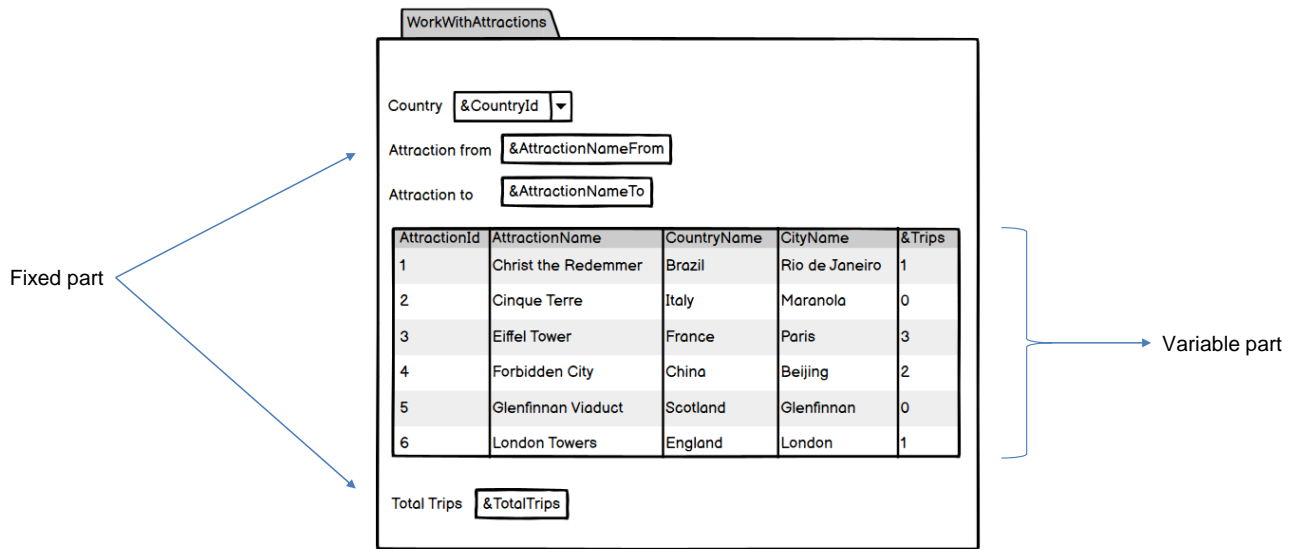


Lógica de carga de datos en la pantalla de un panel

GeneXus™

A continuación veremos cómo se carga la pantalla de un objeto panel, dependiendo de los componentes de interfaz gráfica presentes en ella.

Partes de un objeto panel



En un objeto panel podemos identificar dos partes distintas.

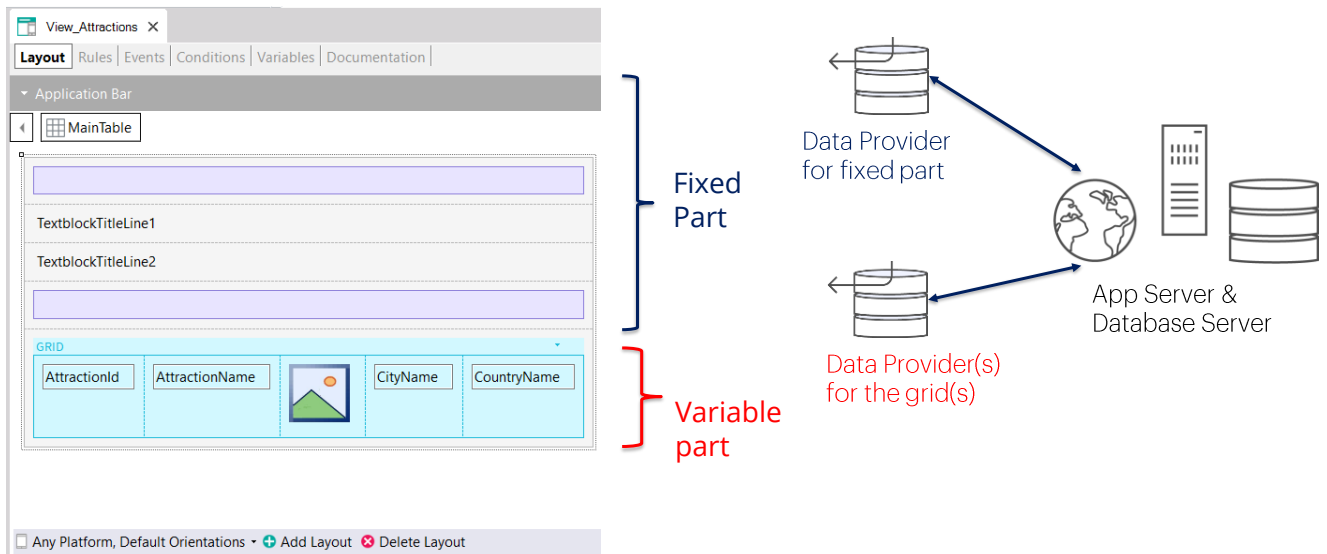
Llamamos parte fija o plana a la parte que contiene todo aquello que esté en el form y que no esté incluido en ningún grid. Esto comprende a controles como textblocks, combos, botones y otros que podemos arrastrar desde la barra de herramientas.

La segunda parte la denominaremos variable, y está compuesta por el o los grids que hayan sido incluidos en el form.

En un objeto panel siempre tendremos una parte fija, y podemos tener una parte variable por cada una de las grillas que tenga el panel.

Los objetos panel se diferencian de los webpanels, en que la parte fija se carga en forma independiente de la parte variable.

Partes de un objeto panel

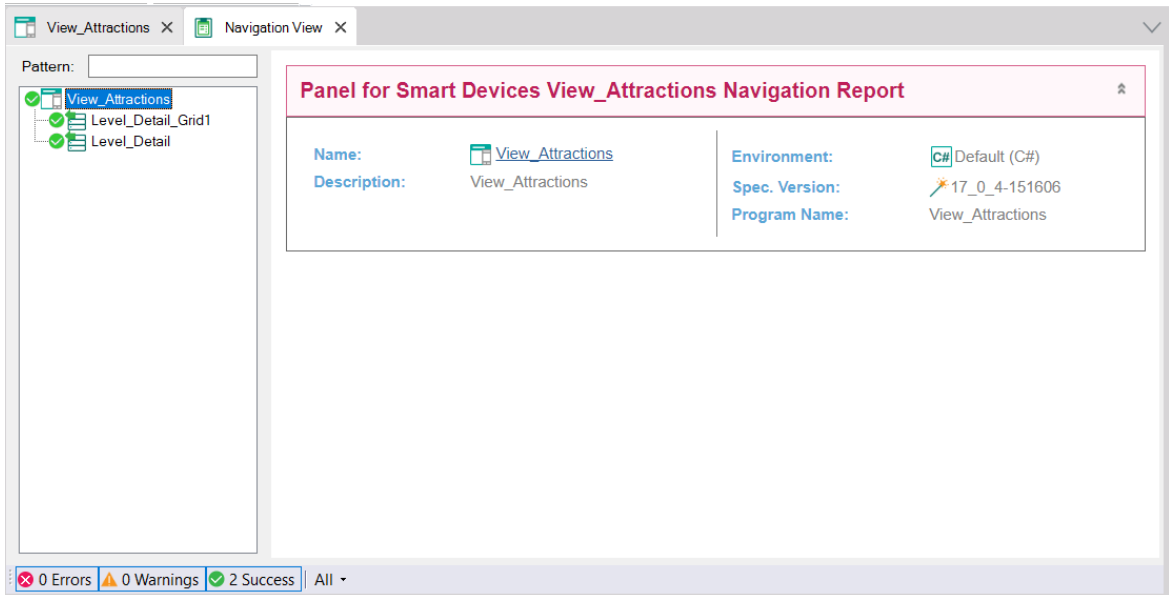


Para cada parte (fija y grid) GeneXus va a generar automáticamente Data Providers independientes que serán publicados como servicios en el servidor y serán los que accederán a la base de datos para obtener la información necesaria para cargar la parte fija y la parte variable del panel.

Estos data providers no los veremos en la Base de Conocimiento ya que GeneXus se encargara de generarlos y mantenerlos, pero si podremos ver a que datos acceden cada uno, si vemos su listado de navegación.

Veamos el listado de navegación del panel View_Attractions que tiene en su parte fija solamente dos textblocks para las líneas del título y tablas separadoras y en la parte variable un grid con datos de atracciones.

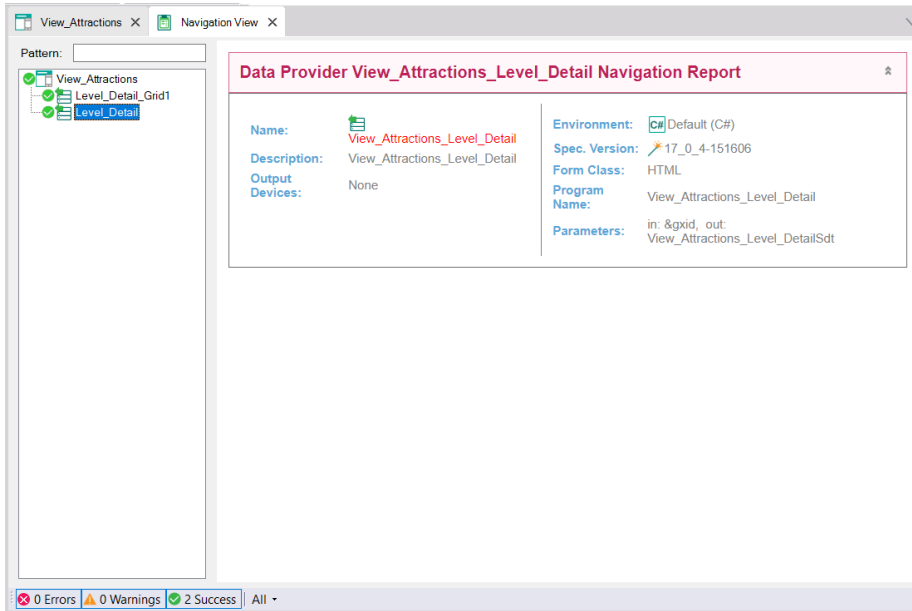
Listado de navegación de un panel



En el listado de navegación del panel View_Attractions vemos que bajo el nodo correspondiente al panel aparecen dos entradas, una con el nombre Level_Detail_Grid1 correspondiente a la parte variable del panel que en este caso está compuesta por el Grid1 y otro llamado Level_Detail que será la que muestre la carga de la parte fija del panel.

Si el panel hubiera tenido más de un grid, aparecería un nodo de detalle por cada grid, ya que cada una tendrá una navegación propia.

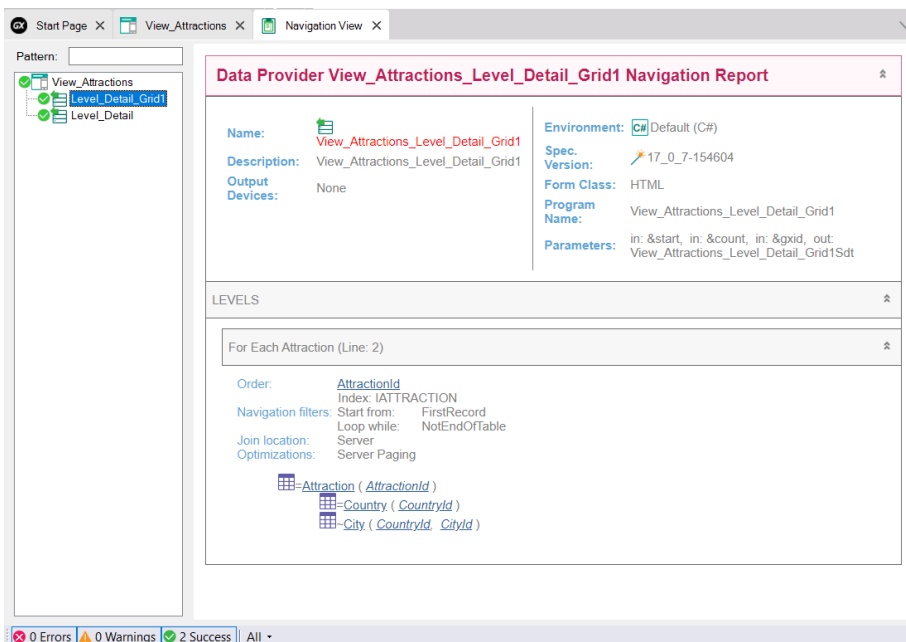
Listado de navegación de un panel



Si seleccionamos el nodo `Level_Detail`, vemos que es el listado de navegación del data provider `View_Attraction_Level_Detail`. El listado aparece vacío, ya que como lo único que tiene la parte fija del panel son dos textblocks con datos fijos para el título, no es necesario ir a la base de datos a recuperar información.

Si en esta parte fija hubiéramos agregado un Dynamic combo box o cualquier otro control que requiriera acceder a datos de la base de datos para mostrar información, aquí aparecería la navegación de acceso a esos datos.

Listado de navegación de un panel



Si ahora seleccionamos el nodo llamado `Level_Detail_Grid1`, vemos que es el listado del Data Provider `View_Attractions_Level_Detail_Grid1`.

Este es el data provider que se creó automáticamente para cargar el Grid1, publicado como servicio en el servidor e invocado por el panel para acceder a la base de datos y recuperar las atracciones.

Si reparamos en donde dice Environment, observamos que el data provider fue generado en C# (.NET) como parte del código del backend.

En la sección Parameters, vemos que el data provider recibe algunos datos del grid y devuelve cargado un SDT de nombre `View_Attractions_Level_Detail_Grid1`, que contendrá los datos para cargar el grid.

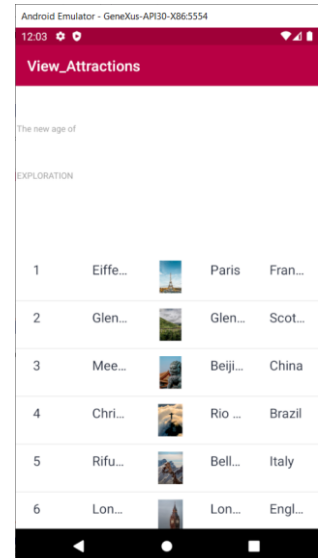
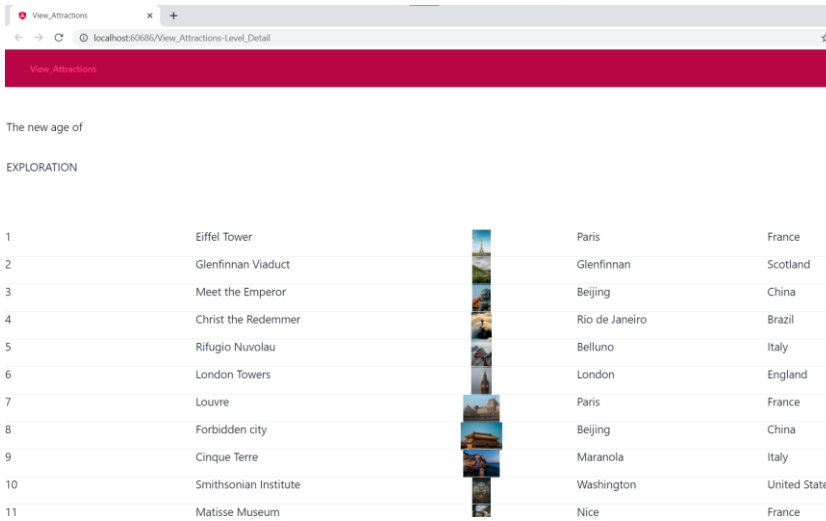
La información del listado es similar a la que veríamos con un web panel con tabla base Attraction, ya que la grilla del panel está recorriendo la tabla Attraction para mostrar las atracciones turísticas y para cada atracción accede a la tabla Country para recuperar el nombre del país y a la tabla City para traer el nombre de la ciudad de la atracción.

Dado que la parte fija y las partes variables del panel se cargan en forma separada, podemos adelantar que las tablas bases de cada parte son completamente independientes entre sí. Es decir que en un objeto panel con un solo grid puede haber 2 tablas bases, una para la parte fija y otra para el grid, a diferencia de los objetos webpanel donde en ese caso la

tabla base era única.

En otro video entraremos en detalle en cómo se determinan la tablas base de cada parte de un panel.

Ejecución del mismo panel en distintas plataformas

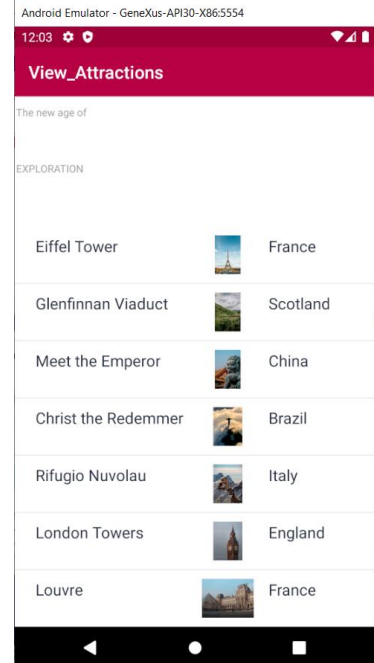
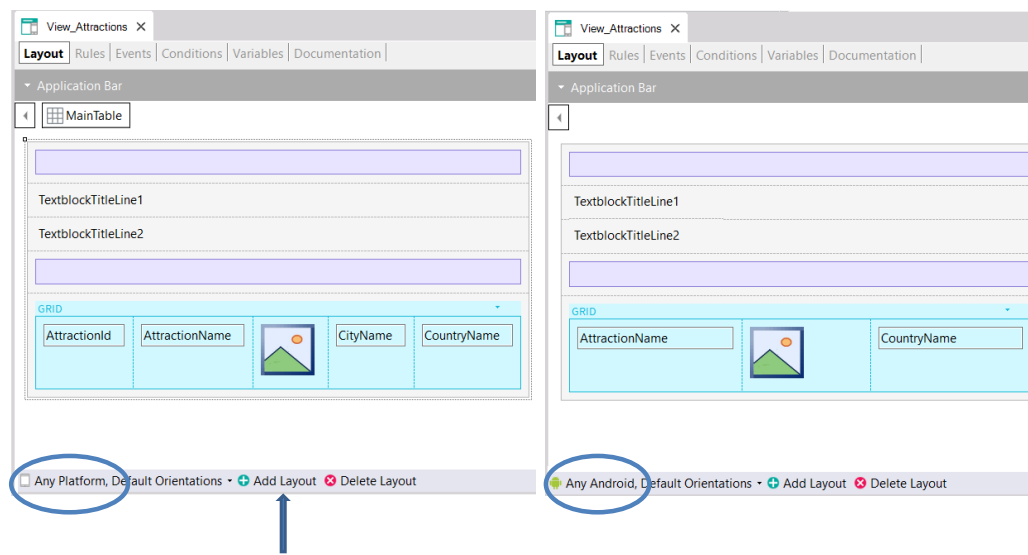


El mismo objeto panel puede ser ejecutado sin cambios tanto en una plataforma web, cuando se genera por ejemplo en Angular, o en plataformas nativas móviles cuando es generado en Android o iOS.

Aquí vemos la ejecución del mismo panel en un desktop web generado con Angular y en un celular Android.

Debido al espacio disponible, la información no se ve igual en cada caso.

Layouts de un panel



Para que el diseño de cada pantalla sea el adecuado, tenemos la posibilidad de crear distintos layouts para cada plataforma. Cada layout será independiente de los otros y en cada uno podremos disponer los elementos de la interfaz gráfica como creamos mejor.

También podemos crear varios layouts para una misma plataforma, por ejemplo si estoy solamente viendo información o si la estoy editando y deseo que el aspecto o la cantidad de información mostrada sea diferente en cada caso.

El form de un panel es un contenedor de uno o varios layouts y cada uno podrá contener una cierta cantidad y distribución de controles diferente, pero la lógica de carga de datos en la pantalla será la misma que la vista anteriormente, es decir que cada layout podrá tener una parte fija y una o más partes variables en caso de contener grids.

Por lo que para cada uno de ellos se creará en forma independiente, los data providers necesarios para obtener la información de la base de datos cuando se requiera.

En próximos videos entraremos más en detalle sobre estos temas.

GeneXus™

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