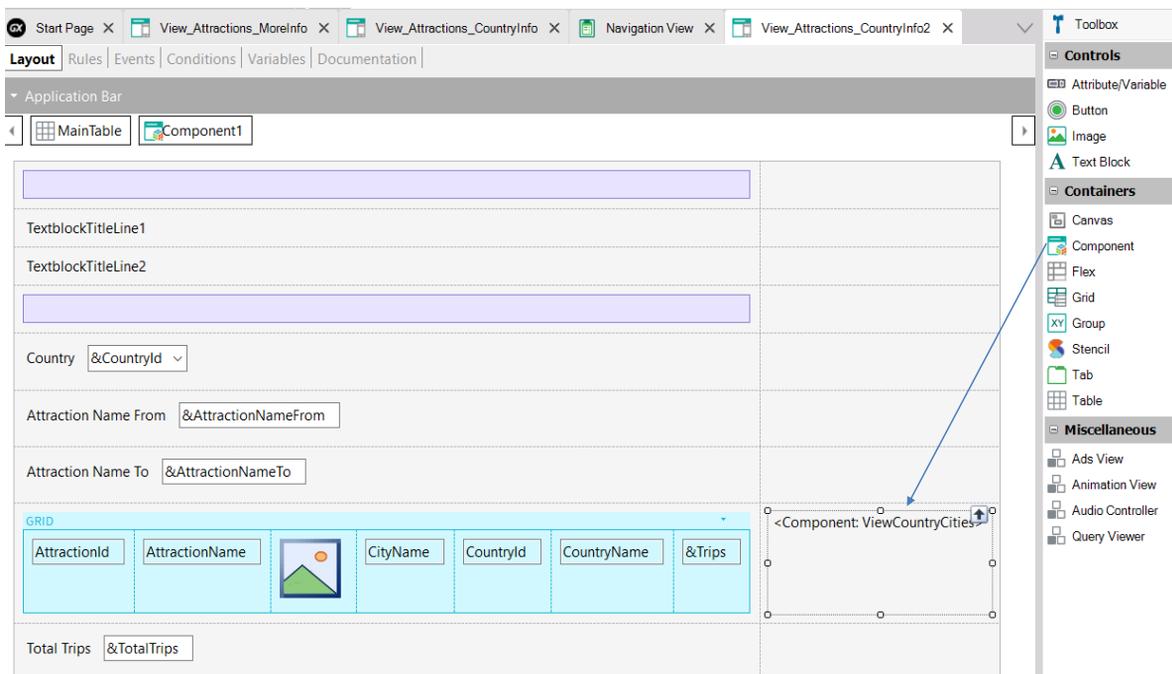
GeneXus™

Neste vídeo utilizaremos os components, que são containers que podemos incluir em um panel e associar outro panel a ele, de forma que será possível executar um objeto panel dentro de outro. Também veremos o objeto Master Panel, que nos permitirá definir uma tela container dentro da qual serão executadas todas as telas de nossa aplicação da agência de viagens, de forma que a aparência e contexto seja o mesmo para todas, por exemplo, que apareça sempre o logotipo da agência, entre outras coisas.

Components

O objeto panel permite encapsular funcionalidade por meio do uso de components, de forma similar aos webpanels com os webcomponents. Vejamos um exemplo de uso.

Uso de um panel dentro de outro panel



Vamos construir um objeto panel que mostra as atrações e os dados de um país selecionado, mas desta vez as informações serão exibidas utilizando um componente dentro do panel que estamos criando.

Para criar o panel que conterá o component, fazemos um Save As do View_Attractions_MoreInfo, com o nome View_Attractions_CountryInfo2.

À direita do grid de atrações, inserimos a partir da barra de ferramentas, um controle Component.

De forma similar ao que fizemos antes, quando clicamos sobre o nome de um país do grid, será aberta a informação desse país.

Criação do painel a ser usado como componente

The screenshot illustrates the development of a panel component in Genexus. The main workspace shows the 'Rules' tab with a parameter rule: `Parm(in:CountryId);`. Below the rules, the 'MainTable' component is visible, containing a 'CountryName' field and a 'GRID' with a 'CityName' field. Two detailed views of the data provider nodes are shown on the right:

- Data Provider ViewCountryCities_Level_Detail:**
 - Name: ViewCountryCities_Level_Detail
 - Description: ViewCountryCities_Level_Detail
 - Output: None
 - Devices: None
 - Environment: ViewCountryCities_Level_Detail
 - Spec: ViewCountryCities_Level_Detail
 - Version: ViewCountryCities_Level_Detail
 - Form Class: ViewCountryCities_Level_Detail
 - Program Name: ViewCountryCities_Level_Detail
 - Parameters: ViewCountryCities_Level_Detail
- Data Provider ViewCountryCities_Level_Detail_Grid1:**
 - Name: ViewCountryCities_Level_Detail_Grid1
 - Description: ViewCountryCities_Level_Detail_Grid1
 - Output: None
 - Devices: None
 - Environment: ViewCountryCities_Level_Detail_Grid1
 - Spec: ViewCountryCities_Level_Detail_Grid1
 - Version: ViewCountryCities_Level_Detail_Grid1
 - Form Class: ViewCountryCities_Level_Detail_Grid1
 - Program Name: ViewCountryCities_Level_Detail_Grid1
 - Parameters: ViewCountryCities_Level_Detail_Grid1

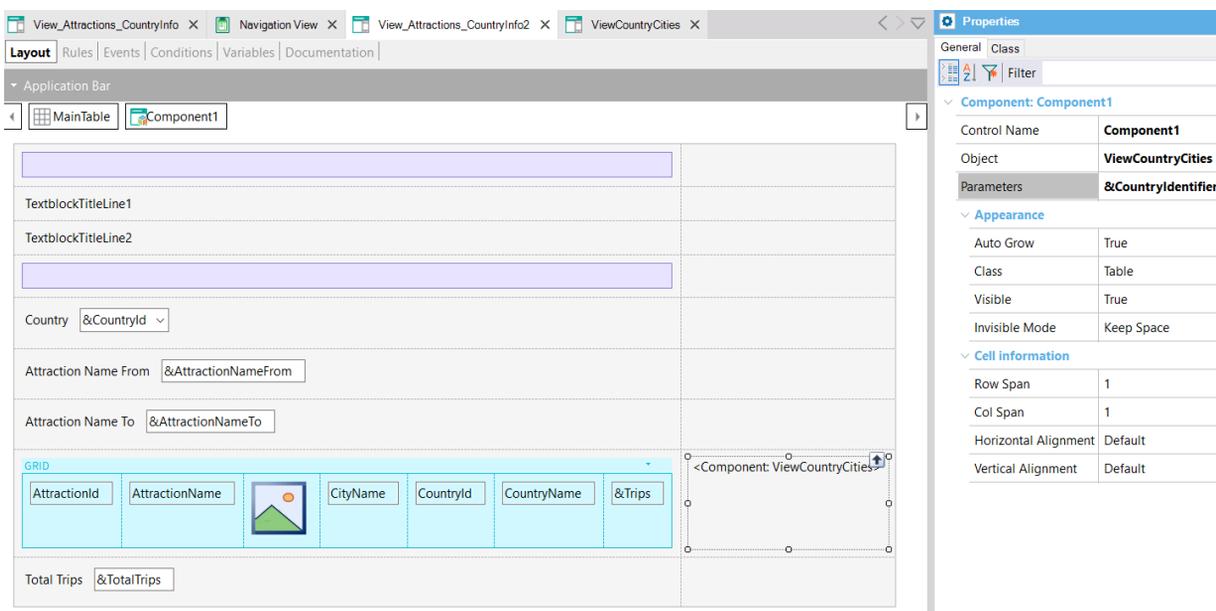
The 'LEVELS' section for 'ViewCountryCities_Level_Detail' shows: 'For First Country (Line: 1)'. The 'LEVELS' section for 'ViewCountryCities_Level_Detail_Grid1' shows: 'For Each City (Line: 2)'. Both levels specify the order as 'CountryId' and include navigation filters and optimization settings.

Agora vamos criar o painel que associaremos ao componente. Colocaremos como nome ViewCountryCities e arrastamos a partir da toolbar o atributo CountryName e um grid com o atributo CityName. Atribuímos ao grid sua transação base como City. Agora adicionamos uma regra Parm com o atributo CountryId.

Se vemos a lista de navegação do painel, no nó Level_Detail vemos que é percorrida a tabela Country, filtrada pelo atributo CountryId presente na regra Parm, portanto irá recuperar um único registro correspondente ao país recebido por parâmetro.

No nó Level_Detail_Grid1 vemos que o grid percorre a tabela City, também filtrada pelo CountryId recebido por parâmetro.

Propriedades do componente



The screenshot displays the GeneXus IDE interface. The main workspace shows a web panel layout with several components: a title bar, two text blocks, a country dropdown menu, two text input fields for attraction names, a grid table, and a total trips label. The grid table has columns for AttractionId, AttractionName, CityName, CountryId, CountryName, and &Trips. A component named 'Component1' is placed over the grid. The Properties panel on the right shows the configuration for 'Component1'.

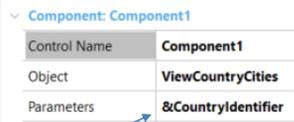
Properties	
General	Class
Filter	
Component: Component1	
Control Name	Component1
Object	ViewCountryCities
Parameters	&CountryIdentifier
Appearance	
Auto Grow	True
Class	Table
Visible	True
Invisible Mode	Keep Space
Cell information	
Row Span	1
Col Span	1
Horizontal Alignment	Default
Vertical Alignment	Default

Nas propriedades do Component1, em Object escolhemos o panel ViewCountryCities e em Parameters adicionamos a variável &CountryIdentifier. Esta variável armazenará o valor do atributo CountryId da linha do grid selecionada.

Uma diferença que vemos em relação aos webpanels é que aqui não é necessário alterar o tipo do panel ViewCountryCities para torná-lo um componente, mas ao inserir um controle Component, posso escolher para inserir qualquer um dos panels que criamos antes.

Criação dinâmica do panel usado como componente

```
34 | Event CountryName.Tap|
35 |     &CountryIdentifier = CountryId
36 |     Component1.Object = ViewCountryCities.Create(&CountryIdentifier)
37 | Endevent
```



Component: Component1	
Control Name	Component1
Object	ViewCountryCities
Parameters	&CountryIdentifier

Agora temos que invocar o panel componente ao clicar sobre o nome do país no grid de atrações, então programamos o evento Tap do atributo CountryName com a invocação.

Como criamos este panel a partir de um save as, já tínhamos código nesse evento, então o excluímos.

Primeiro salvamos o CountryId correspondente ao CountryName selecionado, na variável &CountryIdentifier que definimos antes e então criamos dinamicamente o objeto **ViewCountryCities** associado ao Component1, utilizando o comando Create e passando por parâmetro o país escolhido.

Como o panel que estamos construindo já é main, vamos executá-lo com Run.

Em execução...

View Attractions_Country Info2 x +

localhost:49467/View_Attractions_CountryInfo2-Level_Detail

View Country Cities

The new age of

EXPLORATION

Country (None) v

Attraction Name From Attraction Name From

Attraction Name To Attraction Name To

1	Eiffel Tower		Paris	2	France	6	Country Name China Beijing Shanghai Hong Kong
2	Glenfinnan ...		Glenfinnan	5	Scotland	0	
3	Meet the ...		Beijing	3	China	0	
4	Christ the ...		Rio de ...	1	Brazil	6	
5	Rifugio ...		Belluno	6	Italy	0	
6	London ...		London	7	England	0	
7	Louvre		Paris	2	France	0	
8	Forbidden ...		Beijing	3	China	0	

Total Trips 18

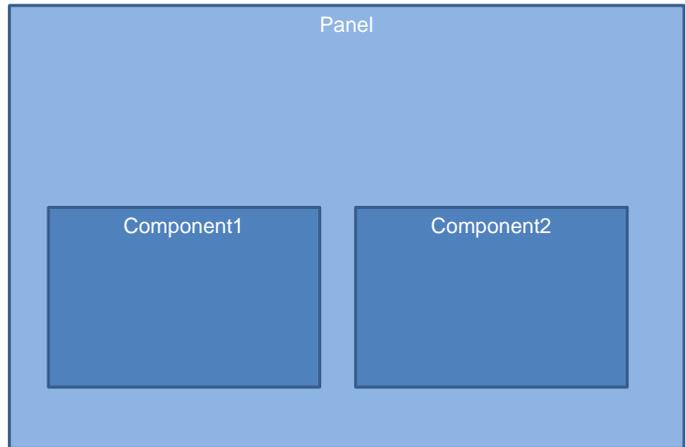
Clicamos sobre o país China e vemos que são apresentadas as cidades da China: Beijing, Shanghai e Hong Kong.

Ordem de disparo dos eventos

ClientStart (Panel)
Start (Panel)
Refresh (Panel)
Load (Panel)

ClientStart (Component1)
Start (Component1)
Refresh (Component1)
Load (Component1)

ClientStart (Component2)
Start (Component2)
Refresh (Component2)
Load (Component2)



Vejamos agora a ordem de disparo dos eventos quando temos painéis dentro de outro panel, ao usar componentes.

Os objetos panels que inserimos como componentes dentro de outro panel têm os mesmos eventos que tinham como panel, ou seja, os eventos do lado do cliente (ClientStart, Back, De usuário, etc.) e os eventos do lado do servidor: Start, Refresh e Load.

Em outras palavras, devemos considerar em que ordem serão disparados os eventos do(s) component(s) que tenhamos no panel, com relação aos eventos do panel host.

Primeiro serão disparados os eventos do panel que contém os components, na ordem usual, que vimos antes, ou seja, o ClientStart, Start, Refresh e Load.

Depois, de cima para baixo e da esquerda para a direita, os eventos de cada panel incluído como componente, também na ordem usual.

Objeto Master Panel

Vejamos agora o que é o objeto Master Panel. Este objeto só é utilizável em aplicações web geradas em Angular.

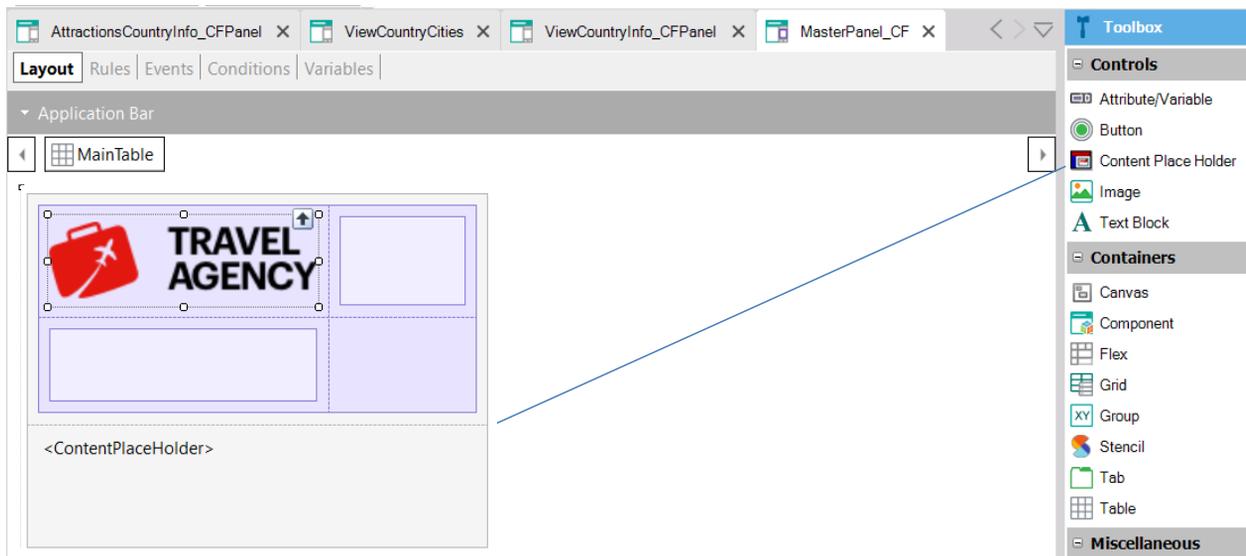
Objeto Master Panel

Properties	
Filter	
Panel: Attractions_CFPANEL	
Name	Attractions_CFPANEL
Description	Attractions_CFPANEL
Module/Folder	FrontendAngular
Qualified Name	Attractions_CFPANEL
Object Visibility	Public
Main program	False
Master Panel	(none)
Caption	Attractions_CFPANEL

Se observamos as propriedades do objeto panel **View_Attraction_MoreInfo**, vemos que uma delas é a propriedade Master Panel, que em nosso panel aparece vazia. Ali se indica qual será a página mestre que conterá o objeto panel e onde o mesmo será carregado. Este conceito é análogo ao do Web Master Panel visto para os webpanels.

No entanto, ao contrário destes últimos, um objeto panel não tem uma propriedade para alterar o tipo, mas o objeto Master Panel é um objeto independente do objeto panel.

Objeto Master Panel (cont.)



Vamos criar um objeto do tipo Master Panel e colocaremos como nome **MasterPanel_TravelAgency**.

Se vamos para a barra de ferramentas, vemos que temos disponível um controle container Content Place Holder, onde serão carregados os objetos panels.

Vamos inserir no webpanel o Content Place Holder. Agora inserimos acima do mesmo uma tabela e dentro dela uma imagem com o logotipo da agência e abaixo e à sua direita inserimos tabelas para nos ajudar a fazer com que o logotipo apareça alinhado à esquerda.

Objeto Master Panel (cont.)

The screenshot displays the design tool interface. On the left, a design canvas shows a table layout with a red suitcase icon and the text "TRAVEL AGENCY". Below the canvas is a content placeholder. On the right, two property panels are visible:

Table: MainTable

Control Name	MainTable
▼ Appearance	
Columns Style	100%
Rows Style	100dip;pd
Width	100%
Height	100%
Auto Grow	True

Table: Table1

Control Name	Table1
▼ Appearance	
Columns Style	320dip;100%
Rows Style	90dip
Width	100%
Height	90dip

On the far right, a class diagram shows the **ImageHeader_TravelAgency** class with the **Scale Type** property set to **Fit**.

Ajustamos as propriedades da tabela container para que a imagem tenha uma boa aparência. E fazemos o mesmo com a Main Table.

Agora devemos modificar o comportamento da imagem para que se ajuste ao tamanho que demos.

Por enquanto, vamos criar uma classe `ImageHeader_TravelAgency` à qual atribuímos a propriedade `Scale Type` no valor `Fit`, mas depois, quando entrarmos no tema `Design`, veremos como se faz isto com um `Design System Object`.

Agora, para a imagem do logotipo, atribuímos a classe que criamos. E salvamos.

Objeto Master Panel (cont.)

Panel: Attractions_CFPanel

Name	Attractions_CFPanel
Description	Attractions_CFPanel
Module/Folder	FrontendAngular
Qualified Name	Attractions_CFPanel
Object Visibility	Public
Main program	True
Master Panel	MasterPanel_CF
Caption	Attractions_CFPanel

Panel: AttractionDetail_CFPanel

Name	AttractionDetail_CFPanel
Description	Attraction Detail_CFPanel
Module/Folder	FrontendAngular
Qualified Name	AttractionDetail_CFPanel
Object Visibility	Public
Main program	False
Master Panel	MasterPanel_CF
Caption	Attraction Detail_CFPanel

Agora atribuímos a propriedade Master Panel aos painéis **View_Attractions_MoreInfo** e **View_Attractions_CountryInfo** com o objeto **MasterPanel_TravelAgency** que acabamos de criar.

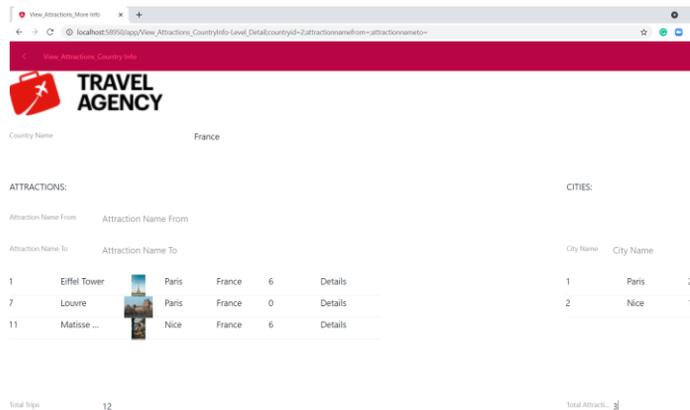
E executamos.

Execução da aplicação com um master panel atribuído



The new age of
EXPLORATION

Country	(None)				
Attraction Name From	Attraction Name From				
Attraction Name To	Attraction Name To				
1	Eiffel Tower	Paris	2	France	6
2	Glenfinnan Viaduct	Glenfinnan	5	Scotland	0
3	Meet the Emperor	Beijing	3	China	0
Total Trips		18			



Vemos que aparece o logotipo da agência e abaixo é carregado panel View_Attractions_MoreInfo. Se clicamos no país France, vemos que também o panel da informação do país tem o mesmo logotipo.

Aqui, não entraremos no detalhe do Design System do objeto Panel, mas podemos antecipar que o uso do objeto Design System Object é similar ao dos webpanels.

Entraremos no tema de design em um próximo vídeo.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications