

Pantallas web con foco en Customer-facing

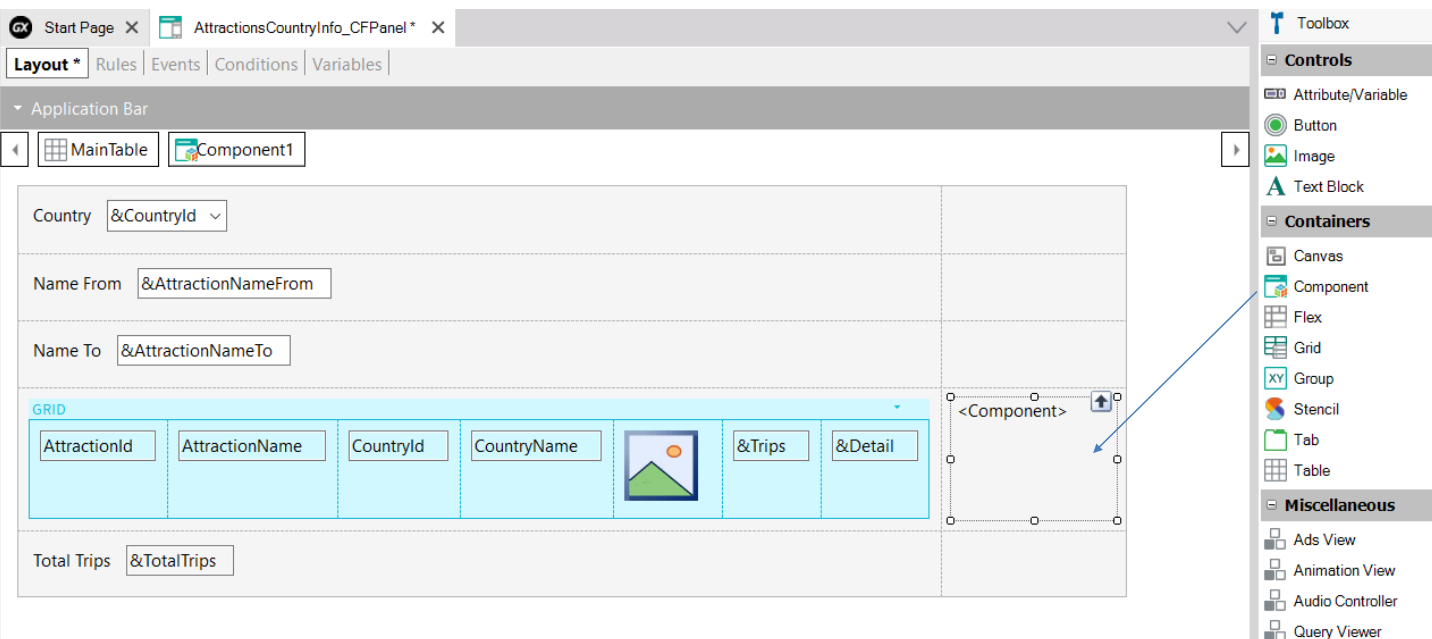
Uso de Components y objeto Master Panel

GeneXus™

Components

El objeto panel permite encapsular funcionalidad mediante el uso de components, en forma similar a los webpanels con los webcomponents. Veamos un ejemplo de uso.

Uso de un panel dentro de otro panel



Vamos a construir un objeto panel que muestre las atracciones y los datos de un país seleccionado, pero esta vez la información será desplegada utilizando un componente dentro del panel que estamos creando.

Para crear el panel que contendrá el component, hacemos un Save As del Attractions_CFPANEL, con el nombre AttractionsCountryInfo_CFPANEL.

A la derecha del grid de atracciones, insertamos desde la barra de herramientas, un control Component.

En forma similar a lo que hicimos antes, cuando hagamos clic sobre el nombre de un país de la grilla, se abrirá la información de ese país.

Creación del panel a ser usado como componente

The screenshot illustrates the creation of a panel component in the GeneXus IDE. The main workspace shows a 'MainTable' with a 'CountryName' field and a 'GRID' containing a 'CityName' field. The 'Rules' tab is active, showing a 'Parm(in:CountryId);' rule. The 'Data Provider ViewCountryCities_Level_Detail' is configured with 'CountryId' as the index and 'CountryId = @CountryId' as the navigation filter. The 'Data Provider ViewCountryCities_Level_Detail_Grid1' is configured with 'CountryId' as the index and 'CountryId = @CountryId' as the navigation filter. Arrows indicate the flow from the main workspace to the data provider configurations.

Ahora vamos a crear el panel que asociaremos al componente. Le ponemos de nombre ViewCountryCities y arrastramos desde la toolbar al atributo CountryName y un grid con el atributo CityName. Asignamos al grid su transacción base en Country.City. Ahora agregamos una regla Parm con el atributo CountryId.

Si vemos el listado de navegación del panel, en el nodo Level_Detail vemos que se recorre la tabla Country, filtrada por el atributo CountryId presente en la regla Parm, por lo cual va a recuperar un único registro correspondiente al país recibido por parámetro.

En el nodo Level_Detail_Grid1 vemos que la grilla recorre la tabla CountryCity, también filtrada por el CountryId recibido por parámetro.

Propiedades del componente

The screenshot shows the GeneXus IDE interface. The main workspace displays a form layout 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
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

En las propiedades del Component1, en Object elegimos al panel ViewCountryCities y en Parameters agregamos la variable &CountryIdentifier. Esta variable almacenará el valor del atributo CountryId de la fila de la grilla seleccionada.

Una diferencia que vemos respecto a los webpanels, es que aquí no es necesario cambiar el tipo del panel ViewCountryCities_CFPANEL para que sea un componente, sino que al insertar un control Component, puedo elegir para insertar cualquiera de los paneles que creamos antes.

Creación dinámica del 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

Ahora tenemos que invocar al panel componente cuando hacemos clic sobre el nombre del país en la grilla de atracciones, así que programamos el evento Tap del atributo CountryName con la invocación.

Primero salvamos el CountryId correspondiente al CountryName seleccionado, en la variable &CountryIdentifier que definimos antes y luego creamos dinámicamente al objeto ViewCountryCities_CFPANEL asociado al Component1, utilizando el comando Create y pasando por parámetro al país elegido.

Como el panel que estamos construyendo ya es main, vamos a ejecutarlo con Run.

En ejecución...

Country (None)

Name From Name From

Name To Name To

Christ the Redemmer	Brazil		1	Details	Country Name	China
Cinque Terre	Italy		0	Details	City Name	Beijing
Eiffel Tower	France		1	Details	City Name	Shanghai
Forbidden city	China		0	Details	City Name	Hong Kong
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

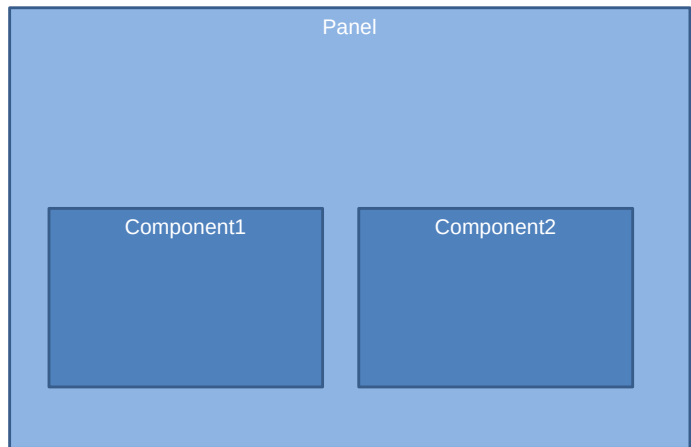
Hacemos clic sobre el país China y vemos que se muestran las ciudades de China: Beijing, Shanghai y Hong Kong.

Orden de disparo de los 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)



Los objetos panels que insertamos como componentes dentro de otro panel, tienen los mismos eventos que tenían como panel, esto es, los eventos del sistema lado del cliente (ClientStart y Back) y los eventos del lado del servidor: Start, Refresh y Load.

Es decir que debemos considerar en qué orden serán disparados los eventos del o los components que tengamos en el panel, con respecto a los eventos del panel host.

Primero se dispararán los eventos del panel que contiene los components, en el orden habitual, que vimos antes, es decir, el ClientStart, Start, Refresh y Load.

Luego, de arriba hacia abajo y de izquierda a derecha, los eventos de cada panel incluido como componente, también en el orden habitual.

Objeto Master Panel

Veamos ahora qué es el 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
--------------	----------------	-----------	-------------	---	--------	---------

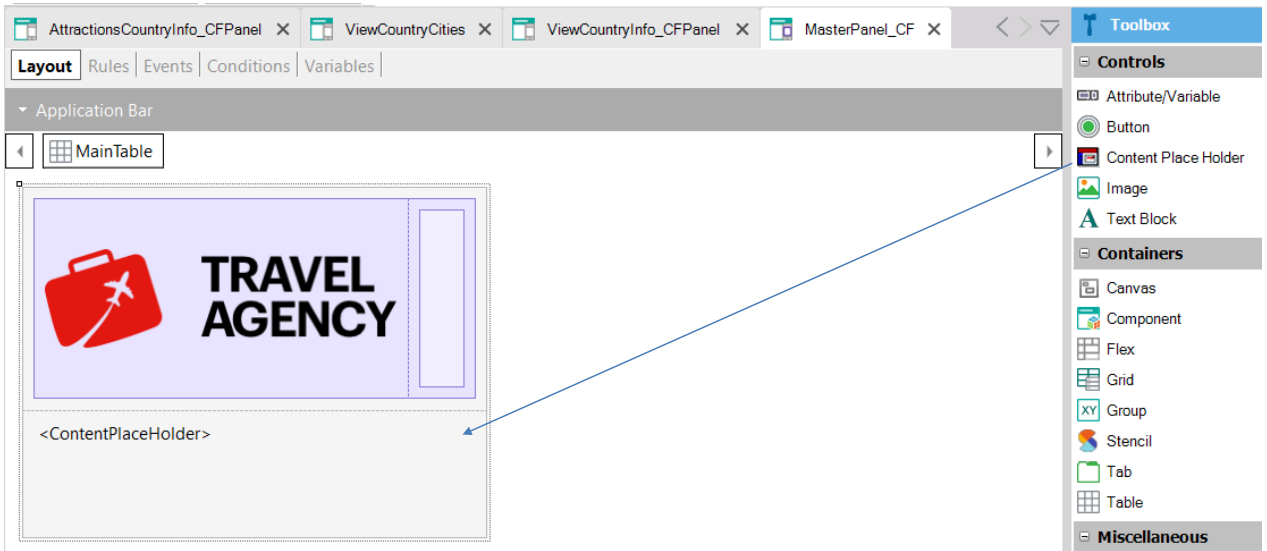
Total Trips &TotalTrips

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

Si observamos las propiedades del objeto panel Attraction_CFPANEL, vemos que una de ellas es la propiedad **Master Panel**, la cual en nuestro panel aparece vacía. Allí se indica cuál será la página maestra que contendrá al objeto panel y donde el mismo se cargará. Este concepto es análogo al de Web Master Panel visto para los webpanels.

Sin embargo, a diferencia de estos últimos, un objeto panel no tiene una propiedad para cambiar el tipo, sino que el objeto Master Panel es un objeto independiente del objeto panel.

Objeto Master Panel (cont.)



Vamos a crear un objeto del tipo Master Panel y le ponemos de nombre MasterPanel_CF.

Si vamos a la barra de herramientas vemos que tenemos disponible un control contenedor Content Place Holder, donde se cargarán los objetos panels.

Vamos a insertar en el webpanel, el Content Place Holder. Ahora insertamos arriba del mismo una tabla y dentro de ella una imagen con el logo de la agencia y a su derecha una tabla para que nos ayude a que el logo nos aparezca alineado a la izquierda.

Objeto Master Panel (cont.)

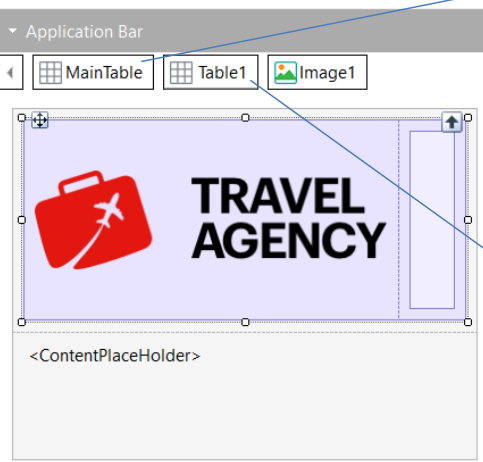
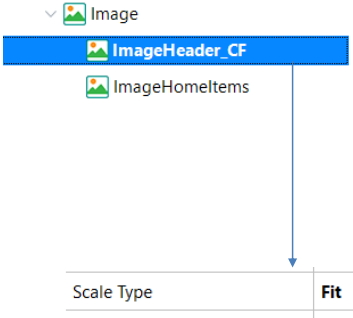


Table: MainTable

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



Ajustamos las propiedades de la tabla y de la MainTable para que la imagen se vea bien. También vamos a crear una clase ImageHeader_CF a la cual le asignamos la propiedad Scale Type en el valor Fit y ahora a la imagen del logo le asignamos esa clase. Y 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

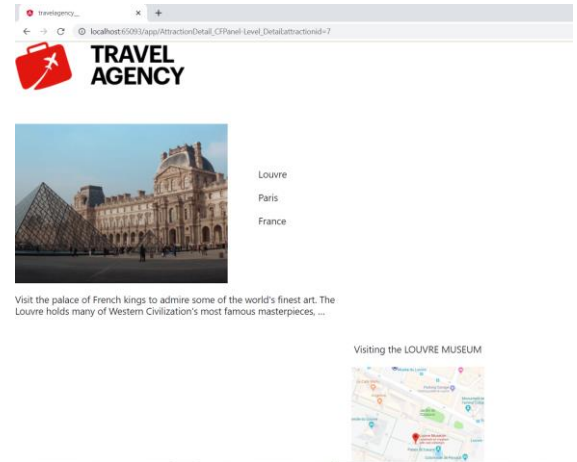
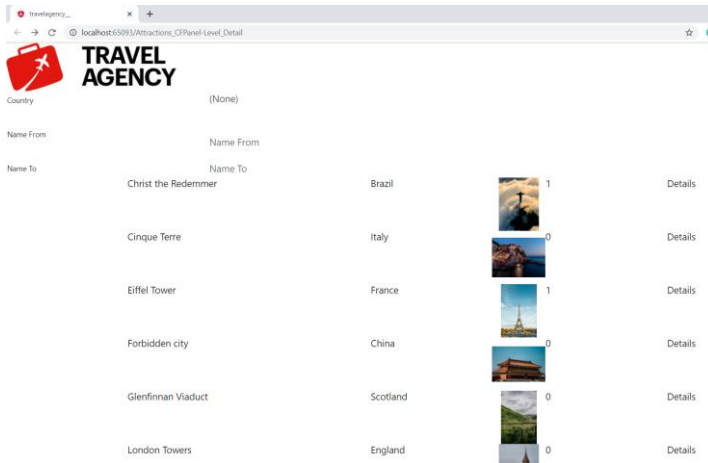
Panel: AttractionDetail_CFPanel

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

Ahora asignamos la propiedad Master Panel a los paneles Attractions_CFPanel y AttractionDetail_CFPanel con el objeto MasterPanel_CF que acabamos de crear.

Y ejecutamos.

Ejecución de la aplicación con un master panel asignado



Vemos que aparece el logo de la agencia y debajo se carga el panel `Attractions_CFPanel`. Si hacemos clic en `Details` del país `France`, vemos que también el panel del detalle de la atracción tiene el mismo logo.

Aquí no entraremos en el detalle del `Design System` del objeto `Master Panel`, pero podemos adelantar que el uso del objeto `Theme` y de las clases es similar que en los `webpanels`. Entraremos en el tema de diseño en un próximo video.



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