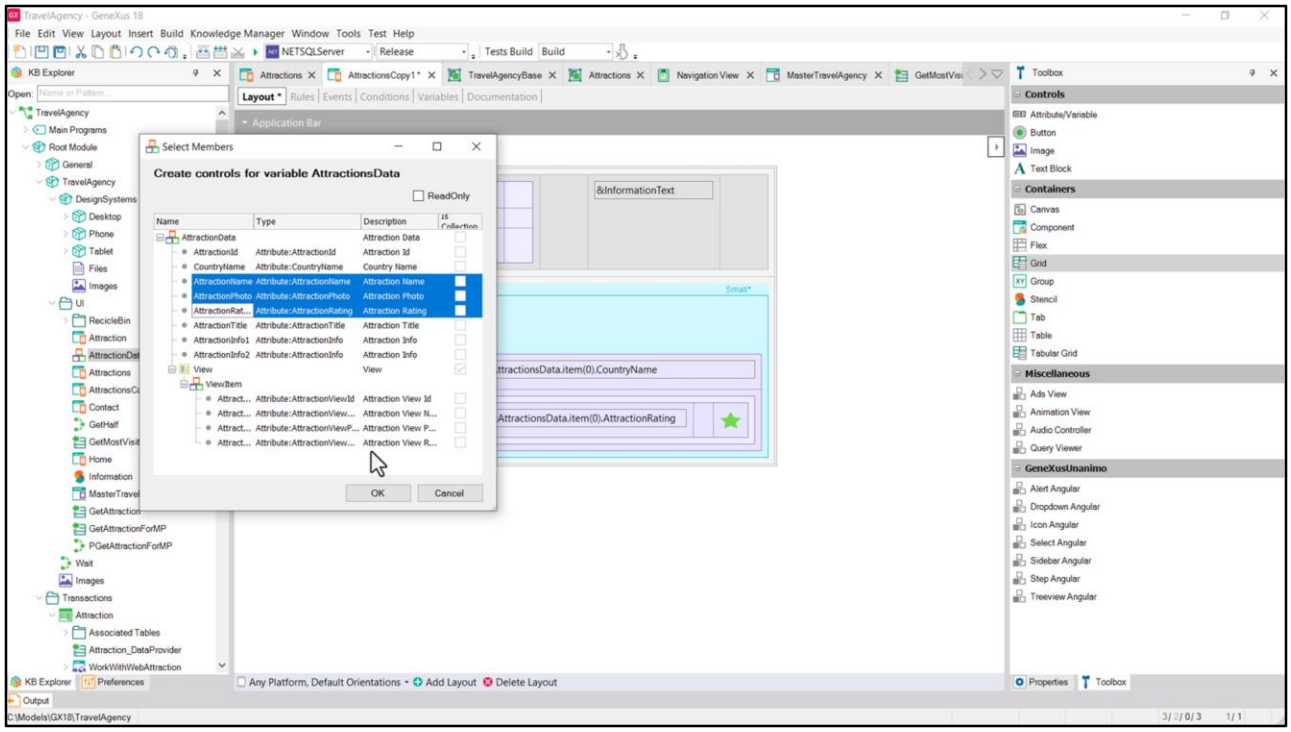
GeneXus by Globant

Attractions Panel: Carousel (Grid control) part 2

Cecilia Fernández

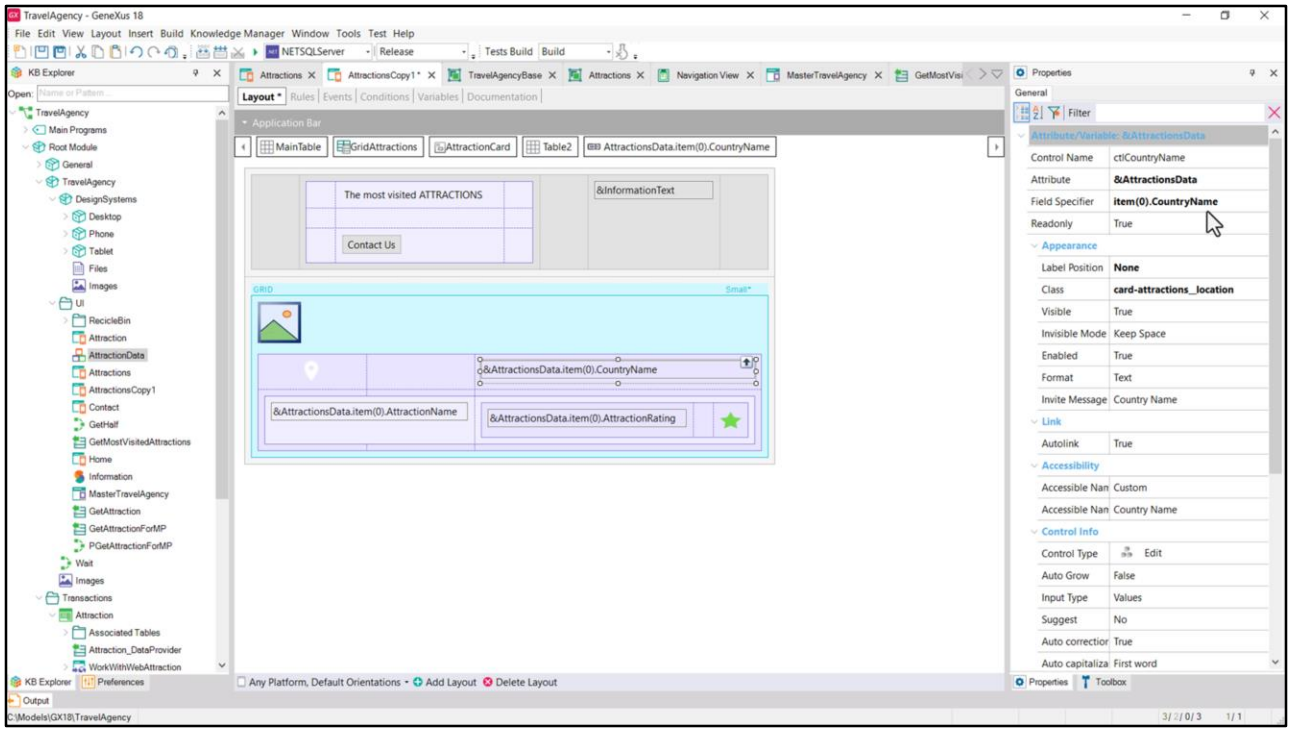


Bueno, vamos a continuar ahora allí donde dejamos en el video anterior.

Como no se los mostré, tal vez se pregunten cómo inserté todos estos campos de cada ítem de la colección.

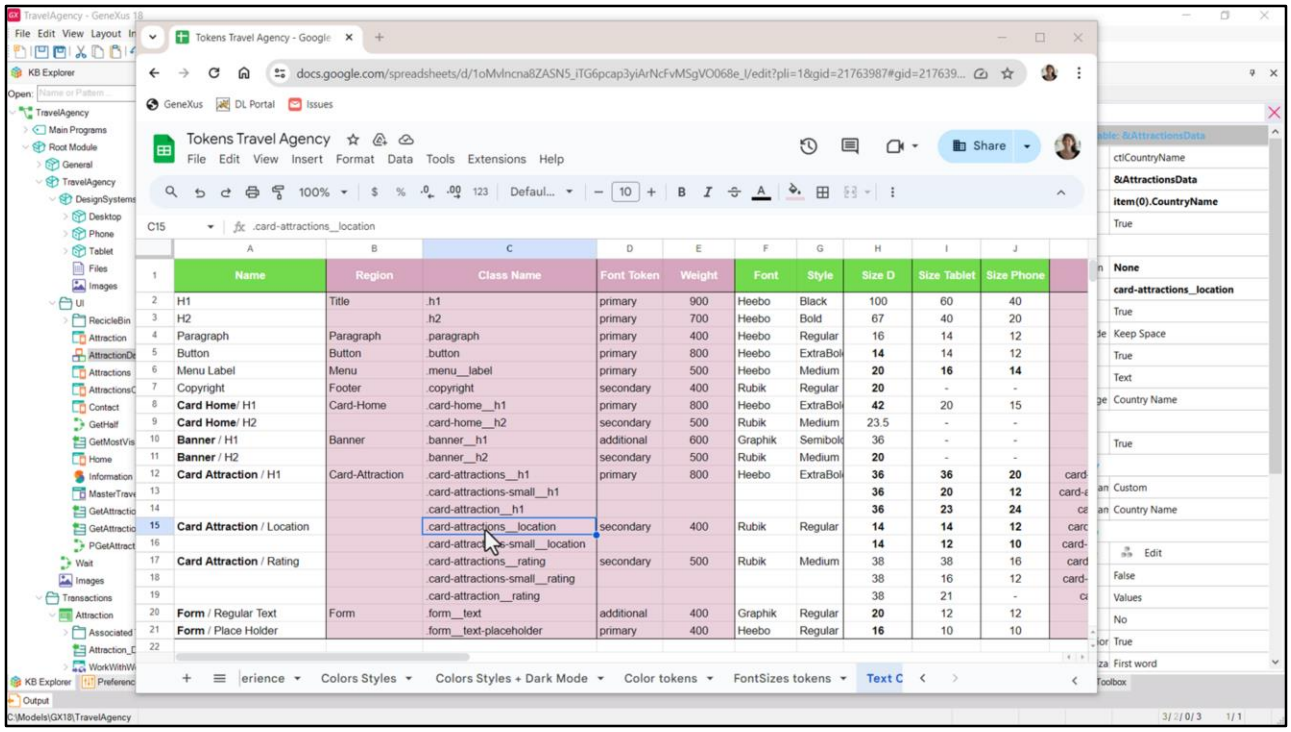
En verdad fue sencillísimo: cuando fui a crear el grid -lo haré con otro- y elegí la variable SDT colección, automáticamente me pregunta cuáles de sus elementos querré insertar en el layout. Vean qué pasa si elijo estos por ejemplo.

Allí vemos que queda igual que el grid de arriba. Voy a eliminar este.

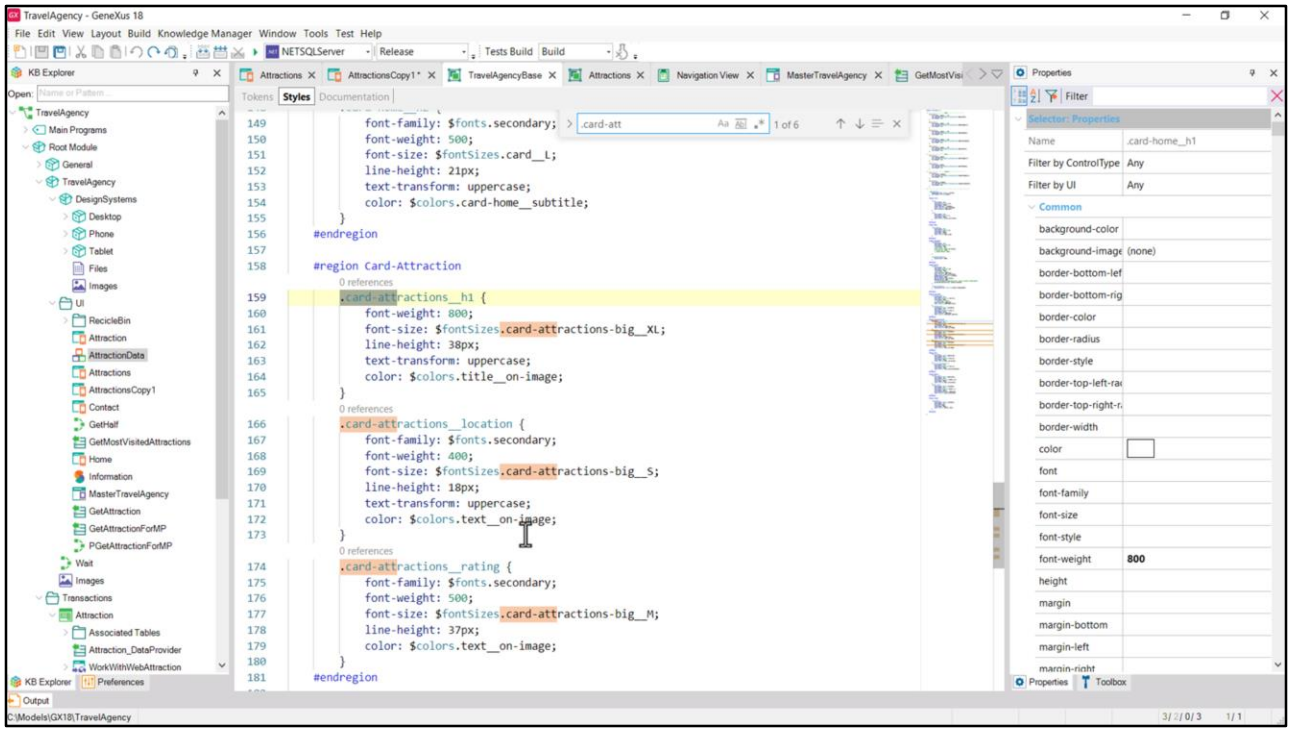


Bien, y si me posiciono sobre uno de los elementos vemos que así queda identificado.

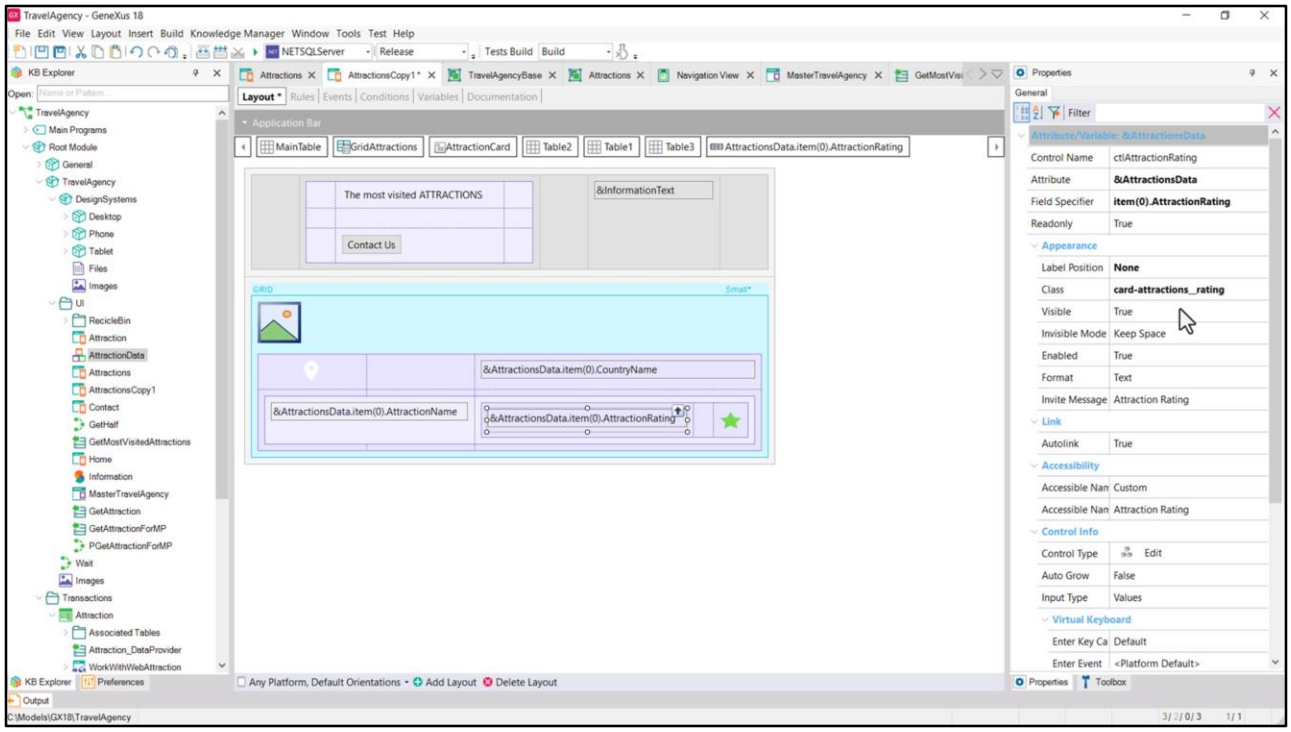
Aprovechemos a observar que le dejé la Label position en None, para que no salga la etiqueta de la variable, que por supuesto es readonly.



Y también veamos cómo le apliqué a todos los textos las clases de la tipografía que habíamos ingresado hace tiempo, en la etapa de preparación, ¿se acuerdan?

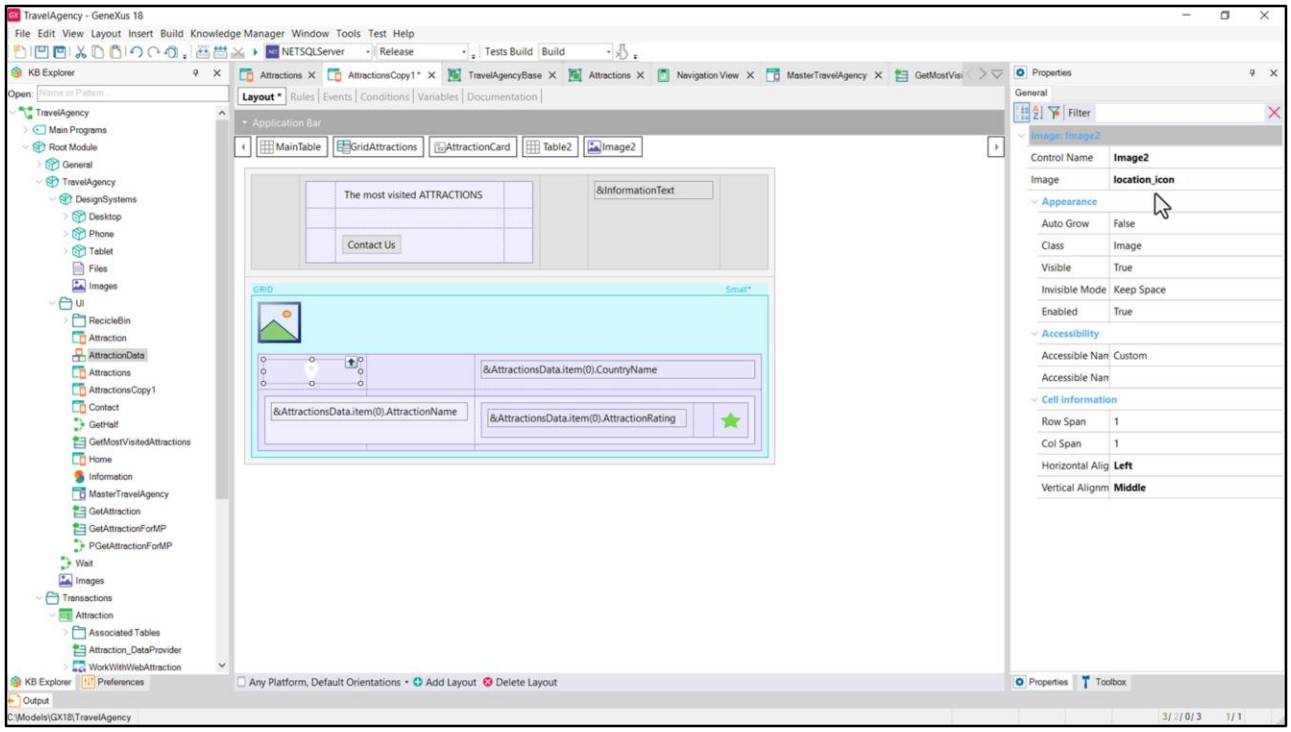


Filtro por card-attraction… y vemos en la región Card Attraction estas tres clases…

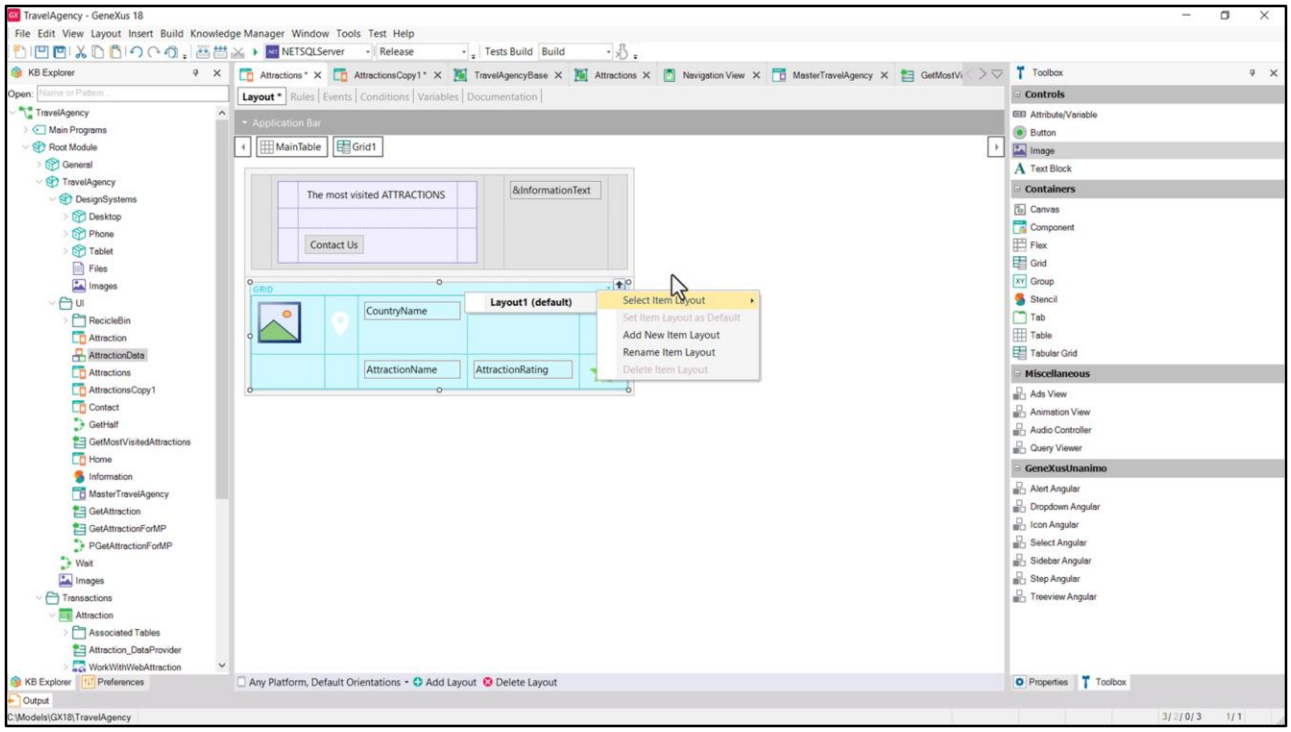


…que son justamente las que aplicamos a estos elementos.

Aquí la del rating… aquí la h1.



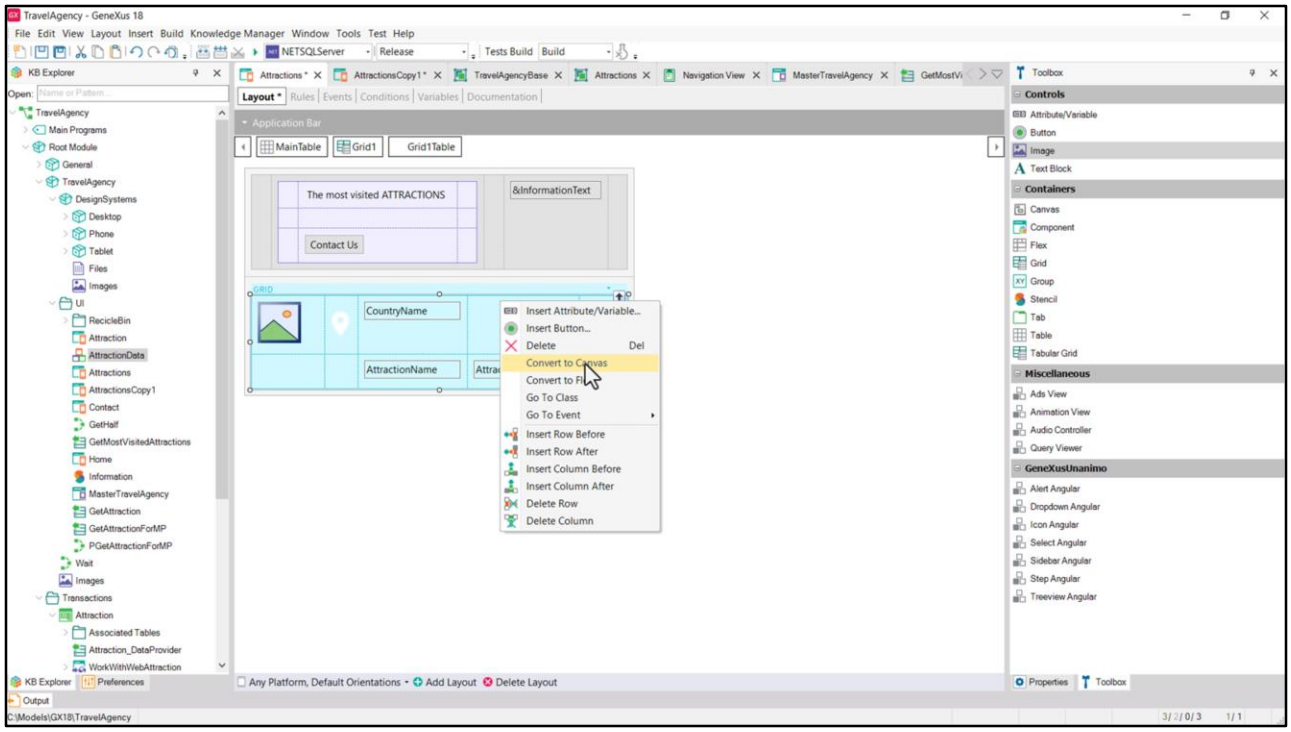
Por otro lado debí descargarme esta imagen de Figma e ingresarla en la KB porque en la etapa de preparación se me había pasado por alto.



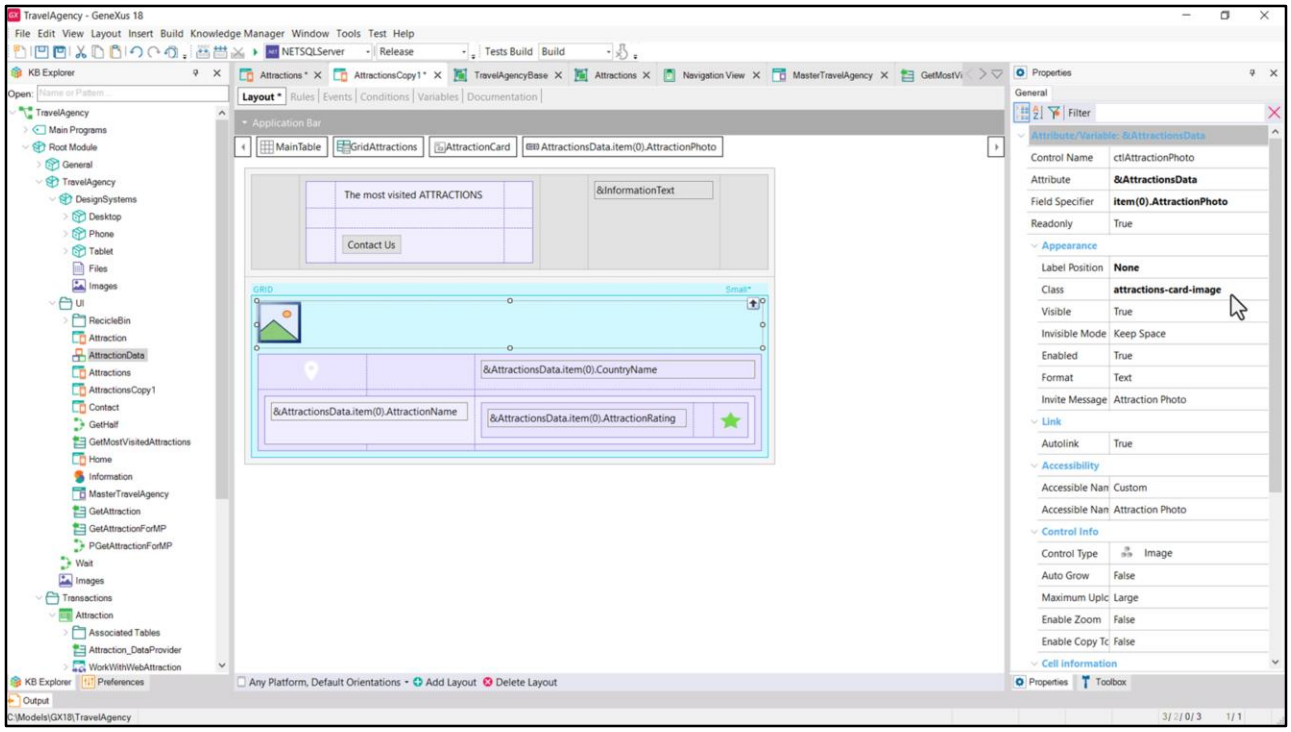
Y lo que hice fue insertar estas dos imágenes.

Podemos hacerlo en nuestro otro grid, el que estamos queriendo implementar con atributos en lugar de la variable SDT, para ver, pero esto es muy sencillo.

Y en verdad el trabajo que nos queda por hacer aquí, a nivel del layout que en este grid por el momento es universal, es colocar todos estos controles de una manera apropiada para poder implementar la card, donde se van a superponer controles…

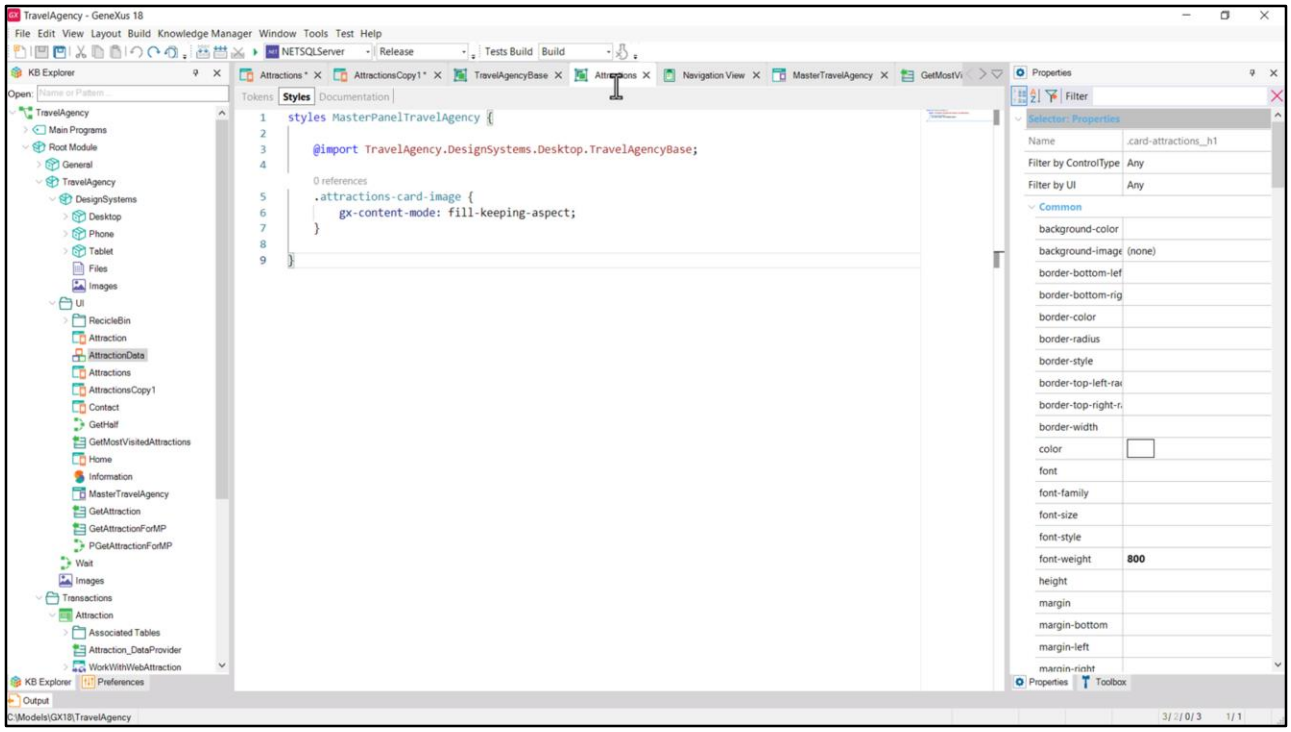


…por lo que ya sabemos que esta tabla en verdad deberá ser un canvas.

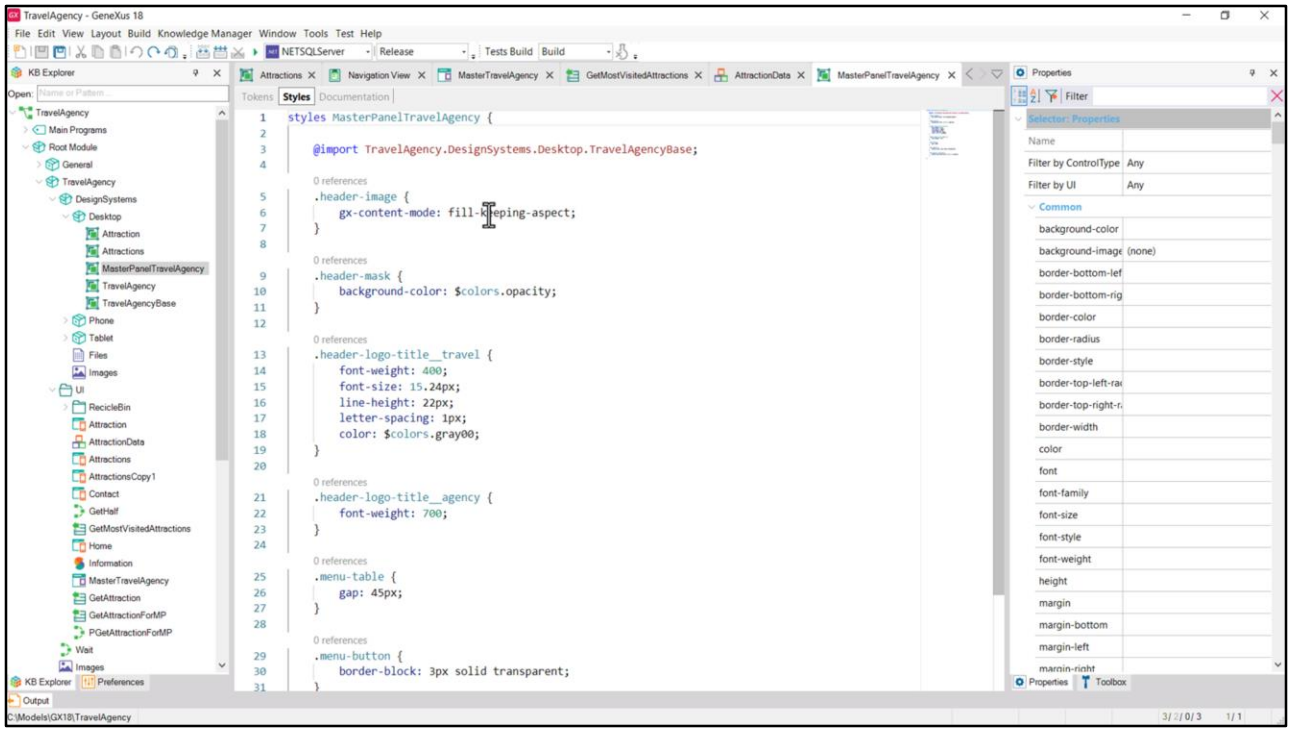


Pero vean cómo estructuré dentro de mi otro grid los elementos internos al canvas.

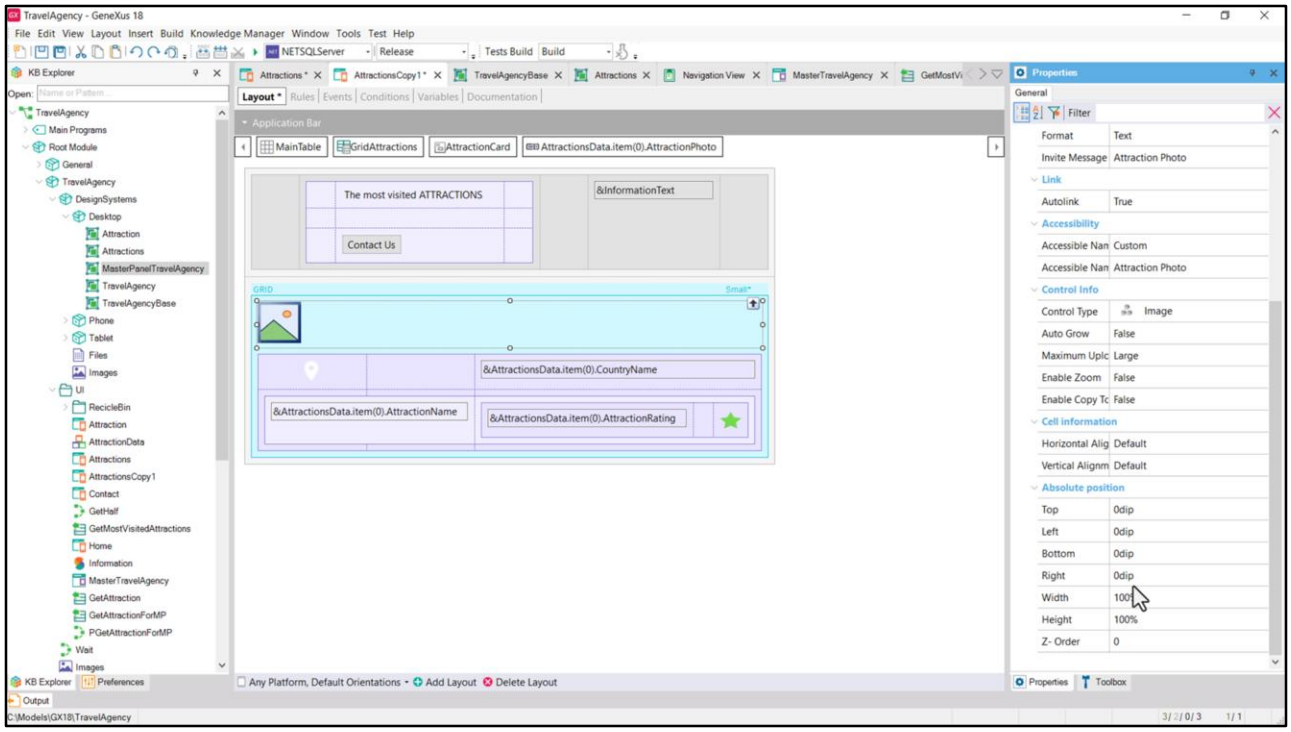
La imagen, suelta, vean que tiene esta clase que creé especialmente dentro de un DSO específico para diseñar lo específico de este panel.



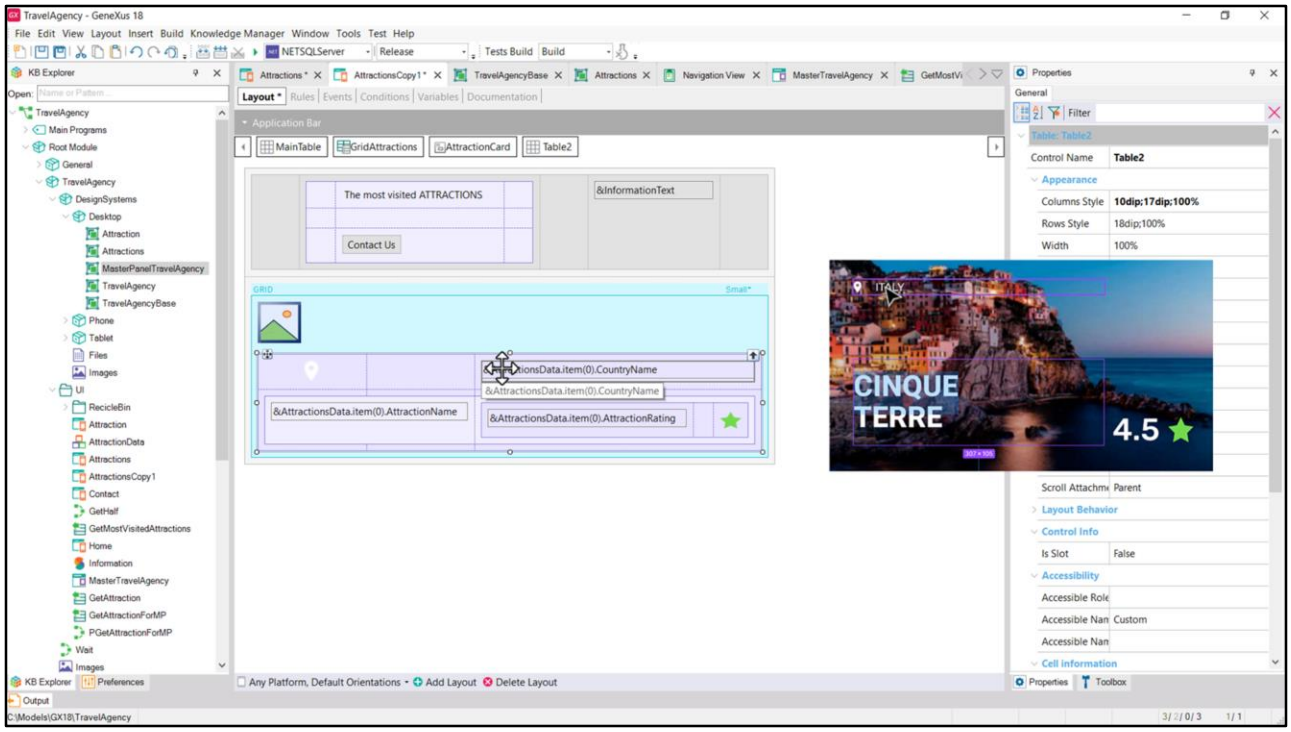
Por eso le llamé igual que al panel: Attractions. Aquí tengo la clase.



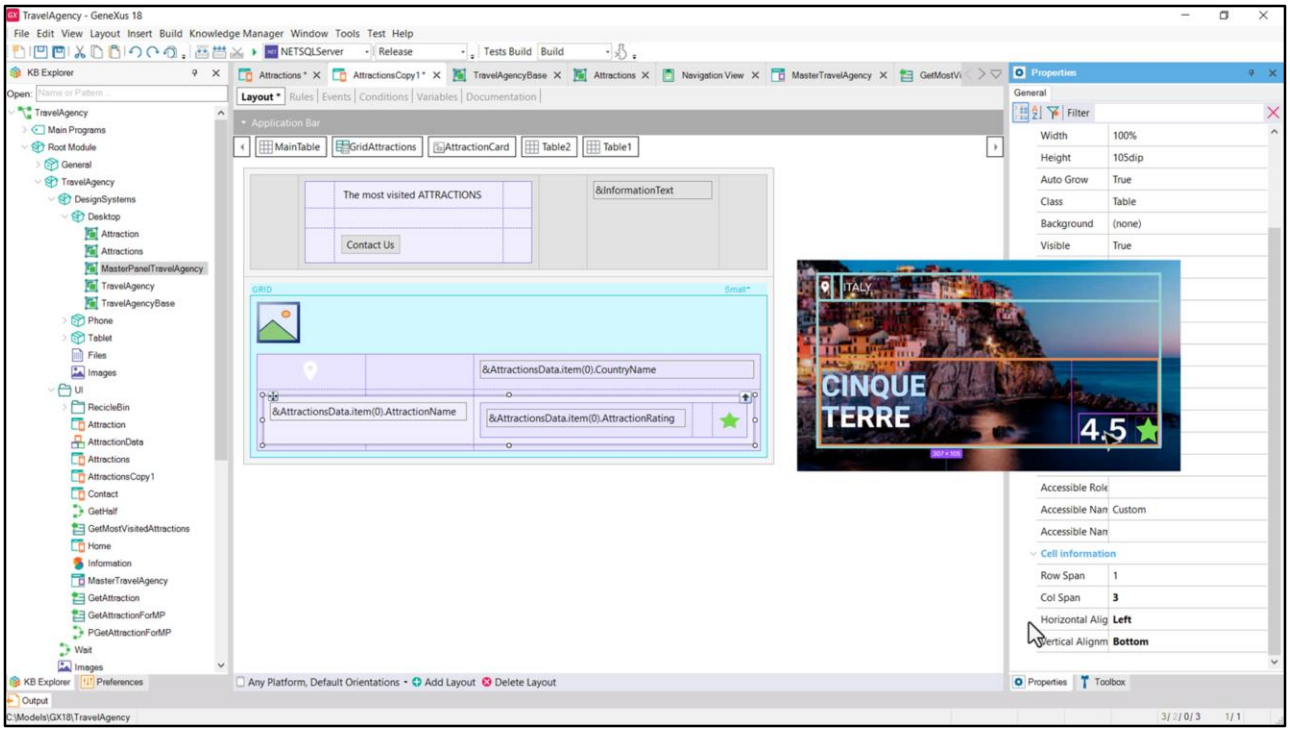
Ya hicimos esto antes, ¿se acuerdan?, para el Header. Pero no quise utilizar aquí su misma clase, por si más adelante necesitara diferenciarlas.



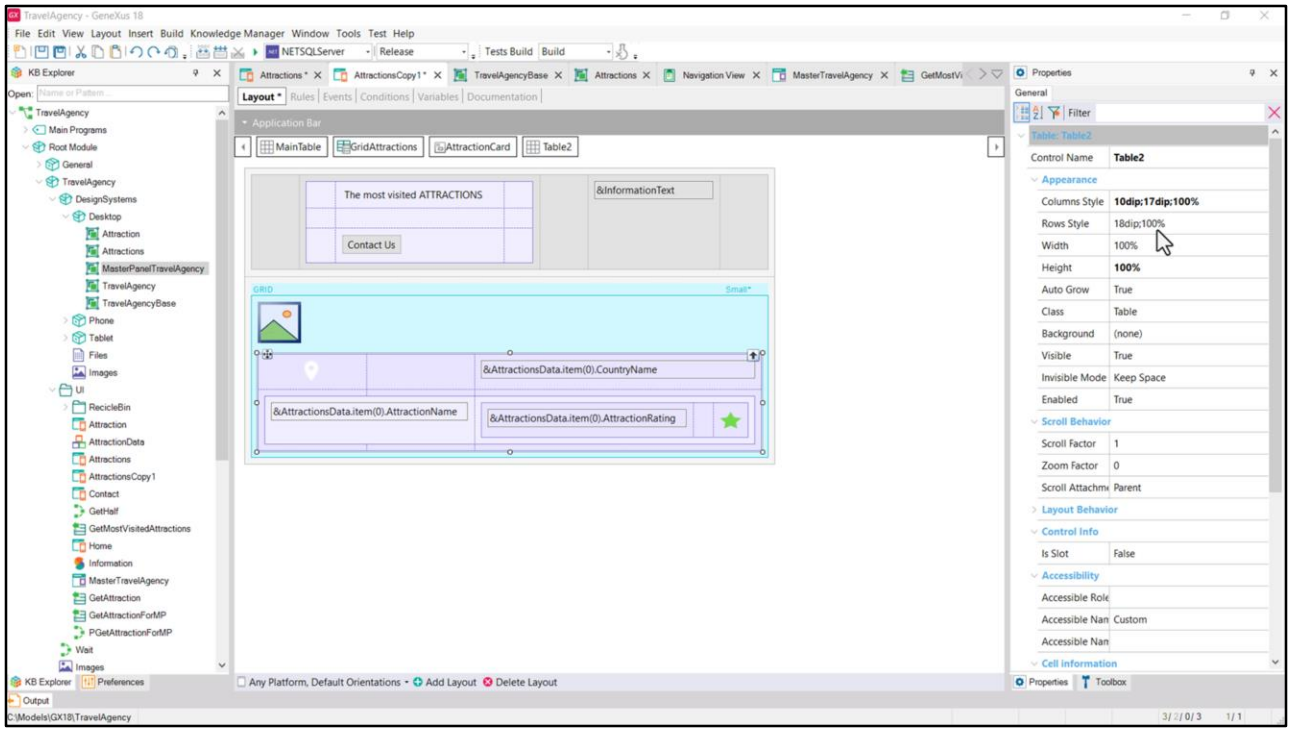
Debemos dar su posicionamiento absoluto y como vemos, será tal que ocupe el 100% de ancho y alto del canvas, pegándose a sus bordes. Y la capa la más profunda.



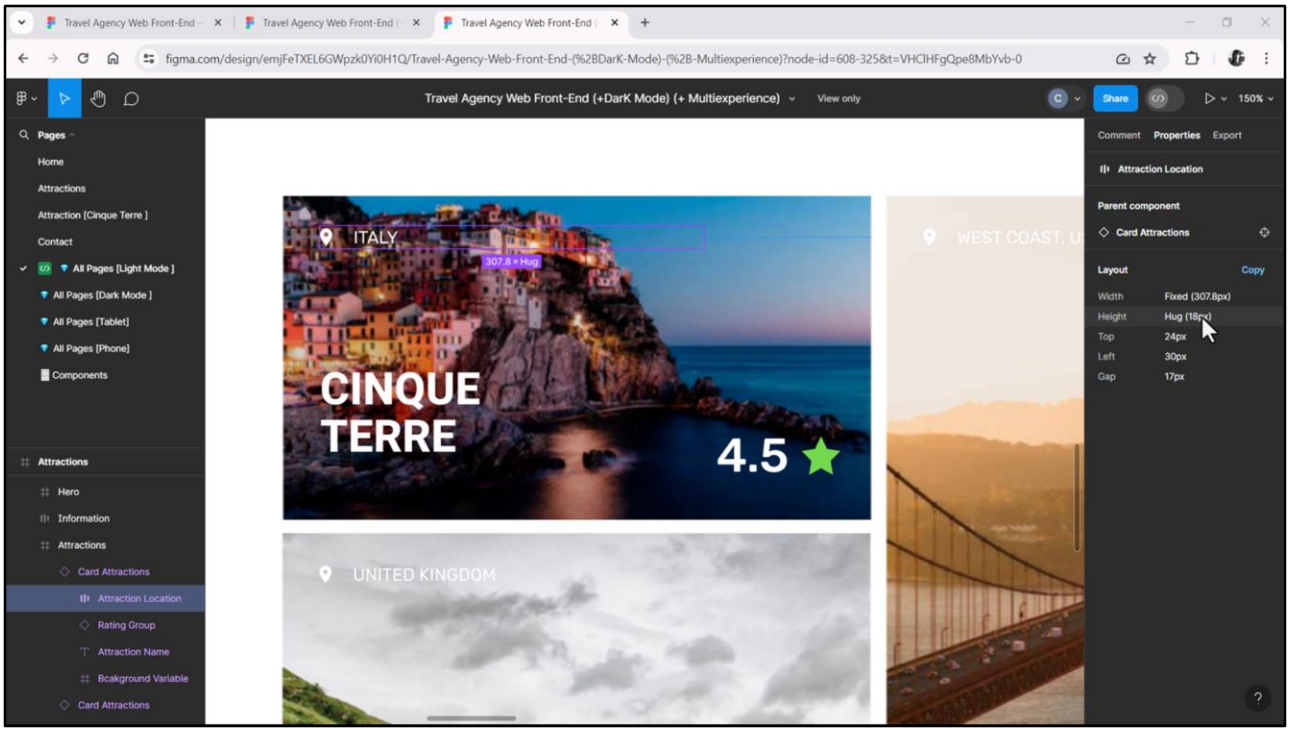
Luego veamos que coloqué todos los otros elementos en otra tabla, con dos filas y 3 columnas. En la primera fila coloqué el ícono, un espacio, y el nombre del país…



Y en la segunda fila otra tabla, que se expande entre las 3 columnas de la tabla contenedora. Se alinea a la izquierda horizontalmente, y verticalmente abajo, en relación a la celda que la contiene, que corresponde a la segunda fila de esta tabla.

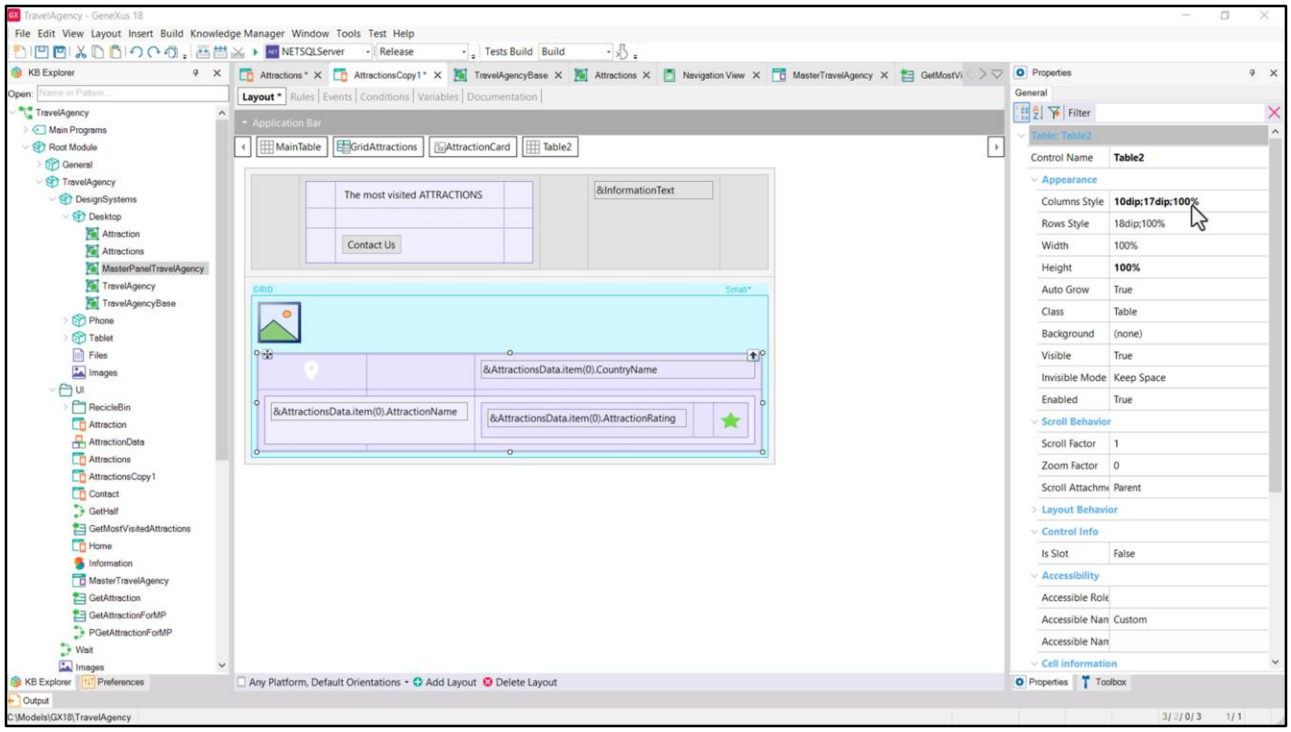


A la que vemos que le dí de alto 100%, porque a la primera fila le coloqué 18 dips…

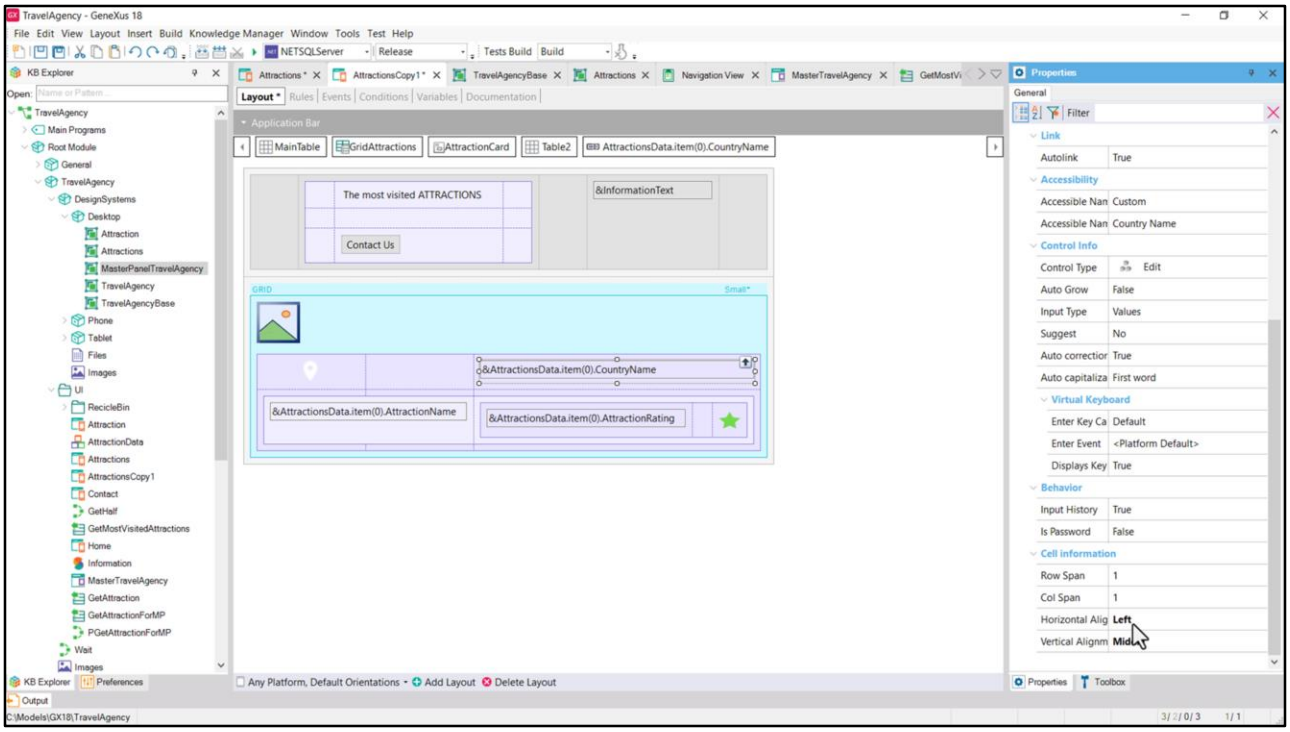


…que extraje de aquí.

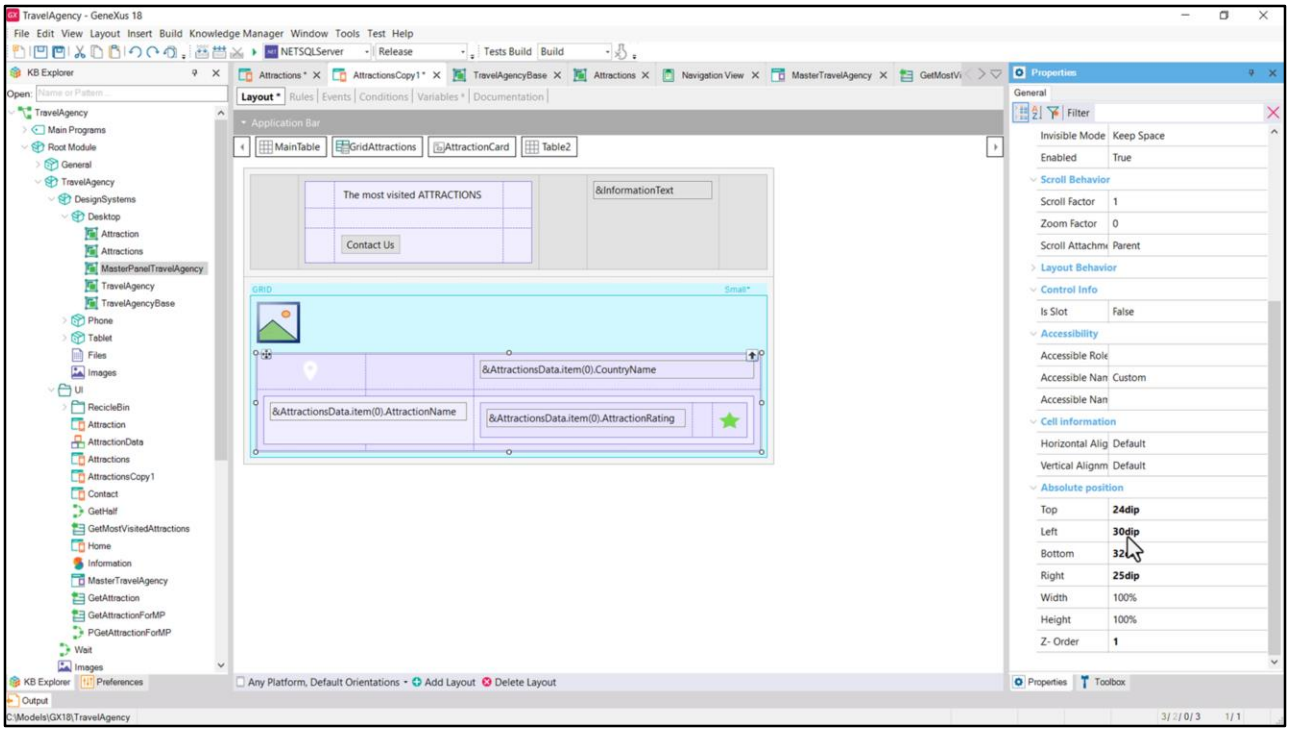
Y ya vayamos viendo estos 24 dips de arriba del canvas, estos 30 de la izquierda y este gap de 17 dips entre el ícono, de casi 10 dips de ancho, y el texto ITALY. Y que están centrados verticalmente, también podemos ver.



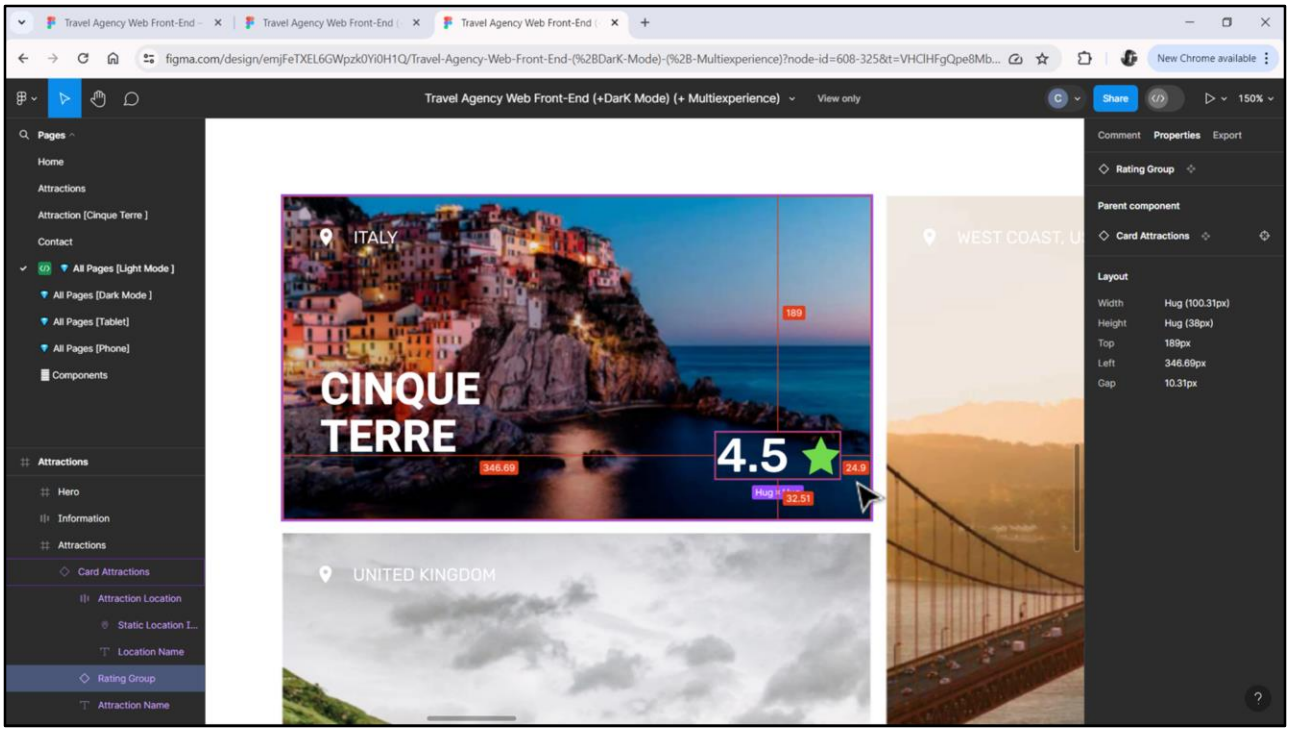
Y es así que dejé 10 dips para la columna del ícono, 17 de aire, y la tercera columna que se expanda al 100% restante.



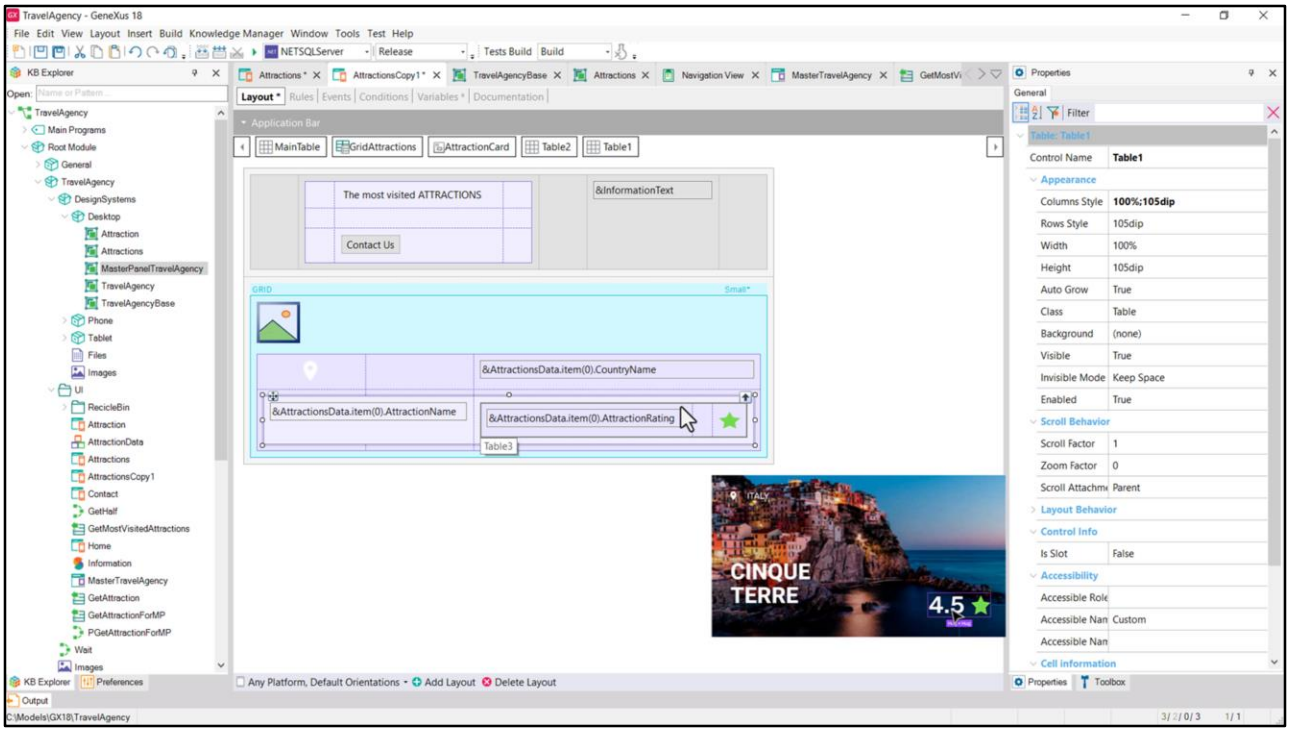
Pero con la condición de que esta variable esté horizontalmente alineada a la izquierda.
Y ambos controles, verticalmente, en el medio.



La tabla está dentro del canvas, así que fue ubicada respecto a sus bordes… Recién veíamos que se iniciaba a 24 dips de arriba y a 30 de la izquierda. Y veamos de dónde salen estos 32 de bottom y estos 25 de la derecha.

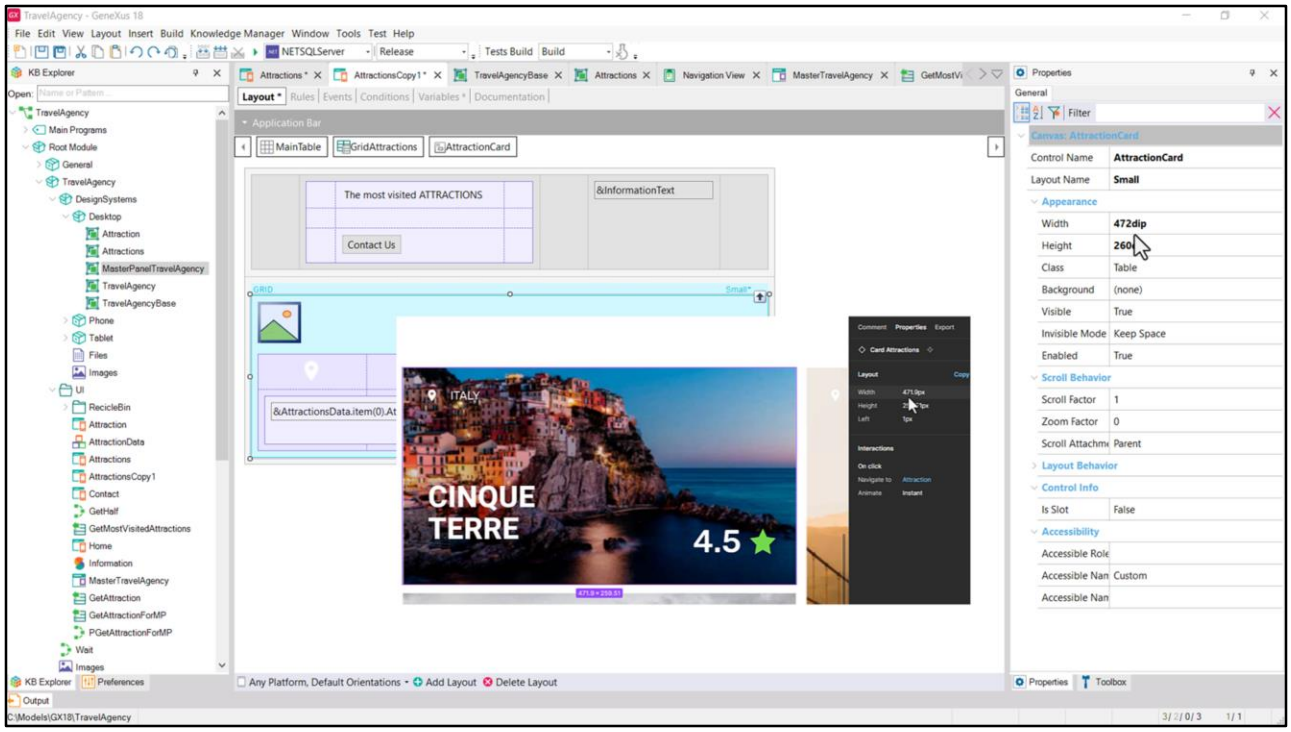


Ahí los vemos. Nosotros los redondeamos.



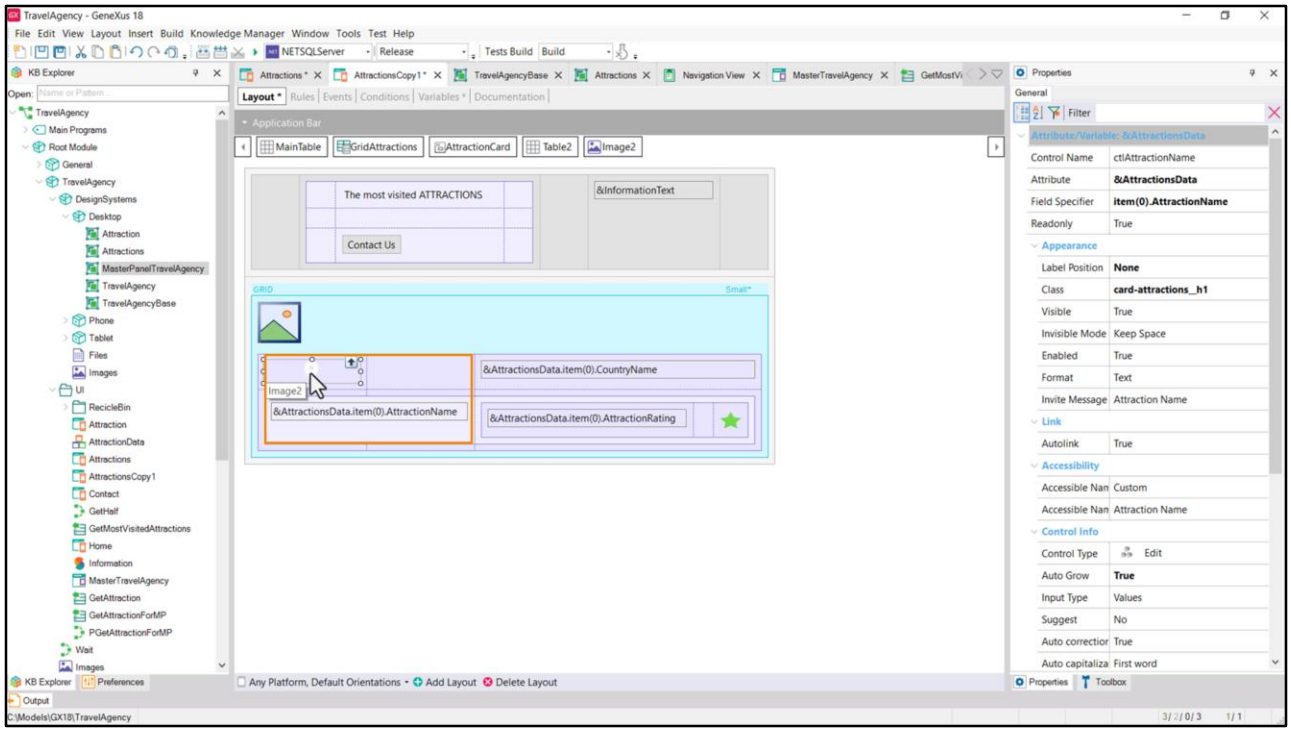
Bien. A la tabla la colocamos con Z order 1, para que esté por encima de la imagen.

Y ahora analicemos esta sub tabla. Vean que tiene 2 columnas, la de la derecha de 105 dips, que es el espacio máximo que podrá ocupar de ancho esta otra tabla; y la de la izquierda, que va a contener a esta variable, que tiene un espacio del 100% restante del ancho de esta tabla para ocupar.



¿Y cuál es el ancho de esta tabla? El 100% de su contendor, que es la fila 2 entera, que como suma las 3 columnas porque se expande en las 3 columnas, es claramente el 100% del ancho de la tabla….

¿Que cuál es, entonces? El que resulta de restar del ancho del canvas… los bordes Left y Right.

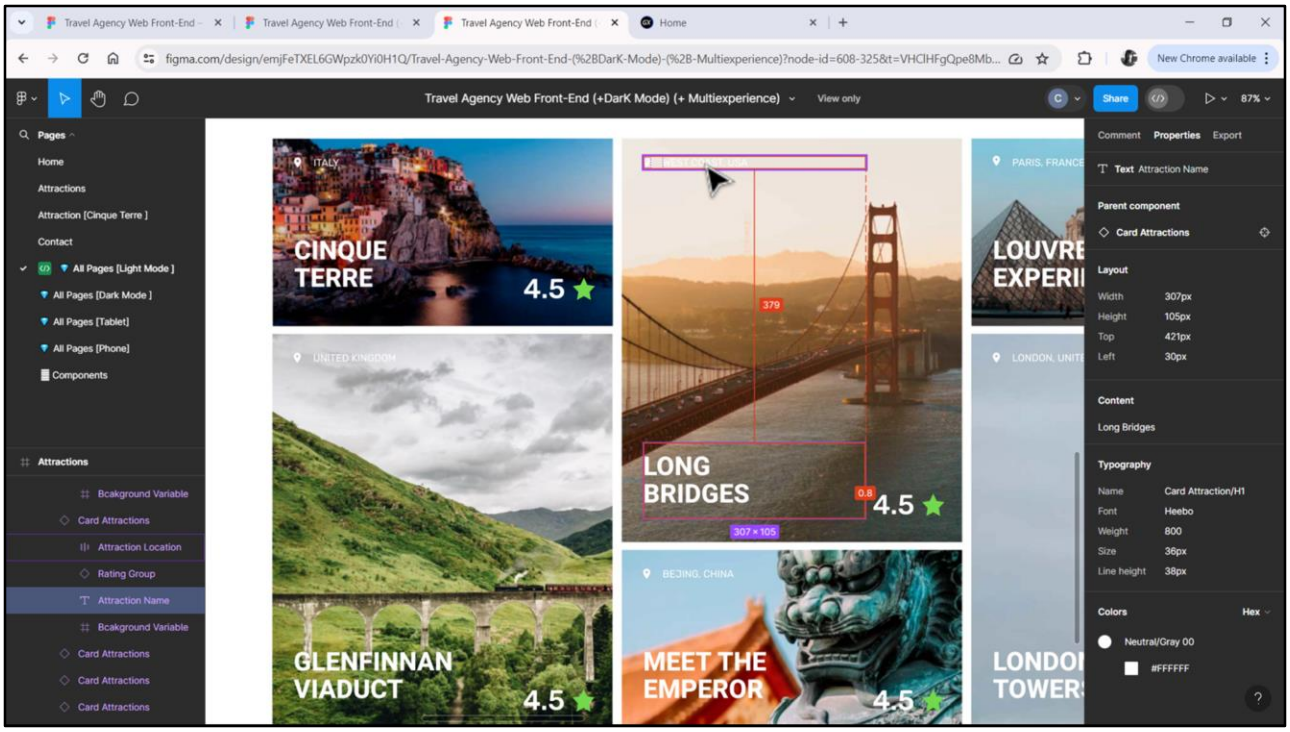


Podría seguir analizando cada valor que di a cada propiedad, pero como es más de lo mismo y no los quiero aburrir vamos a dejar por acá. En el xpz que le dejo pueden investigar todo esto al detalle.

Como vengo repitiendo desde el principio, casi nunca hay una única manera de implementar las cosas. La ventaja de haber colocado este y este control en una tabla, es que modelo la alineación por la izquierda de forma óptima. Imaginen que deba cambiar la distancia del borde izquierdo del Canvas, por ejemplo. Lo hago una única vez, para la tabla. Si tuviera a estos controles ya sea sueltos, ya sea en tablas independientes, tendré que hacerlo para cada uno el cambio.

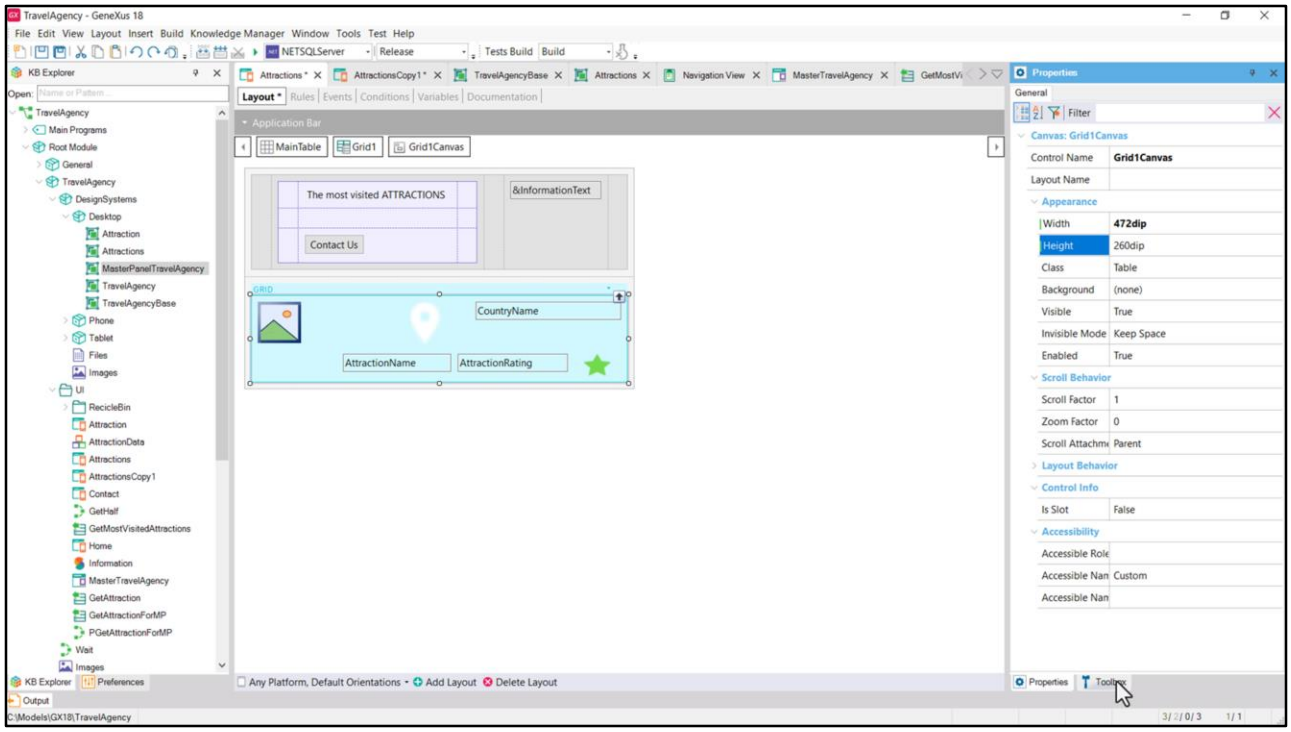
Lo mismo vale para controles que sabemos deben alinearse horizontalmente de algún modo, como estos.

Cuando hay en el diseño una estructura evidente que interrelaciona los controles, entonces conviene utilizar tablas (en las variedades que convengan, como la flex para algunos casos, por supuesto).



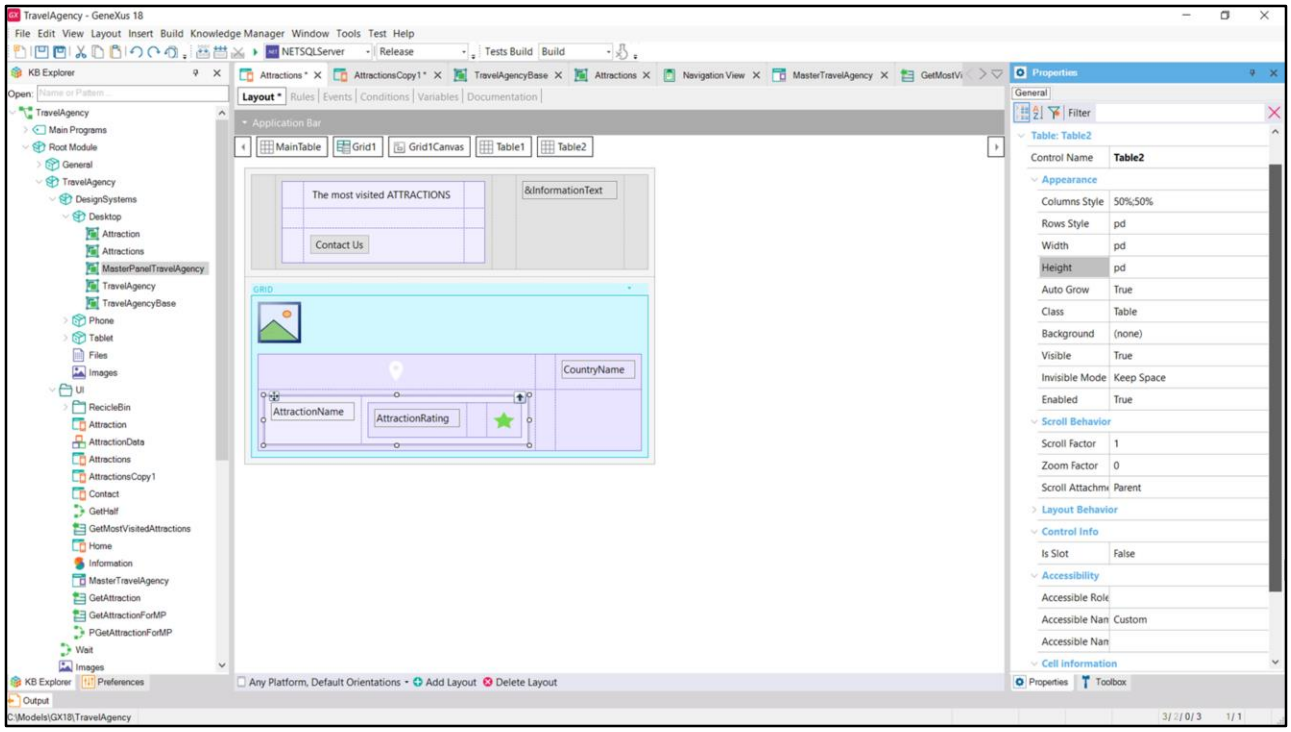
Ahora quiero contarles cómo analicé las diferencias entre las cards pequeñas y las largas. Una sabíamos que tenía 260 dips de alto y la otra 560. Esa diferencia de 300 dips viene dada únicamente por este espacio vertical.

Es decir, para convertir una card larga en una pequeña, lo único que se necesita es quitar este espacio del medio de 300 dips.
Observen que la distancia entre este elemento y este otro es de 80 dips aquí, mientras que aquí es de casi 380. Es decir, 300 es la diferencia entre una y otra.



Entonces, pude dedicarme primero a modelar la card Small como si fuera la default. Y cuando la tuve terminada recién ahí implementé la Large. Y esto mismo podríamos hacerlo, bien rápido, con nuestro grid con atributos.

Aquí establezco las dimensiones del canvas.

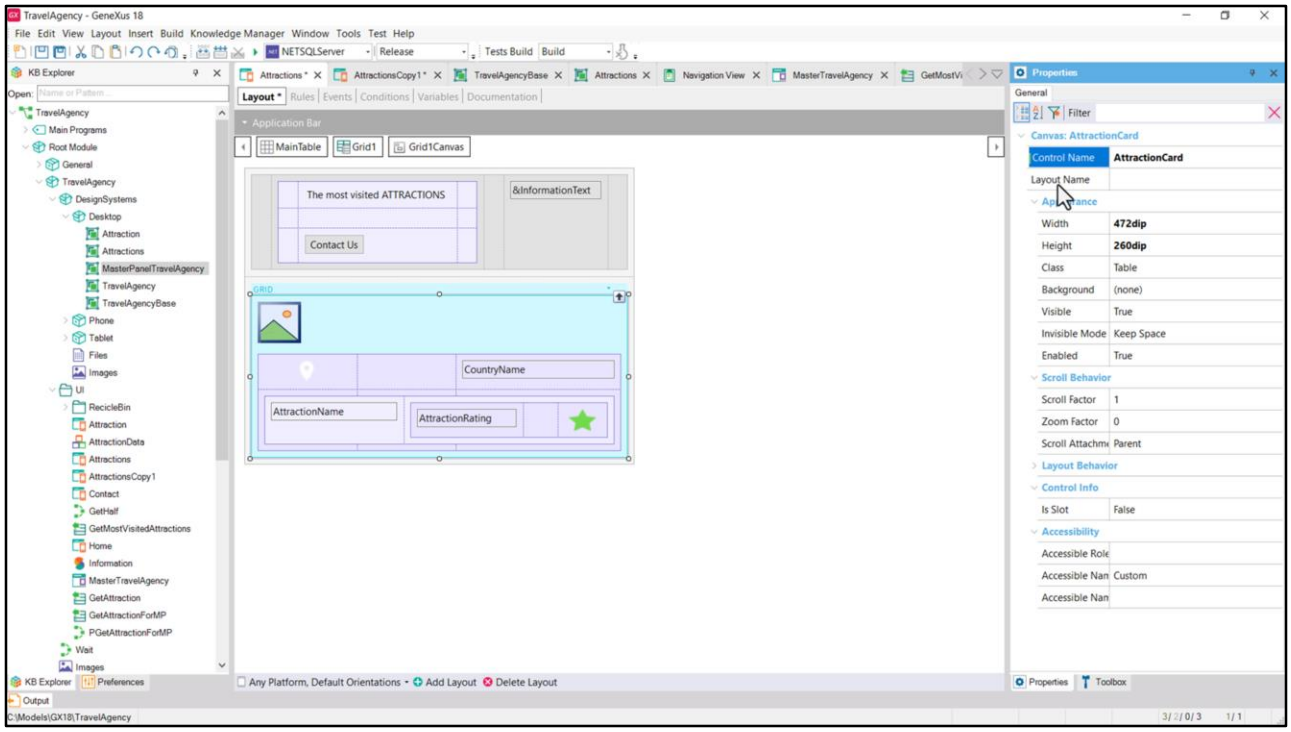


Y luego empiezo a modelar los controles tabla, a colocar entonces nuestros controles imagen y atributos dentro de las tablas que habíamos identificado.

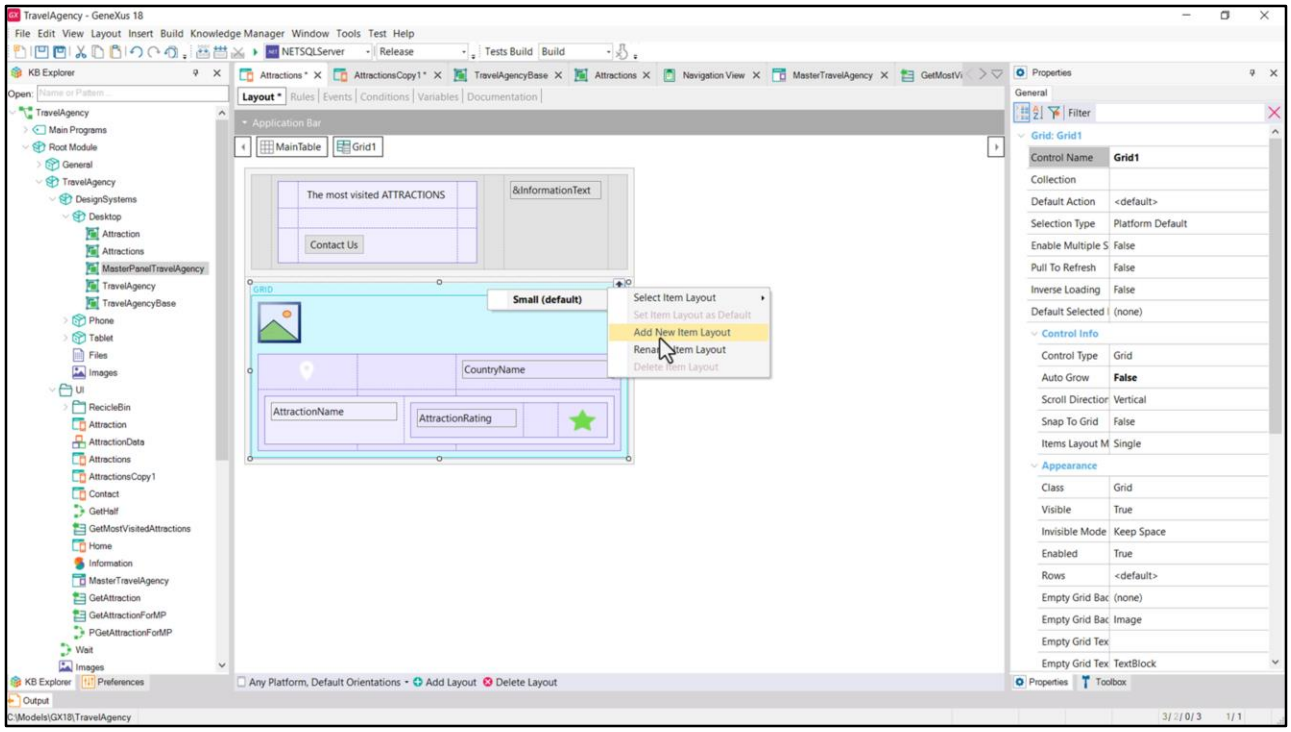
Por último coloco esta como segunda fila de la tabla principal, y hago que se expanda en las 3 columnas.

Bueno, y lo demás es, básicamente es ir copiando las propiedades que teníamos en AttractionsCopy a este otro.

Lo empiezo a hacer rápido pero lo dejo como tarea para uds para no aburrirnos. En verdad nos habría convenido copiar el canvas entero y allí sustituir las variables por atributos.

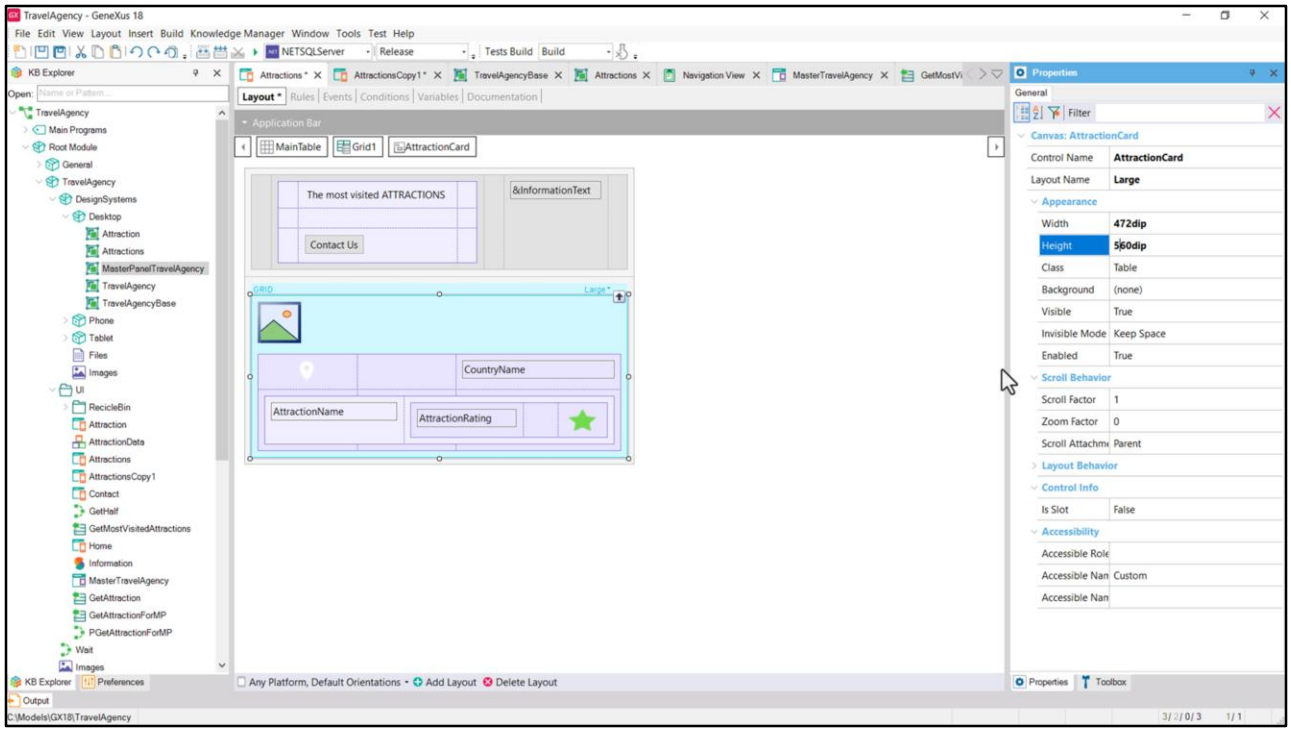


Bueno, aquí ya tenemos la card pequeña lista. Voy a cambiarle el nombre al canvas para que se llame AttractionCard, como en el otro grid.



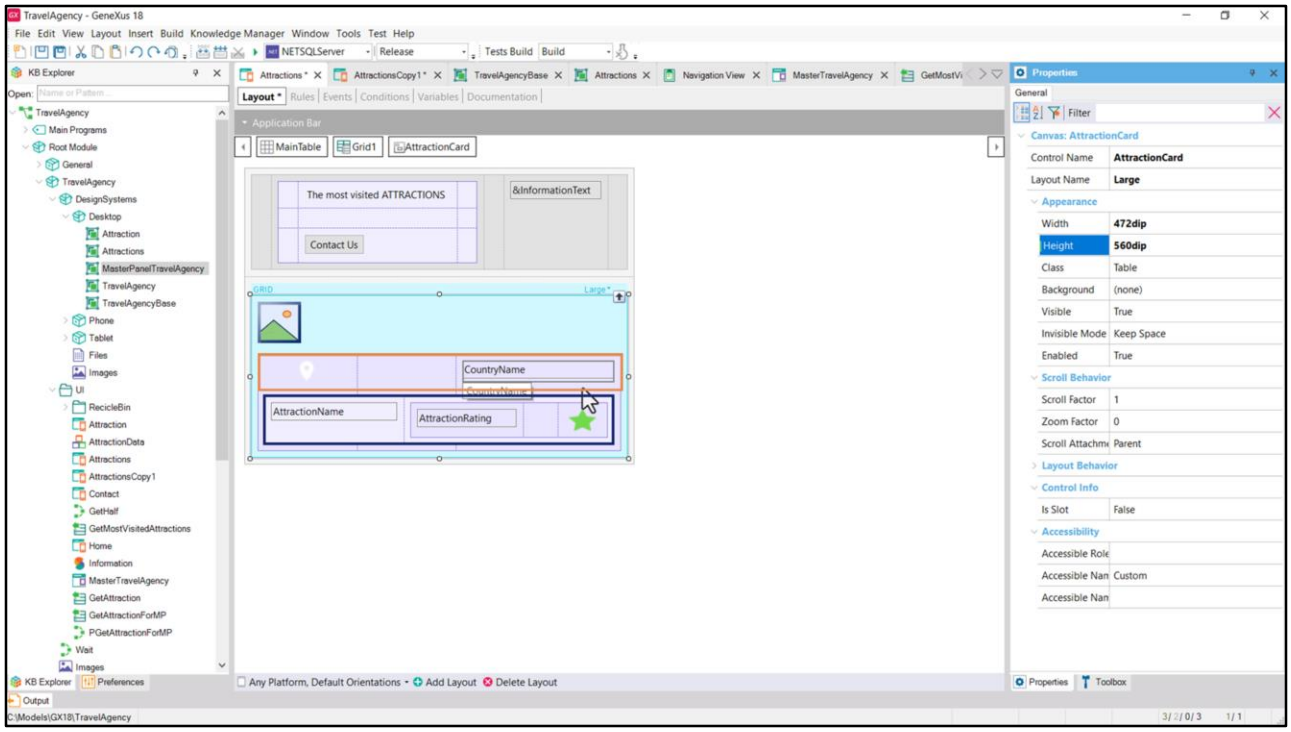
Y voy a cambiarle el nombre a este layout del ítem, para que sea Small. Será el default.

Y ahora voy a agregar otro ítem layout, al que llamaré Large…



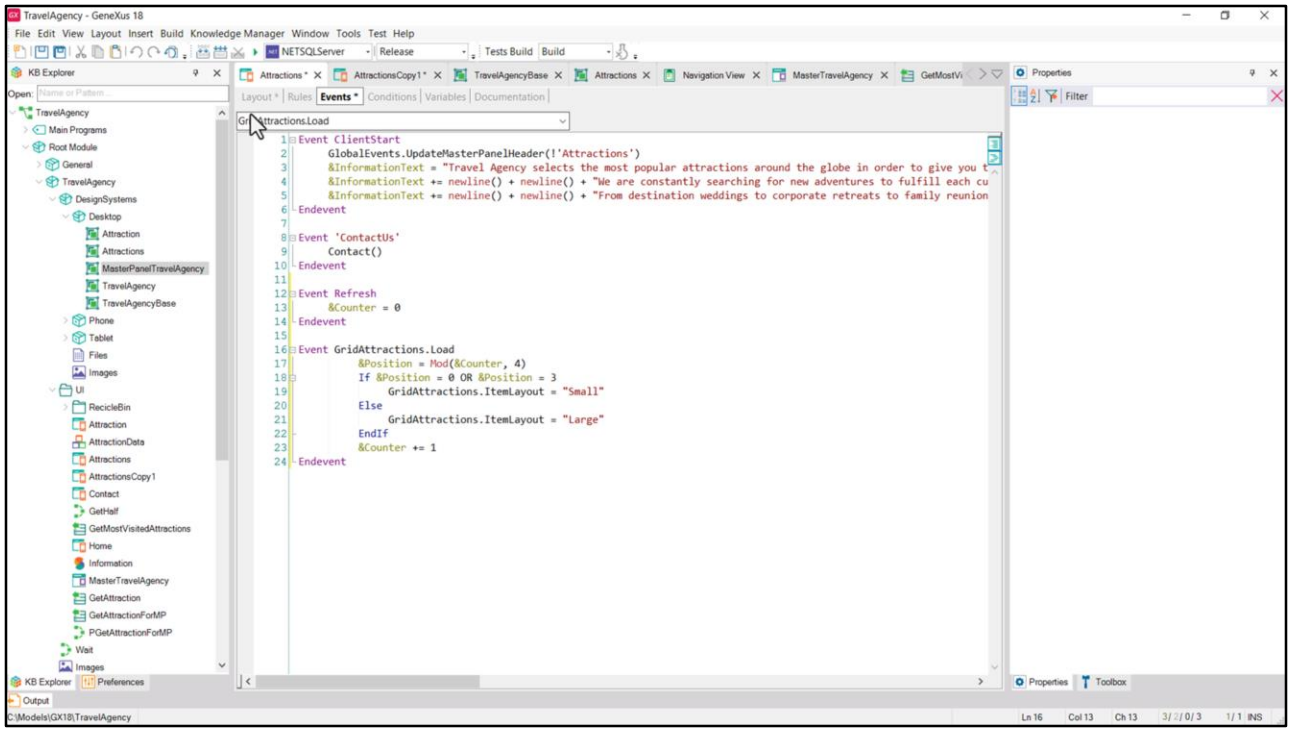
Se inicializa con el layout del único que había. Así que tendrá los mismos controles exactamente, con los mismos valores para las propiedades.

Como ya vimos, la única diferencia entre el layout Small y el Large será en el Height… aquí será de 560 dips.



Analicemos por qué solo haciendo esto ya quedarán 300 dips más entre la fila de esta tabla y la tabla de la fila 2.

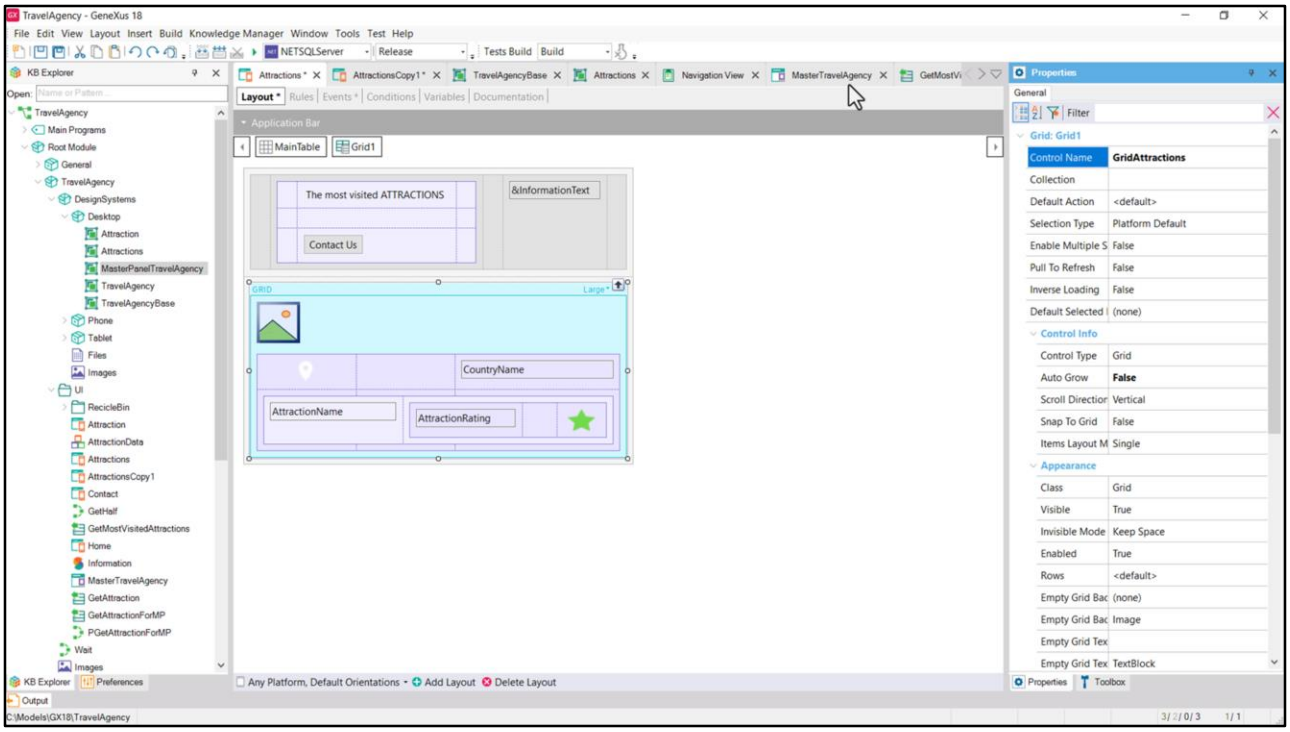
La tabla contenedora tiene fijados sus bordes a estas distancias: de arriba y de abajo. El alto queda relativo al alto del canvas, que es lo que varía entre ambos layouts. Pero además, esta tabla tiene un alto fijo, y está alineada verticalmente abajo, así que va a quedar siempre sobre el borde inferior. Y por tanto el aire que queda para completar el 100% del alto de la fila donde se encuentra la tabla es el que producirá esa diferencia de 300 dips entre un caso y el otro.



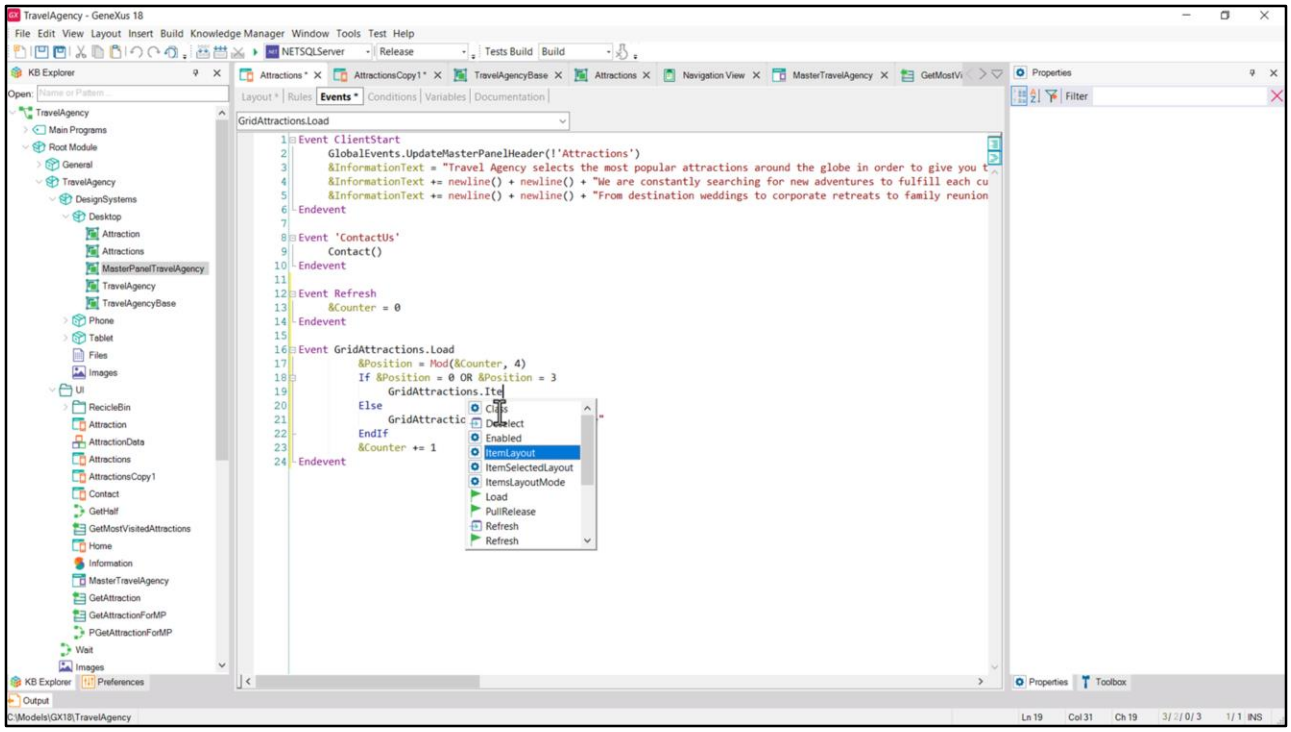
Nos queda copiar lo que hicimos aquí para cargar cada ítem en el grid con el layout que le corresponde.

Copio los dos eventos…

Los pego… quito esto que no va aquí. Dejo en el Refresh únicamente el poner en 0 el contador…

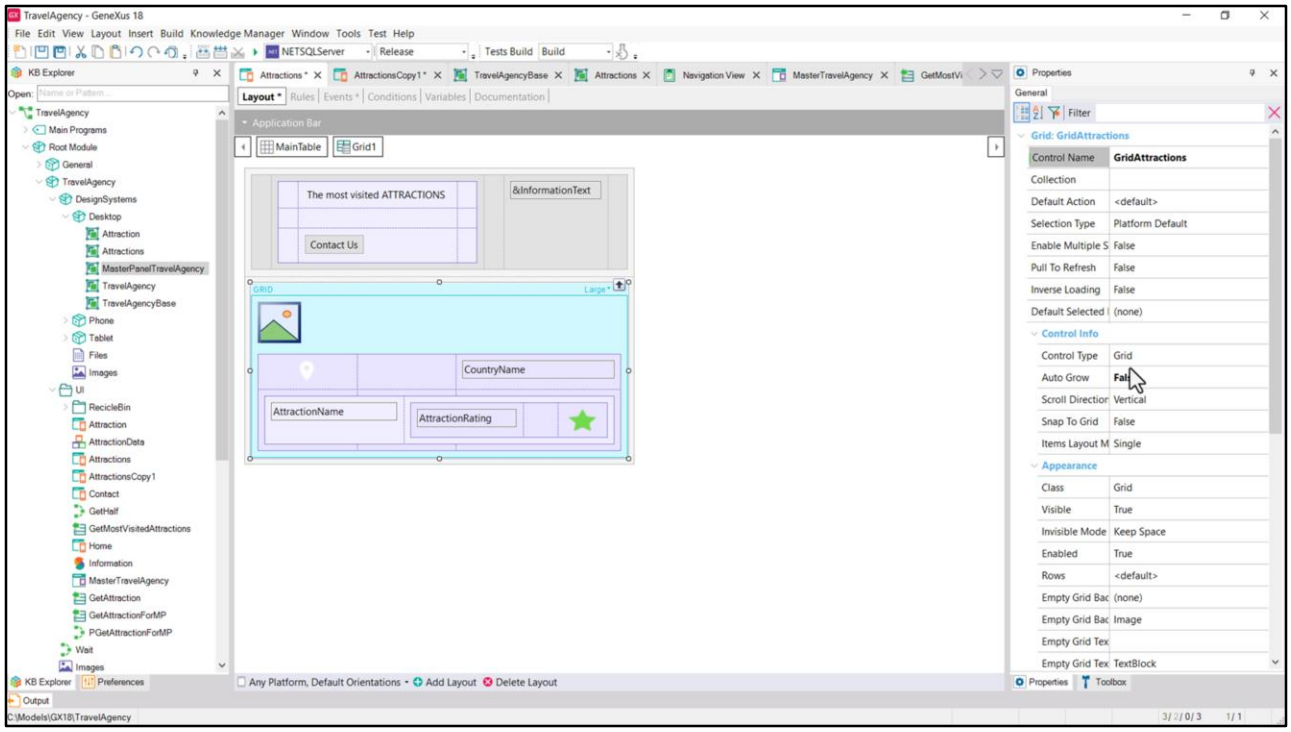


Al Grid le cambio el nombre por el que venía usando…GridAttractions.

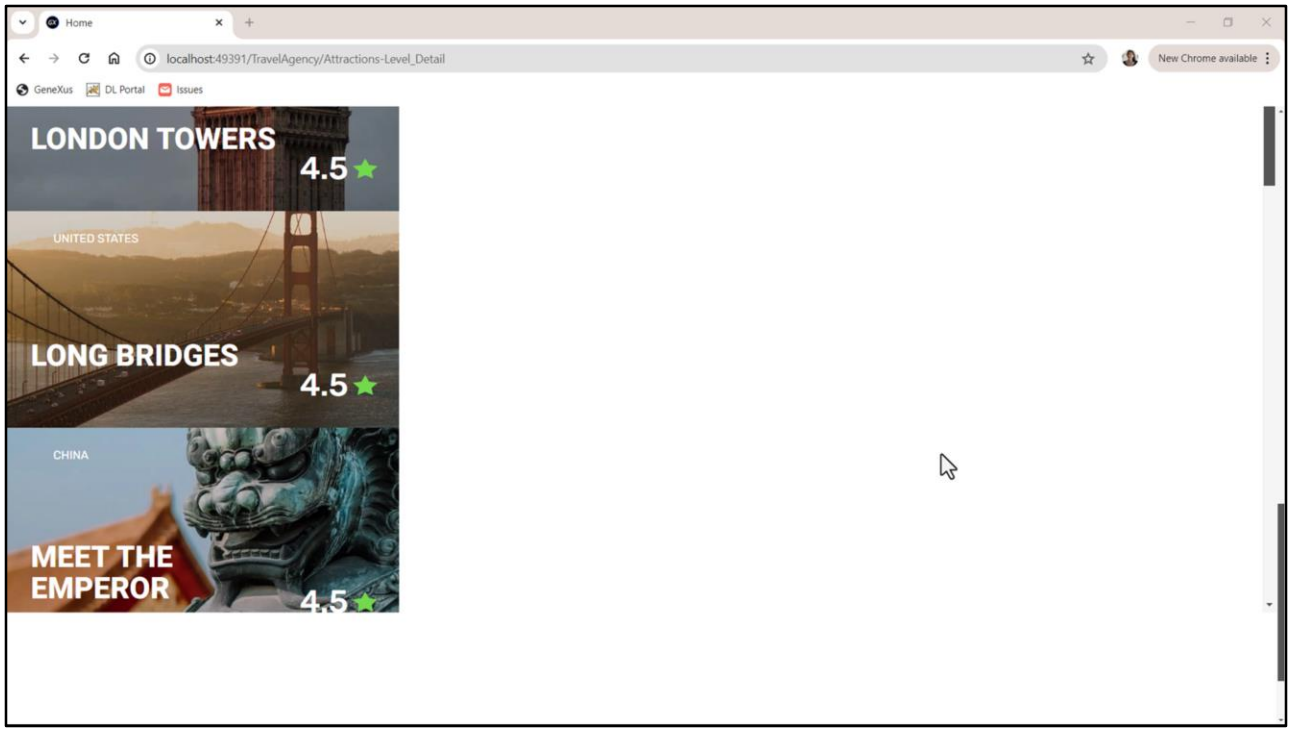


Defino en este panel las variables.

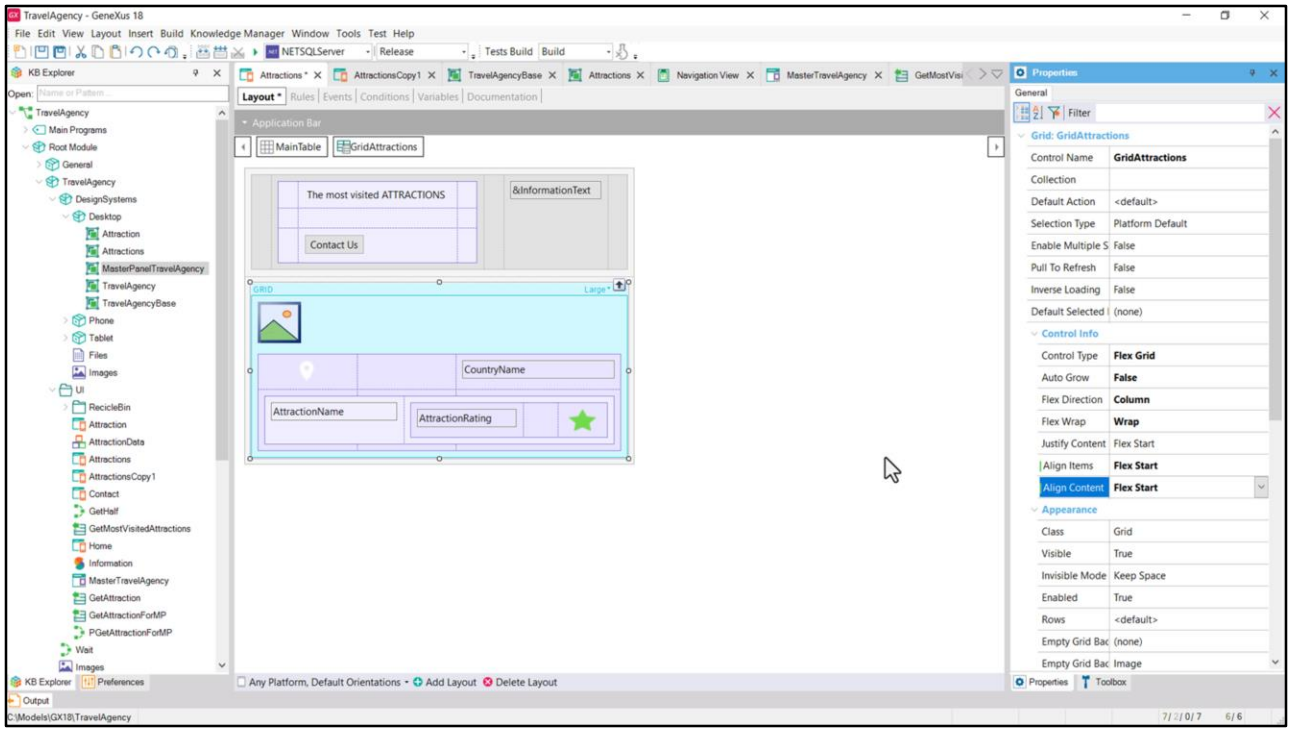
Y esto queda así perfecto, pero para mostrarles… vean que si al nombre del grid lo sigo por punto… encuentro elegible la propiedad… Y allí completo. Coloquemos el signo de exclamación para que no es traduzca este texto ni el otro.



Antes de ejecutar, veamos que el grid lo tenemos como grid estándar. Si ejecuto…

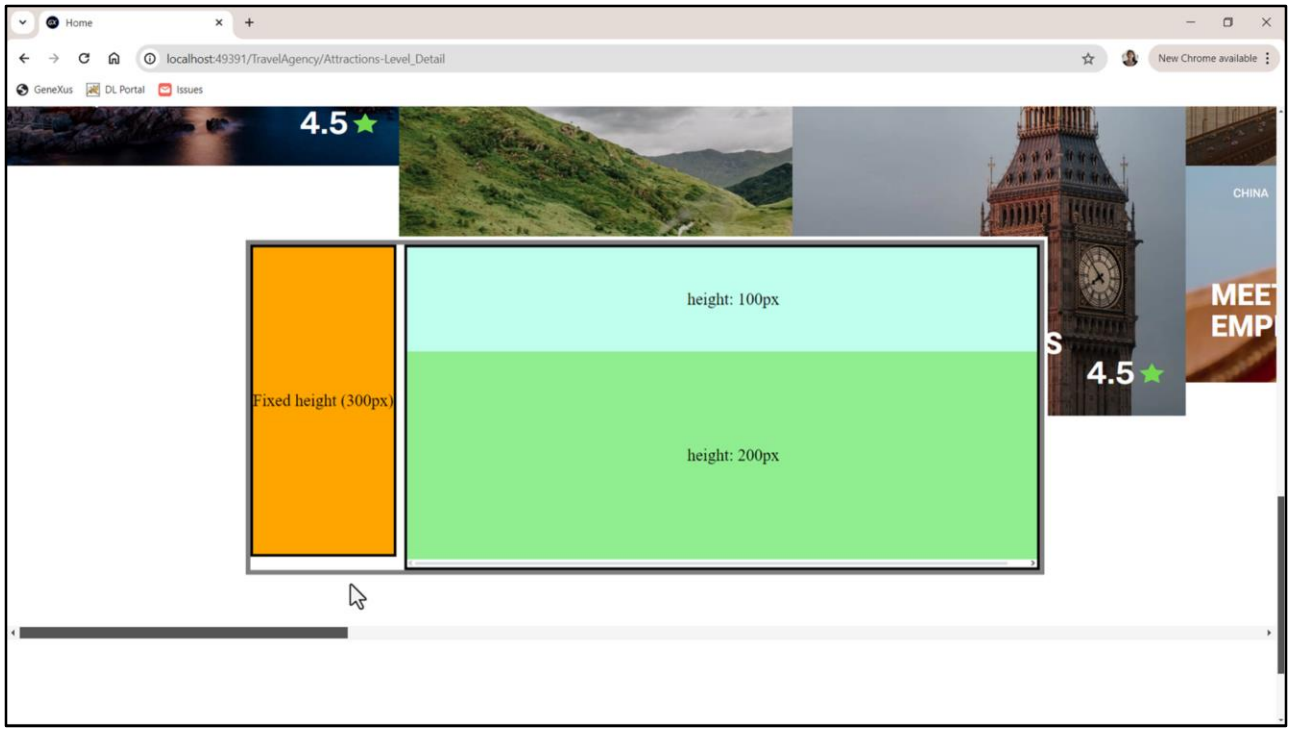


…esto es lo que veo.



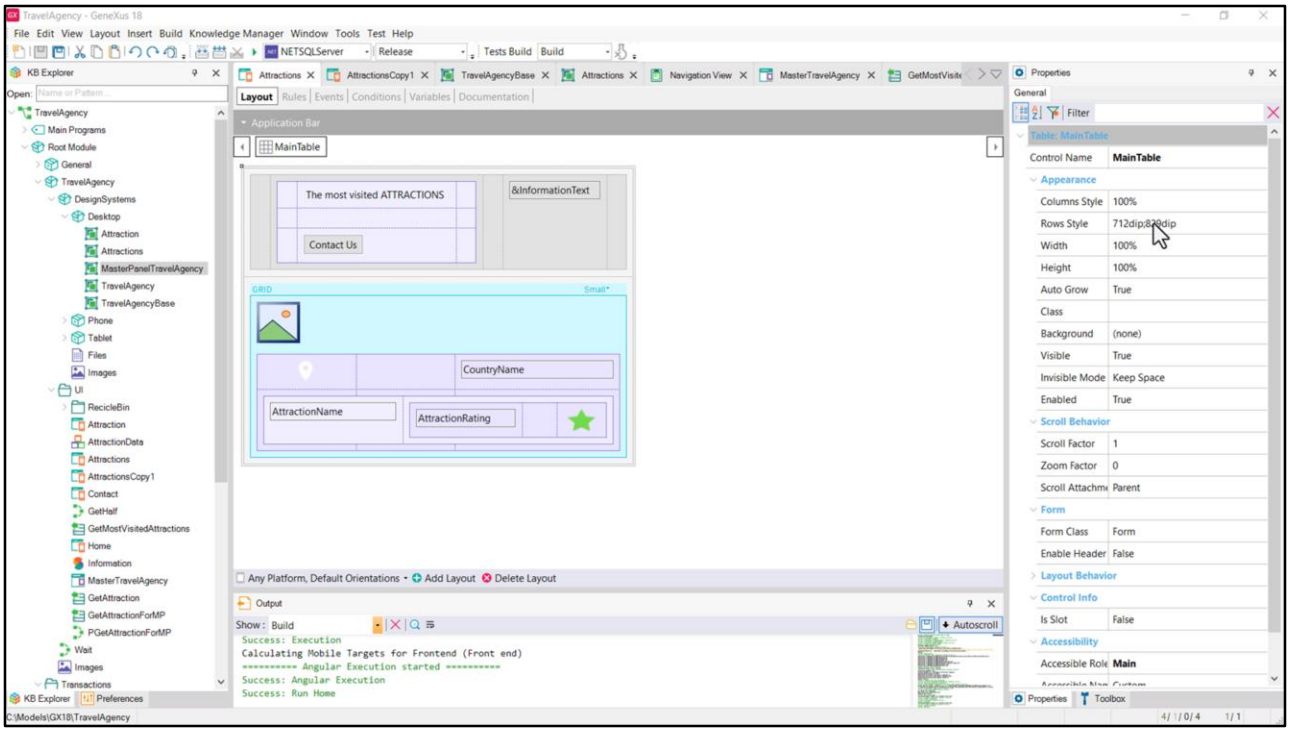
Voy a modificar el grid para que sea Flex. Le coloco dirección Column, Wrap, la justificación del contenido de acuerdo al inicio, es decir, al borde de arriba; la alineación de los ítems respecto al otro eje, al horizontal, también que sea de acuerdo al inicio, o sea, al borde de la izquierda… Y este también.

Ejecutamos…

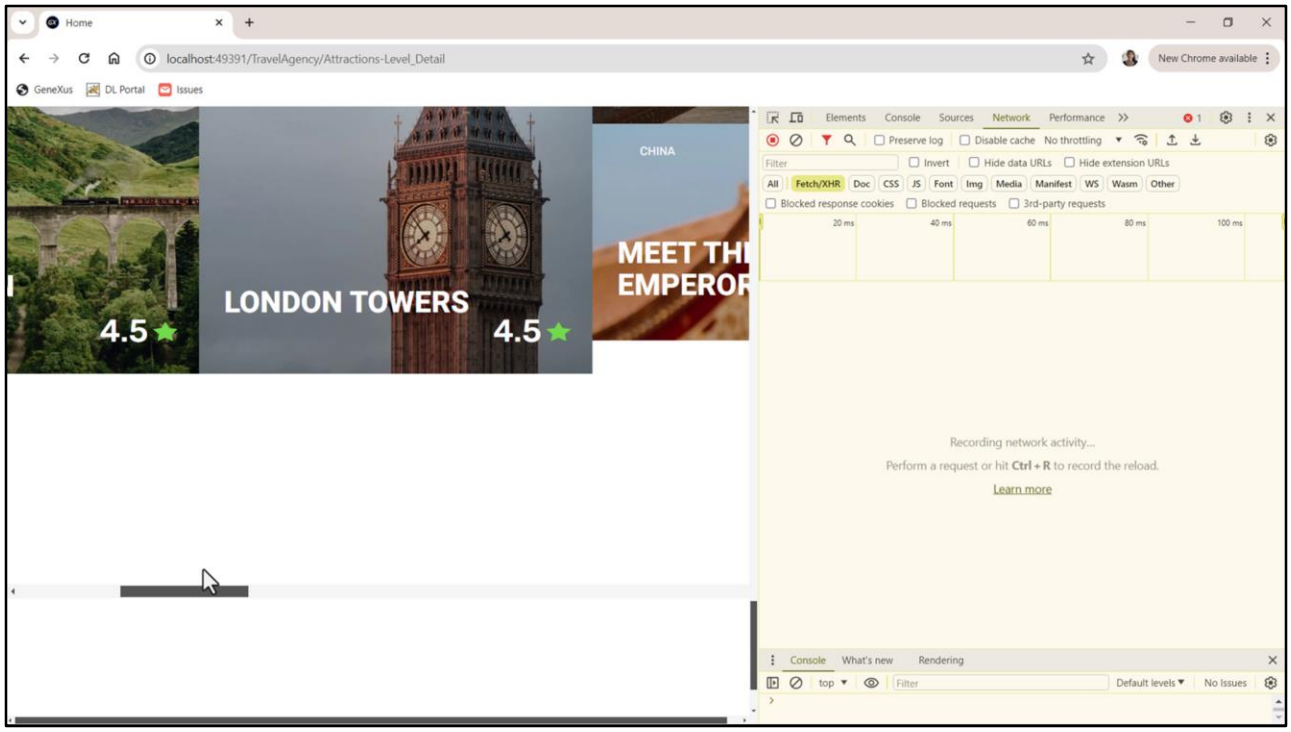


Vemos que en cada columna está quedando una única card en lugar de 2. ¿Por qué?
Tiene que ver con esta barra de scroll, que ocupa espacio aparte en Desktop y hay que tener en cuenta ese espacio.

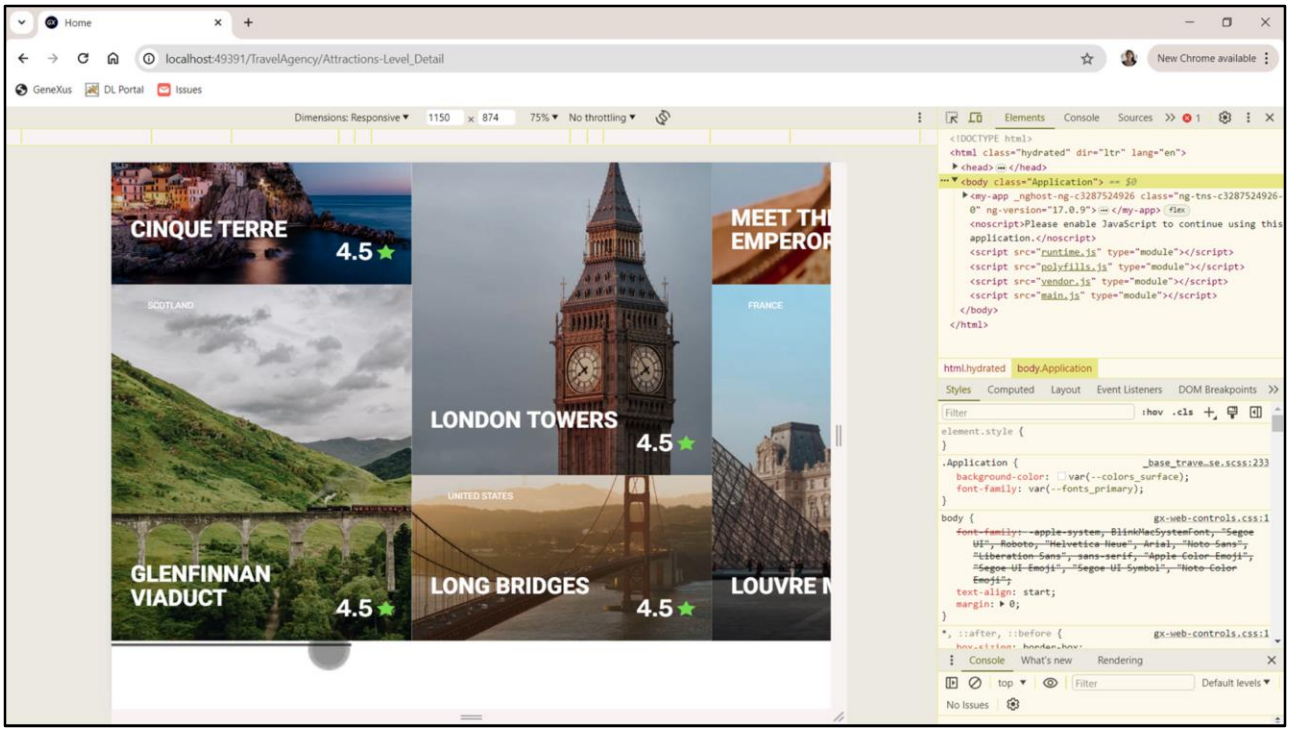
Así, por ejemplo, si en el grid tenemos un ítem de 100 píxeles de alto y otro de 200 píxeles de alto, para que ambos entren el alto del grid no puede ser de 300, tiene que ser mayor, porque tiene que contemplar el alto de la barra de scroll, que queda por fuera.



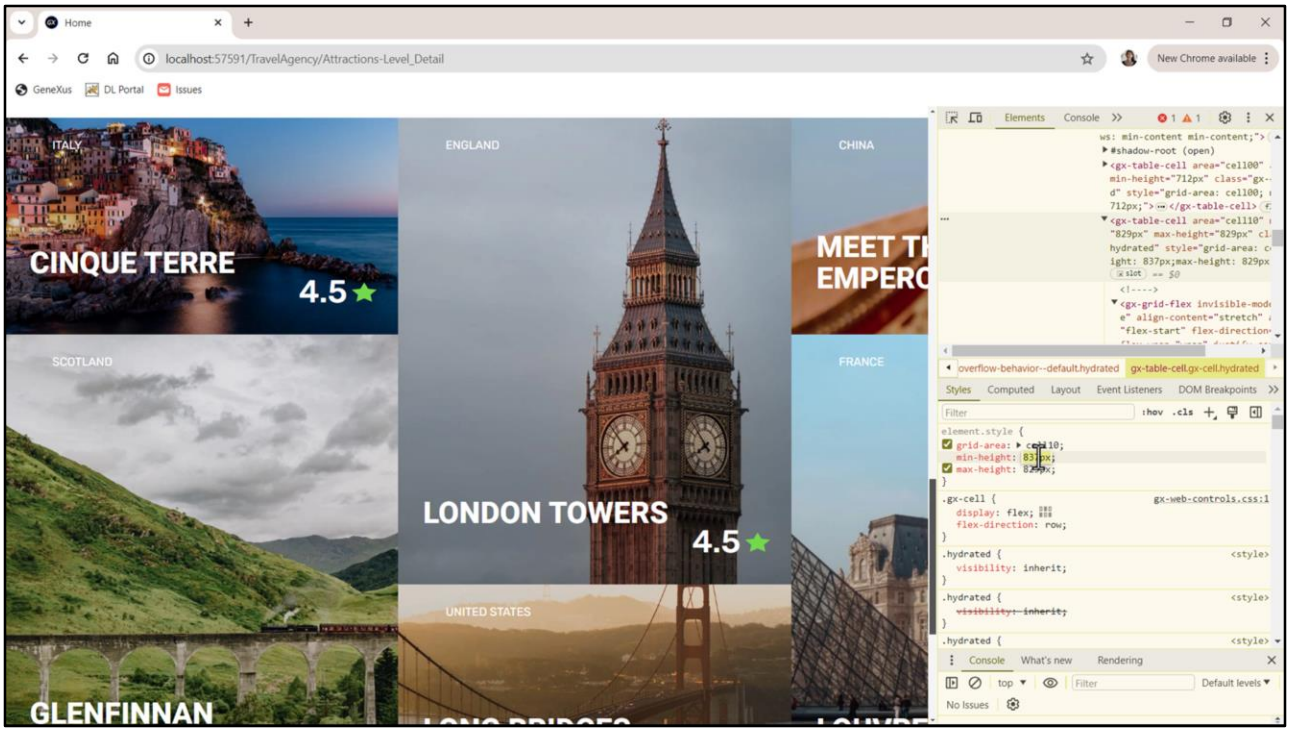
En nuestro caso el alto que le dimos a la fila del grid es de 829 dips, y las 2 cards sumadas dan 820 (260 más 560). Evidentemente no le está dando con esos 9 dips sobrantes para colocar la barra de scroll y por eso no le están entrando las dos cards por columna.



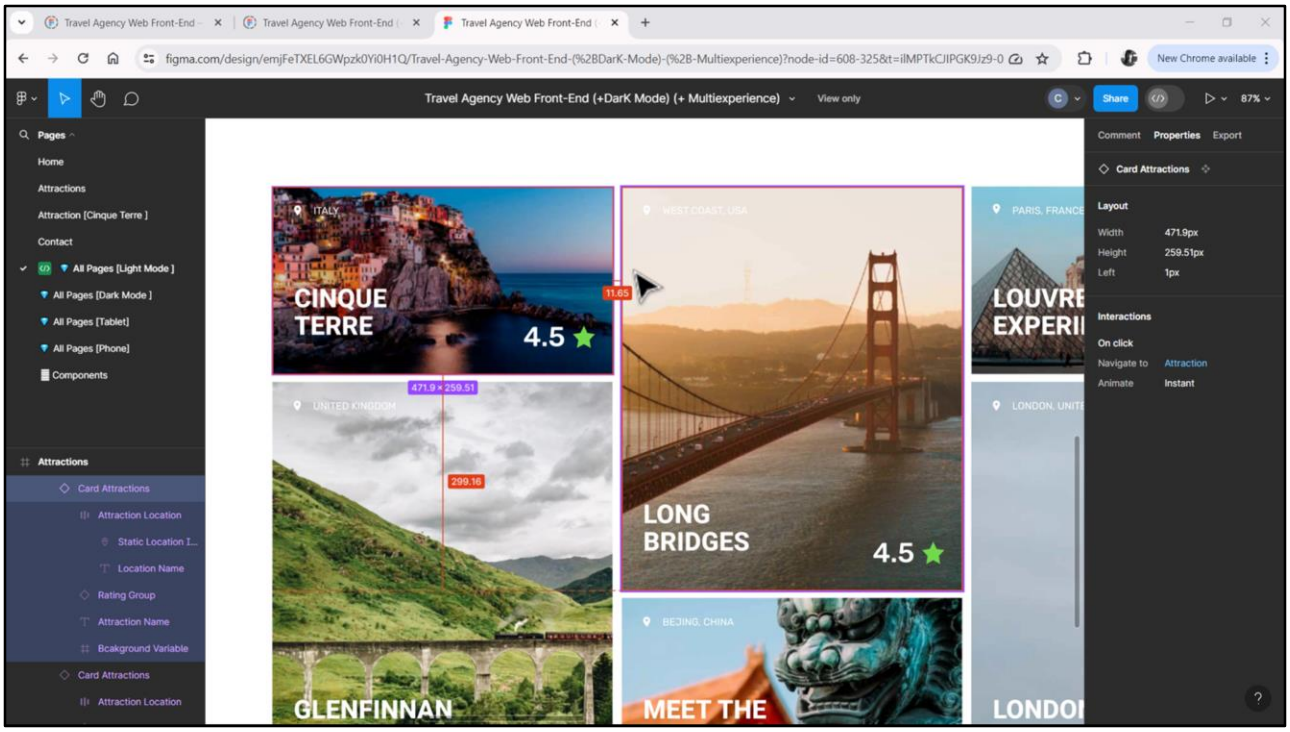
En este F12 que lancé sólo tengo activada la depuración Web para escritorio y no la Web para dispositivo móvil.



Vean que si prendo la depuración mobile ahora sí me están entrando las dos cards, porque en el navegador para mobile la barra de scroll se coloca encima.

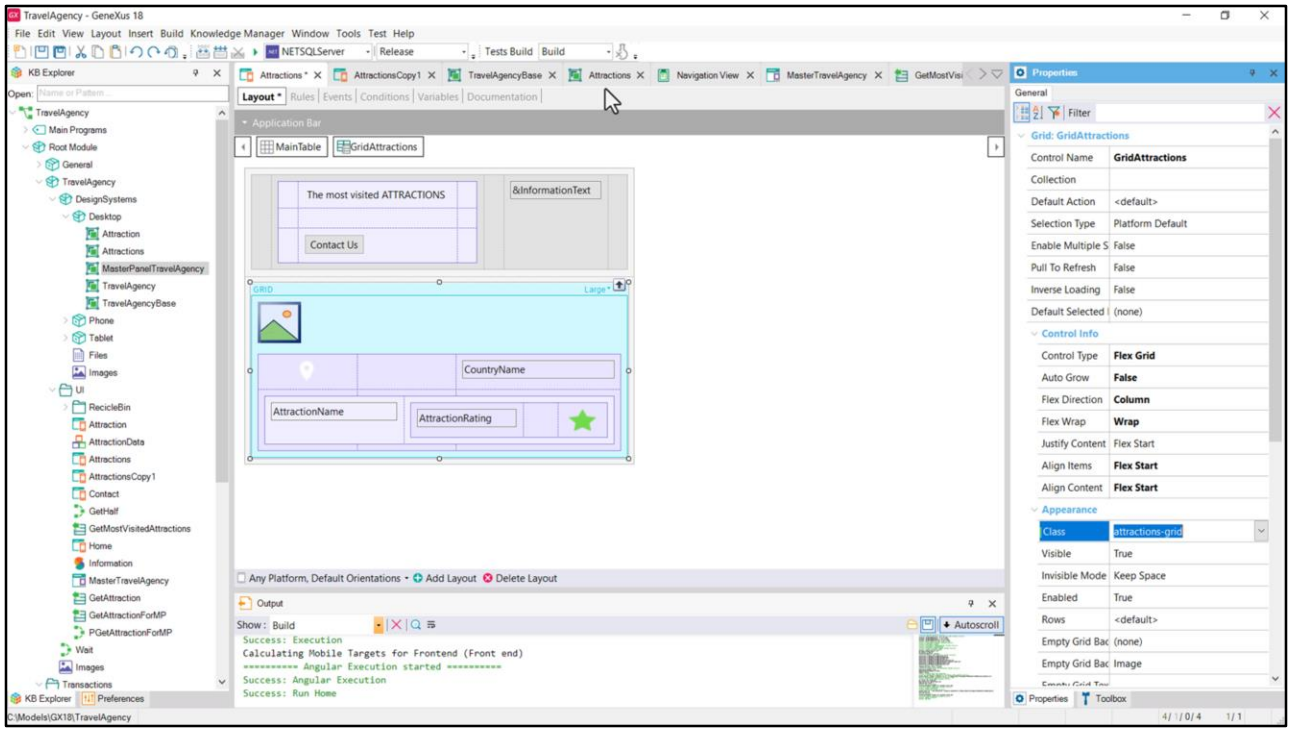


Sin embargo las pantallas que estamos diseñando serán únicamente ejecutadas en Desktop, por lo que necesitamos resolver ese escenario. Claramente tenemos que aumentar el alto de la fila. Vemos que cuando llegamos a este valor, 837, aparecen las dos cards por columna. Y por supuesto si seguimos aumentando vemos cómo se va separando la barra de scroll, así que necesitábamos dedicarle, destinarle,17 dips, porque las dos cards sumaban 820… necesitamos 17 dips para esa barra de scroll.

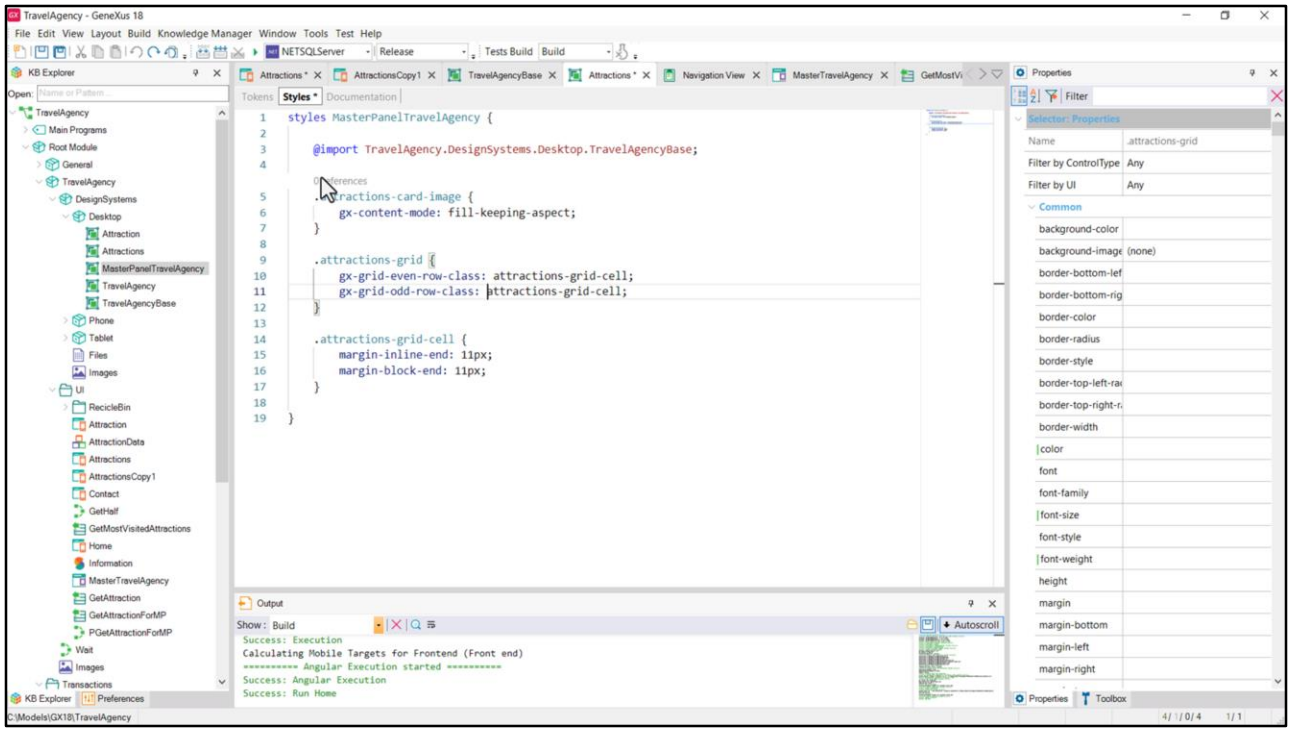


Pero además, nos falta dejar el espacio vacío entre cards…

Vemos que en el diseño de Figma las cards tienen un margen a la derecha de un poco más de 11 píxeles y de abajo de casi 11. Así que podríamos a todas las cards en el grid asignarles un margen de 11 final, en las dos direcciones.



Podemos asignar una clase al grid…

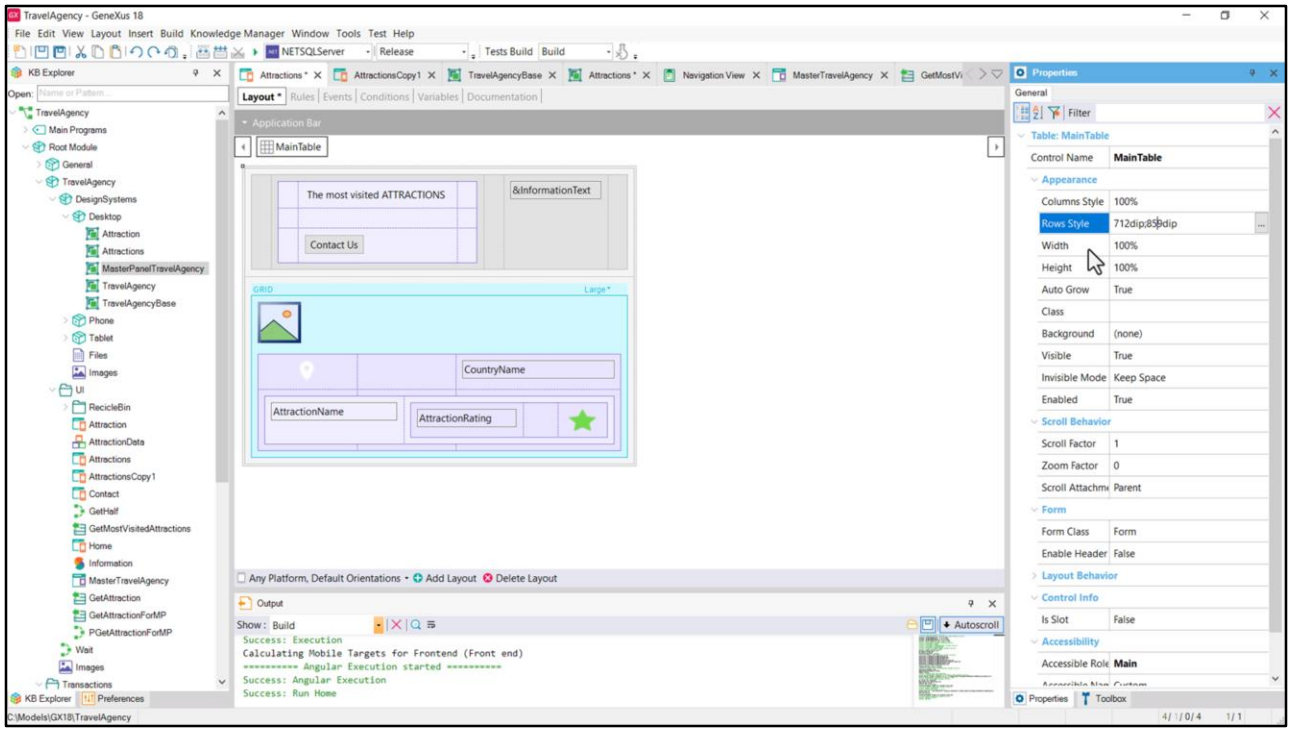


…en la que definamos a través de las gx-properties: gx-grid-**even**-row-class y gx-grid-**odd**-row-class clases que den estilo a la celda del grid cuando se trata de ítem par o ítem impar.

Este row aquí en el nombre, fila, hay que entenderlo como posición del ítem. Es decir, más allá de cómo se presenten los ítems renderizados en la pantalla, algo que depende entre otras cosas del tipo de grid, todo grid, conceptualmente es una lista ordenada de ítems. Entonces está el primero, el segundo, el tercero y así sucesivamente. Después hay que ver cómo ese orden se renderiza, como decía. La propiedad gx-grid-**even**-row-class aplicará para todos los ítems que ocupan posición **par** en esa lista, y la **odd** para los que ocupen posición **impar**. Nos permite esa discriminación, digamos, esa alternancia entre pares e impares.

En nuestro caso queremos que a todos los ítems les aplique la misma clase, a la que llamaré attractions-grid-cell y le definiré estas dos propiedades: margin-inline-end de 11 píxeles y margin-block-end, es decir, en la otra dirección, también de 11 píxeles.

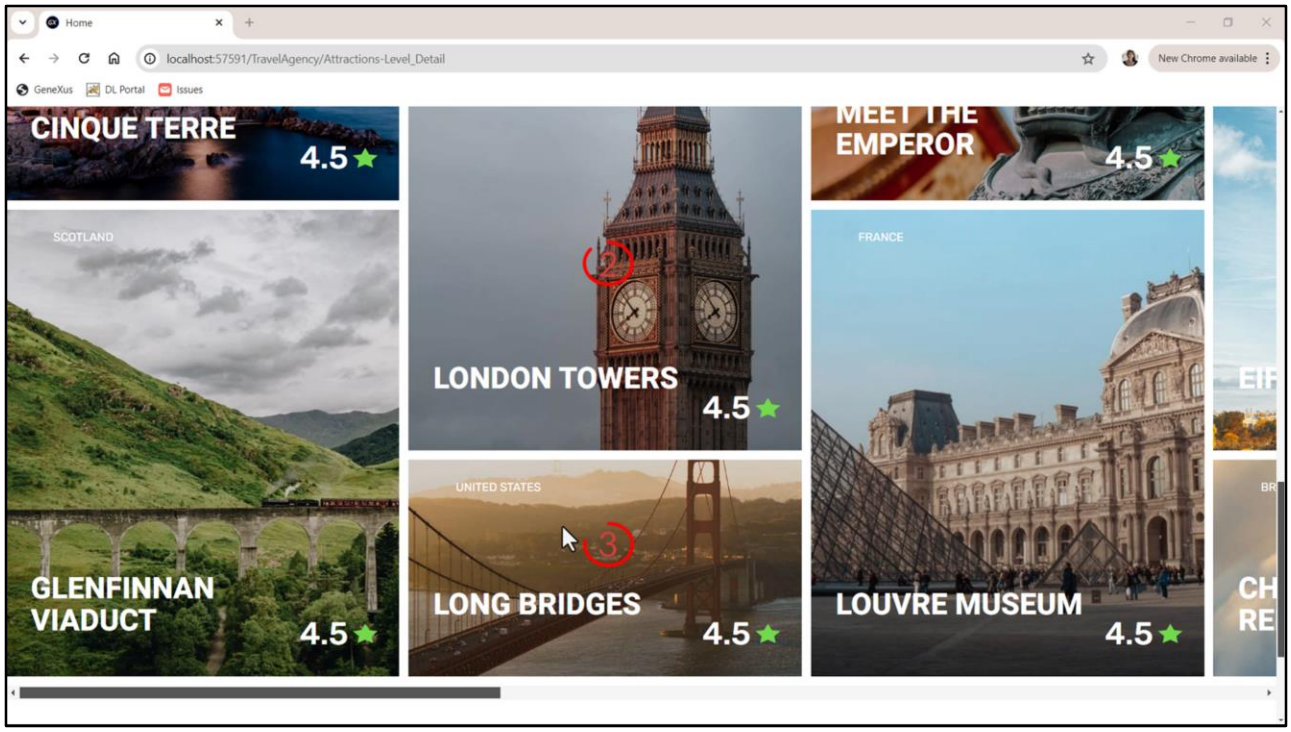
Y luego simplemente establezco que la clase para lo ítems pares y para los impares es la misma y es esta.



Lo que tengo que hacer ahora es calcular el alto que debo darle a la fila del grid. Será de 11 x 2, debido al margen block end de las dos cards más los 820 de alto de las dos cards sumadas, más los 17 de la barra de scroll.

Así que… voy a Attractions…y en la segunda fila… coloco este valor.

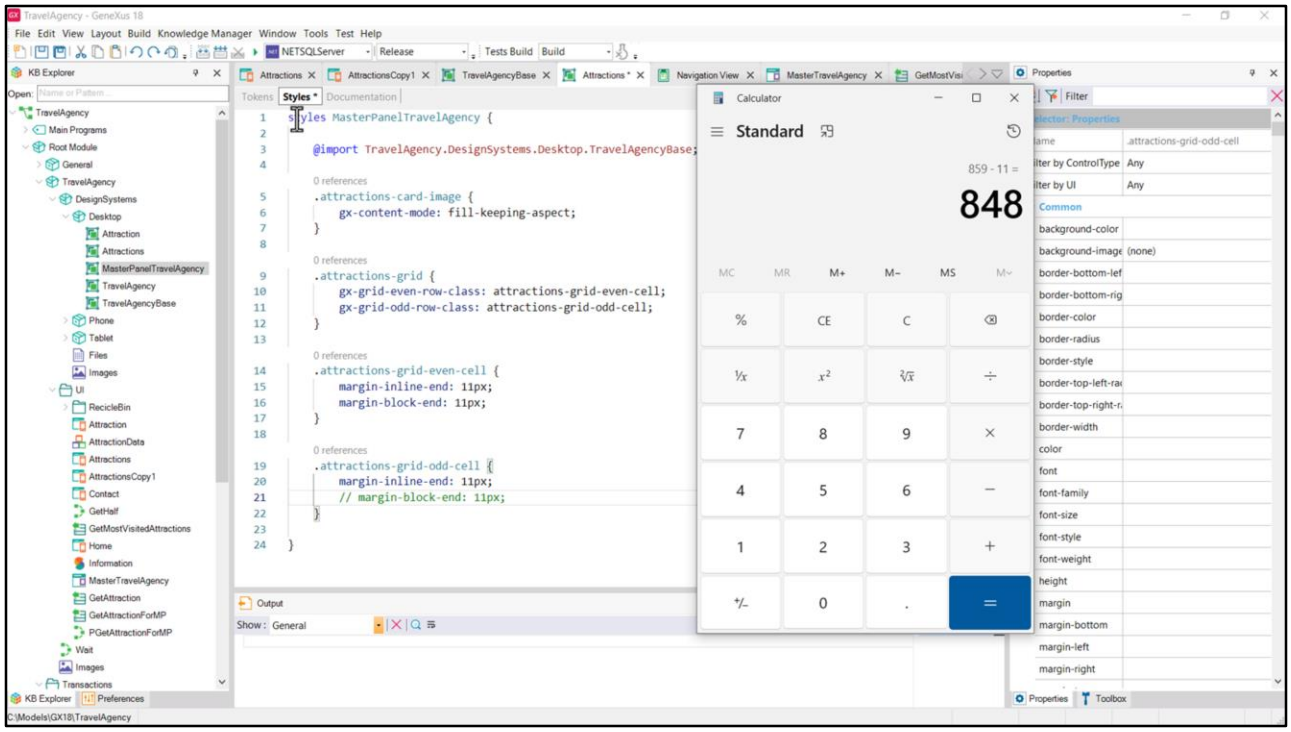
Ejecutemos.



Perfecto.

¿Y si no quisiéramos el margen de abajo para las cards de abajo?

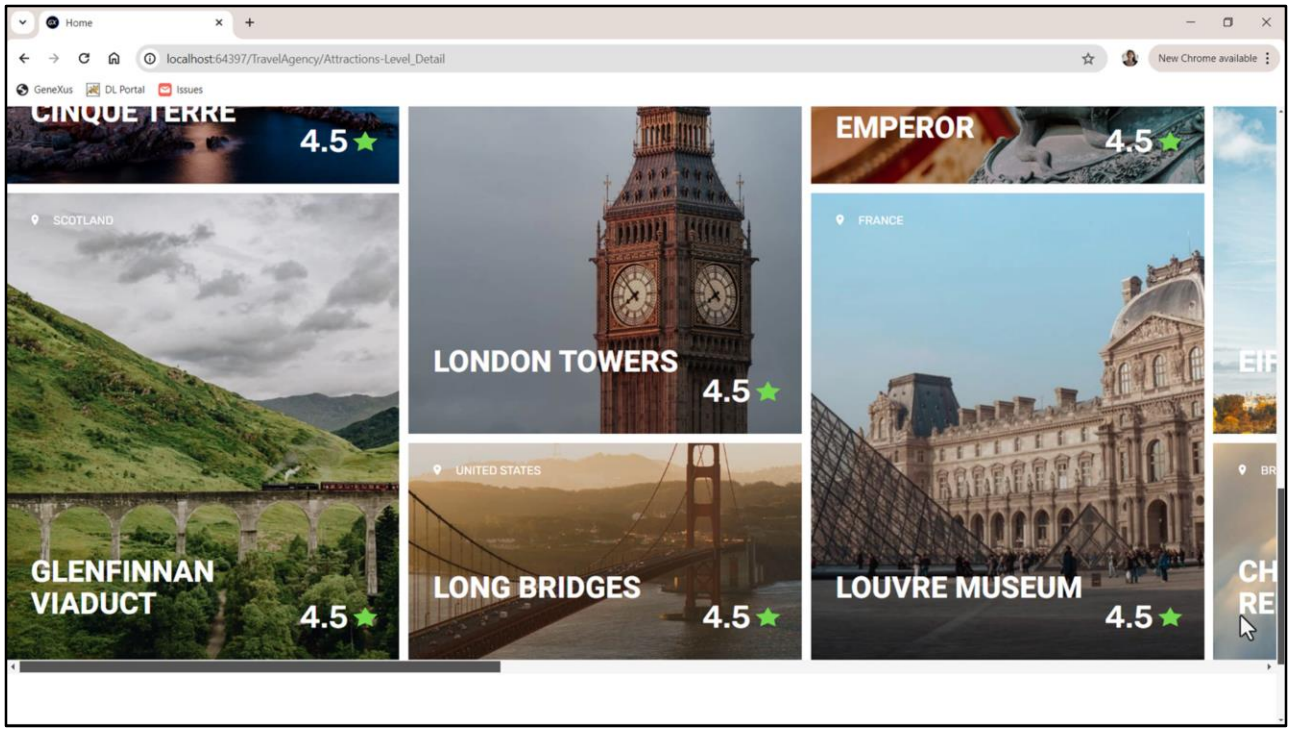
Las cards de abajo serán las impares, si se empieza a contar en 0.
0, 1… 2, 3… 4, 5… y así.



Entonces podemos hacer esto…

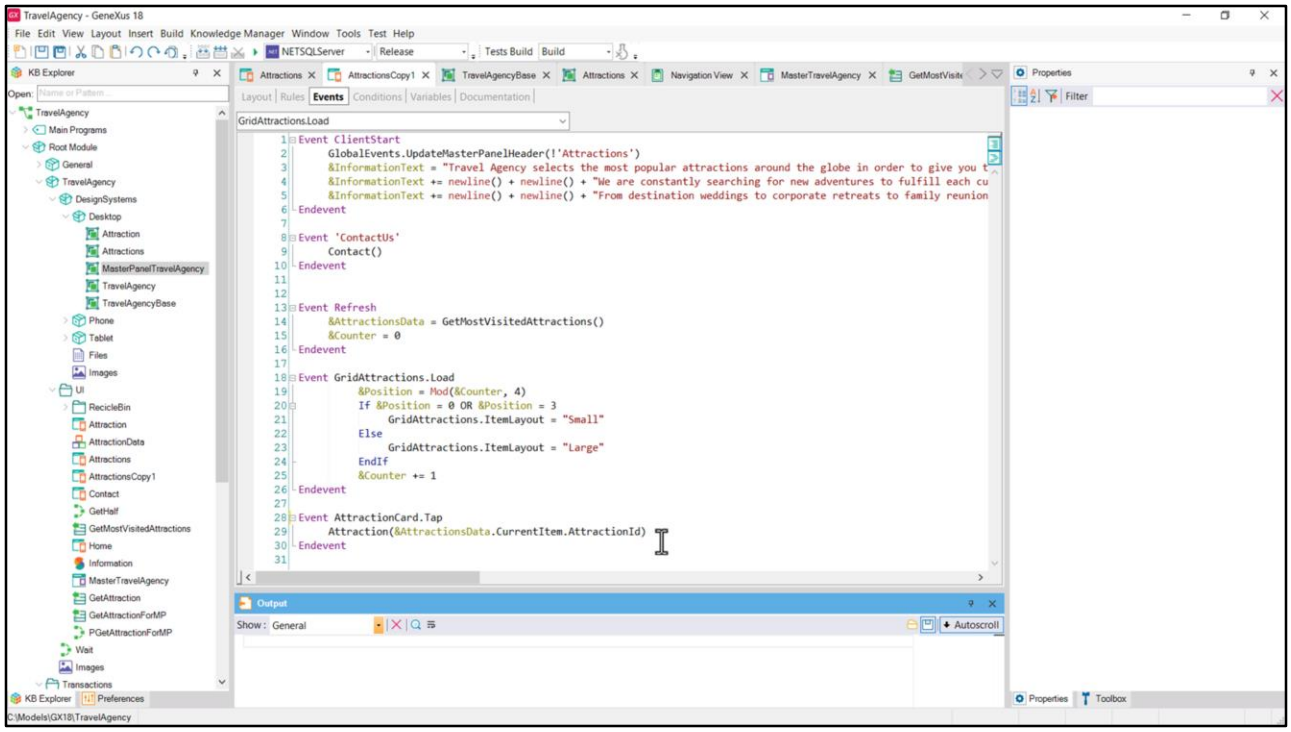
Diferenciamos dos clases: una para las pares y otra para las impares, y a la de las impares le quitamos el margen de abajo.

Y ahora le restamos al alto de la fila estos 11



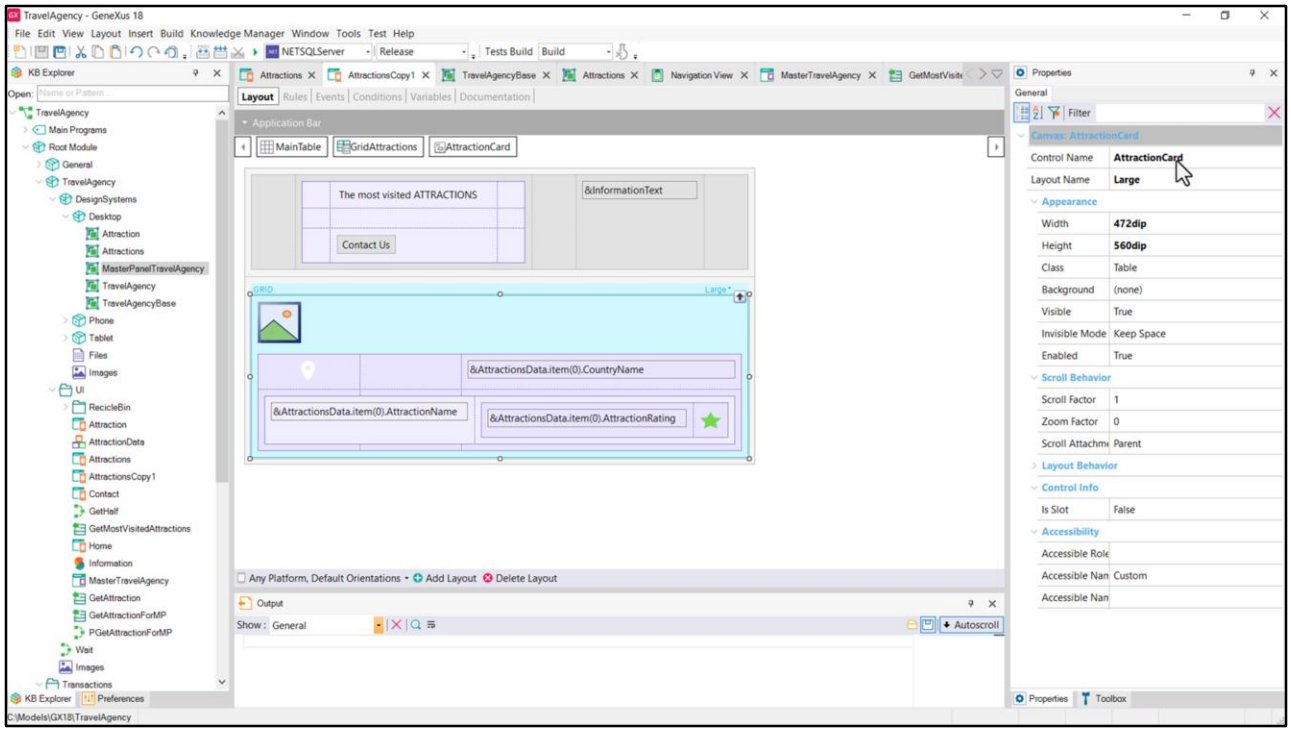
y ejecutamos…

Bien, vemos que ya no quedó el margen de abajo.

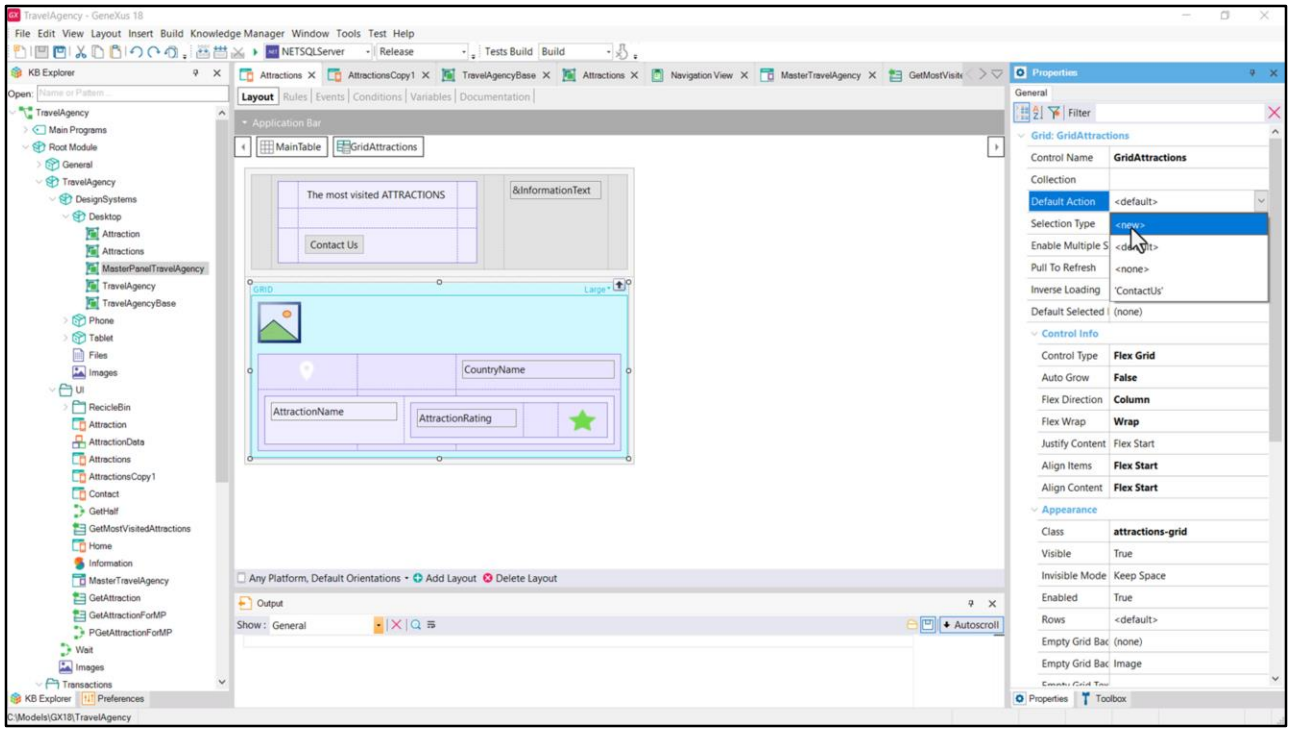


En mi solución con el grid asociado al SDT hagamos lo mismo…

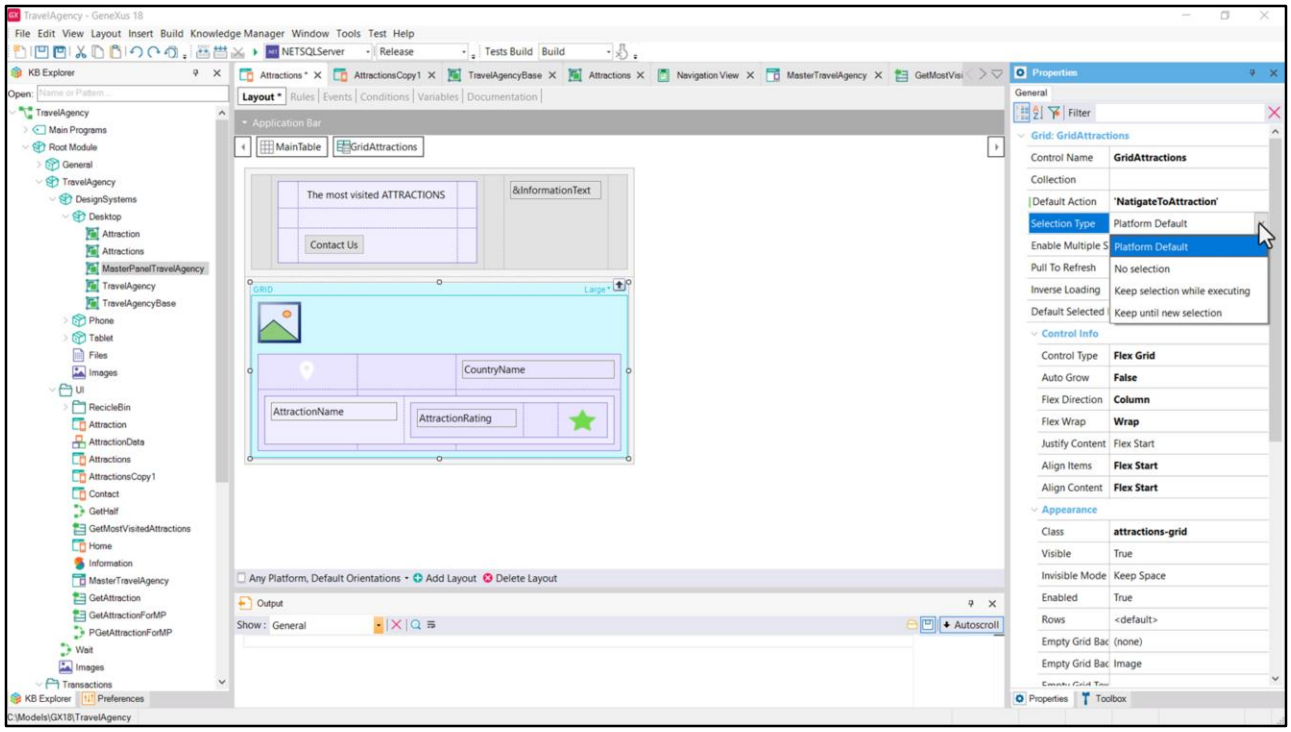
Ahora atendamos a este evento Tap que habíamos asociado al Canvas de cada ítem para poder invocar al panel Attraction pasándole el id de la atracción.



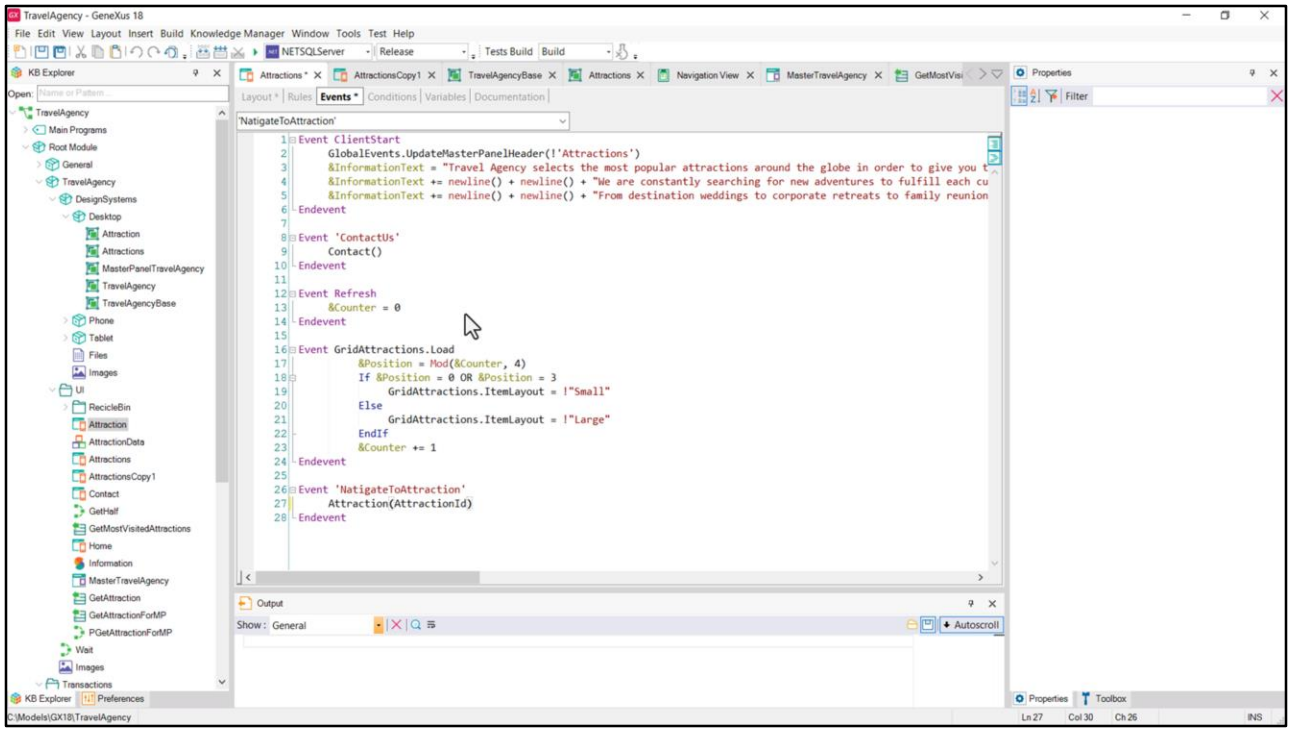
Esta no es la mejor solución, porque tenemos el tap programado para el control de nombre AttractionCard, que en verdad corresponde a dos canvas distintos: al del layout Small y al del layout Large (tuve de hecho que explícitamente darle el mismo nombre, tener ese cuidado).



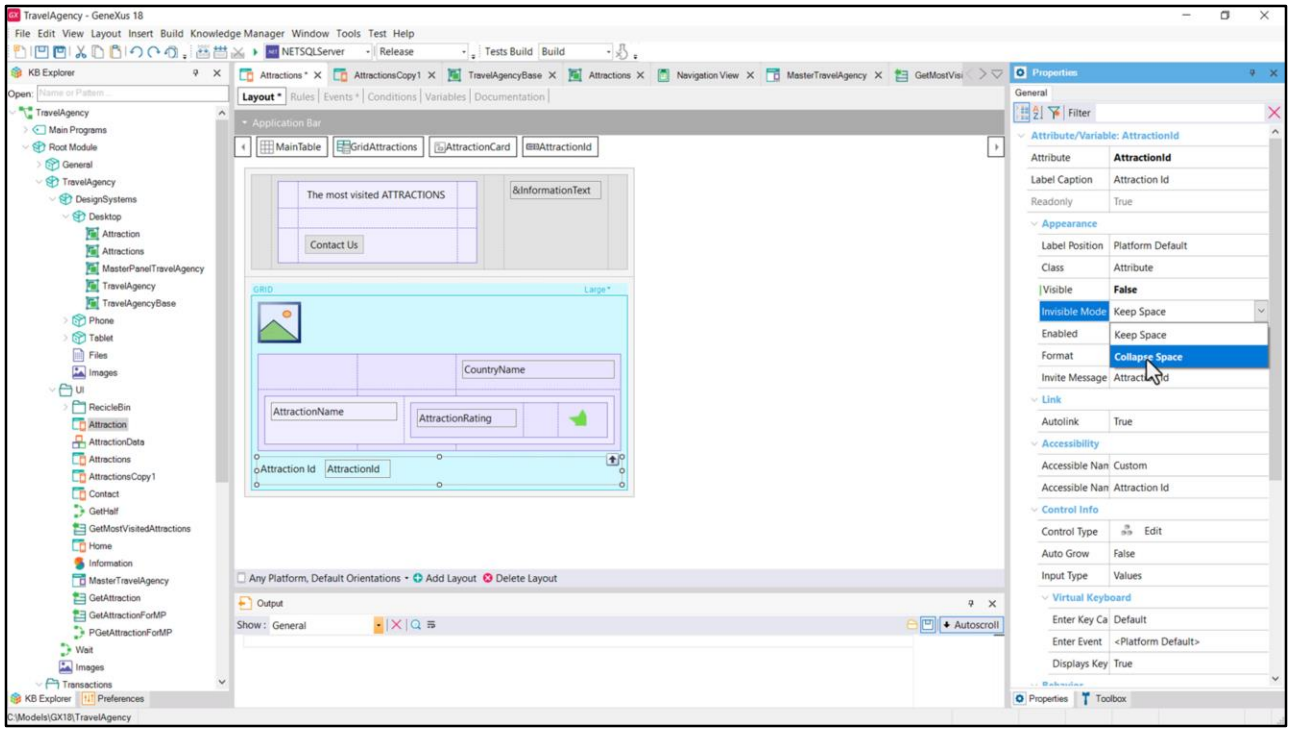
En verdad el grid tiene un evento predefinido que corresponde a hacer tap o click sobre cualquiera de sus ítems. Corresponde a esta propiedad, Default Action. Vamos a usar esta otra solución en nuestro panel Attraction.
Lo que hago es crear un nuevo evento de usuario, que corresponderá a la acción default sobre los ítems del layout. Le doy un nombre….



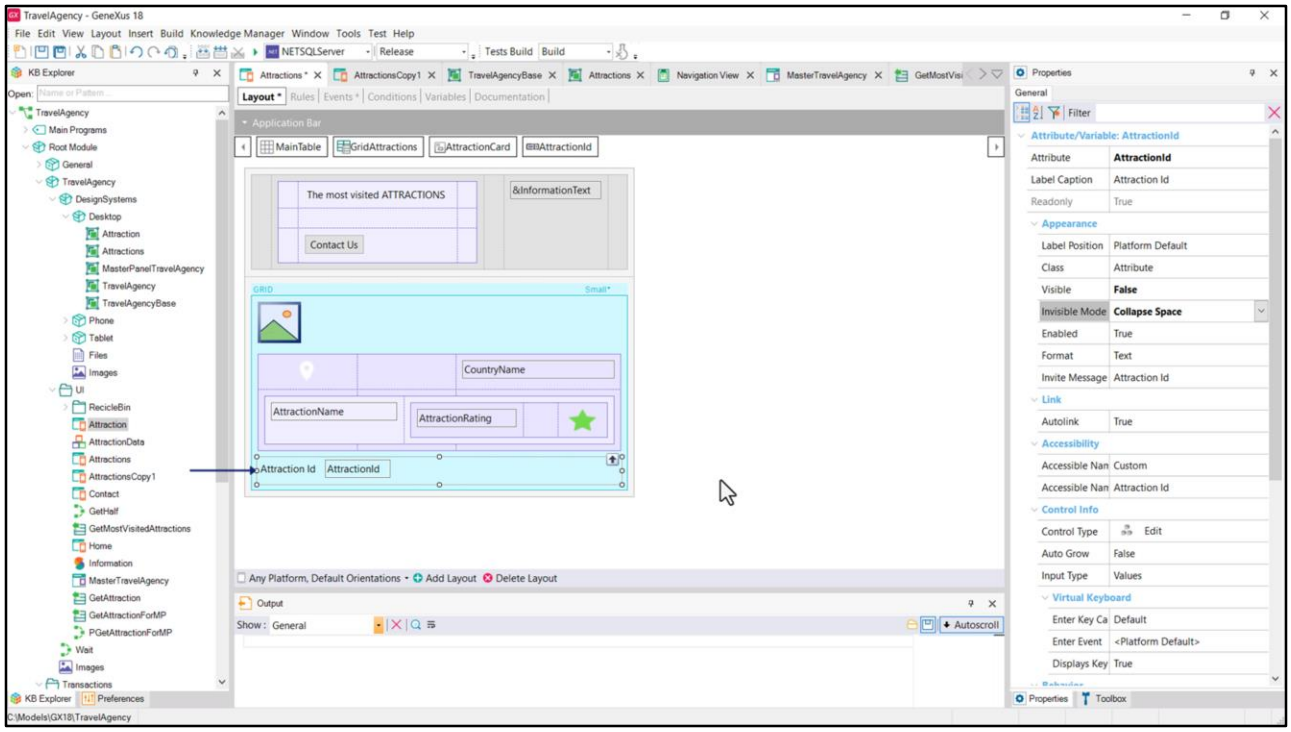
Ya aprovechemos a ver que hay un conjunto de propiedades que permiten definir cierto comportamiento del grid… por ejemplo si se permite seleccionar ítems del grid, y qué tipo de selección se permite, etcétera.



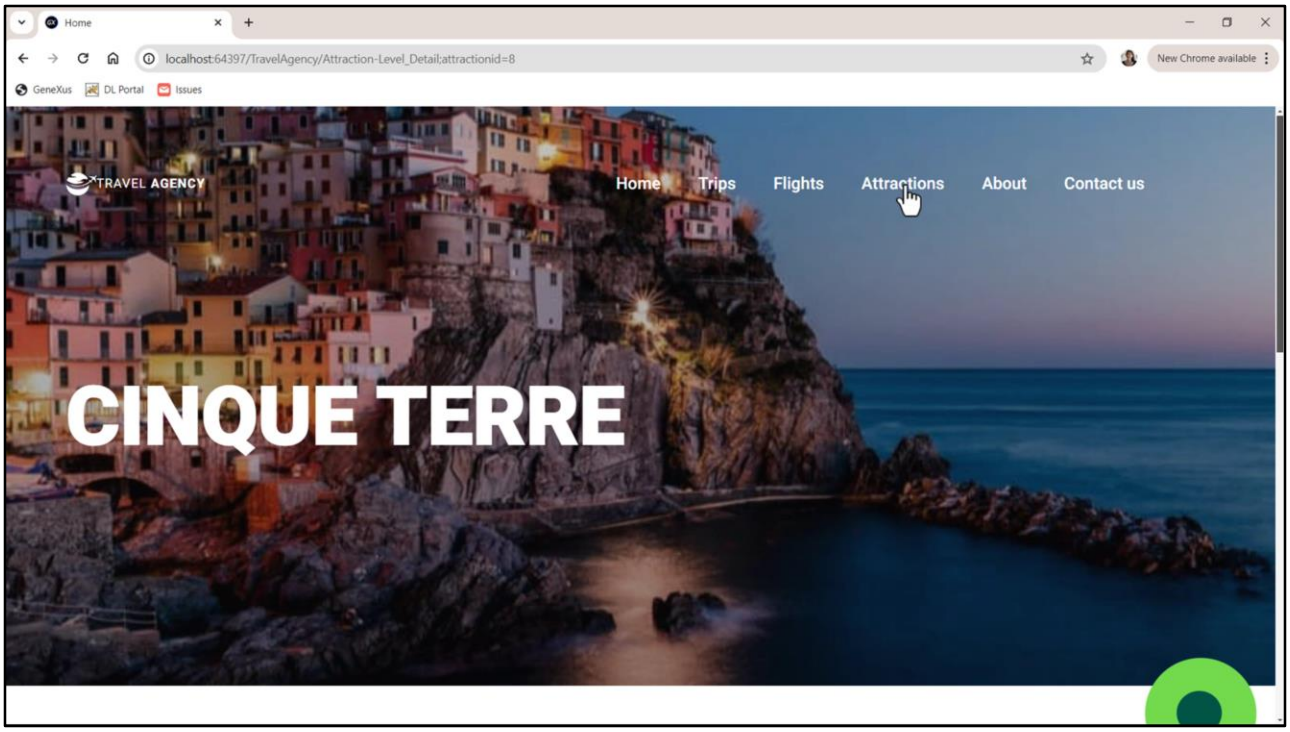
Aquí está el evento y lo que tendremos que hacer es invocar al panel Attraction, pasándole el identificador de atracción, que en este caso estará en el atributo AttractionId. ¿Pero tenemos cargado en el ítem del grid el atributo como para poder pasárselo a este otro panel una vez cargado el grid?



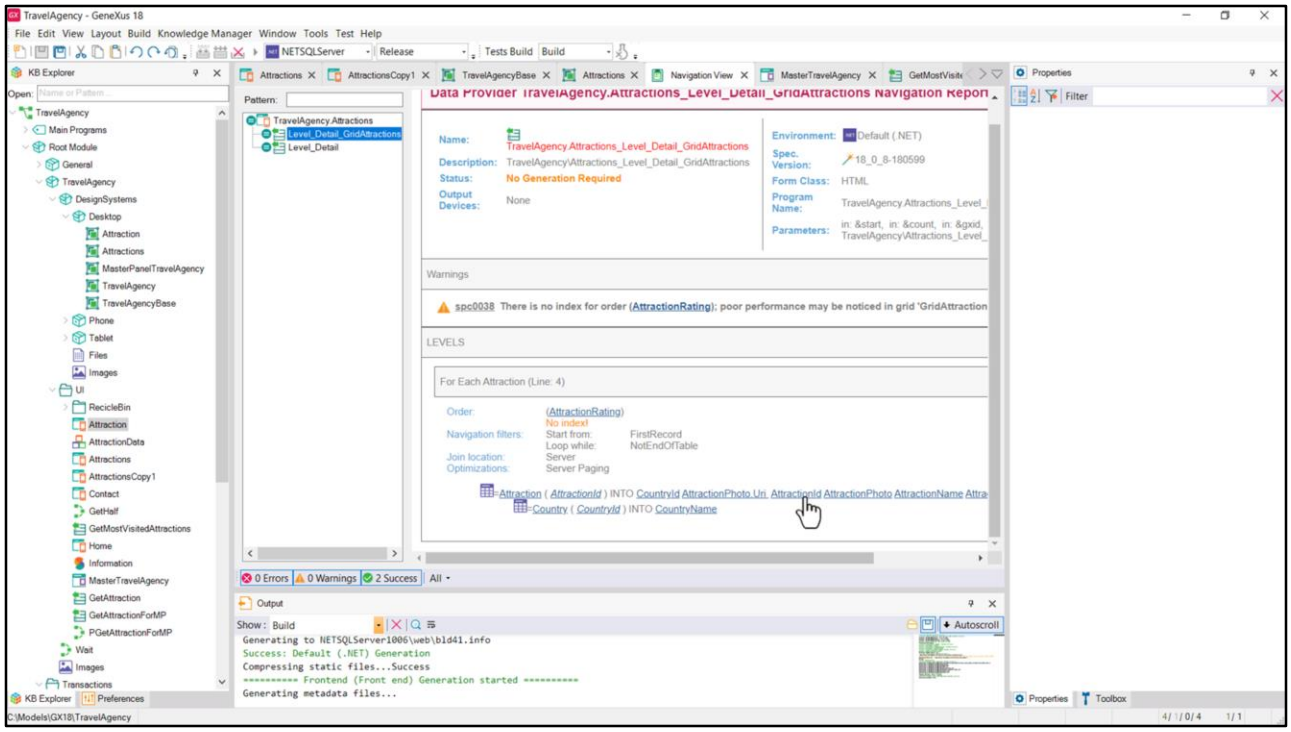
A priori parece que no. Así que si nos queremos asegurar de que se cuente con ese valor, podemos insertar el atributo, colocarlo como invisible y además decir que no se reserve espacio para él en el layout cuando está invisible, justamente.



Bueno, esto vale para el layout Large. Tendremos que hacer lo mismo para el Small. Y esto ya puede empezar a molestarnos, estas duplicaciones, y ya podemos empezar a estar tentados a crear un stencil para estas cards.
Si no lo hacemos, entonces es que tengo que copiarme este control y pegarlo aquí.

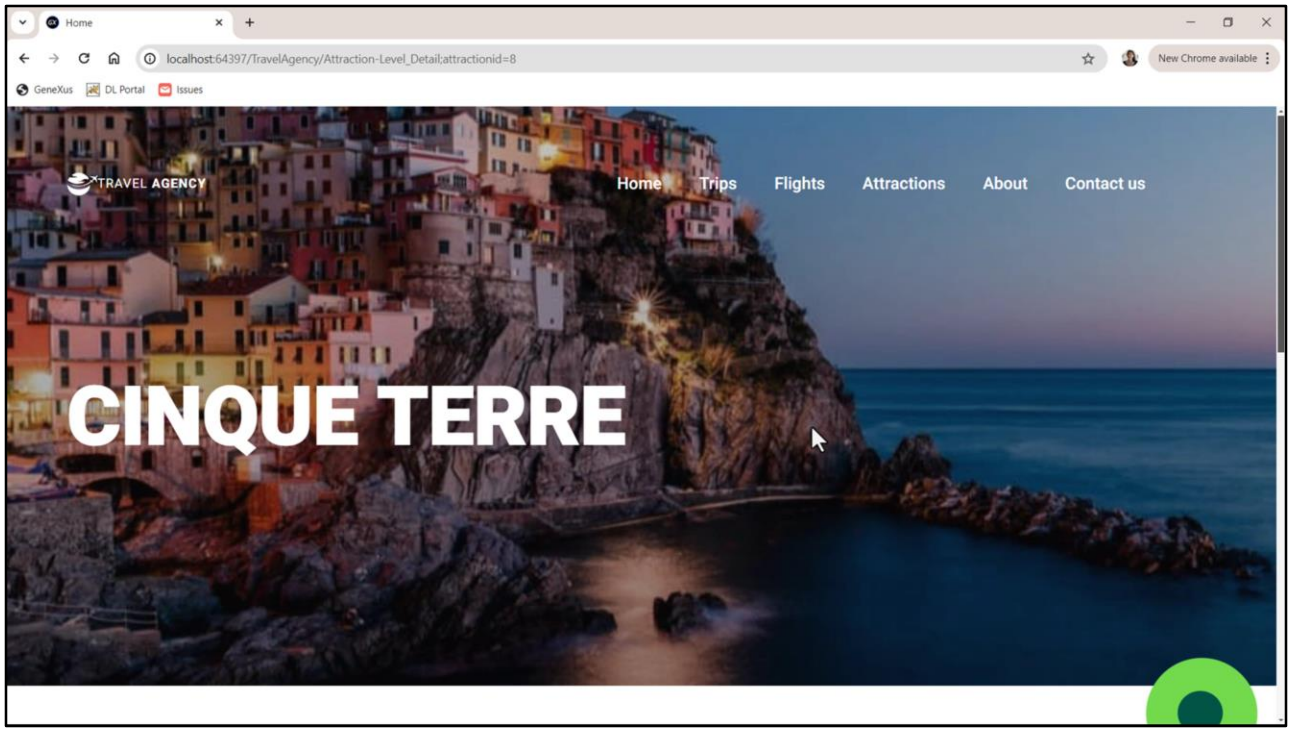


Si ahora ejecutamos… bien… Probemos con otra… la large… perfecto.



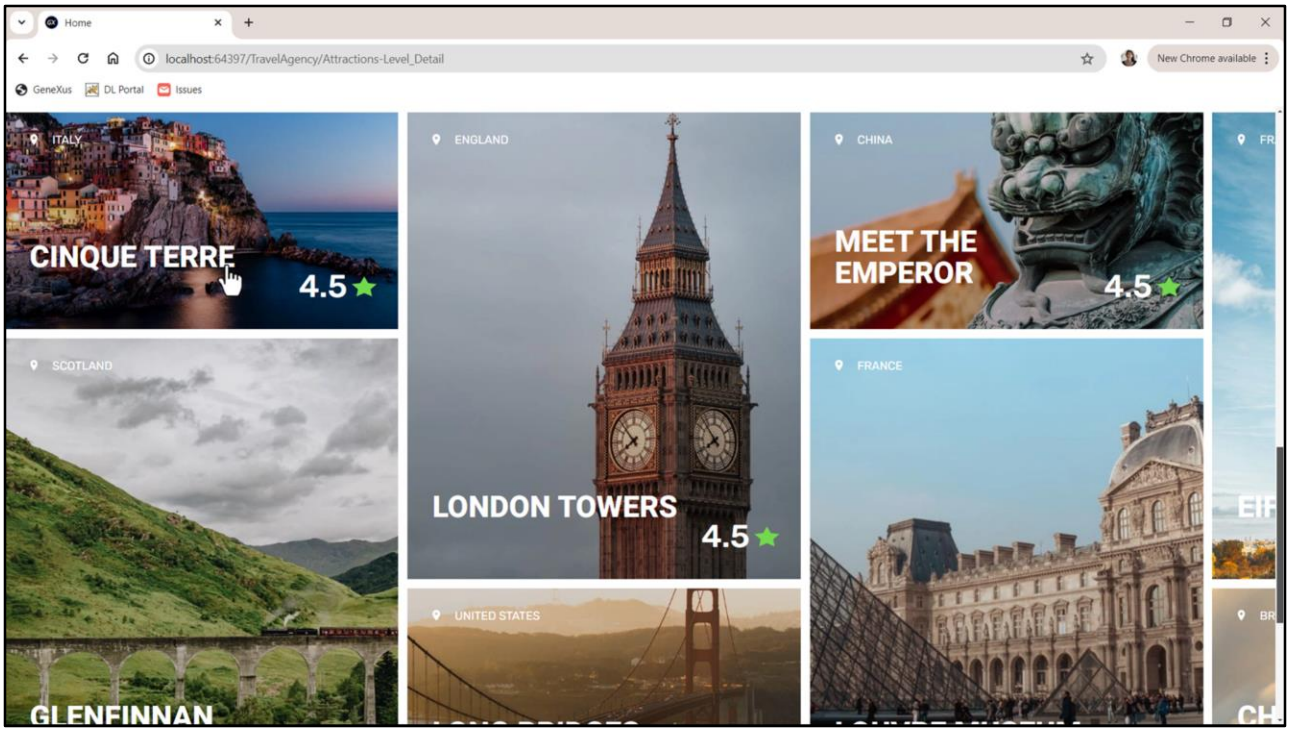
Ahora vamos a ver si era necesario colocar el atributo invisible. Vamos a quitarlo… y quitarlo también del layout Large.

Vamos a ejecutar. Pero antes de que termine veamos el listado de navegación. Vemos en el Data Provider que corresponde al grid que nos informa que sí está trayendo a AttractionId. Lo va a guardar internamente. No teníamos por qué hacer lo que hicimos.

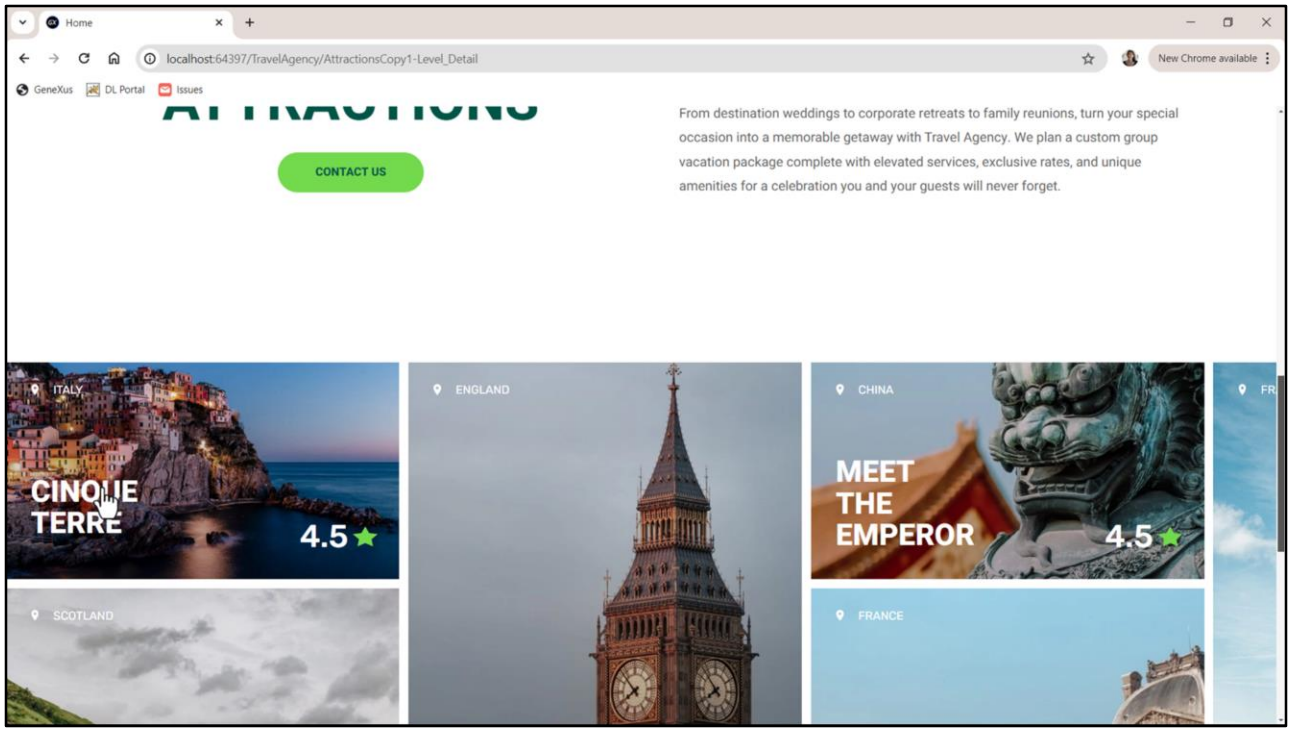


Y vamos a terminar de constatarlo… bien.

Está pasando, efectivamente, el id de atracción porque cuenta con él y eso lo hizo GeneXus automáticamente. Cuento esto porque en los Web panels esto no es así.

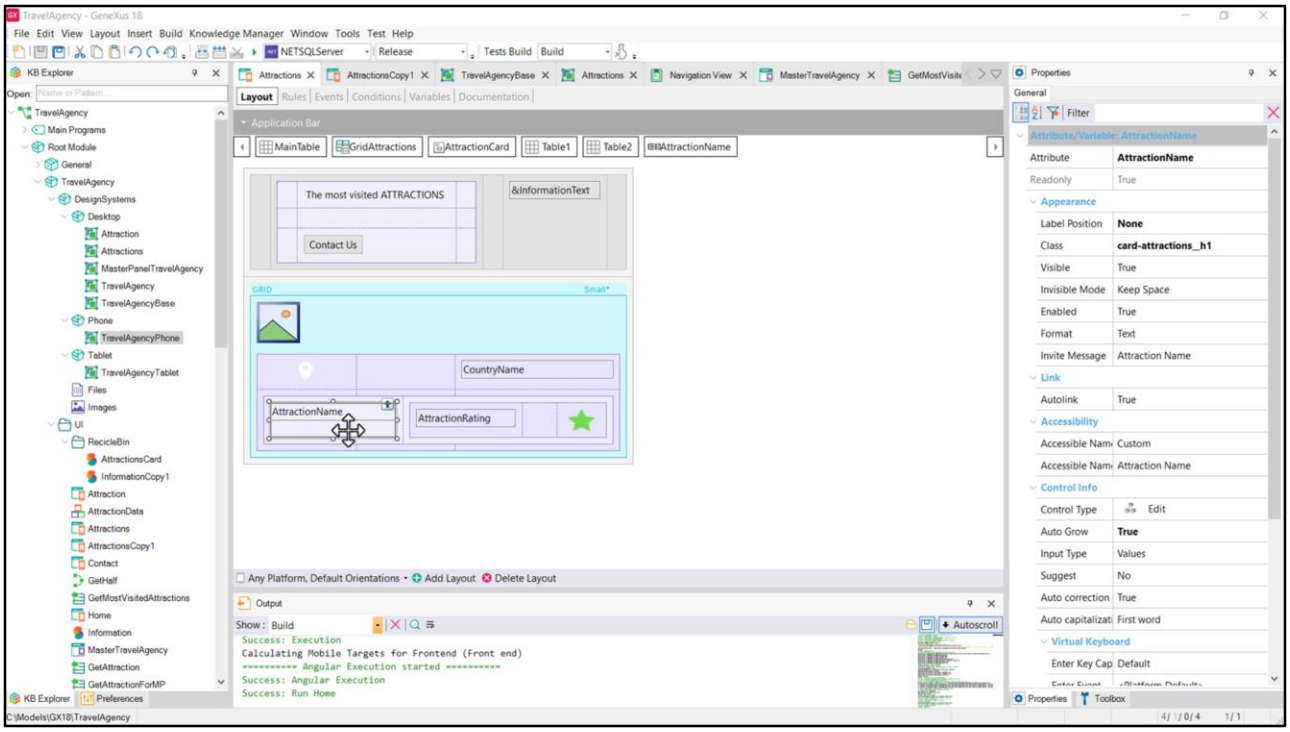


Algo que nos puede llamar la atención es que aquí el texto no está quedando partido en líneas…

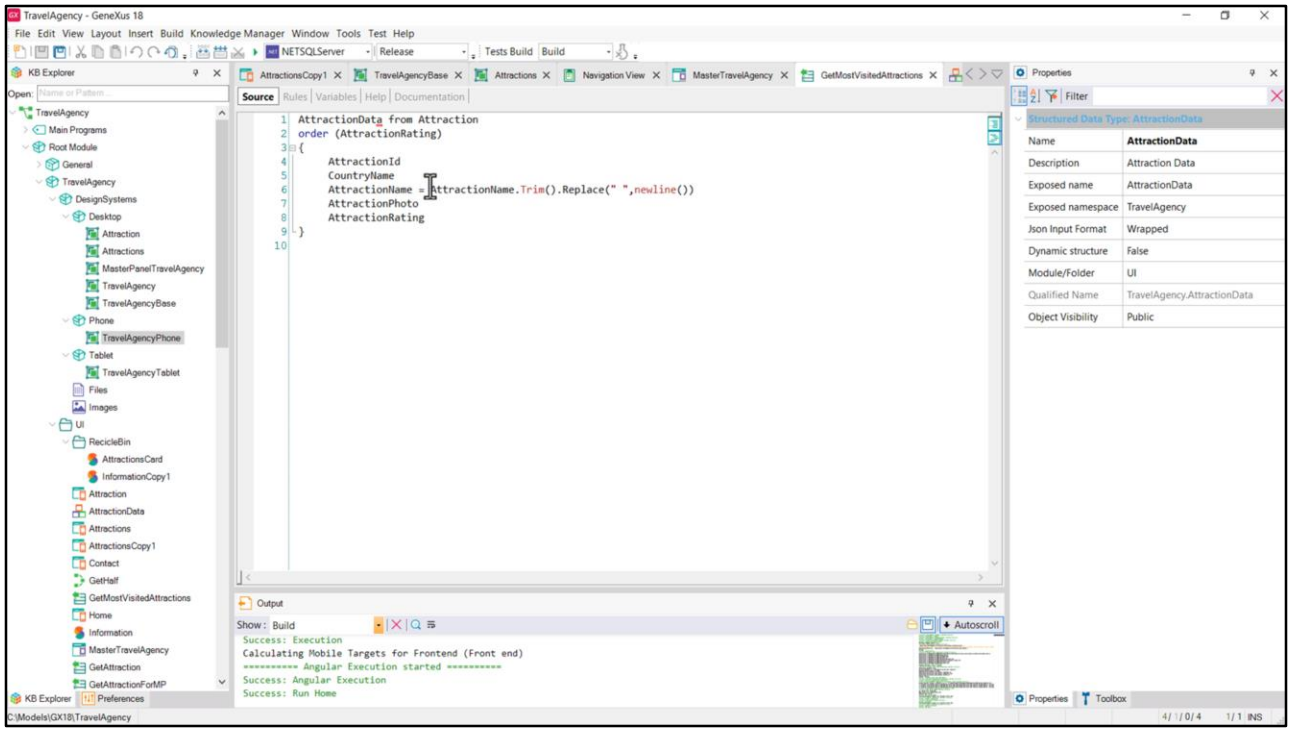


…como sí lo estaba en la solución con el SDT.

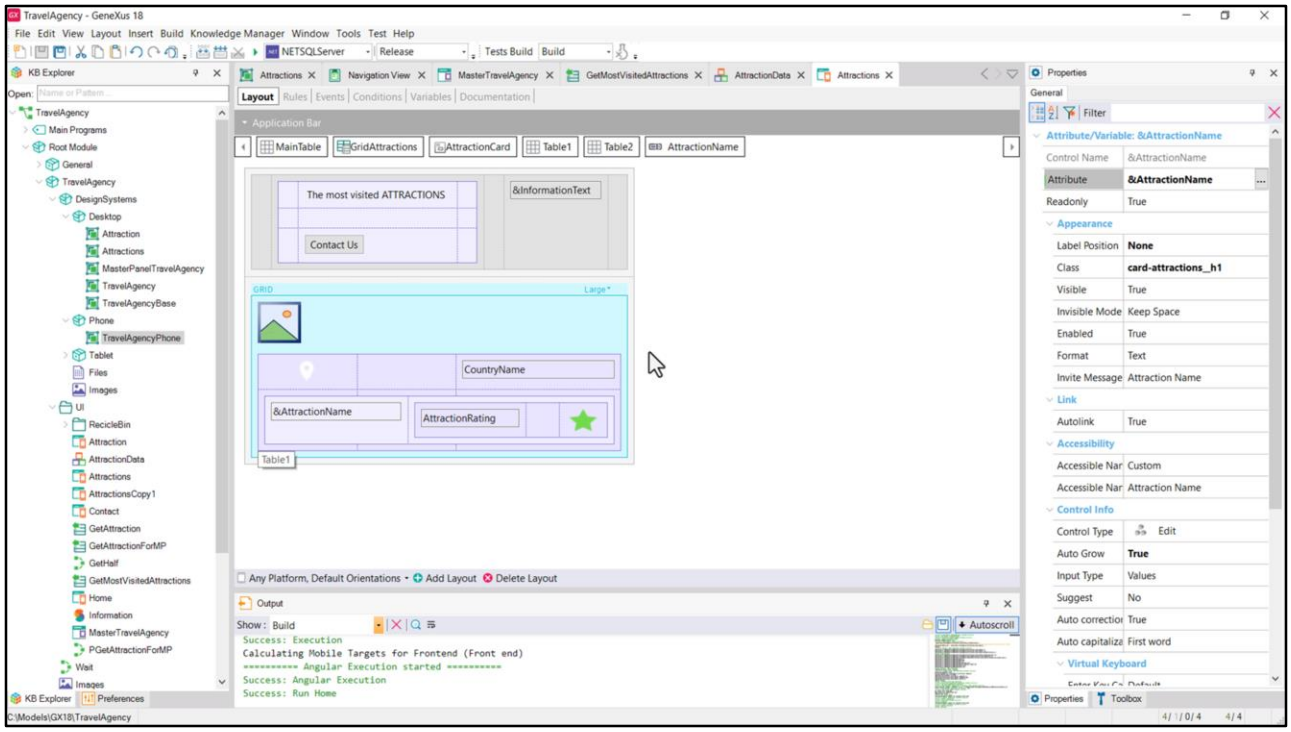
Los que quedan partidos es porque el texto *per se* no entra en el espacio.



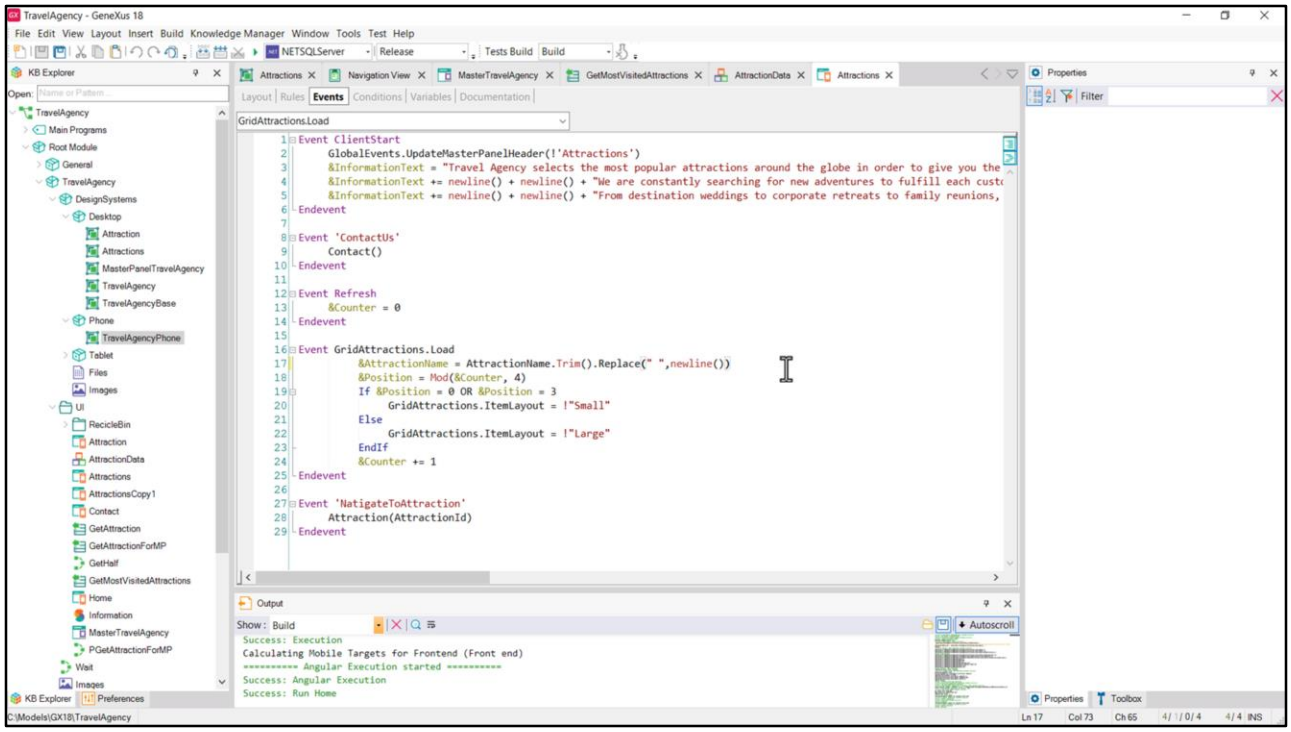
Es que al utilizar el atributo su contenido se muestra tal como está cargado en al base de datos.



En mi solución, en cambio, yo proceso lo que devuelvo al SDT para este campo de esta manera.

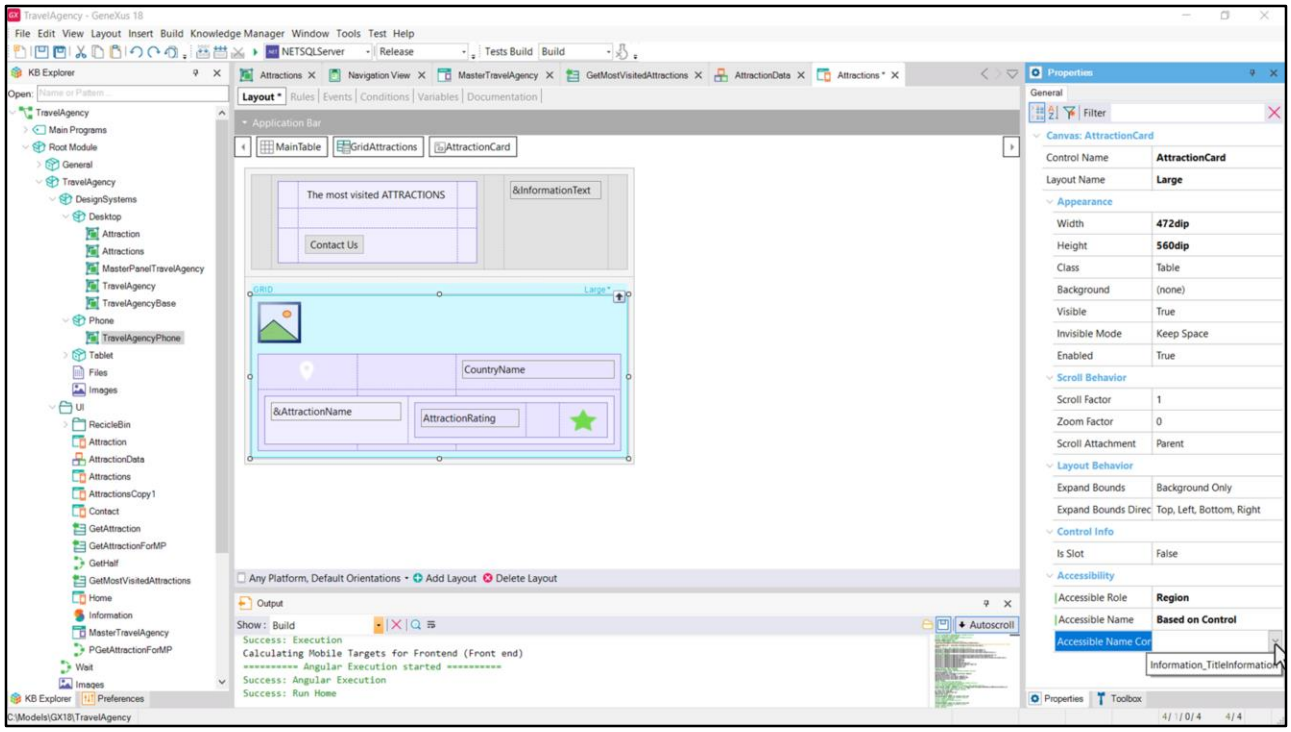


Para adaptar mi solución tendría que colocar una variable aquí en lugar del atributo, variable que me conviene basar en el atributo (por ejemplo llamándole igual, ya automáticamente me va a ofrecer basarla en el atributo de igual nombre) y tendría que hacer lo mismo para el otro ítem layout.

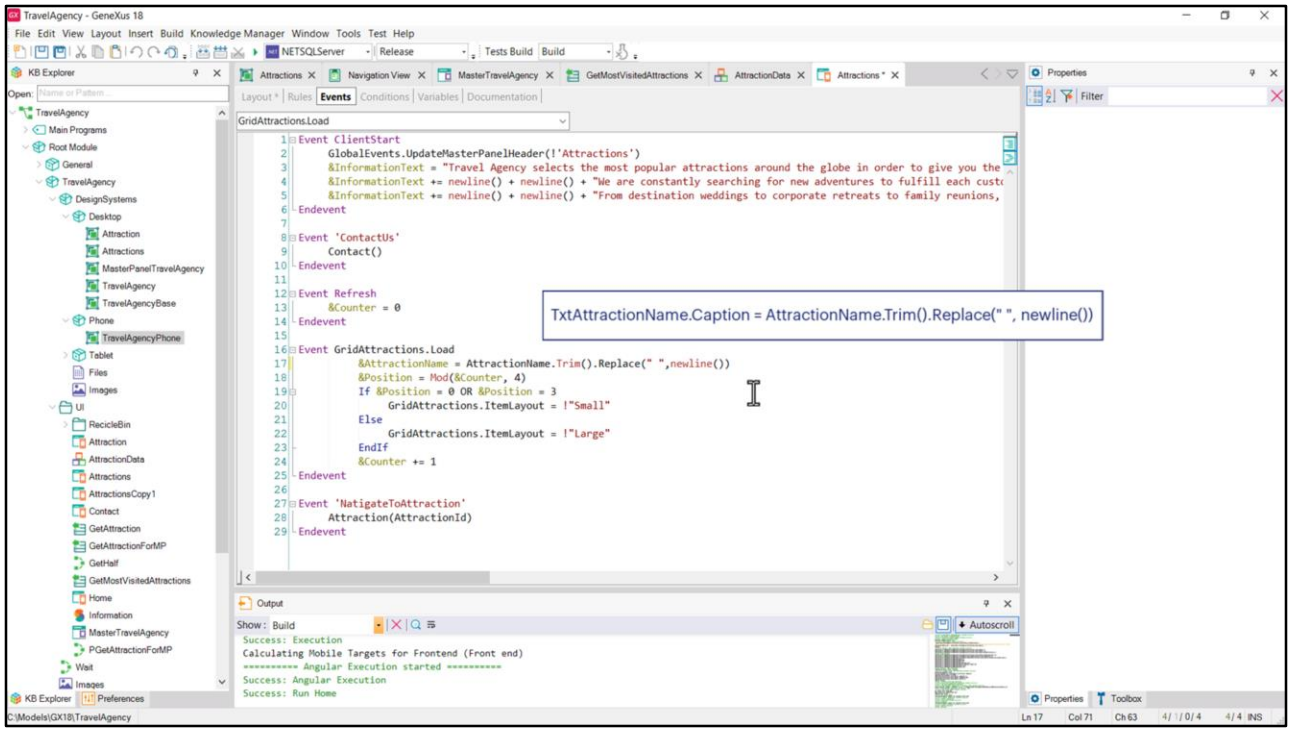


Y en el evento Load asignarle este valor a la variable.

Si ahora ejecutamos, vemos los textos tal como queríamos.

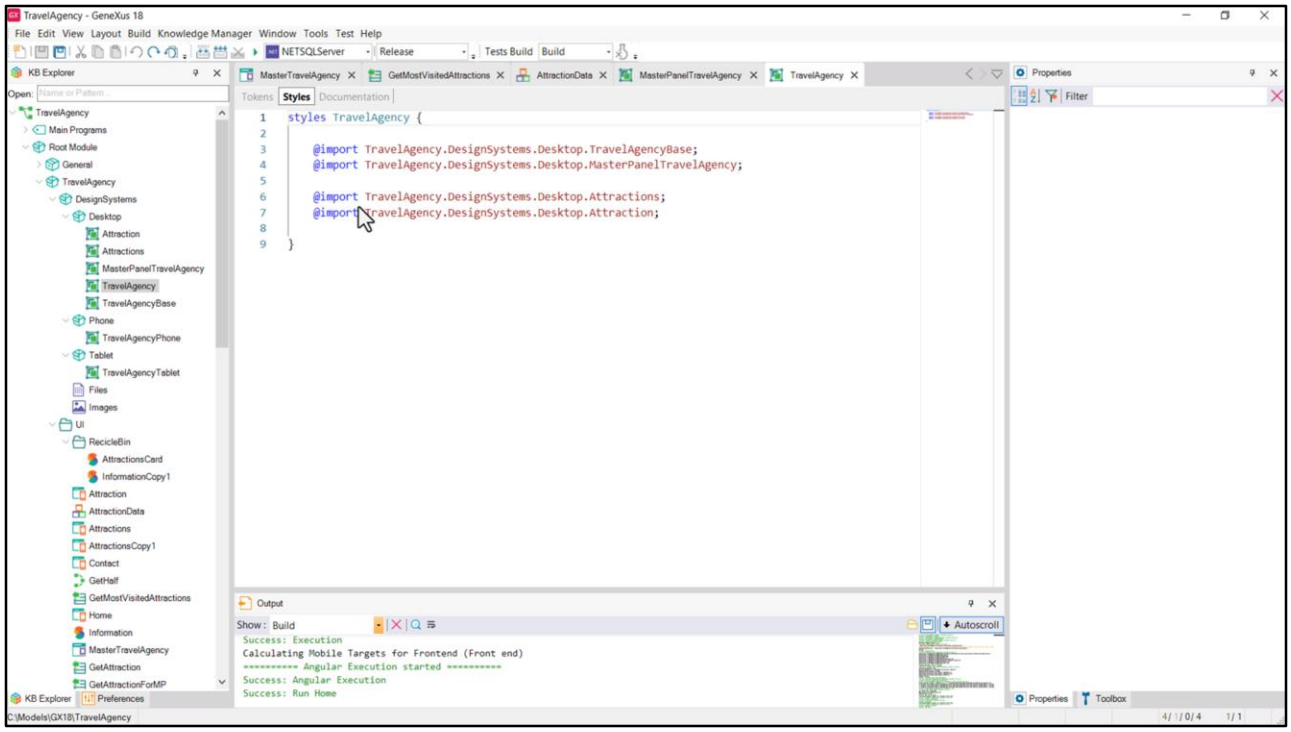


Por otro lado, para atender a la accesibilidad, tendríamos que hacer que cada card muestre como nombre el nombre de la atracción, para lo que aquí deberíamos colocar Region… y en Accesible Name no Custom, sino el basado en un control… pero por ahora sólo podemos basar en control TextBlock y no en variable de texto… por lo que, en verdad, en lugar de la variable AttractionName, deberíamos colocar un TextBlock…

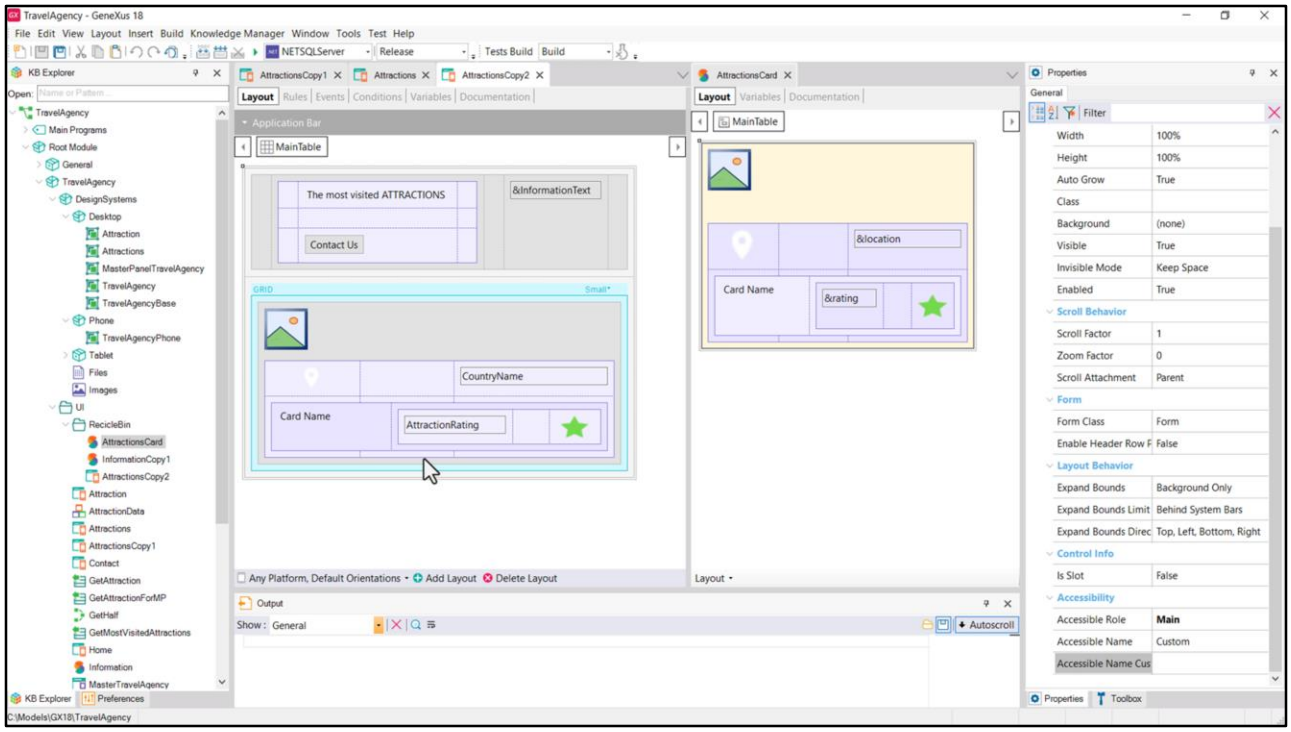


…y hacer esta asignación pero a la propiedad Caption. Para no extenderme más, lo dejo para que ustedes lo hagan.

El grid también tiene propiedades de accesibilidad.



No lo había mencionado explícitamente así que lo hago ahora: creé un DSO por objeto con cuya UI trabajé hasta el momento. Así tengo el del Master Panel, el de Attraction, el de Attractions que fue en el que hicimos todo esto…
El de Base, y en el TravelAgency que será el padre de todos, por supuesto tuve que incluirlos a todos estos otros.



Por último, podríamos haber utilizado un stencil para no duplicar el layout de las cards, dado que la única diferencia entre el ítem Small y el Large era el alto del canvas y nada más.

Entonces podríamos haber insertado para cada ítem layout un control stencil… de ese modo todo lo que hace al layout de la card se resuelve una única vez. Por ejemplo si quisiéramos cambiar la distancia de esta tabla respecto al Top. Lo hacemos aquí y va a repercutir automáticamente en ambos ítem layouts.

Sin embargo hay algunas precisiones que hacer a esta solución, que tienen que ver con cómo se trabaja con valores relativos para la tabla interna al grid, en las que ahora no podemos entrar, y por eso no ahondaré en esta solución. Pero les dejo un xpz en el que podrán investigar todo esto.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com