Components y objeto Master Panel

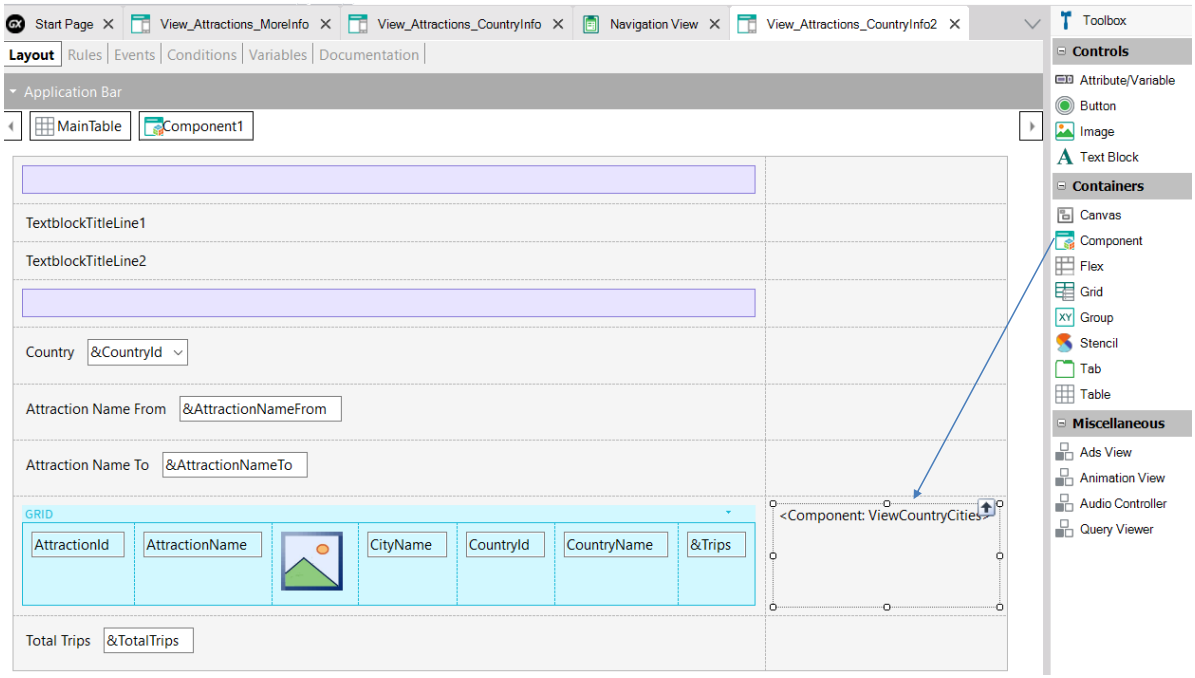
GeneXus™

En este video utilizaremos los components, que son contenedores que podremos incluir en un panel y asociarle otro panel, de forma de que será posible ejecutar un objeto panel dentro de otro. También veremos el objeto Master Panel, que nos permitirá definir una pantalla contenedora dentro de la cual se ejecutarán todas las pantallas de nuestra aplicación de la agencia de viajes, de forma que la apariencia y contexto sea el mismo para todas, por ejemplo, que aparezca siempre el logo de la agencia, entre otras cosas.

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



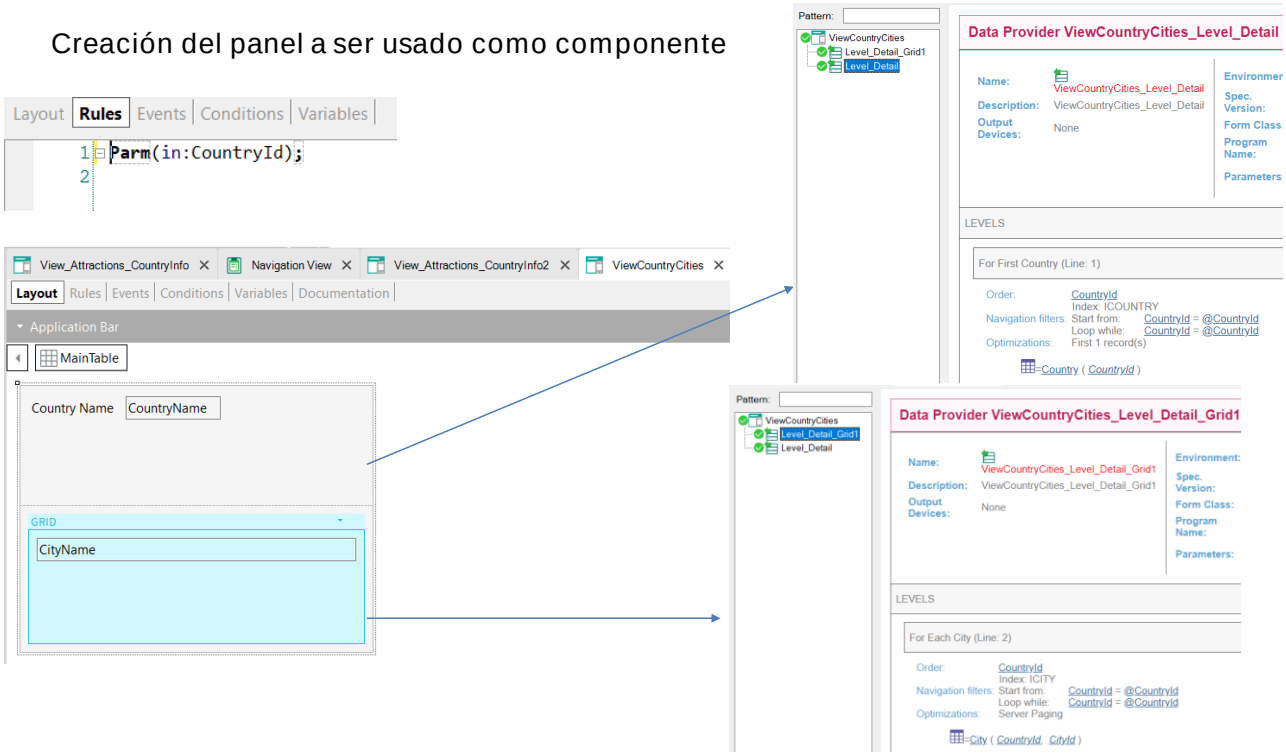
Vamos a construir un objeto panel que muestre las atracciones y los datos de un país seleccionado, pero esta vez la información será desplegada utilizando un componente dentro del panel que estamos creando.

Para crear el panel que contendrá el component, hacemos un Save As del View_Attractions_MoreInfo, con el nombre View_Attractions_CountryInfo2.

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

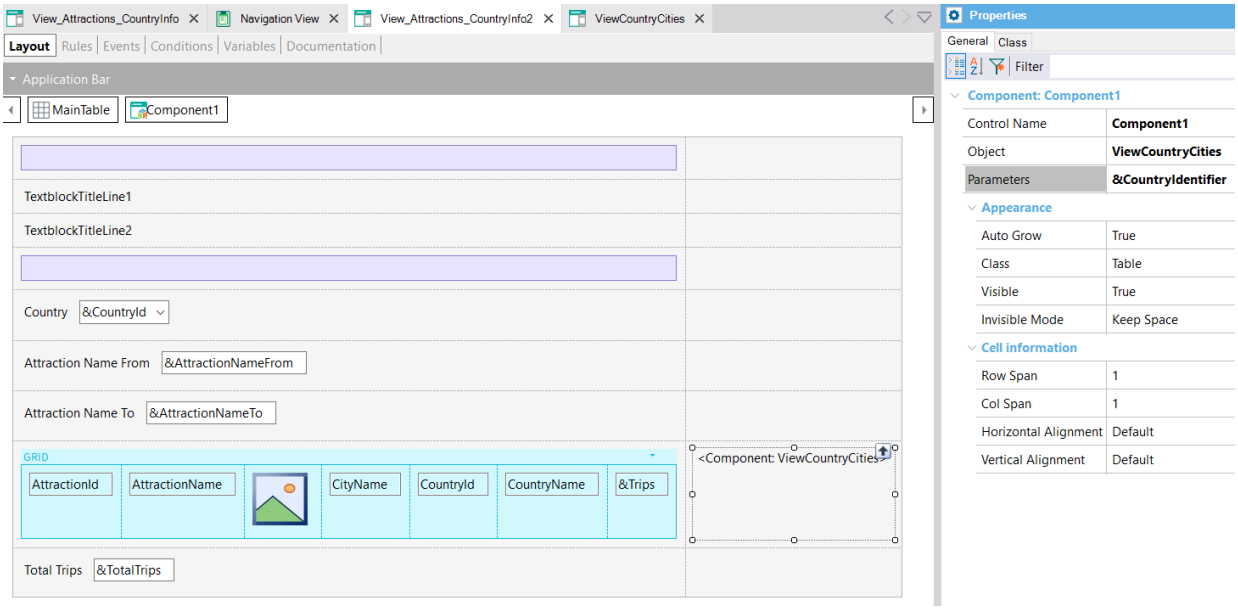


Ahora vamos a crear el panel que asociaremos al componente. Le ponemos de nombre ViewCountryCities y arrastramos desde la toolbar al atributo CountryName y un grid con el atributo CityName. Asignamos al grid su transacción base en City.
Ahora agregamos una regla Parm con el atributo CountryId.

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 City, también filtrada por el CountryId recibido por parámetro.

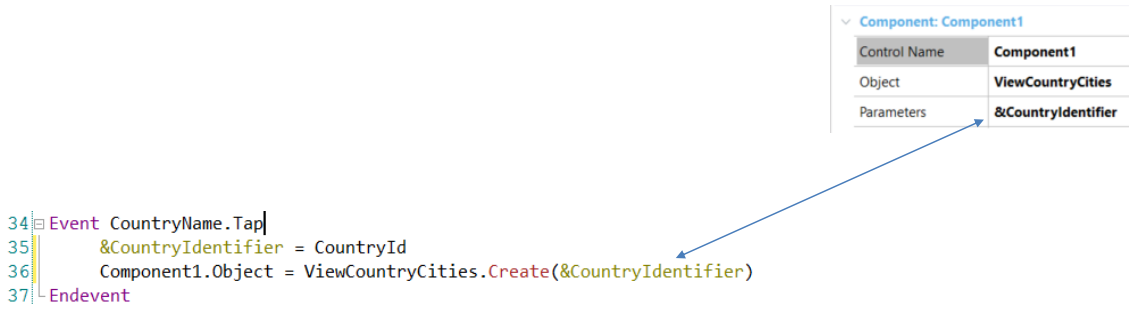
Propiedades del componente



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 web panels, es que aquí no es necesario cambiar el tipo del panel **ViewCountryCities** 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



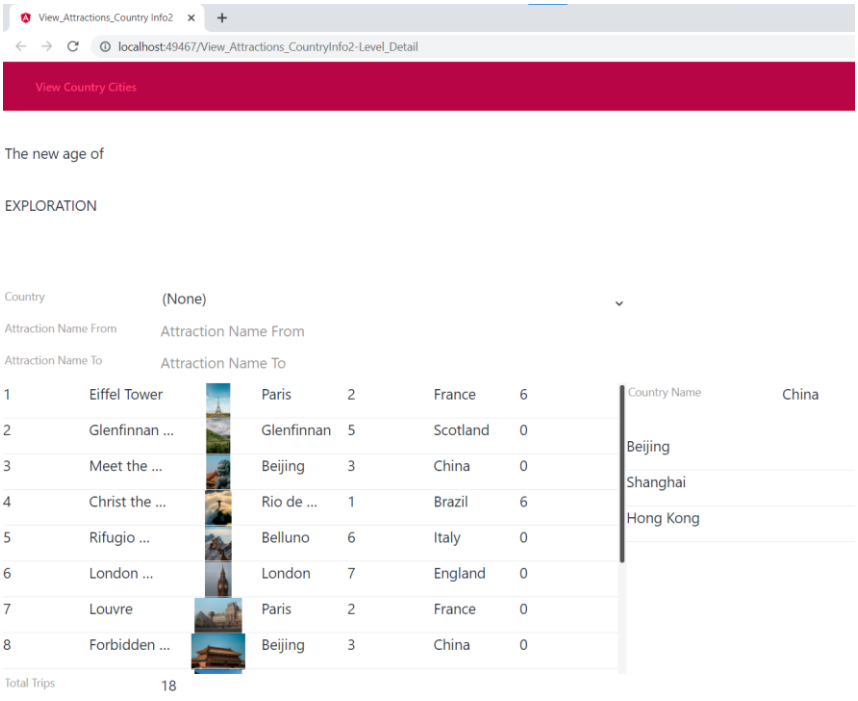
Ahora tenemos que invocar al panel componente cuando hacemos clic sobre el nombre del país en la grilla de atracciones, así que programamos el evento Tap del atributo CountryName con la invocación.

Como este panel lo creamos a partir de un save as, ya teníamos código en ese evento, así que lo borramos.

Primero salvamos el CountryId correspondiente al CountryName seleccionado, en la variable &CountryIdentifier que definimos antes y luego creamos dinámicamente al objeto **ViewCountryCities** 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...



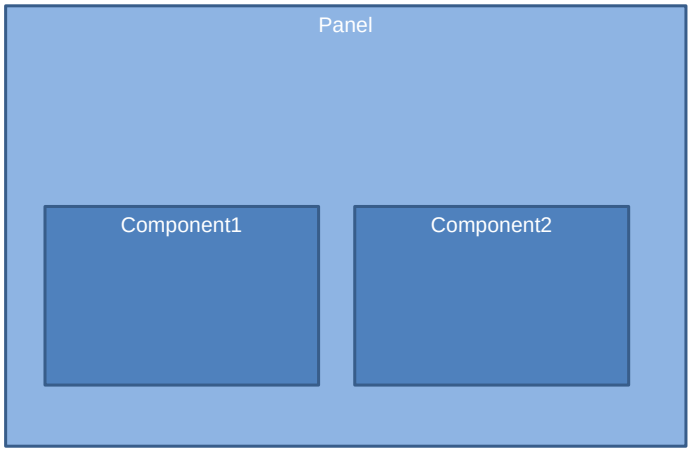
Hacemos clic sobre el país China y vemos que se muestran las ciudades de China: Beijing, Shanghái y Hong Kong.

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)



Veamos ahora el orden de disparo de los eventos cuando tenemos paneles dentro de otro panel, al usar componentes.

Los objetos panels que insertamos como componentes dentro de otro panel, tienen los mismos eventos que tenían como panel, esto es, los eventos del lado del cliente (ClientStart, Back, De usuario, etc.) 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.

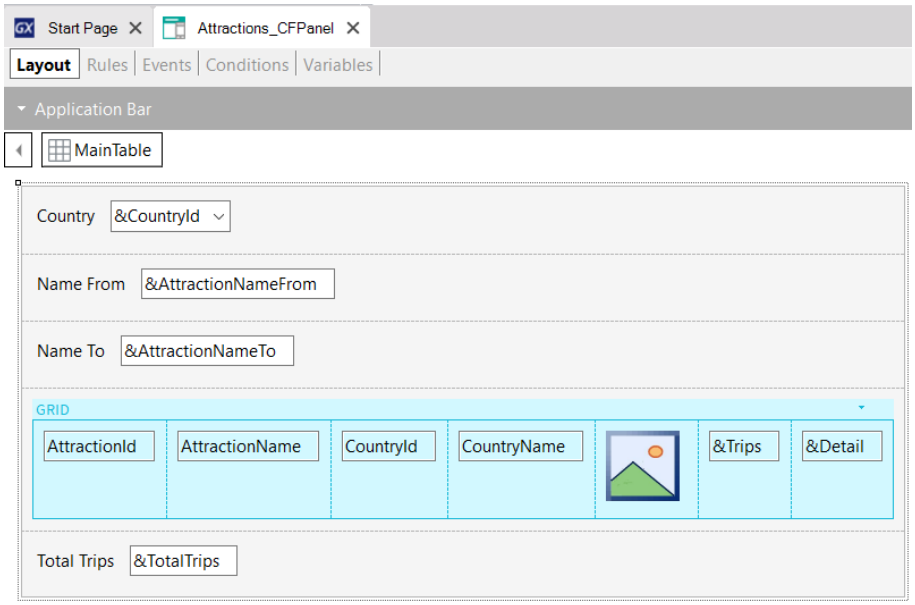
Primero se dispararán los eventos del panel que contiene los components, en el orden habitual, que vimos antes, es decir, el ClientStart, Start, Refresh y Load.

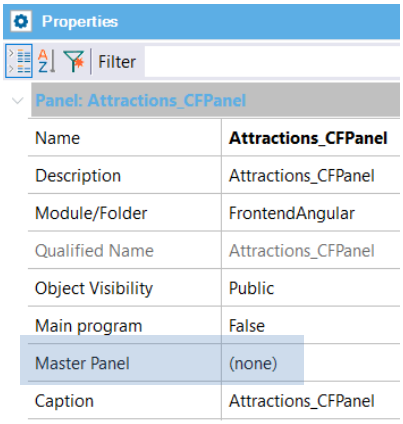
Luego, de arriba hacia abajo y de izquierda a derecha, los eventos de cada panel incluido como componente, también en el orden habitual.

Objeto Master Panel

Veamos ahora qué es el objeto Master Panel. Este objeto es sólo utilizable en aplicaciones web generadas en Angular.

Objeto Master Panel

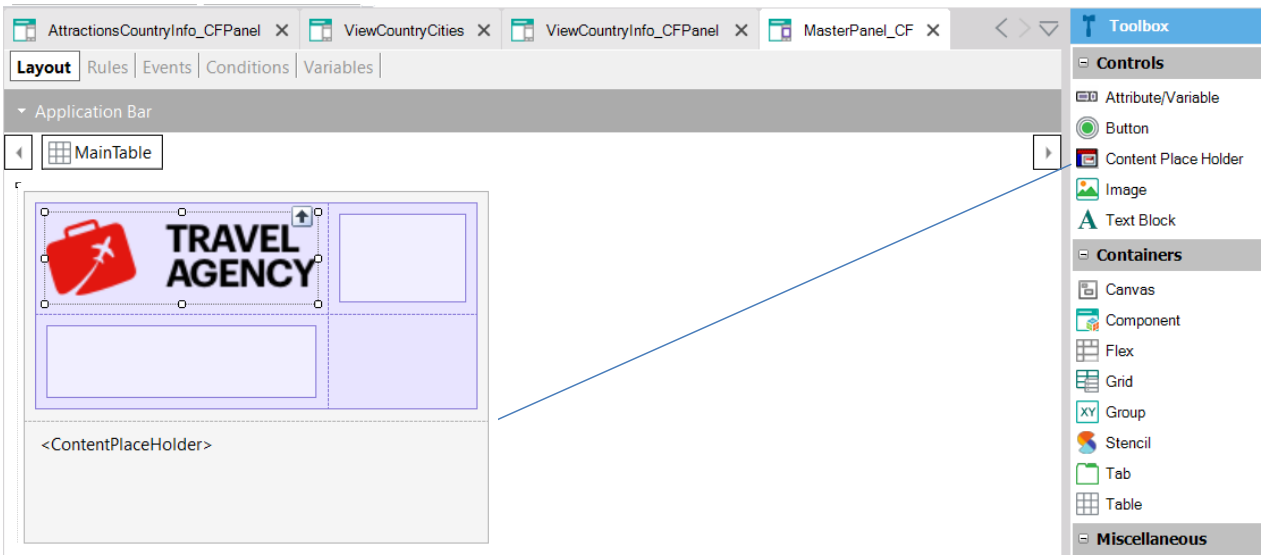




Si observamos las propiedades del objeto panel **View_Attraction_MoreInfo**, vemos que una de ellas es la propiedad Master Panel, la cual en nuestro panel aparece vacía. Allí se indica cuál será la página maestra que contendrá al objeto panel y donde el mismo se cargará.
Este concepto es análogo al de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

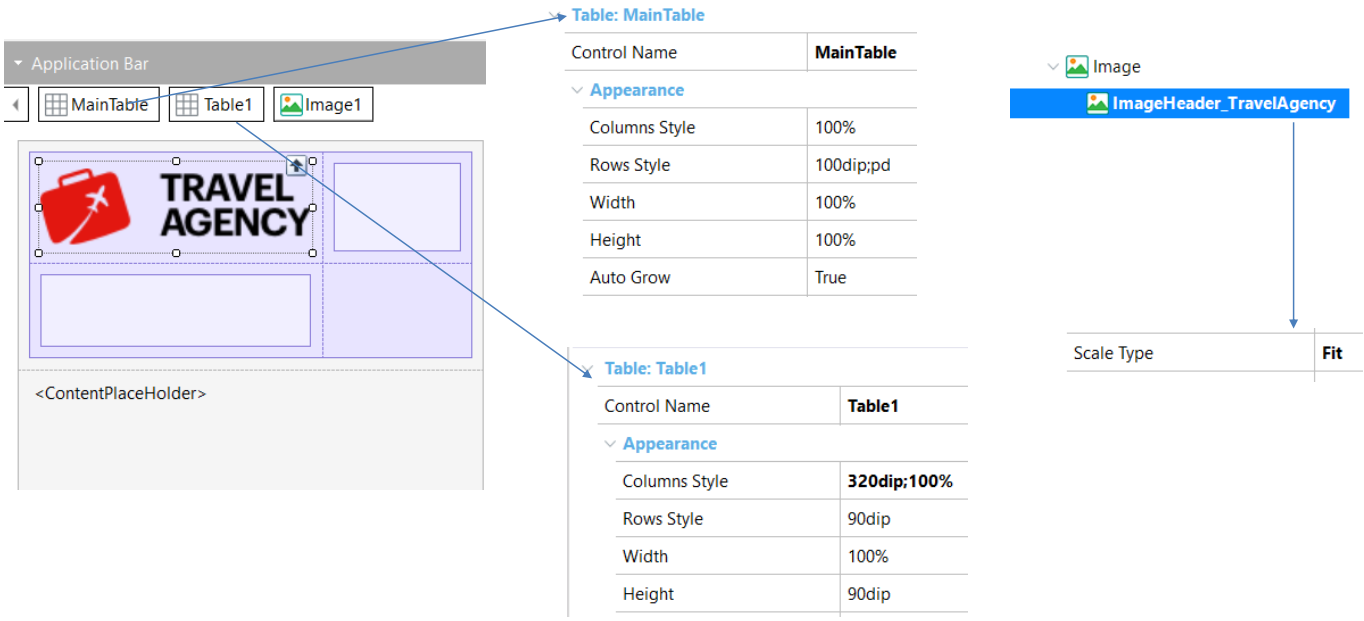
Objeto Master Panel (cont.)



Vamos a crear un objeto del tipo Master Panel y le ponemos de nombre **MasterPanel_TravelAgency**.
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 abajo y a su derecha insertamos tablas para que nos ayuden a que el logo nos aparezca alineado a la izquierda.

Objeto Master Panel (cont.)



Ajustamos las propiedades de la tabla contenedora para que la imagen se vea bien. Y Lo mismo hacemos con la Main Table.

Ahora debemos modificar el comportamiento de la imagen para que se ajuste al tamaño que dimos.
Por ahora vamos a crear una clase ImageHeader_TravelAgency a la cual le asignamos la propiedad Scale Type en el valor Fit, pero después cuando entremos en el tema Diseño, veremos como se hace esto con un Design System Object.
Ahora a la imagen del logo le asignamos la clase que creamos.
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

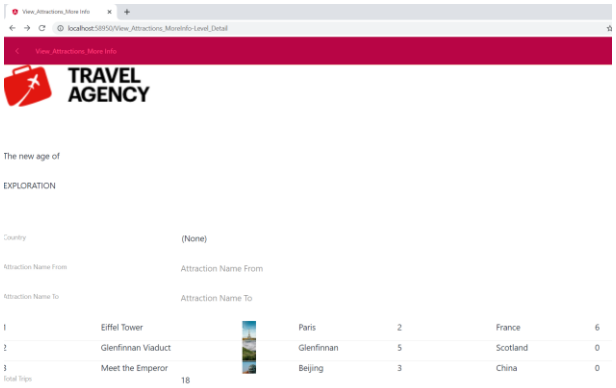
Panel: AttractionDetail_CFPanel

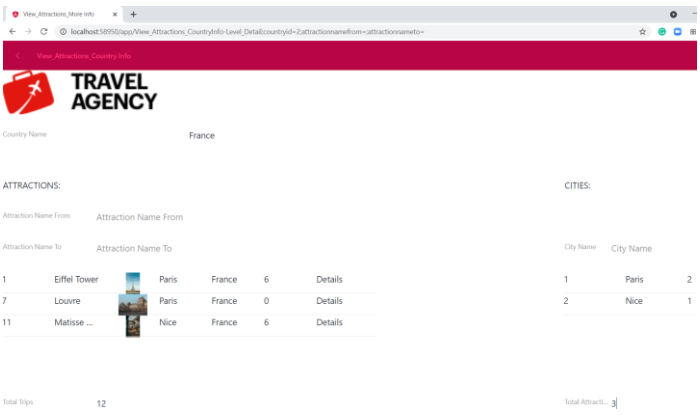
Name	**AttractionDetail_CFPanel**
Description	Attraction Detail_CFPanel
Module/Folder	FrontendAngular
Qualified Name	AttractionDetail_CFPanel
Object Visibility	Public
Main program	False
Master Panel	**MasterPanel_CF**
Caption	Attraction Detail_CFPanel

Ahora asignamos la propiedad Master Panel a los paneles **View_Attractions_MoreInfo y View_Attractions_CountryInfo** con el objeto **MasterPanel_TravelAgency** 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 View_Attractions_MoreInfo. Si hacemos clic en el país France, vemos que también el panel de la información del país tiene el mismo logo.

Aquí no entraremos en el detalle del Design System del objeto Panel, pero podemos adelantar que el uso del objeto Design System Object 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