

# Telas web com foco em Customer-facing

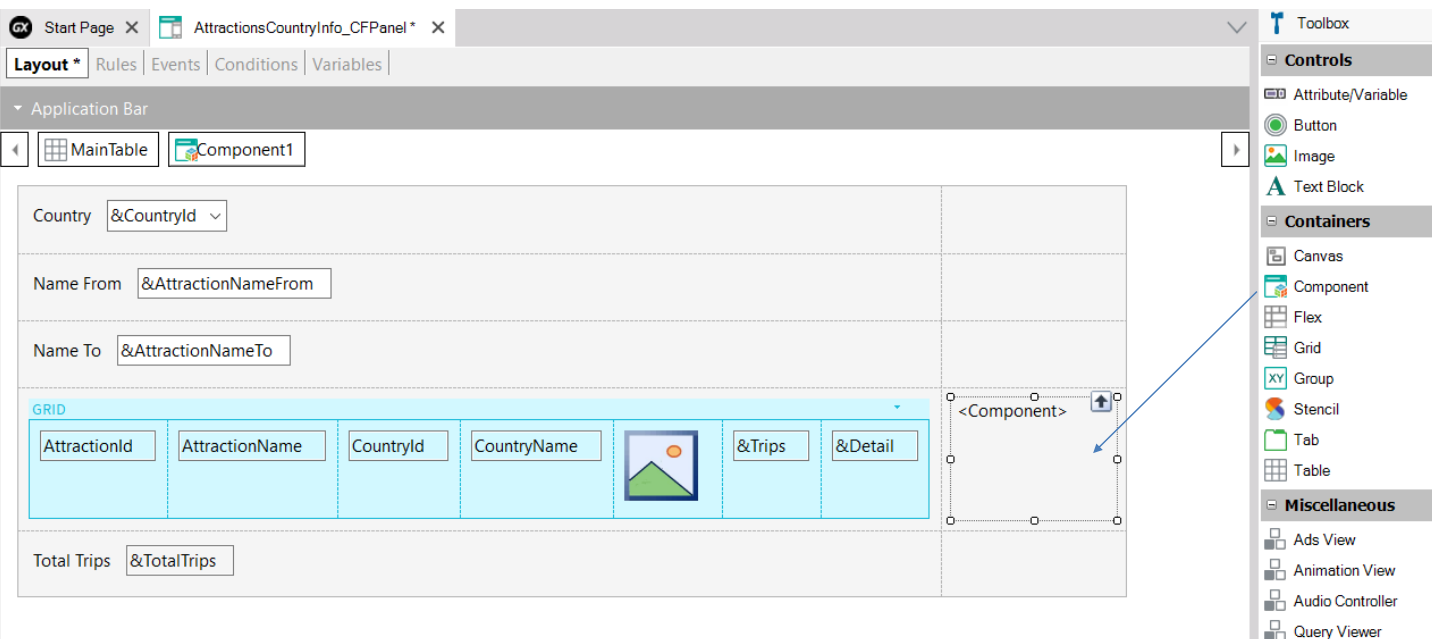
Uso de Components e objeto Master Panel

*GeneXus™*

## Components

O objeto panel permite encapsular funcionalidade mediante o 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 mostre as atrações e os dados de um país selecionado, mas desta vez a informação será exibida utilizando um componente dentro do panel que estamos criando.

Para criar o panel que conterá o component, fazemos um Save As do Attractions\_CFPANEL, com o nome AttractionsCountryInfo\_CFPANEL.

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

De forma similar ao que fizemos antes, quando clicamos sobre o nome de um país do grid, serão abertas as informações desse país.

## Criação do panel a ser usado como componente

The screenshot illustrates the creation of a panel component in GeneXus. The main window shows the 'Rules' tab with a rule named 'Parm(in:CountryId);'. Below it, the 'Layout' tab displays a panel design with a 'Country Name' label and a 'GRID' containing a 'CityName' label. To the right, two panels are shown: 'Data Provider ViewCountryCities\_Level\_Detail' and 'Data Provider ViewCountryCities\_Level\_Detail\_Grid1'. The first panel shows a 'LEVELS' section with a 'For First Country (Line: 1)' loop. The second panel shows a 'LEVELS' section with a 'For Each CountryCity (Line: 2)' loop. Arrows indicate the flow from the main window to the data provider panels.

Agora vamos criar o panel que associaremos ao componente. Colocamos como nome ViewCountryCities e arrastamos da Toolbar o atributo CountryName e um grid com o atributo CityName. Atribuímos ao grid sua transação base em Country.City. Agora adicionamos uma regra Parm com o atributo CountryId.

Se vemos a lista de navegação do panel, no nó Level\_Detail vemos que é percorrida a tabela Country, filtrada pelo atributo CountryId presente na regra Parm, pelo qual vai recuperar um único registro correspondente ao país recebido por parâmetro.

No nó Level\_Detail\_Grid1 vemos que o grid percorre a tabela CountryCity, também filtrada pelo CountryId recebido por parâmetro.

## Propriedades do componente

The screenshot shows the GeneXus IDE interface. The main workspace displays a form with the following elements:

- Country:
- Name From:
- Name To:
- GRID:
 

AttractionId	AttractionName	CountryId	CountryName		&Trips	&Detail
- Total Trips:

The Properties panel on the right shows the configuration for **Component1**:


Control Name	Component1
Object	ViewCountryCities
Parameters	&CountryIdentifier
<b>Appearance</b>	
Auto Grow	True
Class	Table
Visible	True
Invisible Mode	Keep Space
<b>Cell information</b>	
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 a respeito dos webpanels, é que aqui não é necessário mudar o tipo do panel ViewCountryCities\_CFPANEL para que seja um componente, mas ao inserir um controle Component, posso escolher para inserir qualquer um dos painéis que criamos antes.

## Criação dinâmica do panel usado como componente

```
41 Event CountryName.Tap
42     &CountryIdentifier = CountryId
43     Component1.Object = ViewCountryCities_CFPANEL.Create(&CountryIdentifier)
44 Endevent
45
```



Component: Component1	
Control Name	Component1
Object	ViewCountryCities
Parameters	&CountryIdentifier

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

Primeiro salvamos o CountryId correspondente ao CountryName selecionado, na variável &CountryIdentifier que definimos antes e, depois, criamos dinamicamente o objeto ViewCountryCities\_CFPANEL 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...

Country (None)

Name From

Name To

Name	Country	Image	Count	Details
Christ the Redemmer	Brazil		1	Details
Cinque Terre	Italy		0	Details
Eiffel Tower	France		1	Details
Forbidden city	China		0	Details
Glenfinnan Viaduct	Scotland		0	Details
London Towers	England		0	Details
Long Bridges	United States		0	Details
Louvre	France		0	Details
Matisse Museum	France		1	Details
Meet the	China		0	Details

Total Trips 3

Country Name China

City Name Beijing

City Name Shanghai

City Name Hong Kong

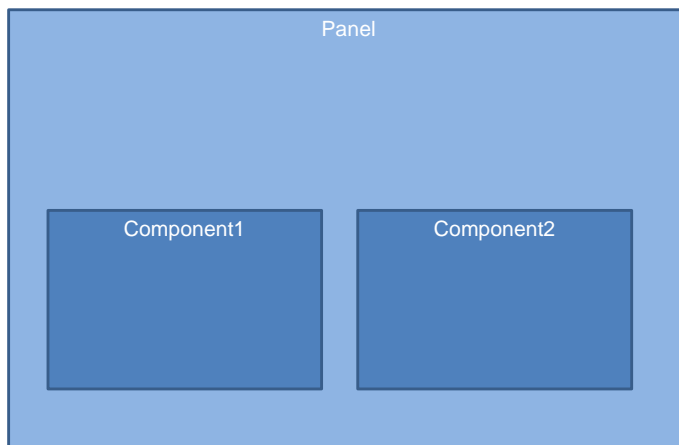
Clicamos sobre o país China e vemos que se mostram as cidades da China: Beijing, Shanghai e Hong Kong.

## 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)



Os objetos panels que inserimos como componentes dentro de outro panel, têm os mesmos eventos que tinham como panel, isto é, os eventos do sistema do lado do cliente (ClientStart e Back) e os eventos do lado do servidor: Start, Refresh e Load.

Ou seja, devemos considerar em que ordem serão disparados os eventos do ou dos components que tenhamos no panel, com relação aos eventos do panel host.

Primeiro serão disparados os eventos do panel que contém os components, na ordem habitual 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 habitual.



## Objeto Master Panel

Vejamos agora o que é o objeto Master Panel.

## Objeto Master Panel

Start Page X Attractions\_CFPANEL X

Layout Rules Events Conditions Variables

Application Bar


MainTable

Country &CountryId

Name From &AttractionNameFrom

Name To &AttractionNameTo

GRID

AttractionId	AttractionName	CountryId	CountryName		&Trips	&Detail
--------------	----------------	-----------	-------------	---	--------	---------

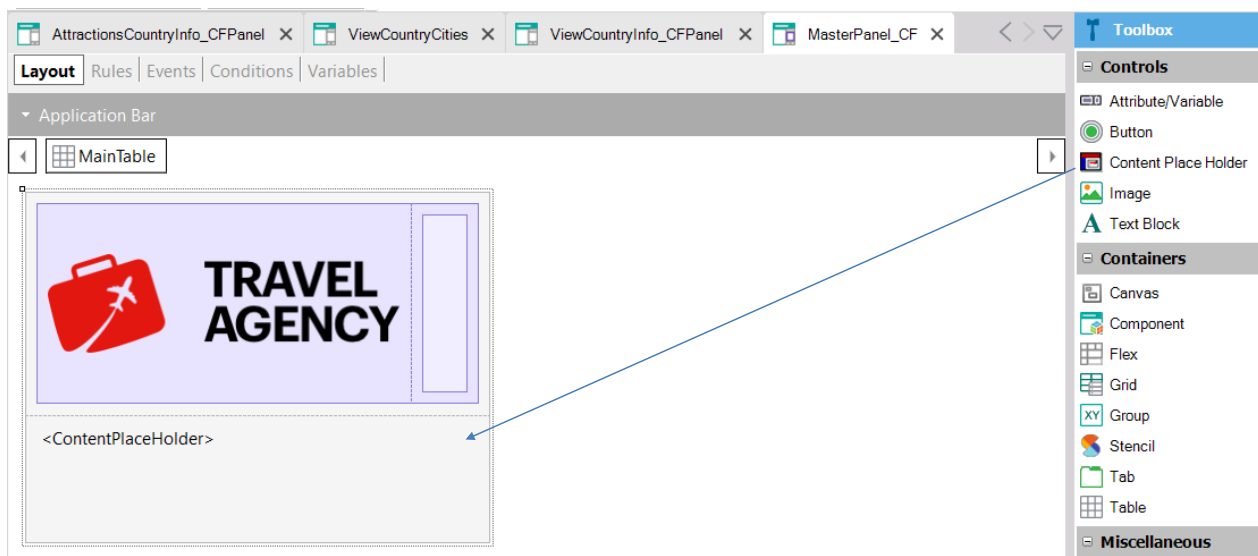
Total Trips &TotalTrips

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 Attraction\_CFPANEL, vemos que uma delas é a propriedade **Master Panel**, a qual em nosso panel aparece vazia. Ali se indica qual será a página mestra que conterá o objeto panel e onde o mesmo será carregado. Este conceito é análogo ao de 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.)

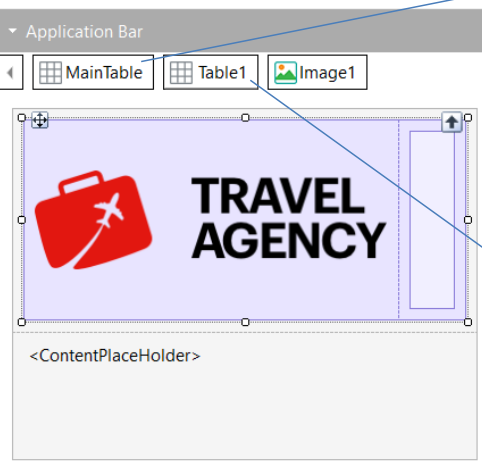


Vamos criar um objeto do tipo Master Panel e chamá-lo de MasterPanel\_CF.

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

Vamos inserir no form do Master Panel o Content Place Holder. Agora, inserimos acima do mesmo uma tabela e dentro dela uma imagem com o logo da agência e, à sua direita, uma tabela, para que nos ajude a fazer com que o logo apareça alinhado à esquerda.

## Objeto Master Panel (cont.)



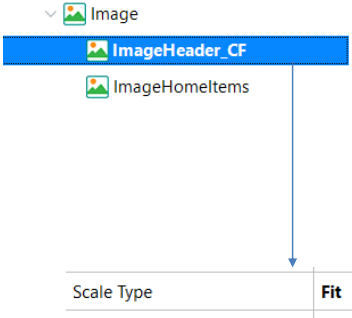
**Table: MainTable**

Control Name	MainTable
<b>Appearance</b>	
Columns Style	100%
Rows Style	100dip;pd
Width	100%
Height	100%
Auto Grow	True

**Table: Table1**

Control Name	Table1
<b>Appearance</b>	
Columns Style	<b>320dip;100%</b>
Rows Style	90dip
Width	100%
Height	90dip



```

classDiagram
    class Image
    class ImageHeader_CF
    class ImageHomeItems
    ImageHeader_CF --|> ImageHomeItems
    ImageHeader_CF --> ImageHomeItems : Scale Type
    
```

Ajustamos as propriedades da tabela, para que a imagem fique bem visível. Também vamos criar uma classe ImageHeader\_CF, à qual atribuímos a propriedade Scale Type no valor Fit e, agora, para a imagem do logo atribuímos essa classe. E salvamos.

## Objeto Master Panel (cont.)

### Panel: Attractions\_CFPanel

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

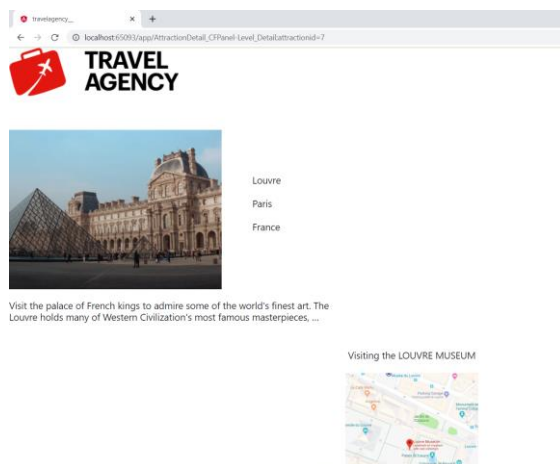
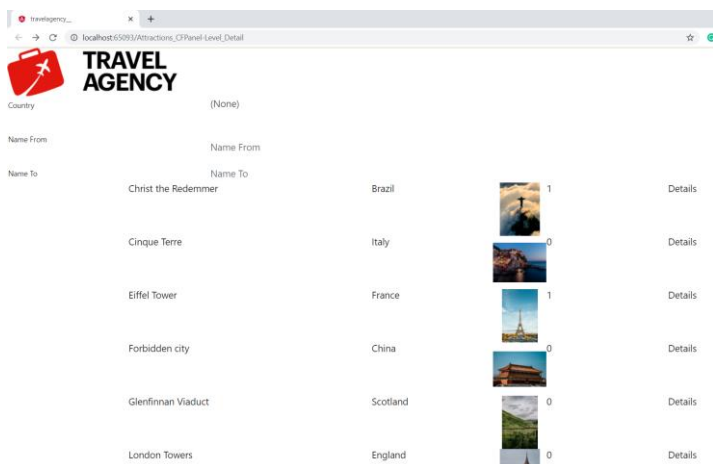
### Panel: AttractionDetail\_CFPanel

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

Agora atribuímos a propriedade Master Panel aos panels Attractions\_CFPanel e AttractionDetail\_CFPanel com o objeto MasterPanel\_CF que acabamos de criar.

E executamos.

## Execução da aplicação com um master panel atribuído



Vemos que aparece o logo da agência e, embaixo, se carrega o panel `Attractions_CFPANEL`. Se clicamos em `Details` da atração Louvre, vemos que também o panel do detalhe da atração tem o mesmo logo.

Aqui não entraremos em detalhes do Design System do objeto `Master Panel`, mas podemos adiantar que o uso do objeto `Theme` e das classes é similar ao dos `webpanels`.

Entraremos no tema de desenho em um próximo vídeo.



[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)  
[training.genexus.com/certifications](http://training.genexus.com/certifications)