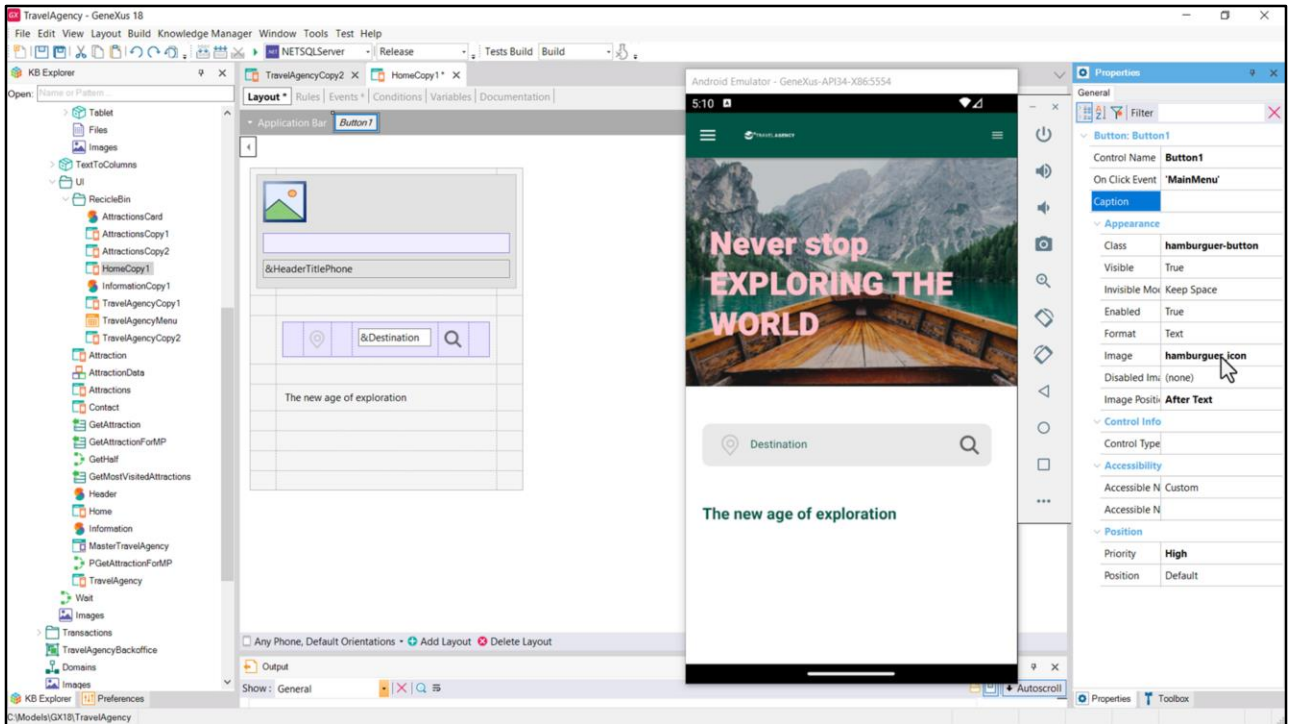


From Travel Agency Web to Native Mobile (part 3)

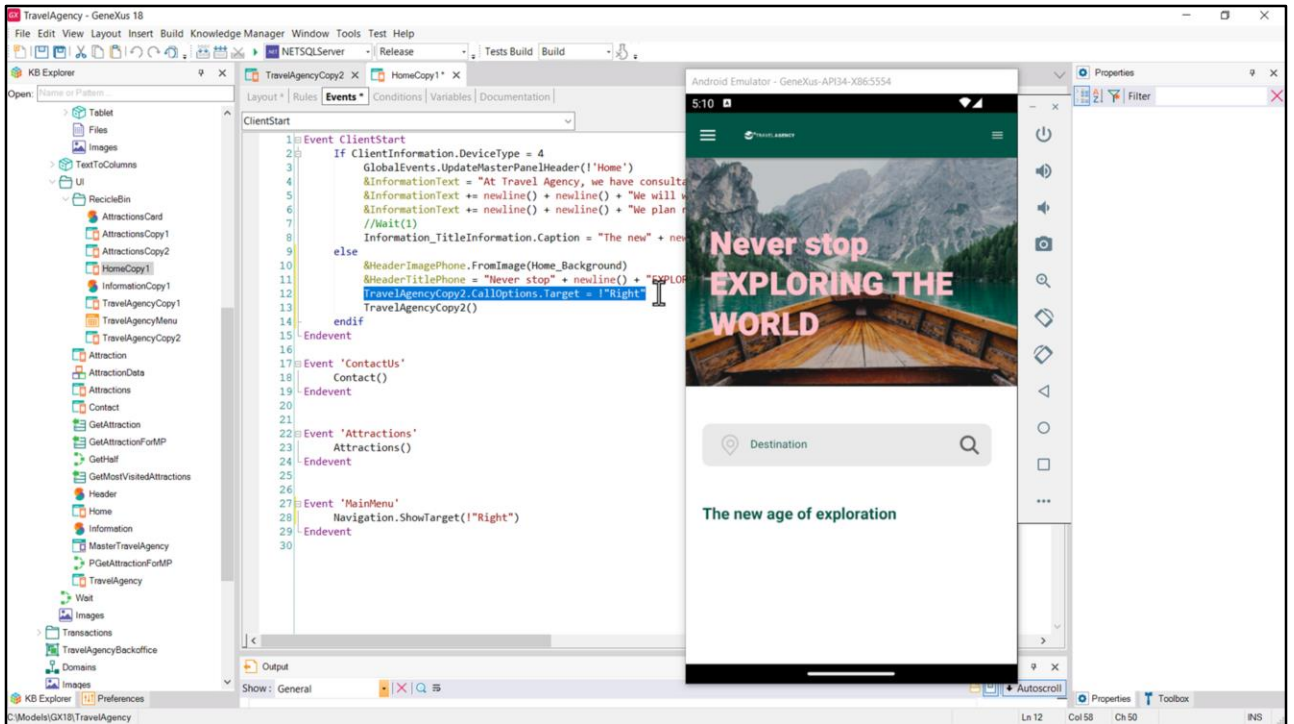


Cecilia Fernández

Continuemos donde interrumpimos en el video anterior.

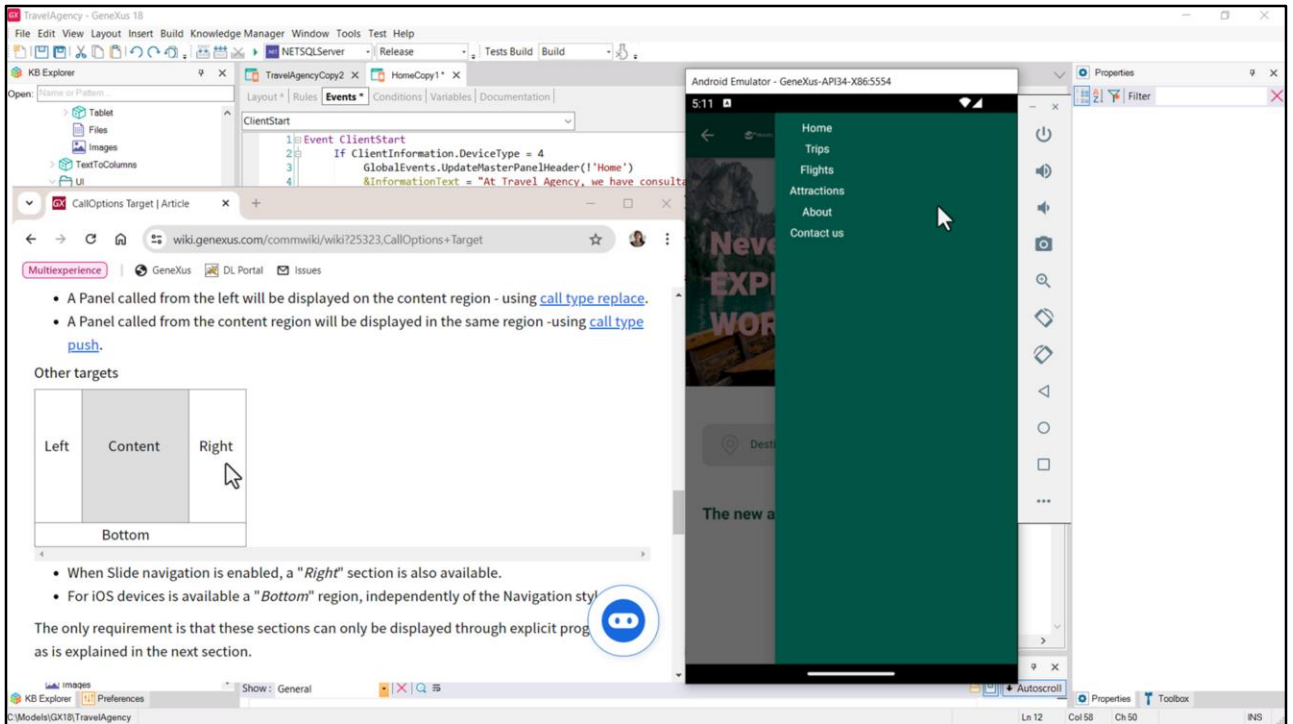


Veán qué pasa si a esta copia del panel Home le coloco un botón en la Application Bar al que le quité el Caption porque le asigné la imagen hamburguesa que me descargué de Figma.



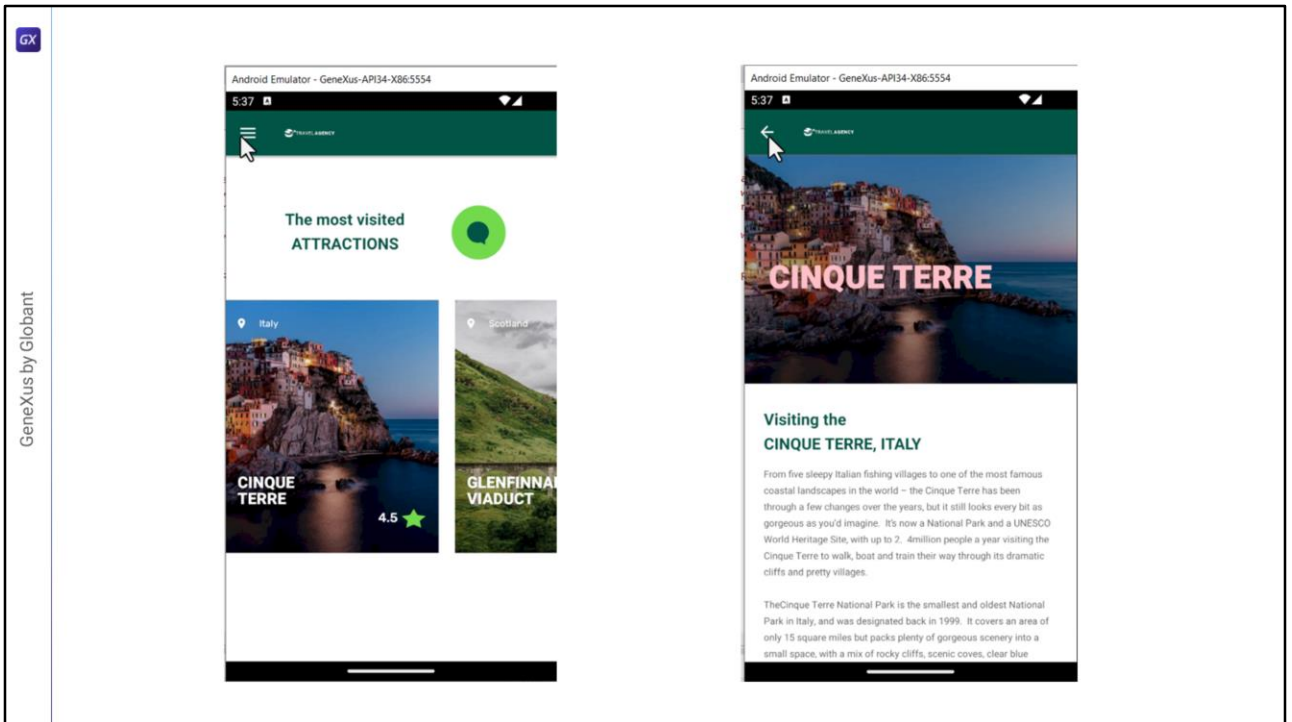
Y lo que hago en el ClientStart es invocar al menú (a esta copia del menú, que es idéntica a la del menú que teníamos, pero ahora invoca a la copia de Home y no a Home).

Pero vean que antes de invocar al menú le indico que su **target** no será el central, el que es por default, sino el de la **derecha**.



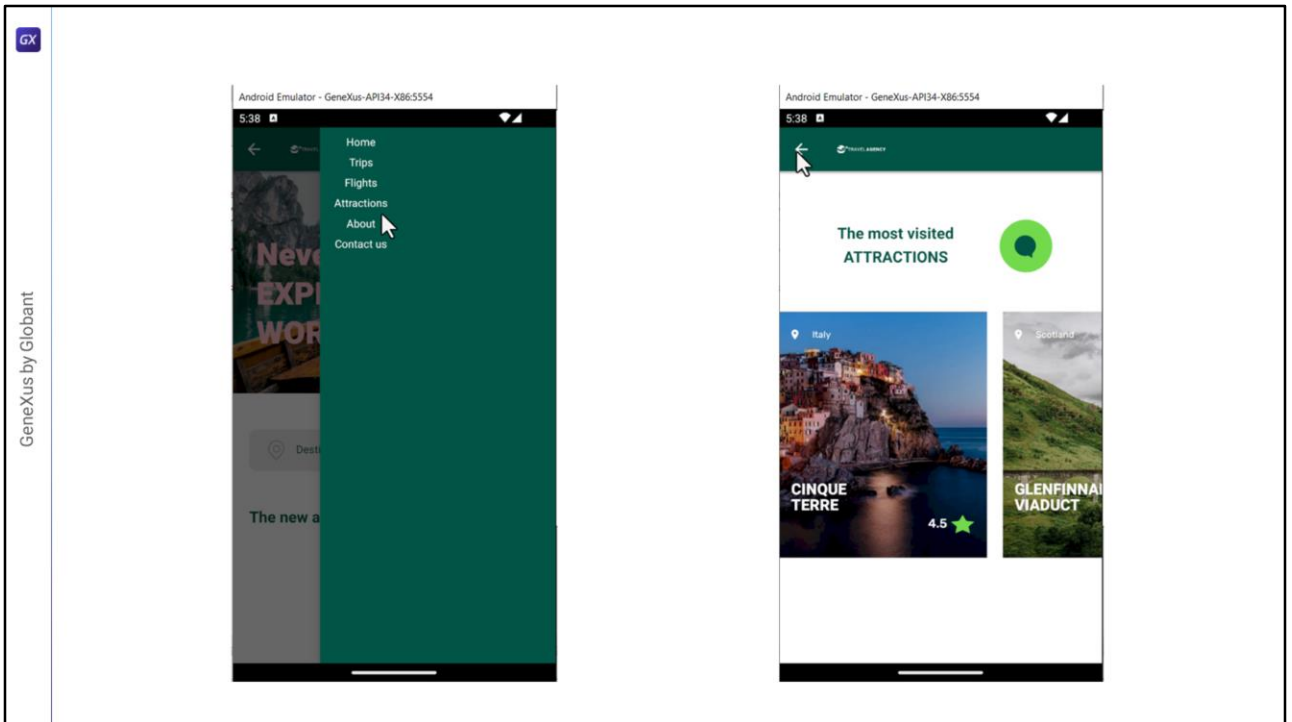
Quando el estilo de navegación es **Slide** para un teléfono Android, tendremos dos targets obligatorios: el de la izquierda, que es el del menú hamburguesa, y el central que es en el que todo panel se cargará por defecto.

Aquí estamos queriendo utilizar un target a la derecha, para lo cual, para hacer que se despliegue, tenemos que utilizar el objeto externo **Navigation**, que se encuentra en el módulo GeneXus.



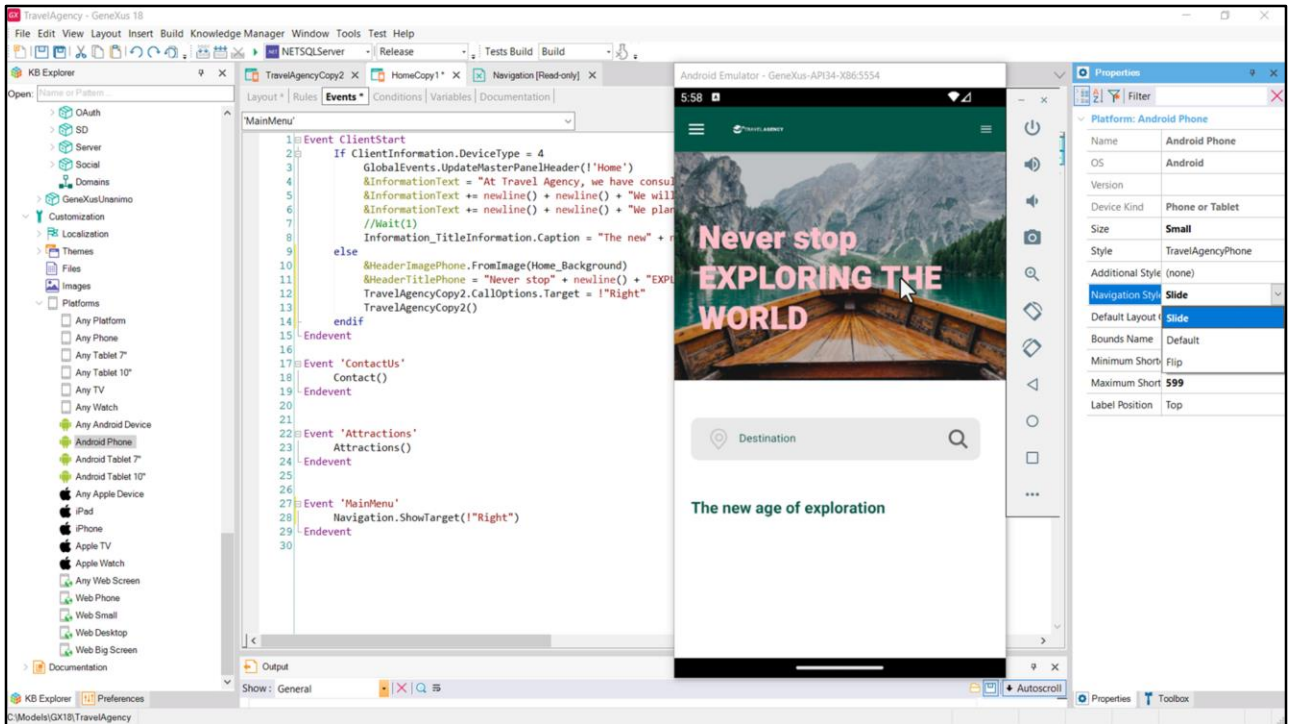
Sin embargo no es tan sencillo implementar el menú hamburguesa como puede parecer así a primera vista.

No voy a entrar aquí en detalles, pero alcanza con ver que, mientras que si utilizo el menú propio de la plataforma, cuando elijo Attractions vuelve a aparecer el menú hamburguesa y recién si me meto en profundidad en esta otra pantalla aparece el botón de back para volver a quien lo llamó, y allí de vuelta recupero el menú hamburguesa...



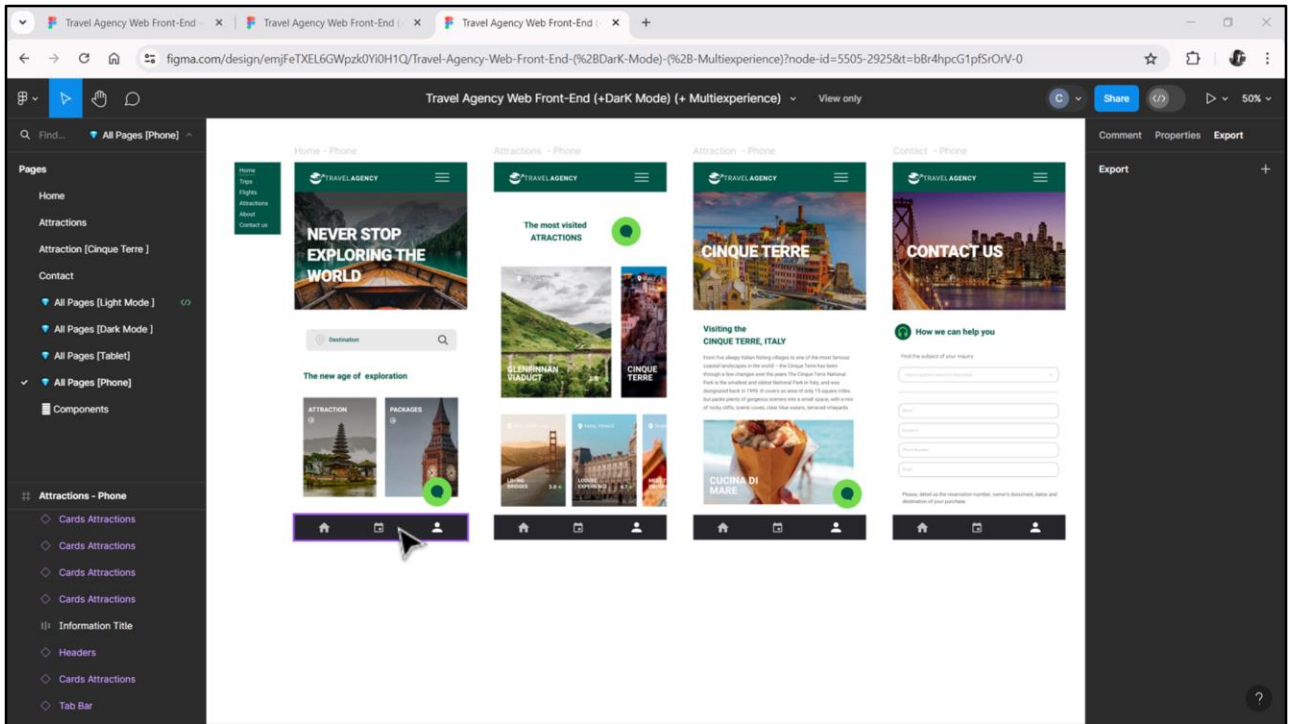
... si voy por este otro lado, el del menú manual, en realidad aquí estoy en un panel que no tiene la semántica del menú. Es un panel como cualquier otro, que está en el mismo stack de invocaciones, por lo que, cuando llame a Attractions, vemos que ya allí aparece el botón de back (por supuesto que si me meto en profundidad esto también sucede; pero vean que hago un back a Attractions y en el siguiente back me lleva al panel del que partí).

Tendría que hacer varias cosas más, aquí, para emular al menú hamburguesa, como invocar con CallOptions Type Replace. No lo vamos a ver.



Porque aunque lo resolviéramos, no nos va a funcionar la solución cuando quitemos el estilo de navegación **Slide**. Observemos que en esta solución tenemos una duplicación: la solución automática, con **Slide Navigation**, y la manual, abriendo el panel en el **target derecho**. Entonces quisiéramos quitarle el estilo Slide para que no coloque más el menú hamburguesa a la izquierda. Sin embargo, por el momento el **target right** sólo funciona con estilos de navegación a pantalla partida. El **Flip** no es uno de esos.

Así que si fuera indispensable que el menú hamburguesa estuviera a la derecha, contraviniendo las convenciones de Android, tendríamos que buscar otra solución.



Para no irnos de este tema de los menús.

Vean que en el diseño en Figma tenemos doble menú de la aplicación. Tenemos el hamburguesa, y tenemos este otro menú en forma de **tabs**, que invoca a objetos que aún no tenemos ni siquiera diseñados, pero que existirán. Este menú en forma de tabs no es propio de las aplicaciones Web, sino de las nativas.

Several ways to show a Menu


This documentation is valid for: [GeneXus 18 Help](#) [GeneXus 17 Help](#)

There are several ways to show the same information in a [Menu object](#), the visualization can be changed using the [Control property](#) in the Menu's root item.

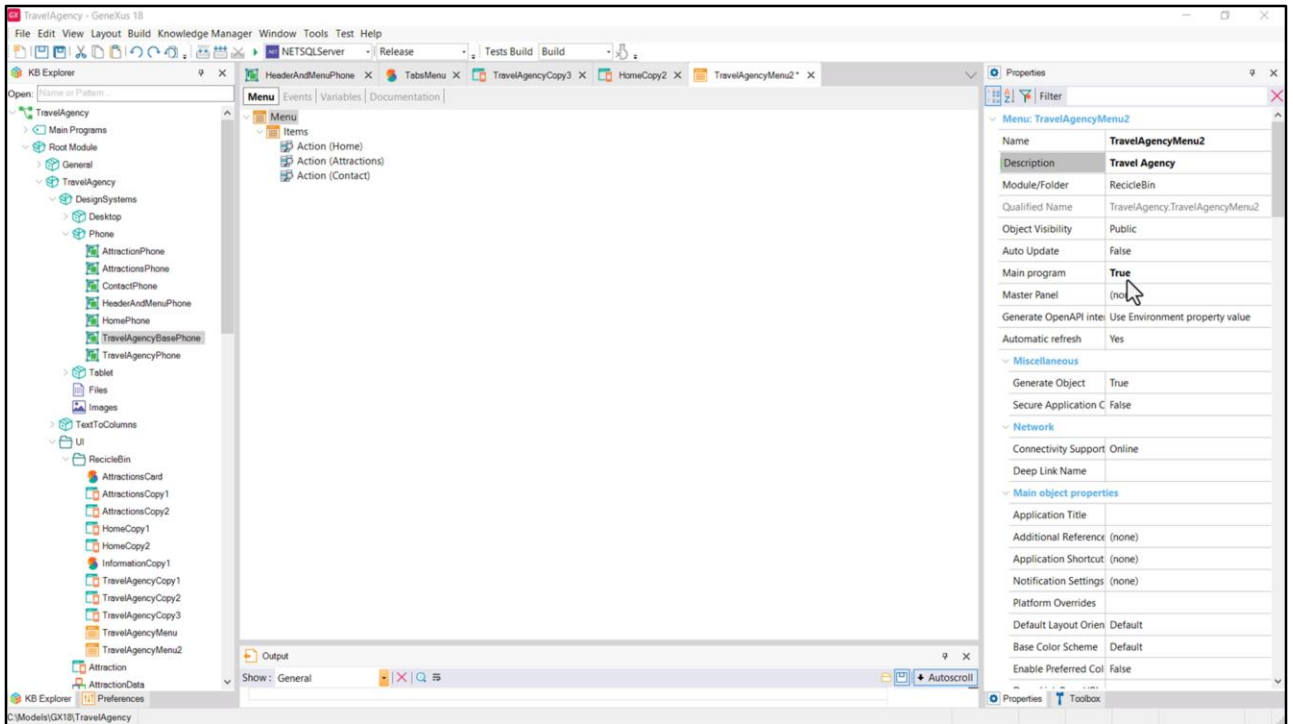
The values are:

List
Menu items are displayed as a list. The content of the selected item is shown on the right of the screen when running on a tablet device.
This is the default for iOS when using [Split Navigation Style](#).

Table
Menu items are displayed as table cells. No content is displayed while the Menu is visible.
This is the default for Android.

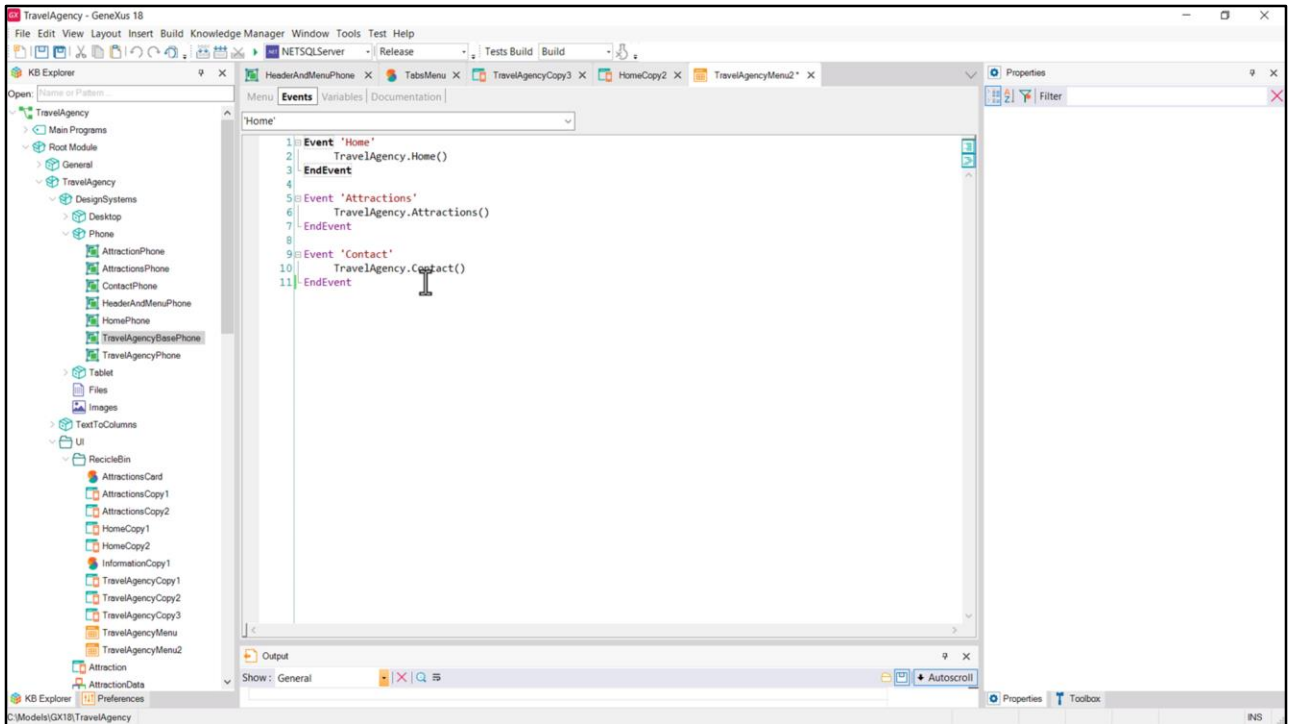


Les había mencionado hace un rato al objeto Menú, que antes existía solamente para aplicaciones móviles, pero que se extendió y ahora también vale para Angular.

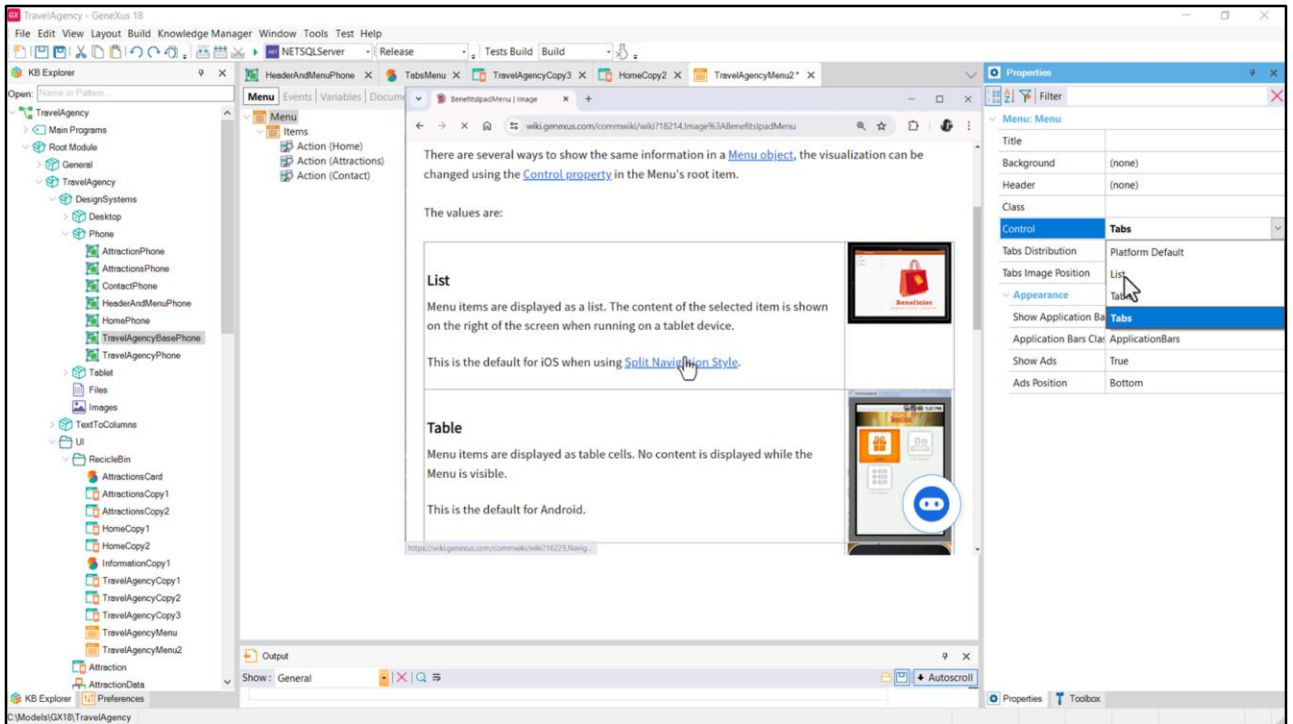


El **objeto menú** es bien específico: implementa un menú de acciones y por eso tiene la propiedad **Main Program** en **True** por defecto, porque se parte de la base de que un menú será el punto de entrada de la aplicación. A través de ese menú y sus navegaciones serán alcanzables todas las pantallas de la aplicación.

Aquí agregué tres acciones, que corresponderán (sólo para probar)...

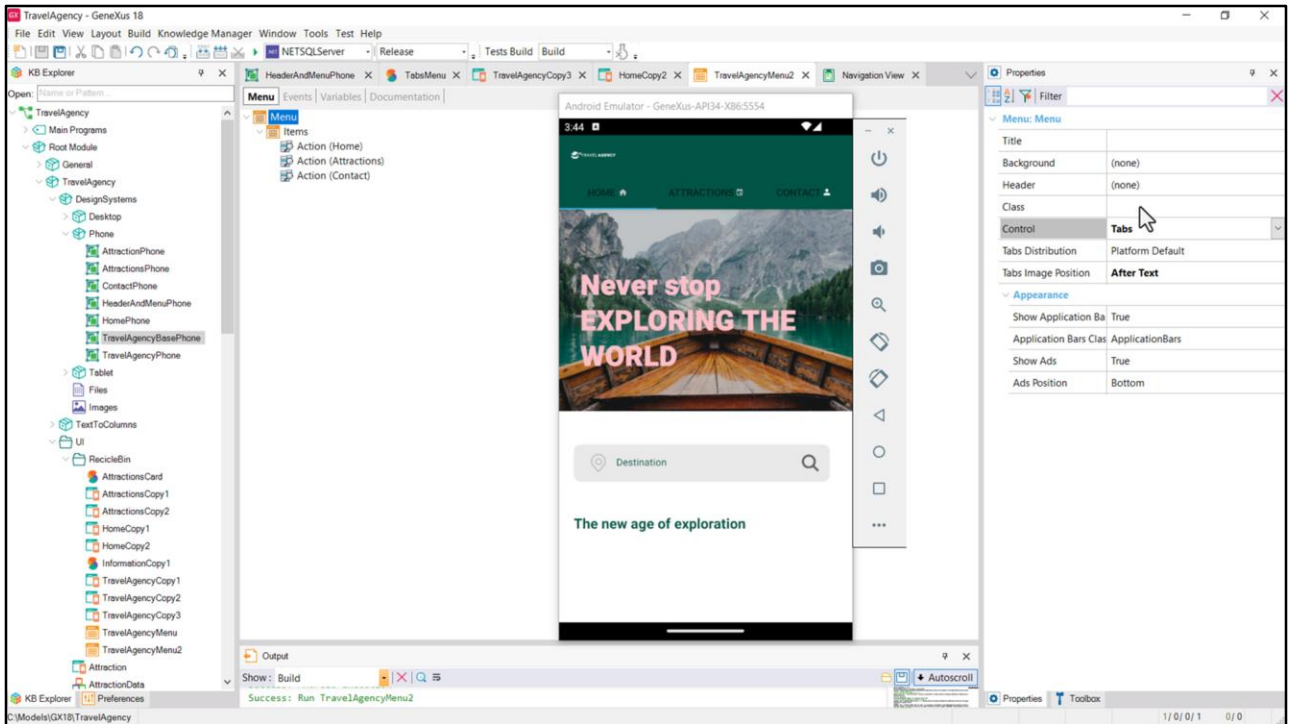


...a llamar a estos tres paneles. Y cuando se ejecute el menú, por defecto se invocará al primer ítem especificado.



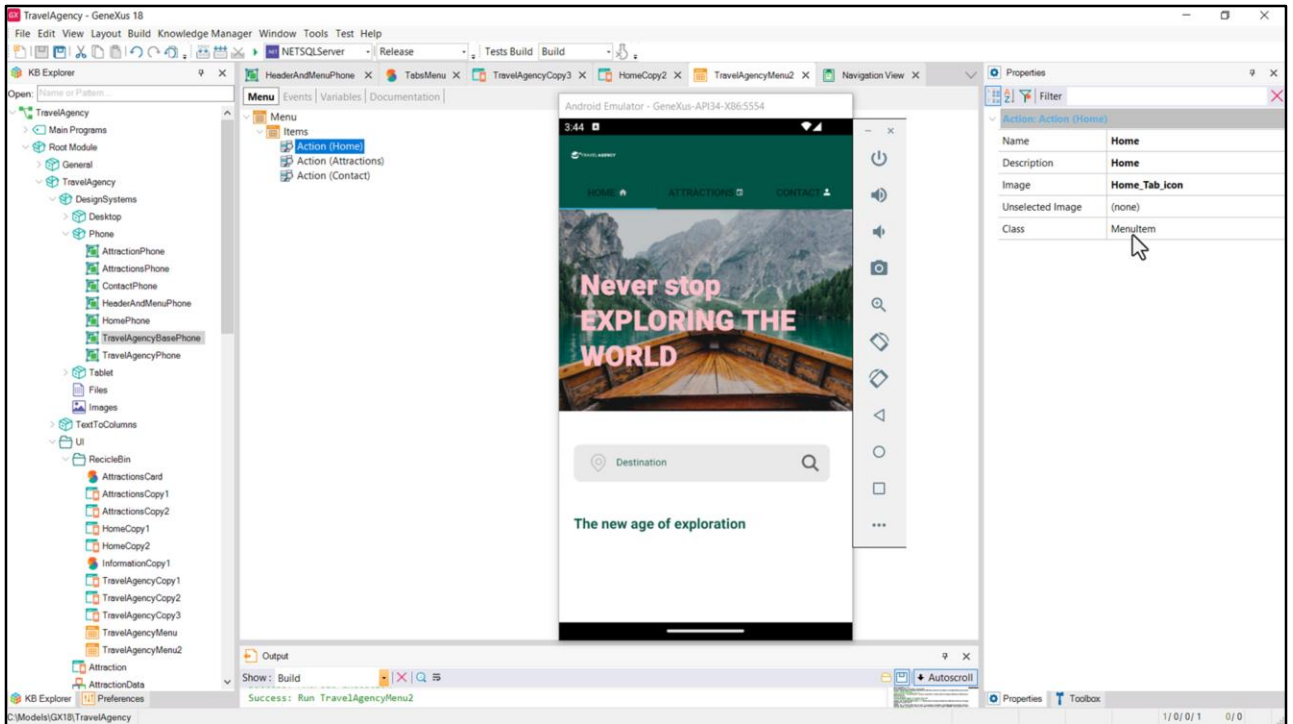
Pero veamos además que ofrece tres formas de mostrar esos ítems: o como **listado** de ítems, que es similar al que habíamos implementado en el panel (aquí tenemos esta copia), como **tabla** de ítems, o como **tabs**.

Y en cada caso, utiliza la forma nativa de la plataforma para mostrarse.



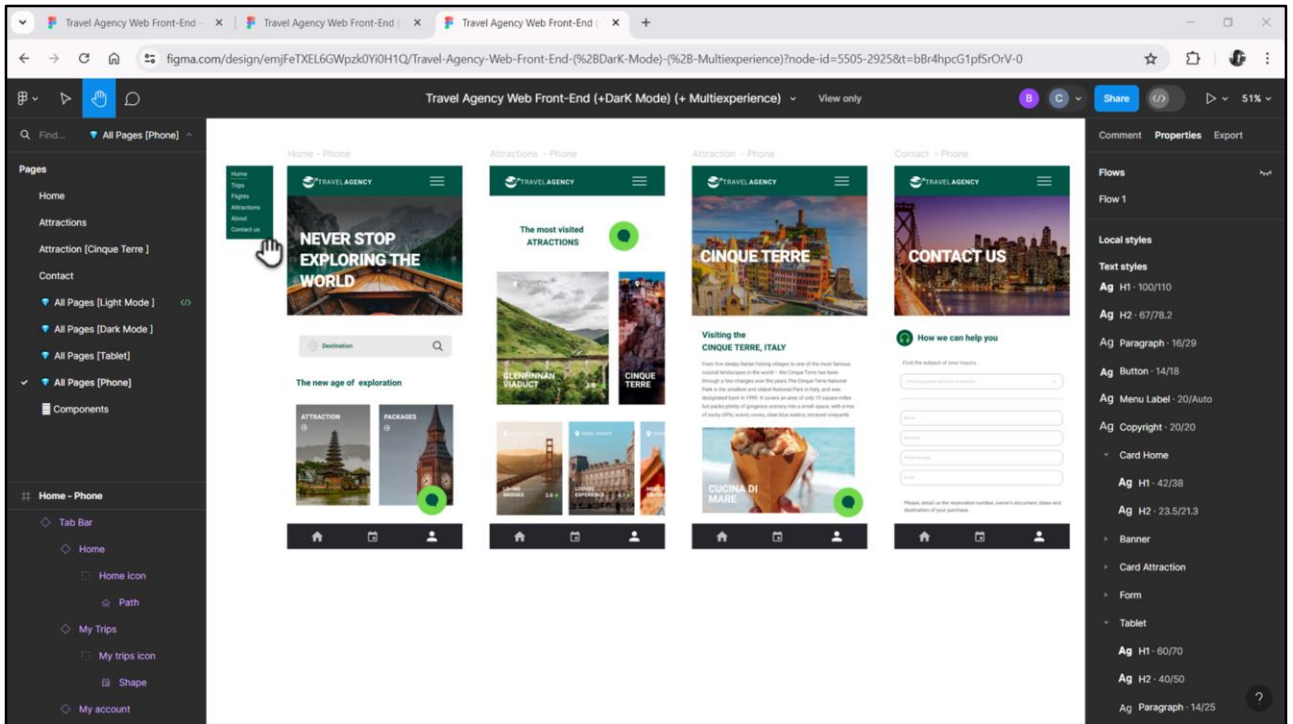
Por ejemplo, para el caso de Android los tabs se muestran arriba.

Por supuesto tendremos clases para dar estilo al menú de ítems: al conjunto...

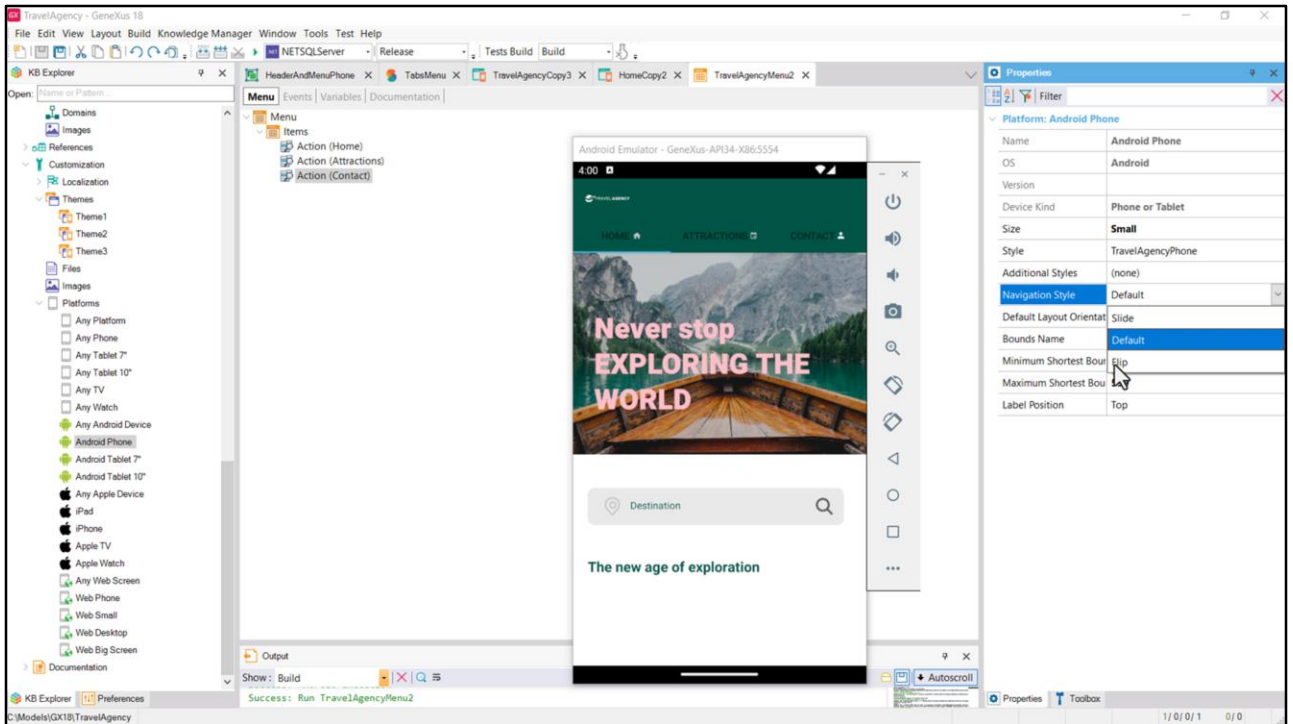


...y a cada ítem.

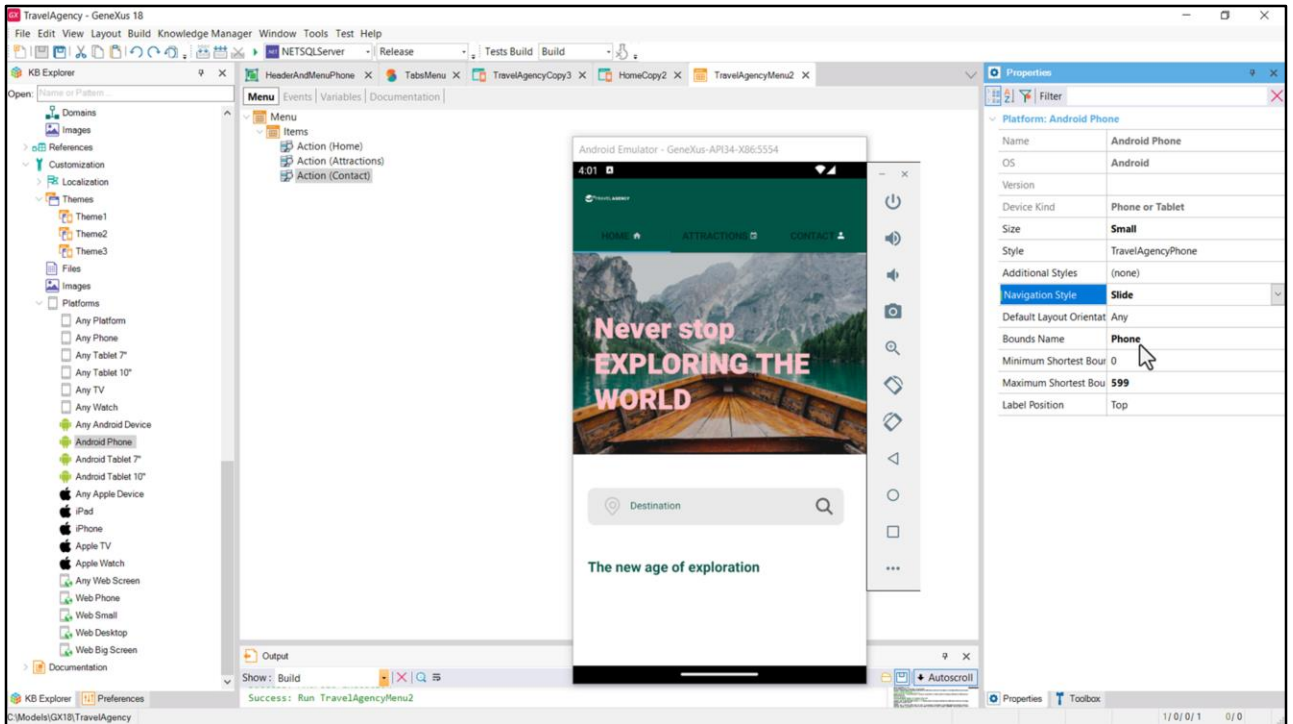
También observen que a cada ítem le asocié una imagen, que, como imaginarán, descargué de Figma e incorporé en la KB.



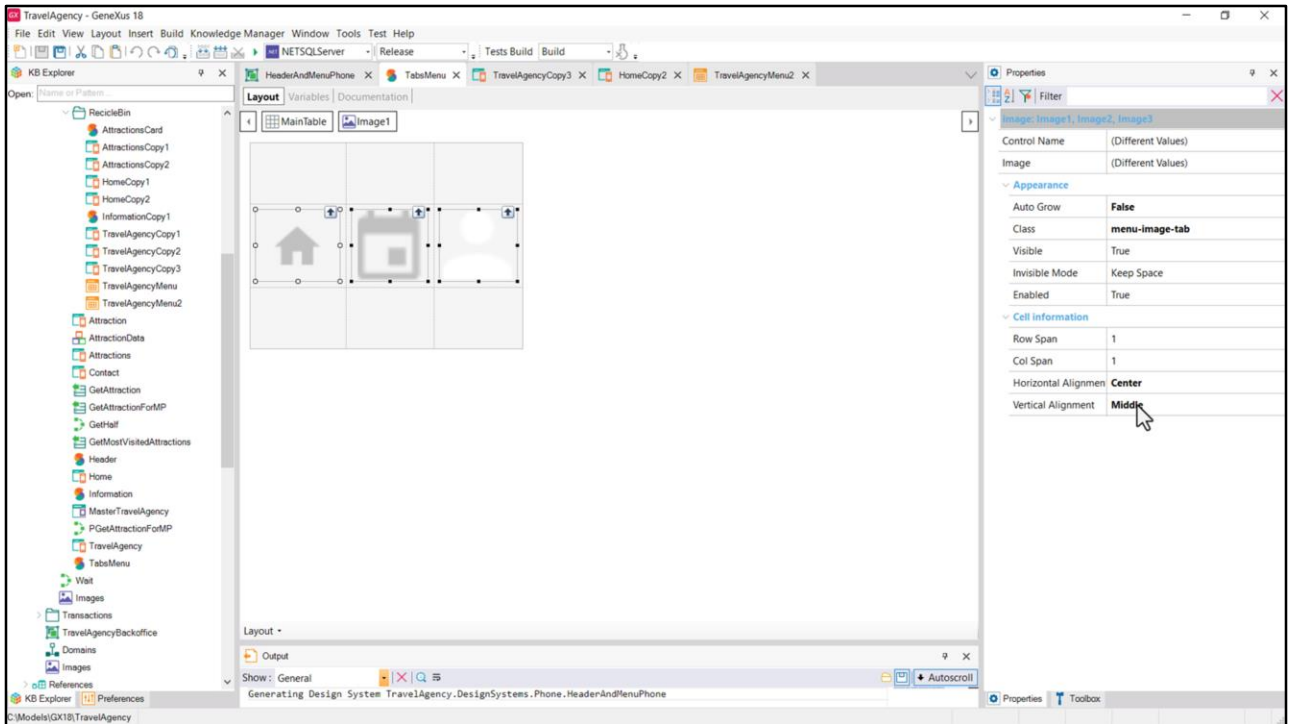
Tendremos que elegir si este menú con tabs será el punto de entrada de la aplicación o si será el menú hamburguesa, porque, al menos por el momento, no tenemos la posibilidad de tener los dos a la vez de forma automática.



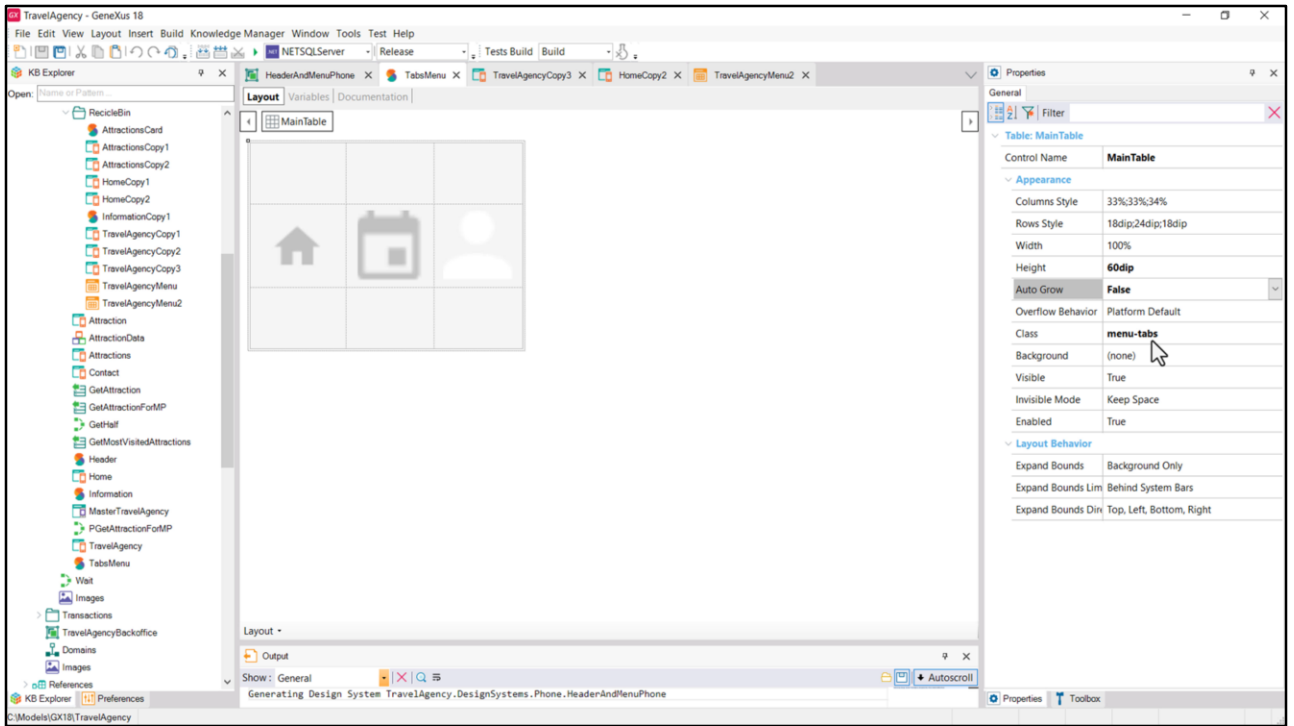
Así, si optamos por tener los **tabs**, entonces el estilo de navegación no será **Slide**, y este será nuestro objeto main, punto de entrada. Aquí tendremos que implementar de forma manual el menú hamburguesa (abriéndose por la izquierda).



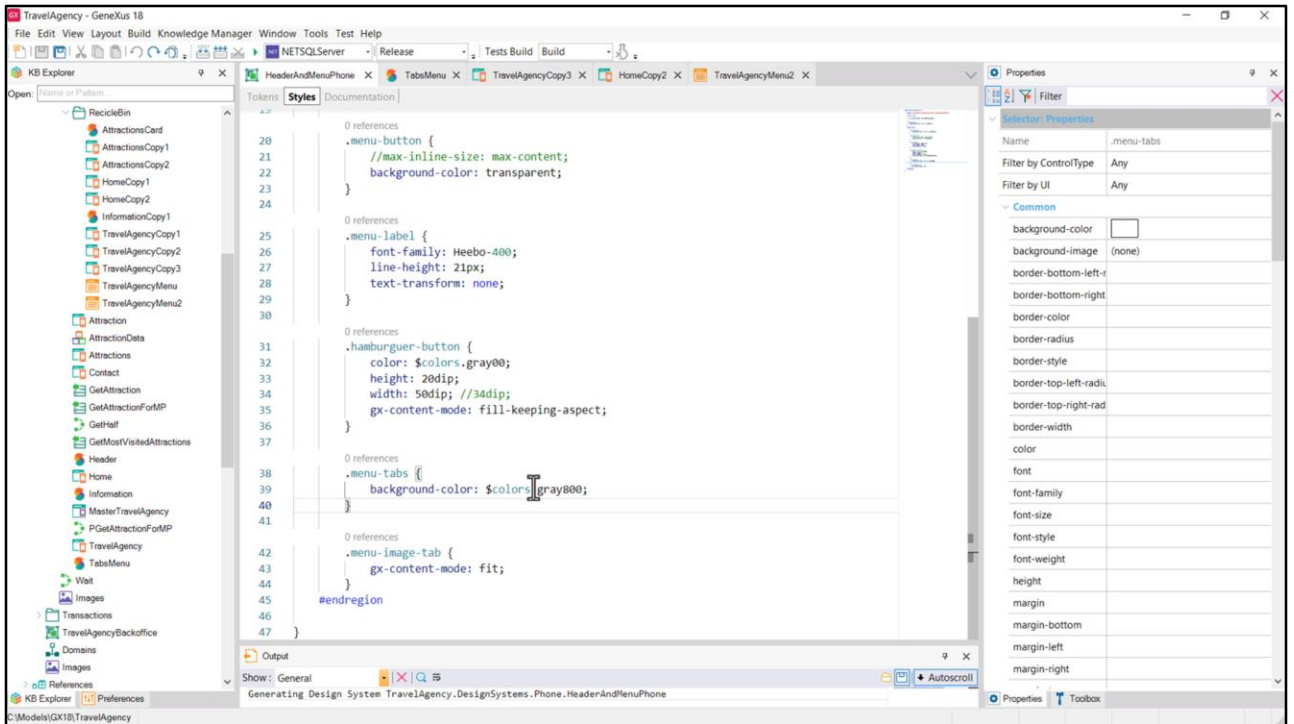
Si, en cambio, optamos porque el punto de entrada sea el hamburguesa, entonces dejaremos como estilo de navegación **Slide** y tendremos que implementar manualmente los **tabs**.



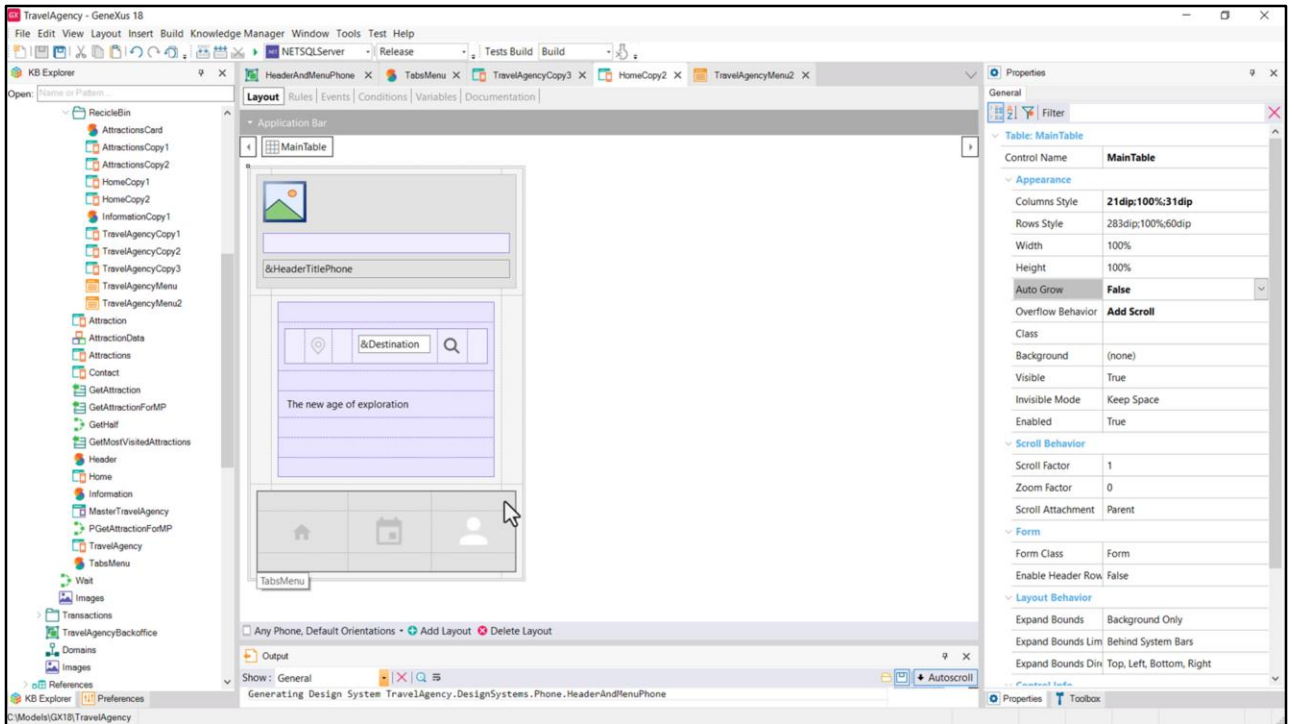
Por ejemplo...como los necesitaremos en todos los paneles, creé un Stencil donde inserté las 3 imágenes, centrándolas y dándoles las dimensiones que tomé de Figma, como ya sabemos hacer...



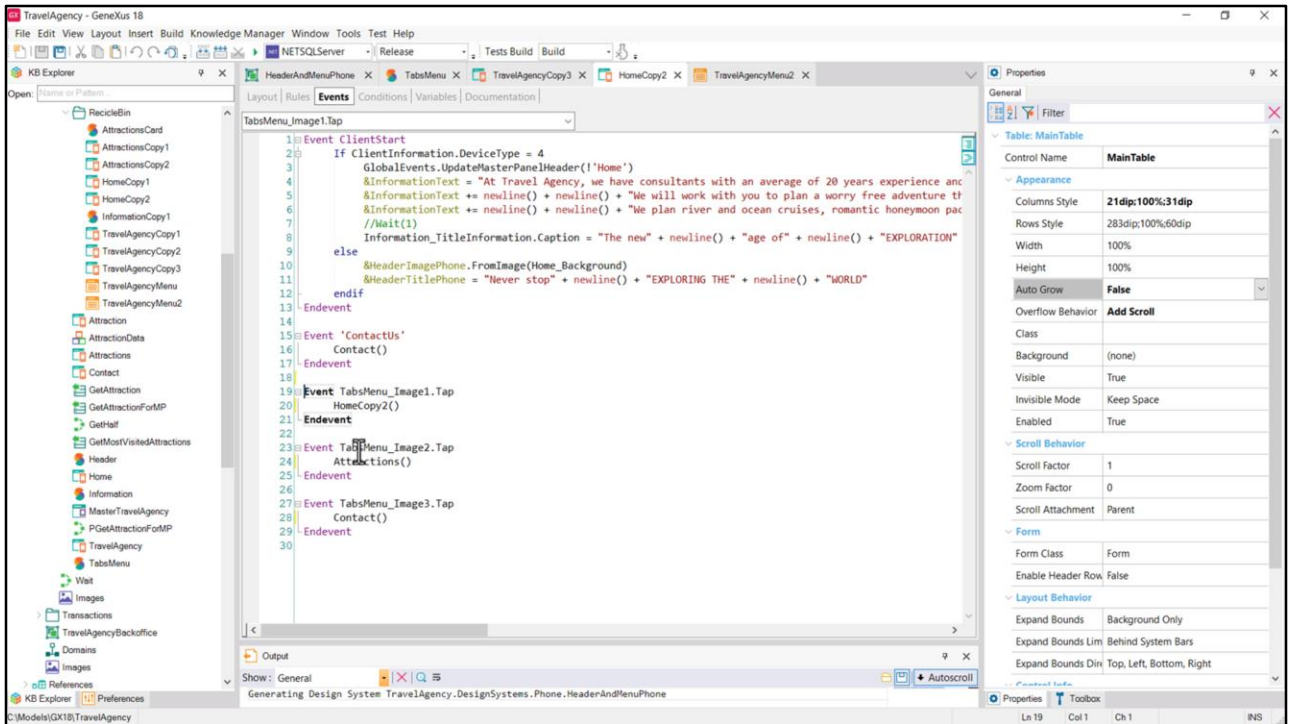
Le asigné esta clase...



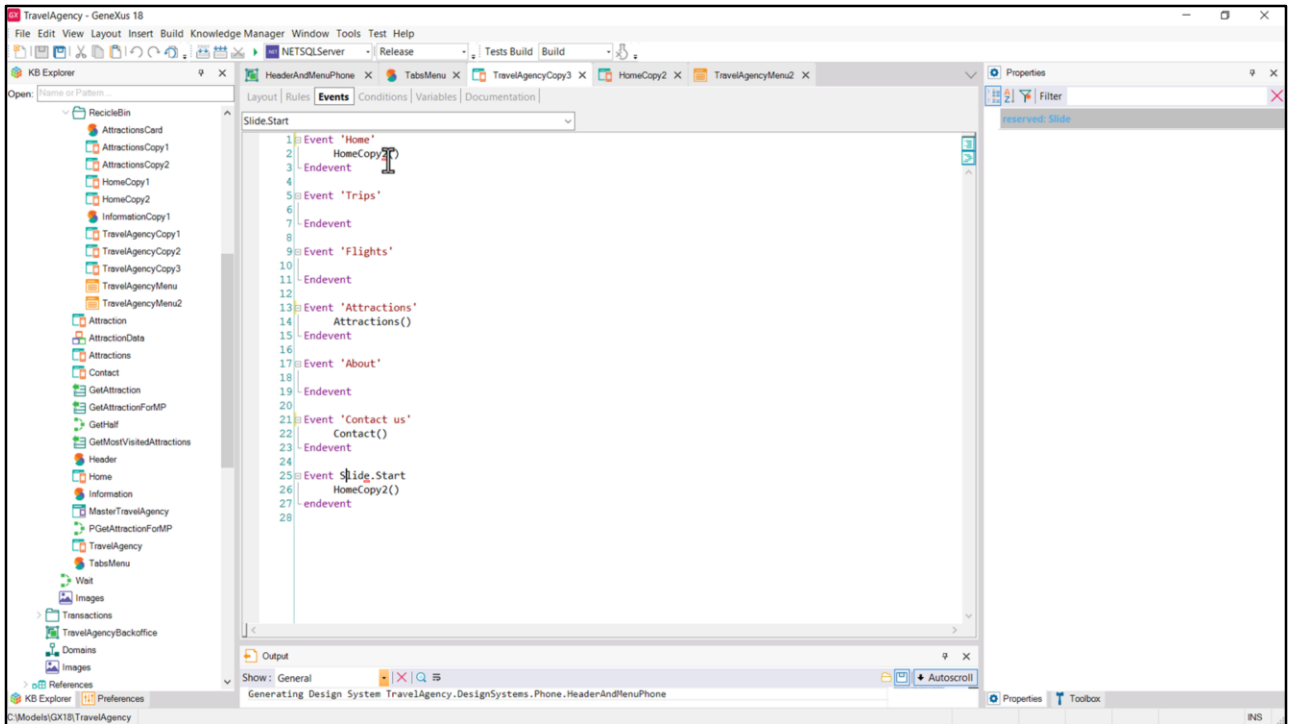
...donde asigno el color de fondo que tomé también de Figma... y luego lo que tendría que hacer es insertar el Stencil en todos los paneles.



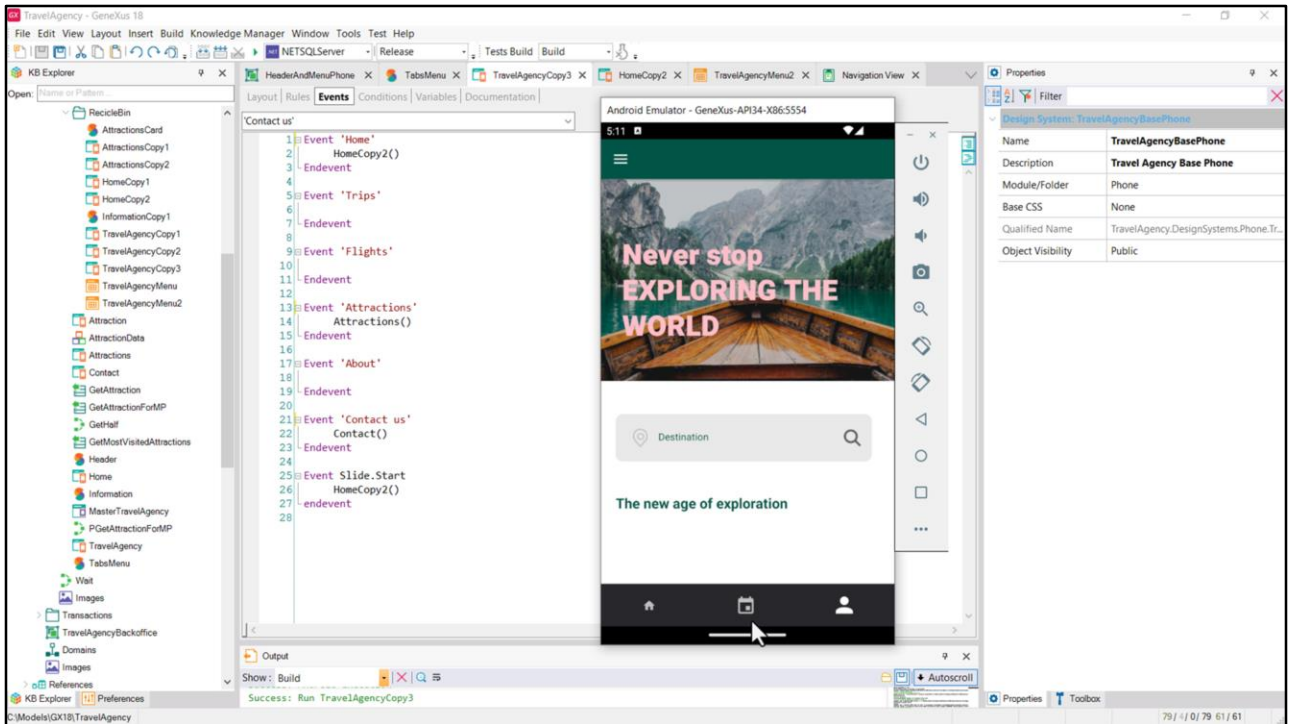
Aquí probé sólo con esta copia del Home, donde vean que reacomodé los controles del layout y dejé la tabla main con el stencil del Header en la primera fila, y el stencil con la imitación de los tabs en la última, ambas de alto fijo, y la del medio de alto variable.



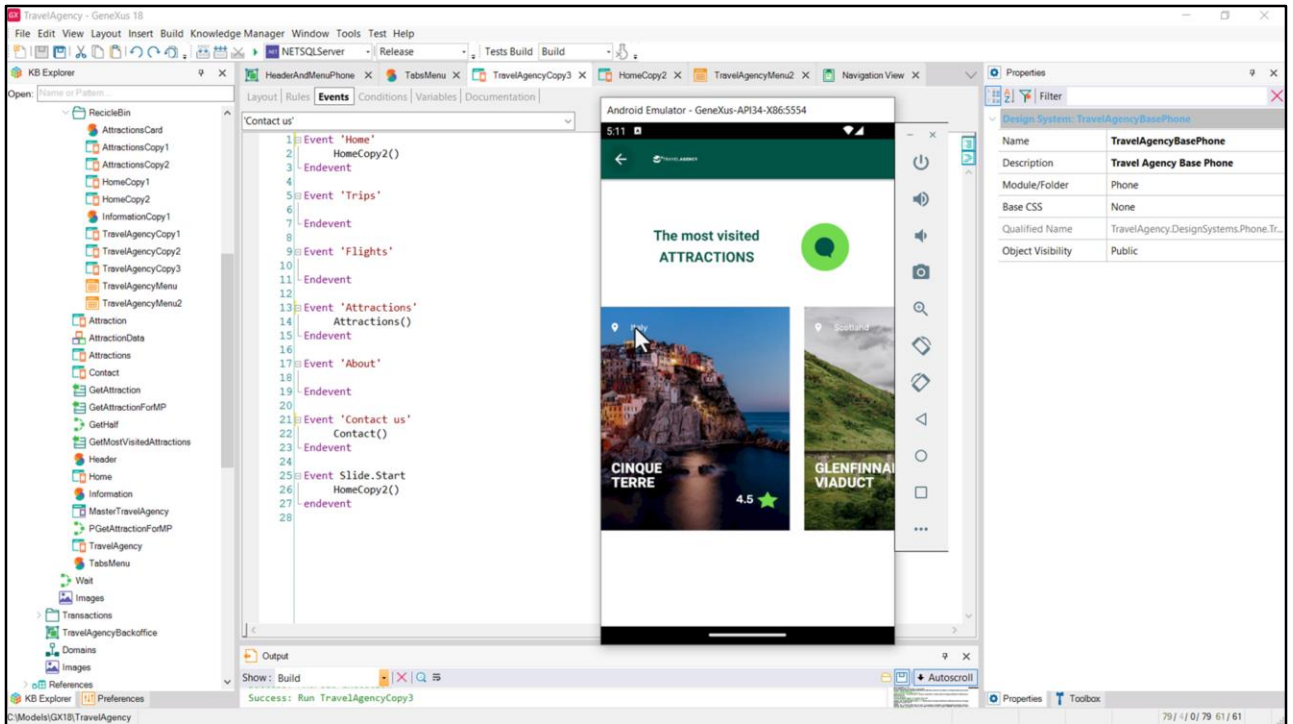
Y sólo por probar, en el evento tap asociado a cada una de las imágenes invoco a los únicos paneles que tengo por el momento.



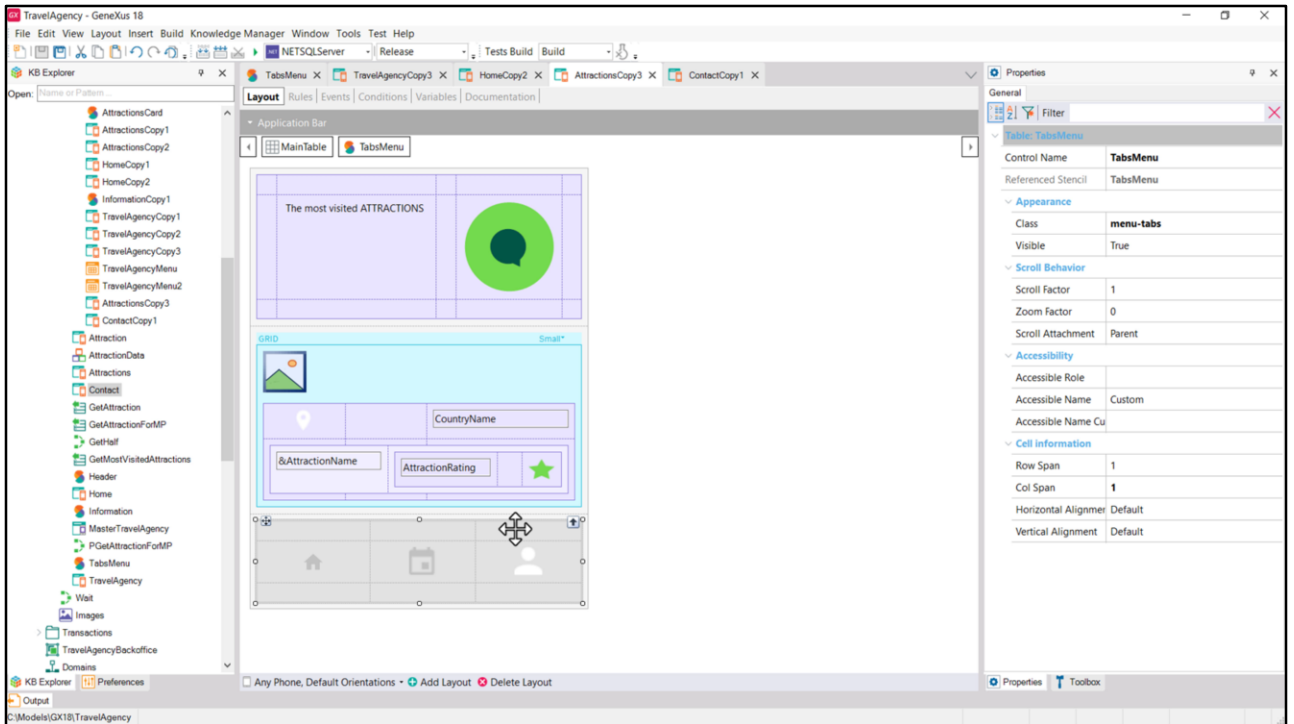
El main de mi aplicación ahora será esta copia del menú, que hará de hamburguesa, donde invoco a esta copia de Home... Y así, si probamos...



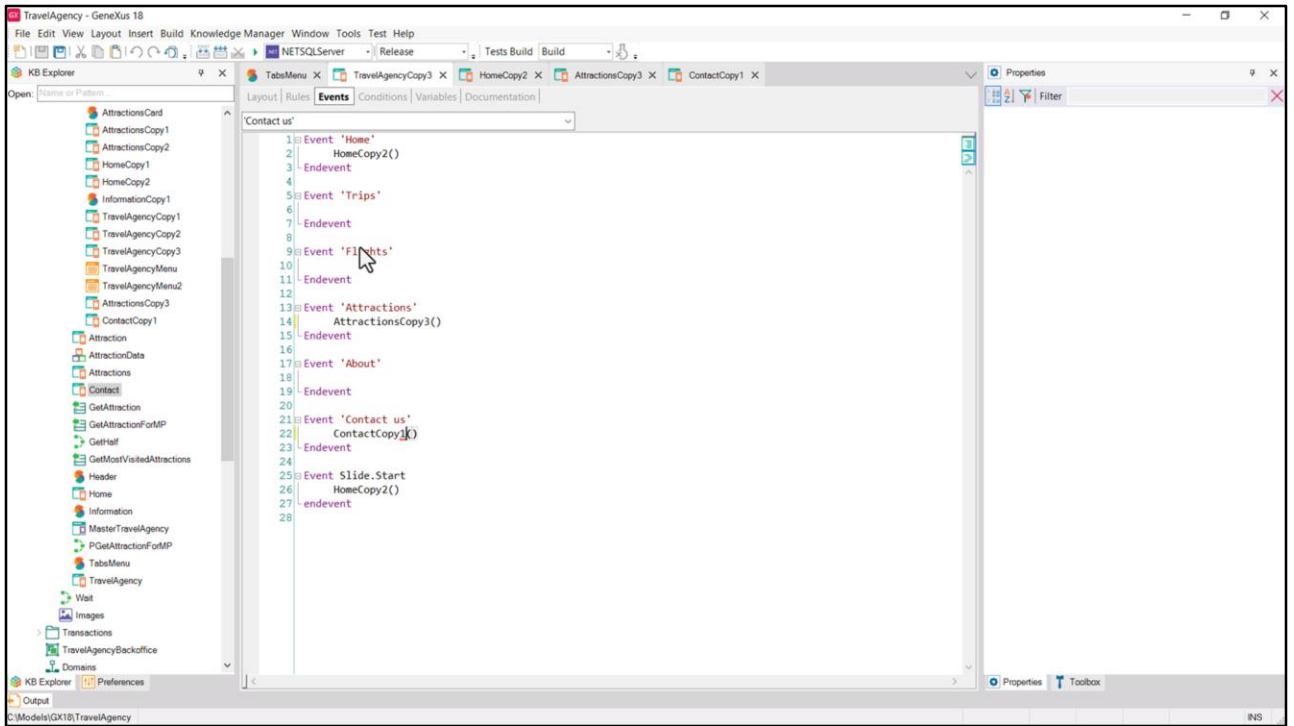
Tenemos el menú hamburguesa automático y aquí el menú de tabs manual...



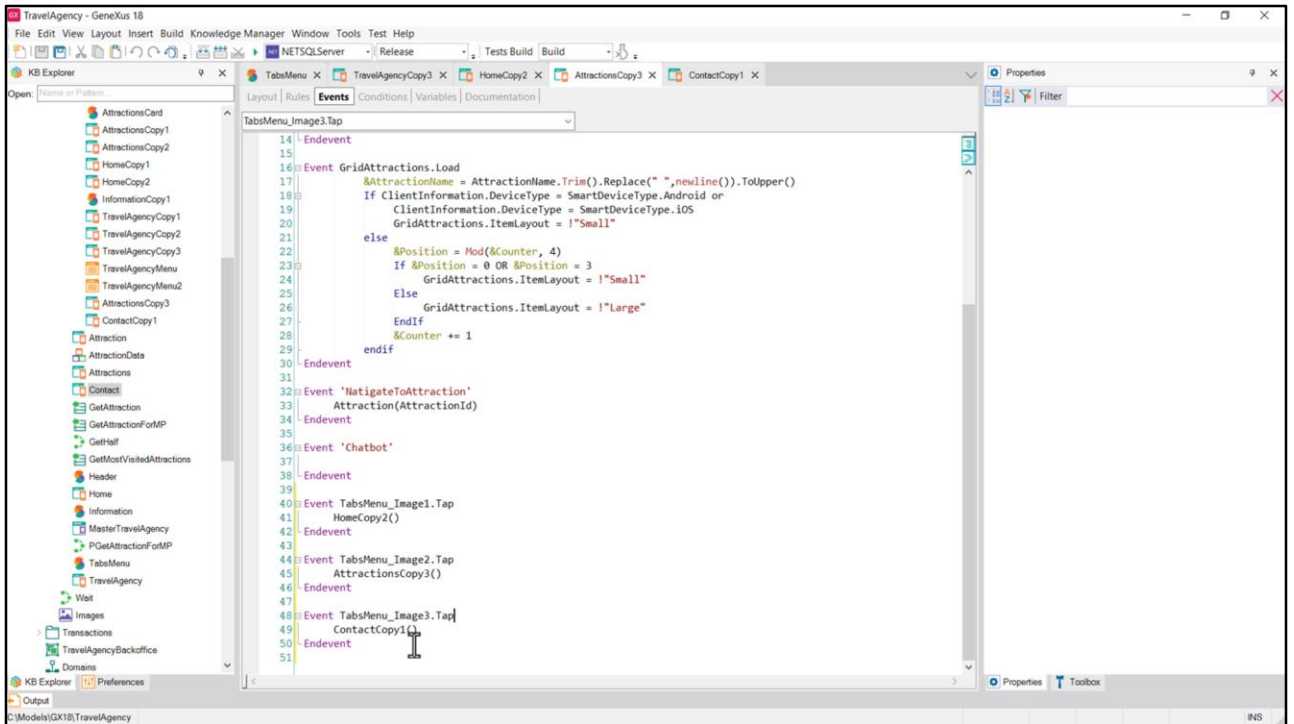
Aquí no tengo el menú porque no agregué aún el stencil al panel Attractions... ni tampoco lo agregué al de Contact.



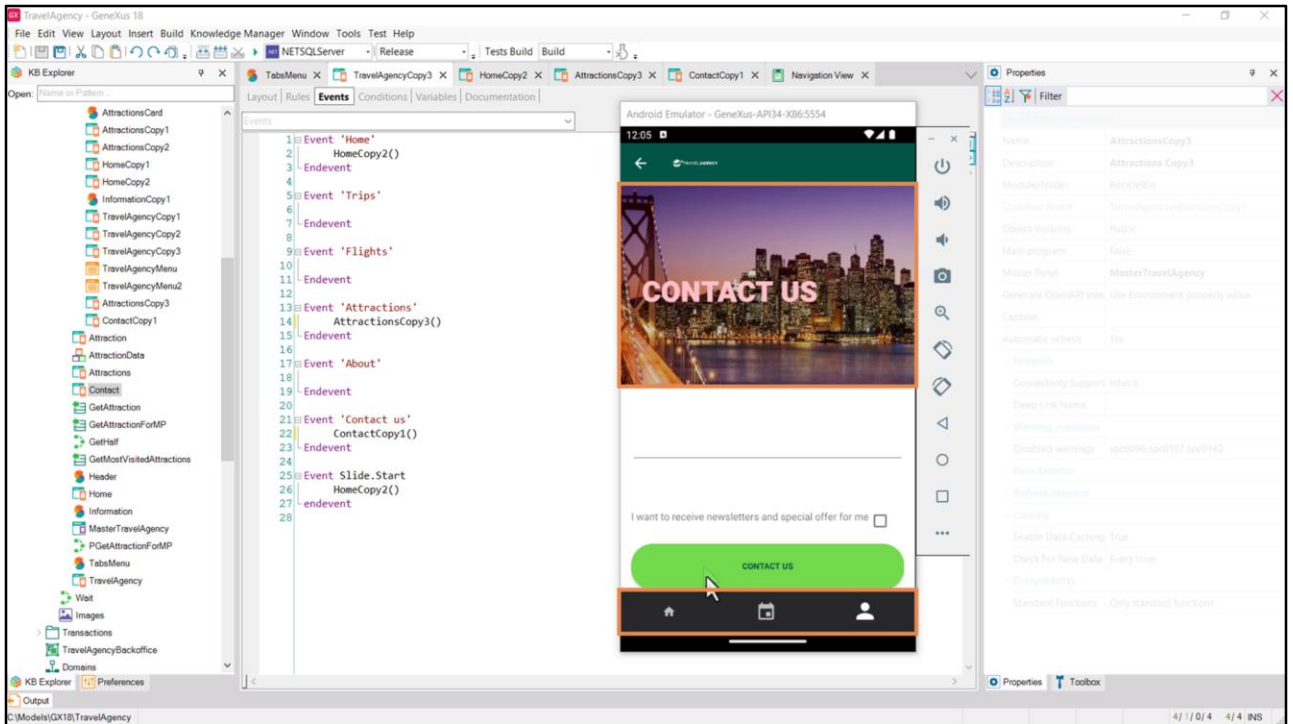
Aquí me hice copias de esos dos (para dejarles la evolución de los objetos indentificable en el xpz que les incluyo como material) y ahora sí los agregué...



...y cambié el Menú para invocarlos...

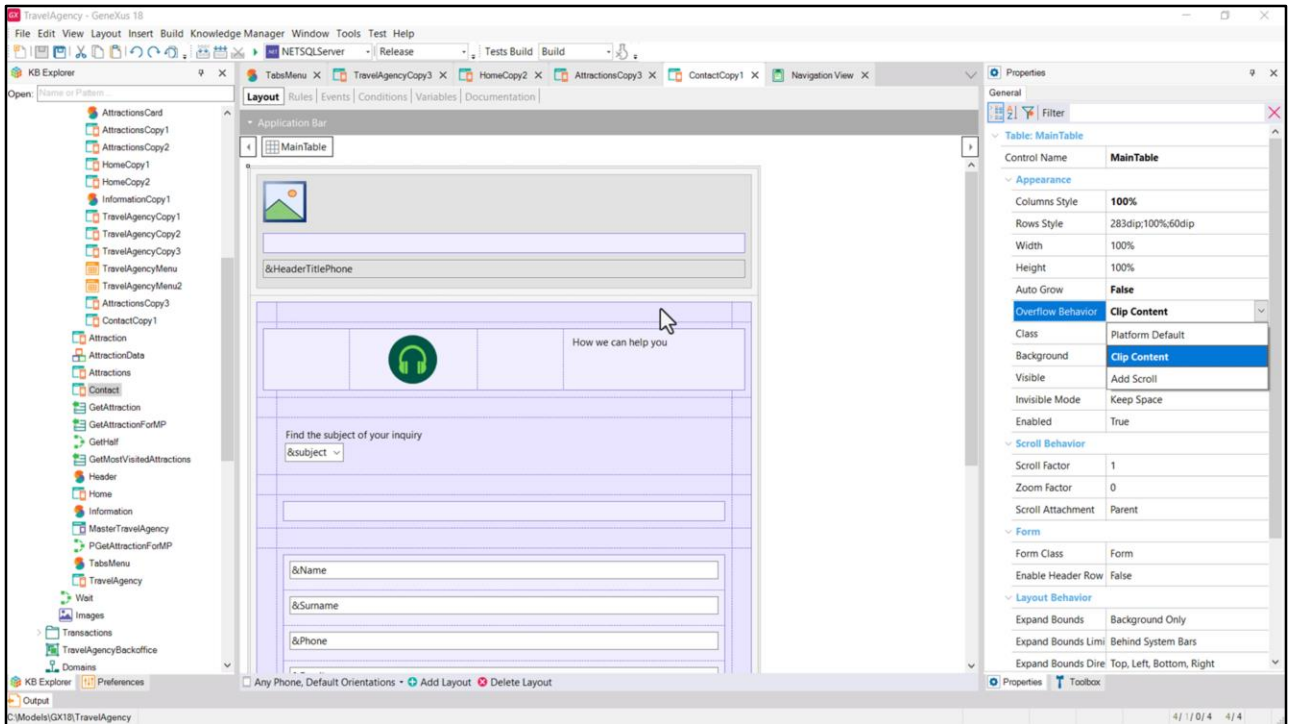


...y también desde las opciones de los tabs invoco a estos objetos.

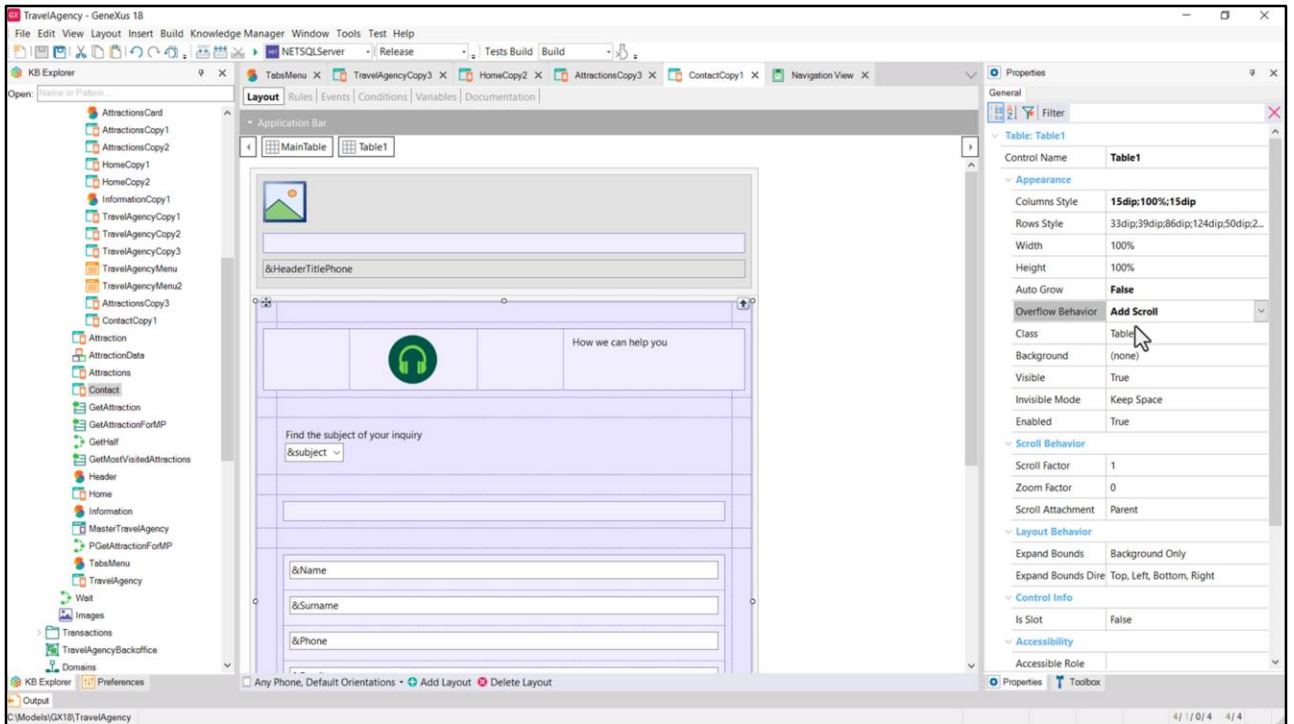


Y ahora así vemos las pantallas.

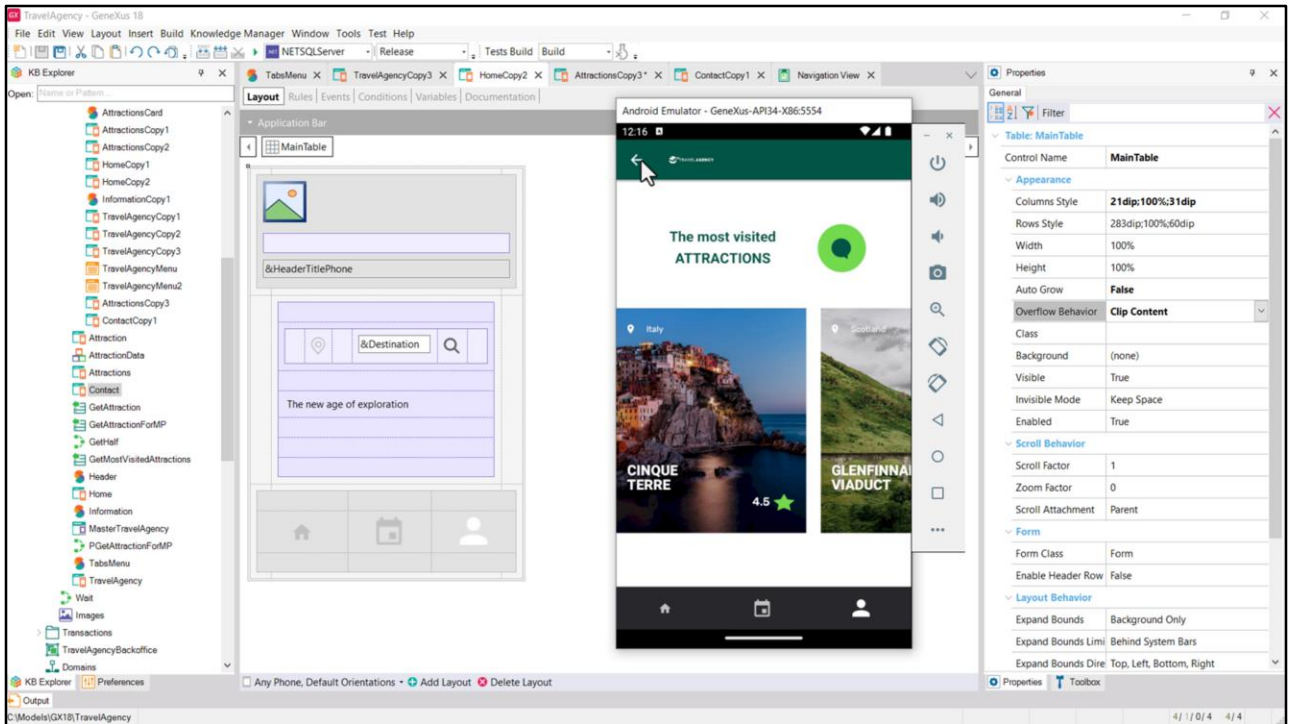
Algo que antes me había pasado desapercibido y ahora tuve que atenderlo: necesito que el Header y los tabs queden fijados arriba y abajo y el contenido scrolleable sea el del medio. ¿Cómo lo conseguí?



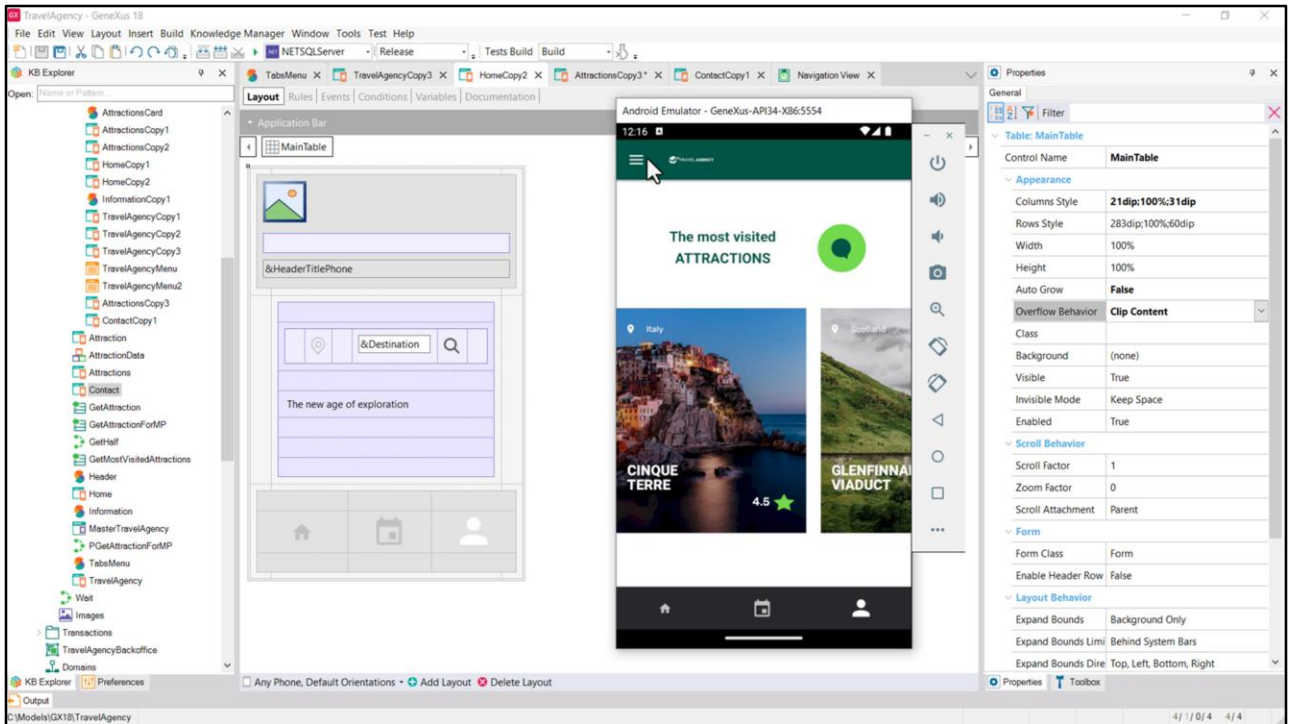
Colocando a la tabla main el Auto Grow en False y el comportamiento cuando no entra todo el contenido en la pantalla: así en lugar de scrolleable.



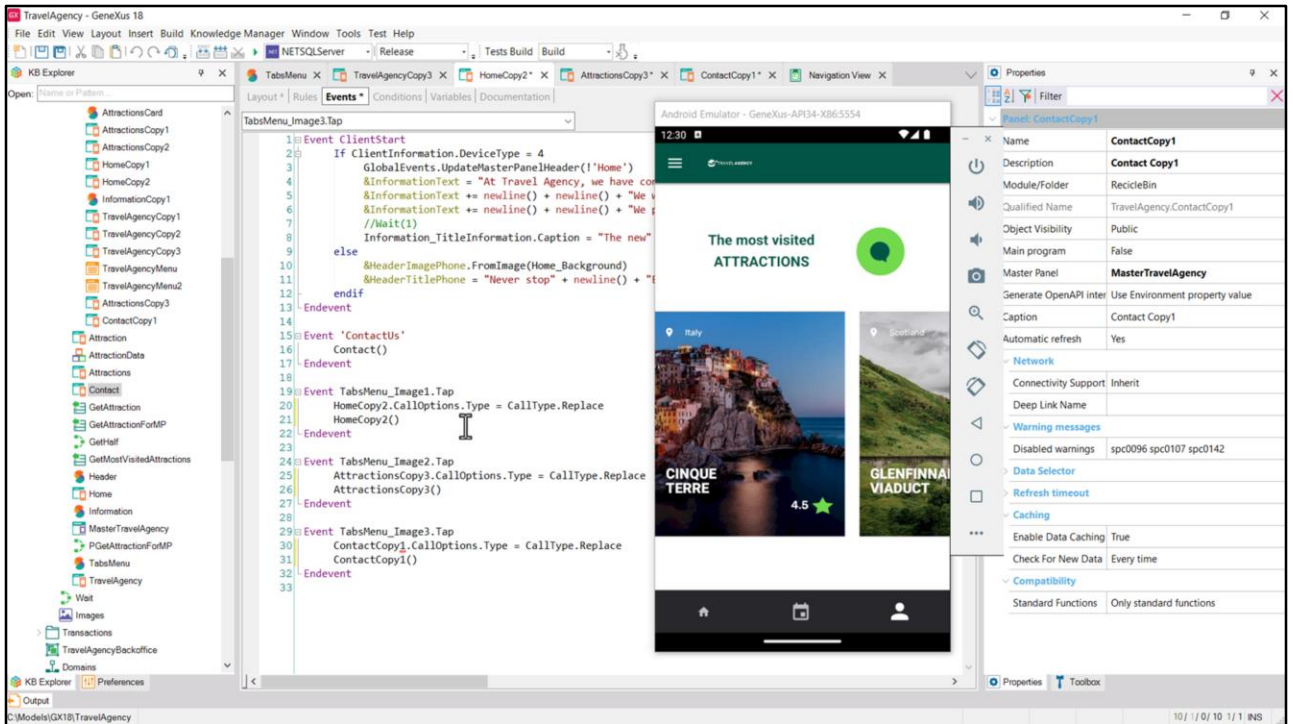
Y, en cambio, para la tabla interna colocamos el Auto Grow también en False, pero con Scroll.



Por otro lado, una consecuencia de que se trate de una imitación del menú de tabs y no del propio menú de la aplicación, es que cuando desde él se llama a un objeto, aparece el botón de back. No queremos esto, queremos que se comporte igual que el menú hamburguesa...

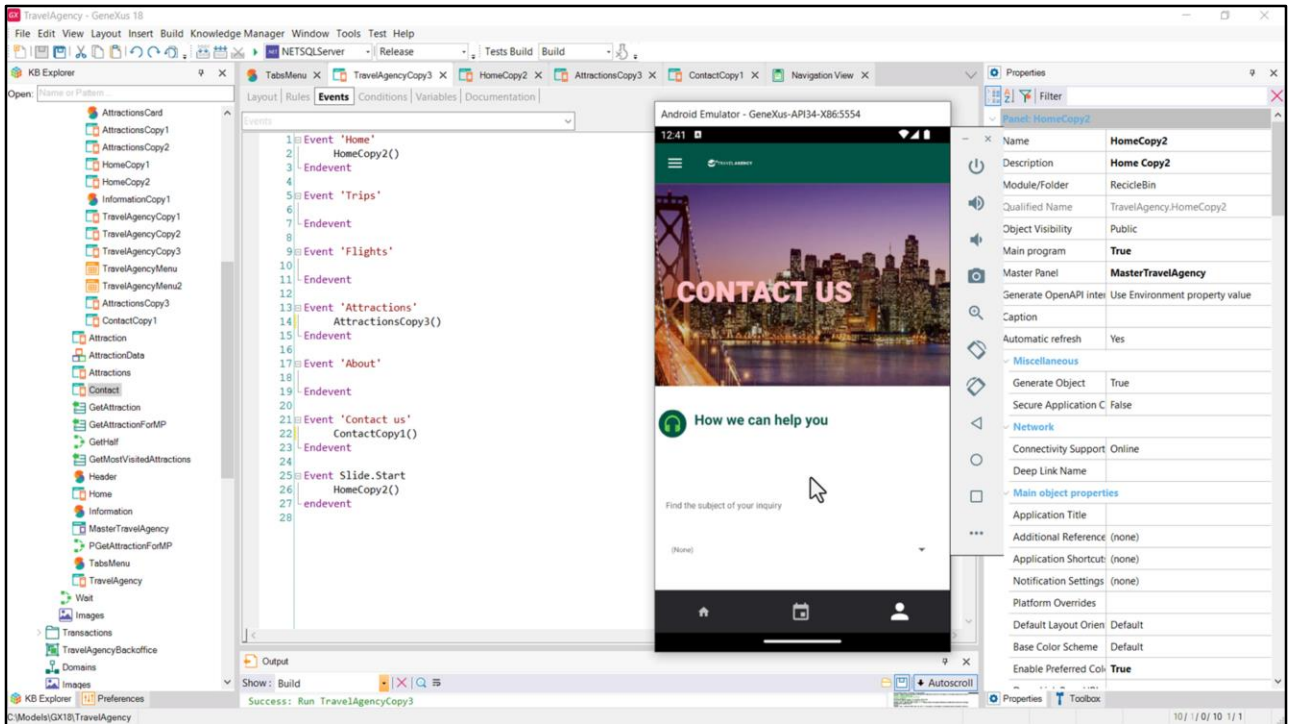


...que sí es el menú de la aplicación, y por ese motivo al invocar a objetos del menú no aparece el botón de back.



Para imitar esto invocamos con **CallOptions Replace**. Es decir, antes de invocar al objeto, le decimos que sustituya en el stack de invocaciones a actual.

Y así lo hacemos para los tres paneles.



Y así no perdemos nunca el menú hamburguesa.

Por supuesto, si nos metemos en profundidad vamos a tener el botón de back.

GeneXus for Mobile V18 Course

training.geneXus.com/en/learning/courses/geneXus-for-mobile/v18/course-geneXus-for-mobile-geneXus-18/~26080/invocations-between-mobile-objects

Multitexperience | GeneXus | DL Portal | Issues

GeneXus for Mobile course

Version: GeneXus 18

Invocations between Mobile objects

We will study the syntax of invocations to different objects, we will see the possibility of specifying in runtime the transition with which the invoked object will be opened and closed, the behavior with respect to the type of call, and the size that the layout of the object invoked will occupy in the screen using Call Options.

The diagram illustrates the state of a mobile application during an invocation. On the left, a flow shows 'Object A' calling 'Object B'. Below this, an event handler for 'NewEvent' is shown with the code: `B.CallOptions.Type = CallType.Push` and `B(%param)`. In the center, a 'STACK' is depicted as a vertical container with three slots. The bottom slot contains object 'X' (green), and the middle slot contains object 'A' (pink). A red play button icon is positioned to the left of the stack. To the right, a mobile phone screen displays object 'A'. A hand cursor points towards the stack.

Watch on YouTube

Multiple Layouts in a Grid
Grids, Types and Functionalities
Navigation styles of a mobile application

Logic and behavior

- Work With Pattern in Mobile
- Data loading logic and base tables in a Panel
- Events in a Panel
- Invocations between Mobile objects

Advanced aspects of Design

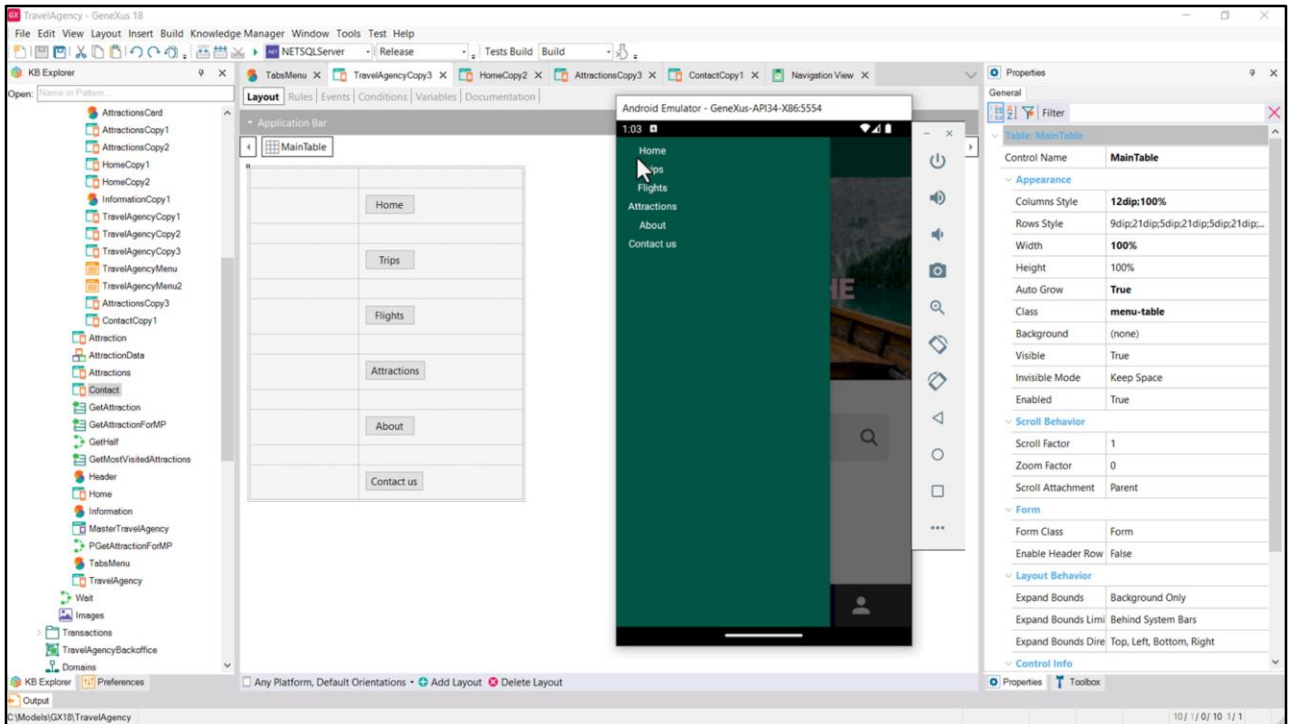
- Importing a design from Figma

Offline Applications

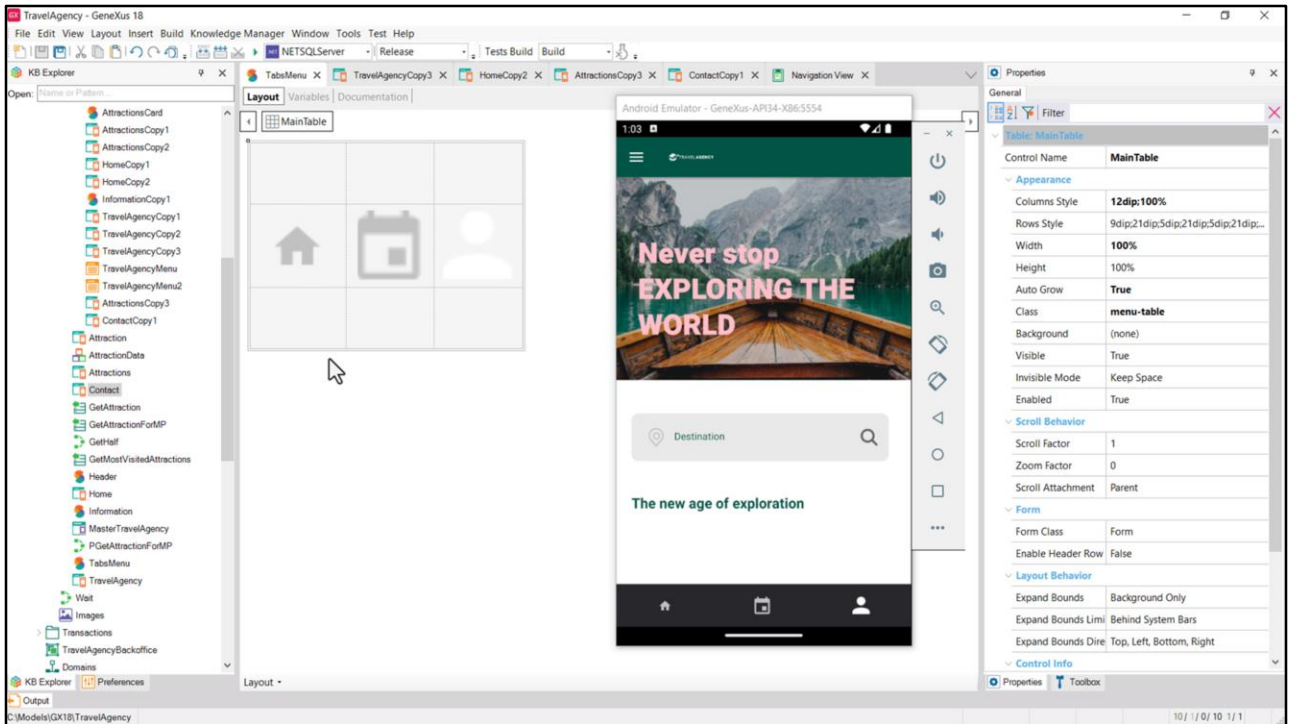
- Introduction to Offline Applications
- Generation of the Offline Database object
- Offline Database Synchronization

Total length of videos: 5h

Vuelvo a recomendar ver este video para entenderlo bien.



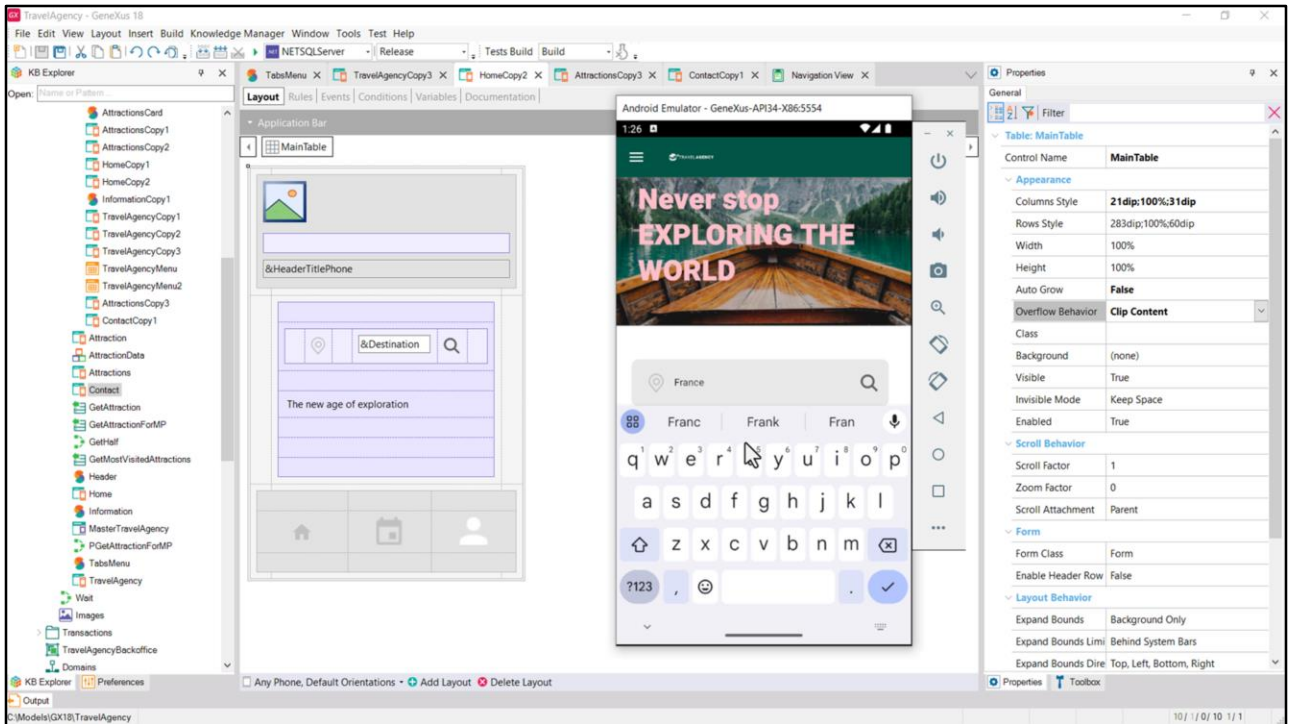
Resumiendo: para esta solución que elegimos tenemos este objeto como el punto de entrada de la aplicación, menú hamburguesa automático (por el estilo de navegación Slide)...



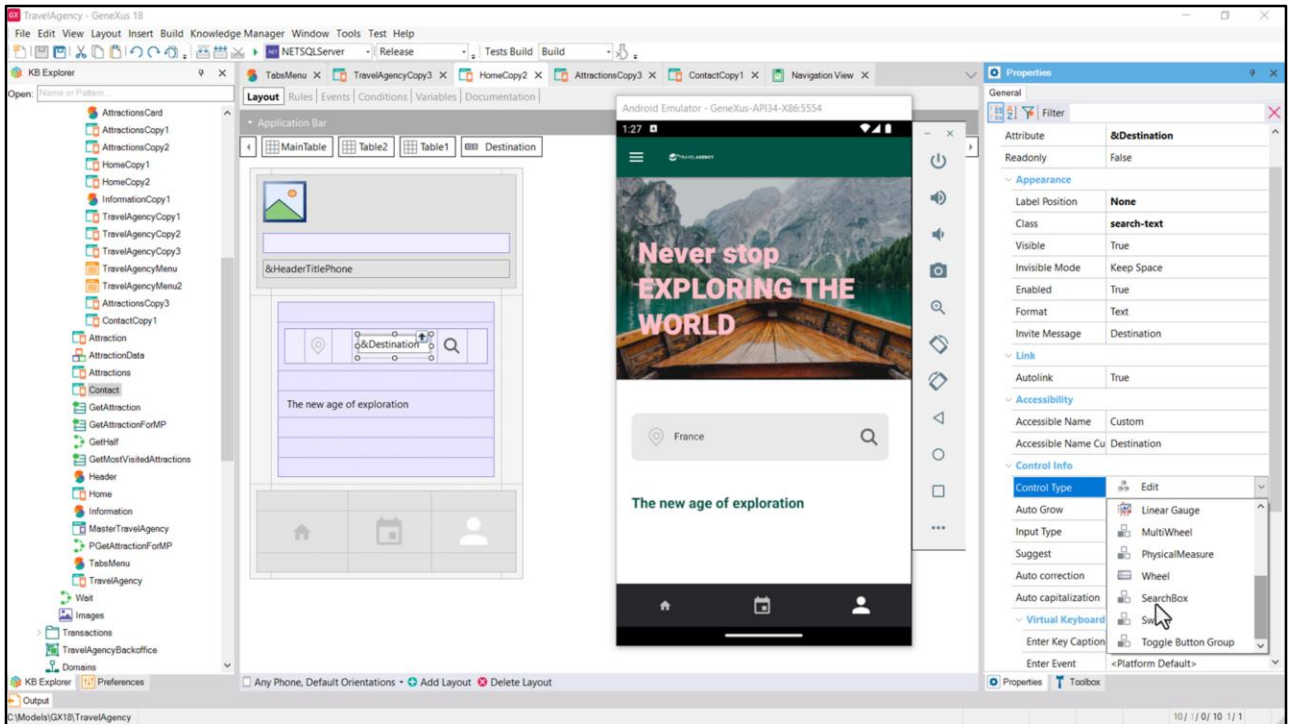
...y el menú de tabs programado y diseñado manualmente.

More about Controls

Voy a mostrarles algunas cuestiones para que las investiguen ustedes porque no quiero irme tan lejos.



Por ejemplo, vean que en este campo ingresaría el destino al que quisiera ir y esperaría que me llevara a alguna pantalla donde me mostrara su información (atracciones turísticas y demás). No tengo implementada esa pantalla, pero lo que les quiero mostrar es que en GeneXus tenemos incorporada esa funcionalidad.



Es a través del Control Type del control, eligiendo Search-Box.

The screenshot shows a web browser window displaying a Wiki page. The browser's address bar shows the URL: `wiki.genexus.com/commwiki/wiki?31862.HowTo%3A+In-app+search+in+Native+Mobile+applications`. The page header includes the GeneXus logo (by Globant), a search icon, and buttons for 'Sign up' and 'Login'. Below the header, there are navigation options: 'Page Tools', 'Page Info', and 'Other document versions'. The main content area features the title 'HowTo: In-app search in Native Mobile applications' with a blue checkmark icon. Below the title, it states 'This documentation is valid for: GeneXus 18 Help, GeneXus 17 Help'. The text explains that the document covers creating in-app search in Native Mobile applications. It notes that displaying content quickly and in a friendly way is vital, and GeneXus provides components for this, including 'Search Pattern'. A section titled 'SearchBox control' describes its purpose: to improve search mechanisms by storing the last keyword. A yellow warning box states: 'Warning: When you use a search-box control, the [Enter Event property for Attribute/Variable controls](#) should not be set because the keyboard enter action will always execute a search.' Below this, a section 'Adding the control' lists step 1: 'Create a string variable in a [Panel](#) ([Character](#), [VarChar](#) or [LongVarChar](#)).' A blue 'Ask here!' button is located in the bottom right corner of the page content.

Aquí tienen esta página del wiki para investigarlo...

Mobile applications with GeneXus

training.genexus.com/en/learning/courses/genexus-for-mobile/mobile-applications-with-genexus-course-v15/define-a-custom-search-in-the-app

Multitexperience | GeneXus | DL Portal | Issues

Updated to GeneXus Upgrade 10.

Define a custom search in the app

SearchBox control

Variables

- Standard Variables
- vSearchText
- VarChar(80)

Play now!

Look for songs available

Control Info

Control Type	SearchBox
Type	Dynamic
Results Panel	SearchResult

Event TextLookFor.Tap
@vSearchText.DoSearch()
Endevent

Watch on YouTube

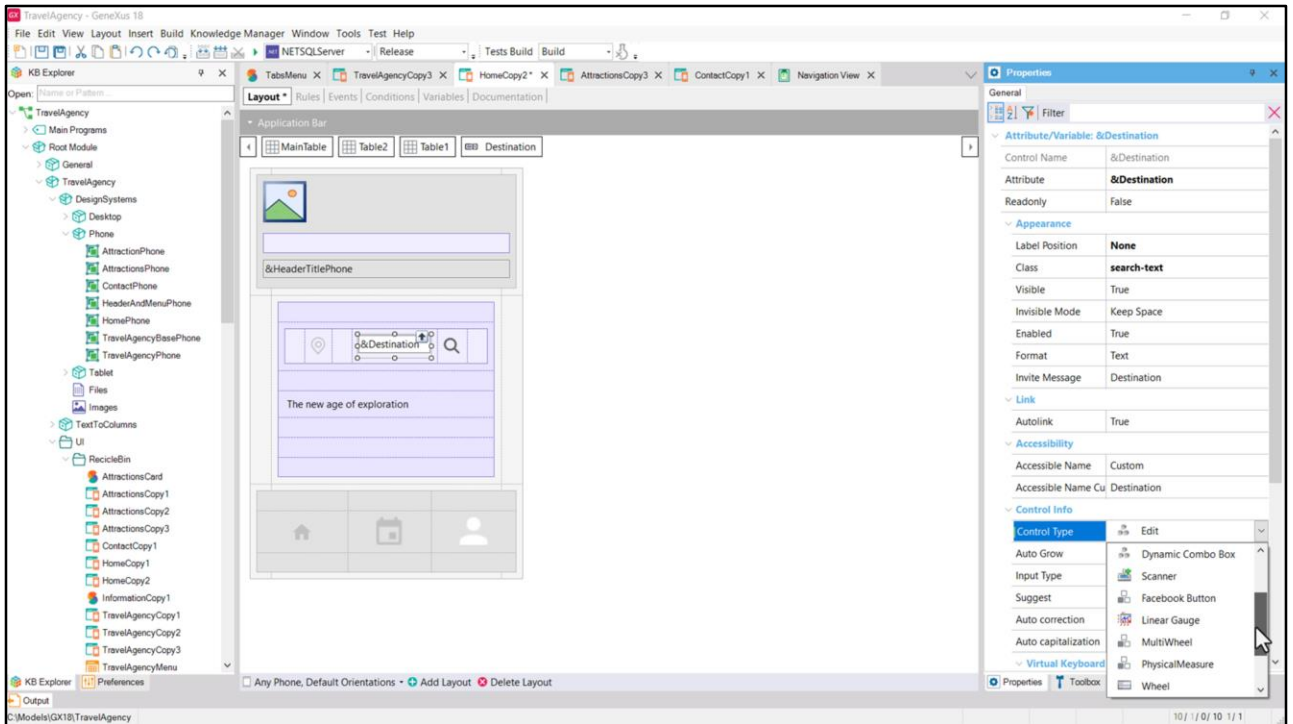
Click [here](#) for a PDF transcript of the video.

How? Control type SearchBox for string variable and DoSearch event or Results Panel property. In addition, Search external object.

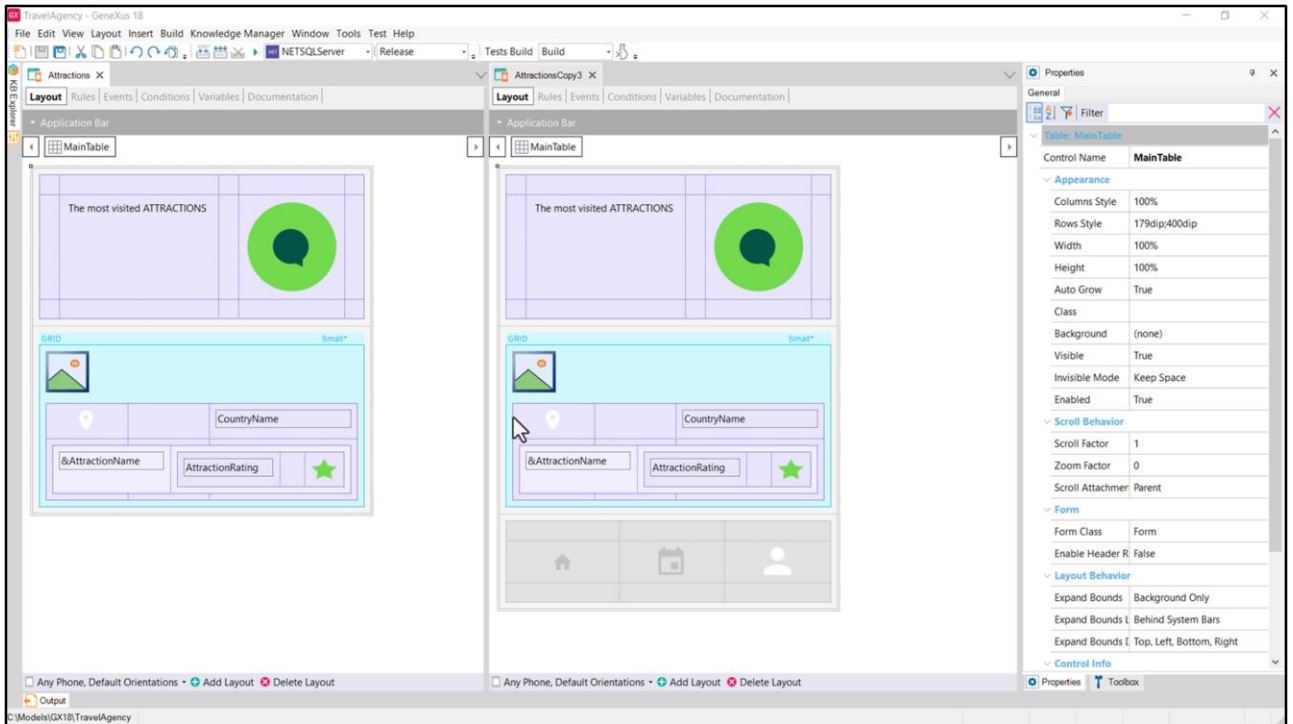
- Integrating animations provided by ARIS
- More about external objects
- Sharing content and Deep Linking
- Audio playback queues and player
- Recording audio that can be played on all platforms
- Change in the notification provider
- Grids: inverse loading and refresh from a pull
- ControlValueChanged event for att/var editable control
- Possibility to open a screen in a drop-down window on the right
- Define a custom search in the app
- Show web content on mobile app
- How to add fonts and use them in our mobile apps
- Floating position for att/var label and SD Flex Grid
- Runtime permissions
- SD Components and Global Events. New Chronometer control type
- Short list about more new features for Smart Devices

training.genexus.com/en/learning/courses/.../show-web-content-on-mobile-app

...y también tienen este video.

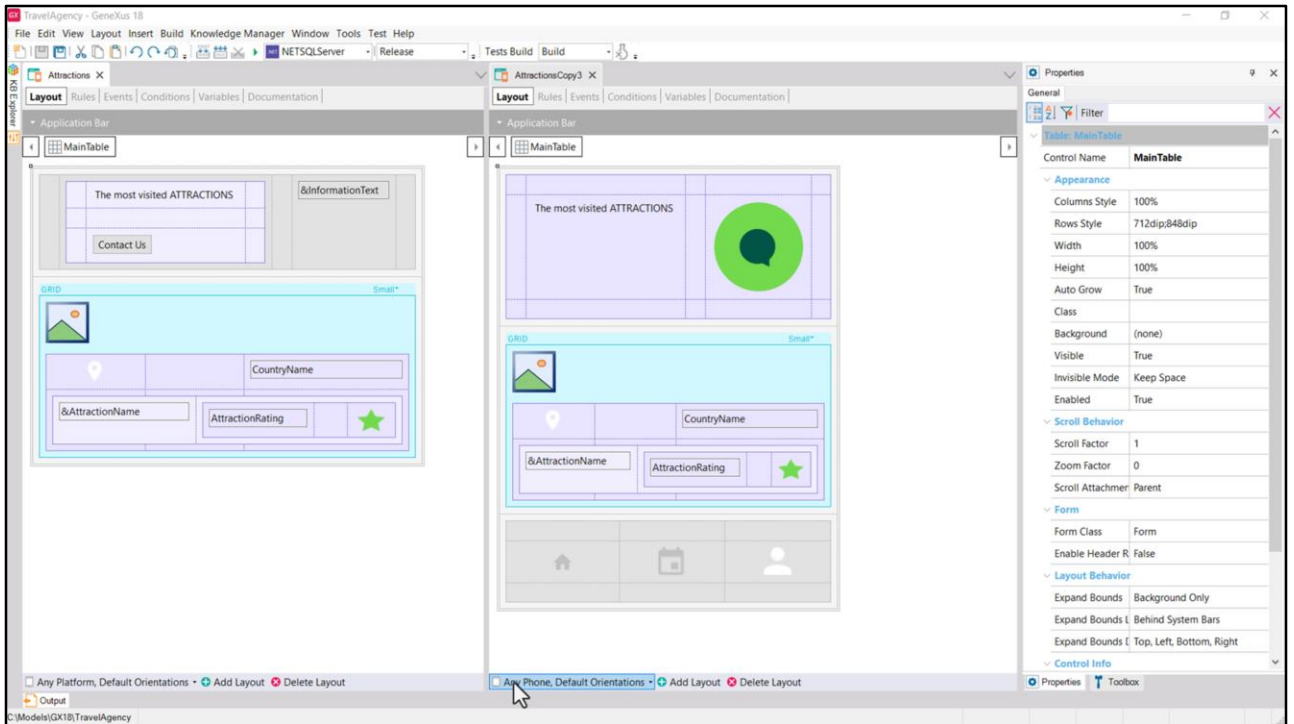


Observen que para cada control de tipo Atributo/Variable que por defecto es de tipo Edit, tienen como opciones no sólo que en su lugar se trate de un ComboBox, un Radio Button, un Check box, un combo dinámico, sino todas estas otras opciones.



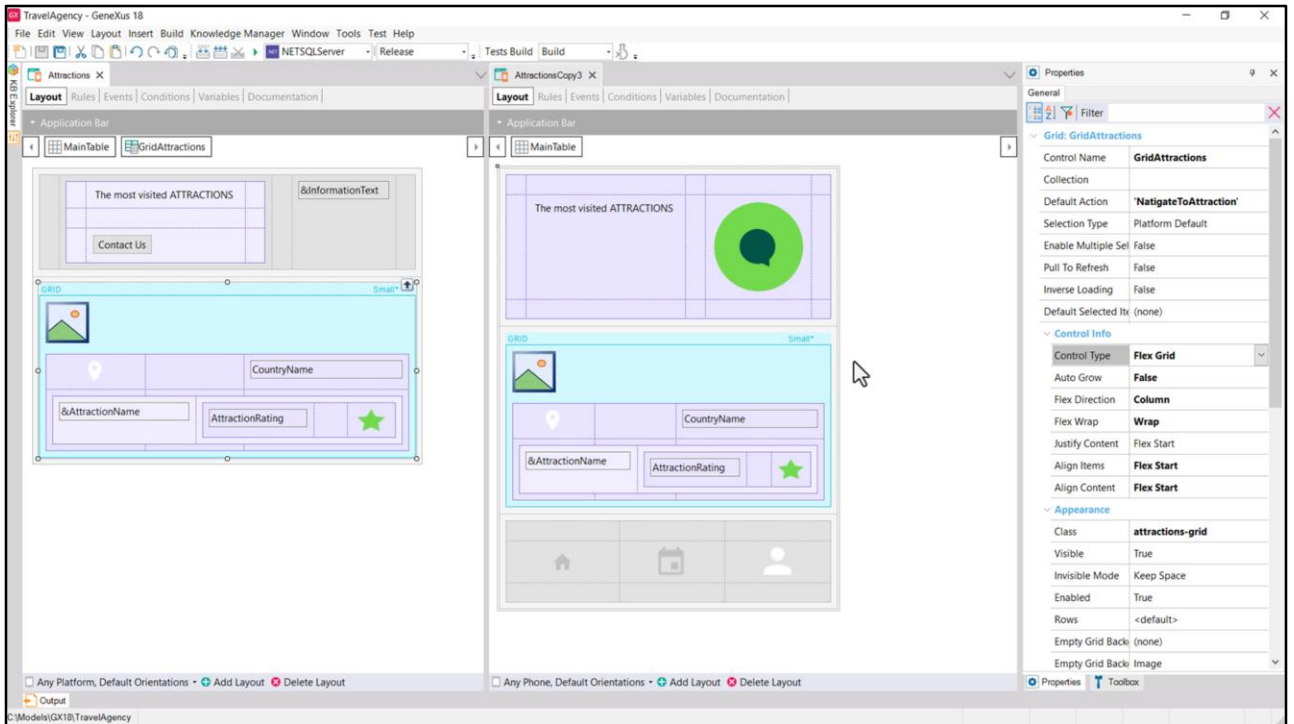
Y por otro lado no quiero dejar de mencionar el caso del Grid.

Aquí tengo el panel Attractions y aquí la copia de la que partí para agregar los tabs. El grid está idéntico en ambos para este layout.

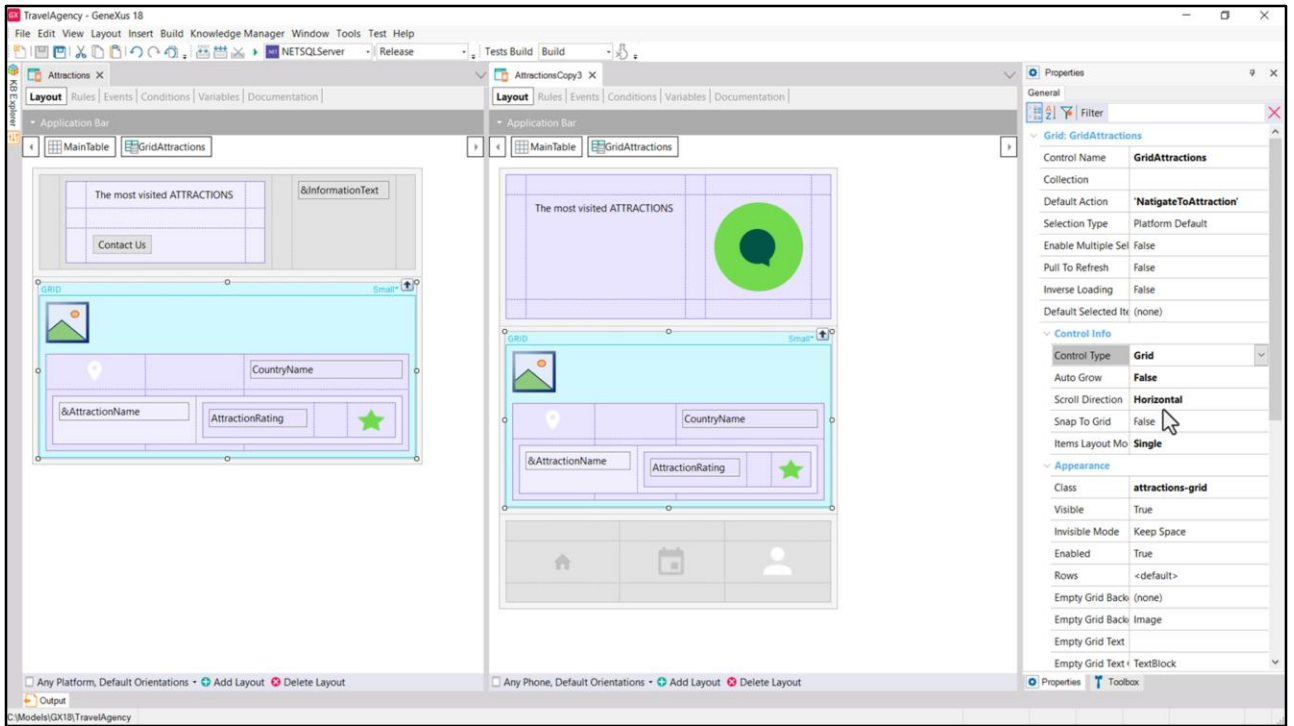


Lo que hice cuando me creé el layout para Any Phone fue inicializarlo con el layout para Any Platform, que era el que nos dedicamos a implementar en los módulos anteriores del curso, para Angular tamaño Desktop.

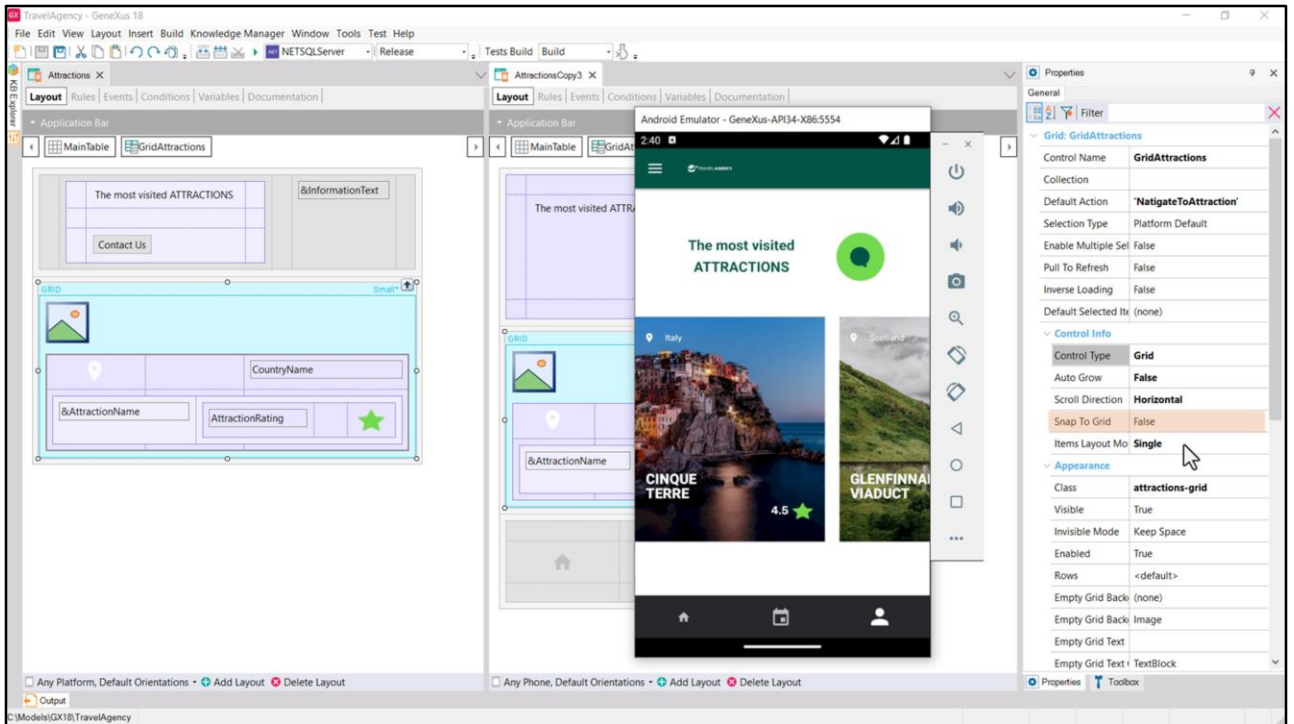
Voy a dejar de este lado ese layout, y aquí el de Any Phone, para esta copia, así los podemos comparar fácilmente.



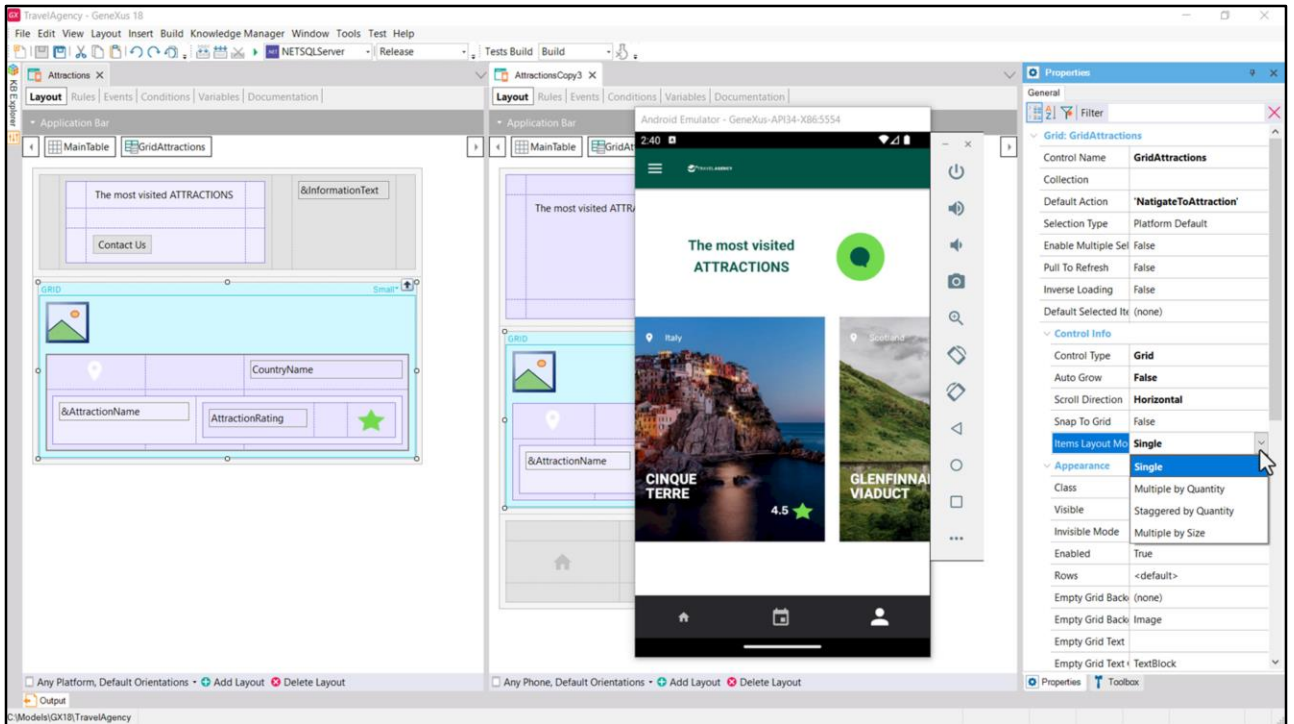
Entonces lo que hice para poder mostrarles a ustedes de manera bien rápida un panel que luciera bien para Android Phone fue simplemente cambiar el tipo de grid de Flex...



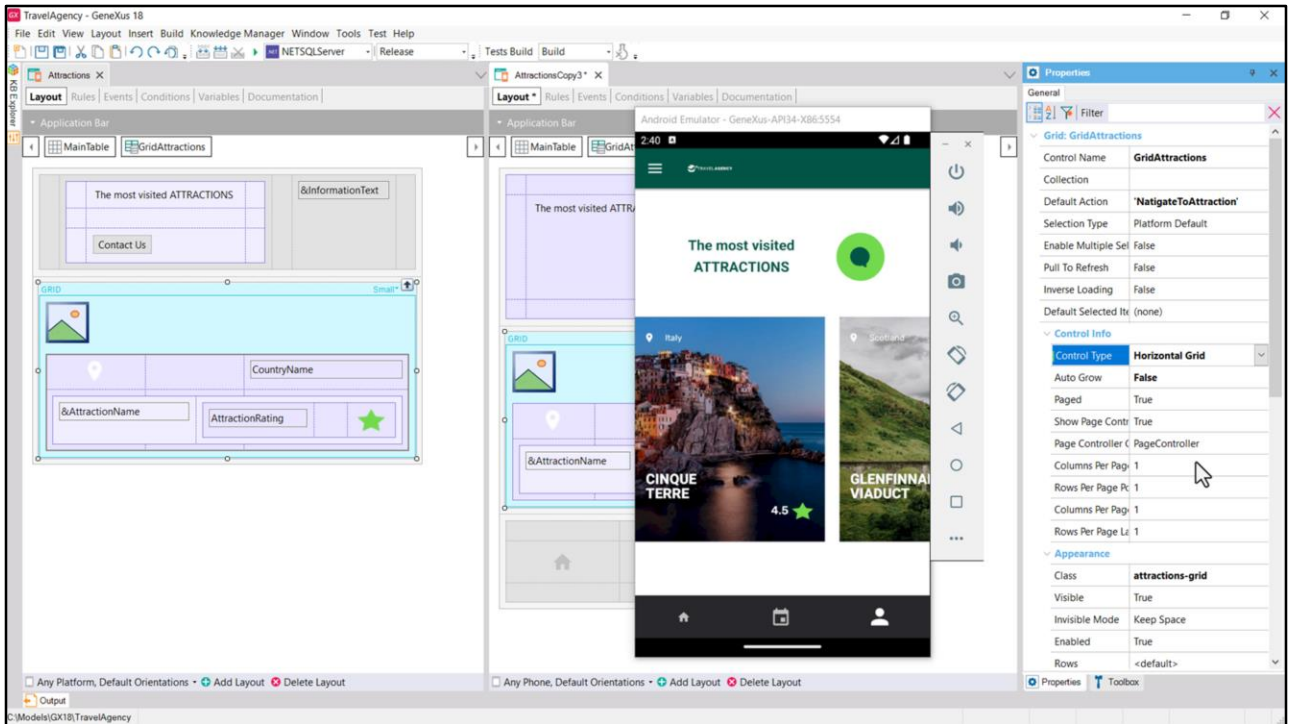
...a grid estándar pero con dirección de Scroll horizontal...



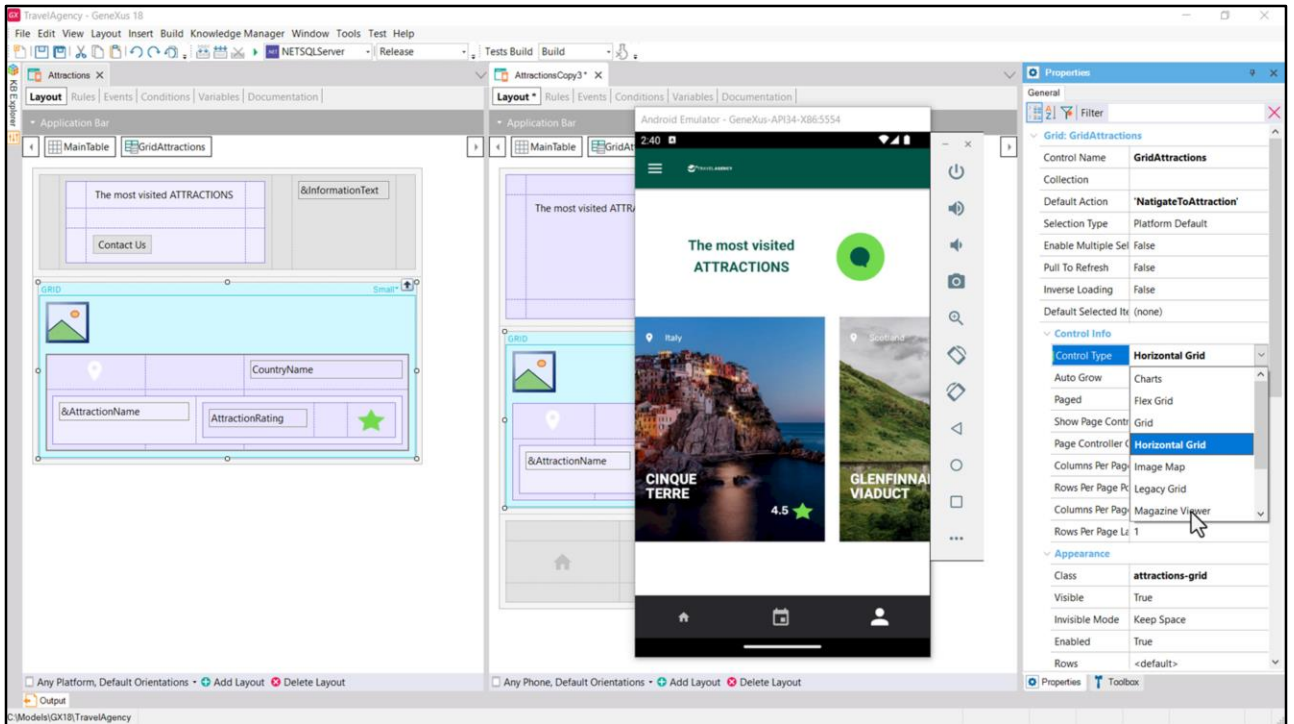
Si observan los ítems quedan fijos en el lugar en el que nos detenemos en el scroll, así queden partidos en la pantalla. Este comportamiento lo da esta propiedad.



Por otro lado vean todas estas opciones que nos ofrece la propiedad Item Layout Mode...



Y también vean que podríamos haber elegido en su lugar el Horizontal Grid.



Será bueno entender las diferencias, así como también las particularidades de todos estos otros tipos de grids, para elegir en cada caso el que mejor se adecúe a nuestras necesidades.

GeneXus for Mobile V18 Course

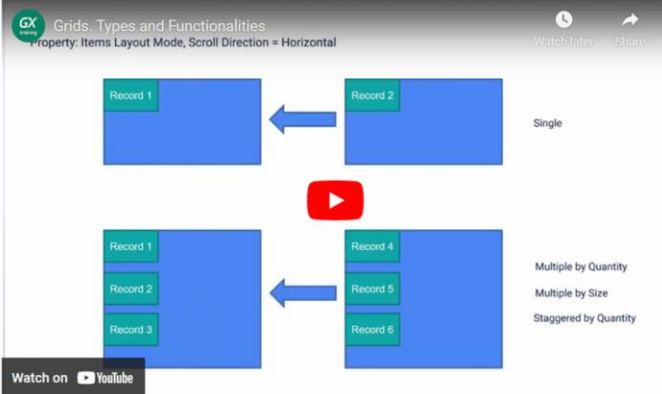
training.geneXus.com/en/learning/courses/geneXus-for-mobile/v18/course-geneXus-for-mobile-geneXus-18/~26080/grids-types-and-functionalities

Multixperience | GeneXus | DL Portal | Issues

Grids. Types and Functionalities

In this video, we will see how to configure a Grid to adopt a different behavior than the default one, as well as other types of Grids available to develop our application.

Total length of videos: 5h



Property: Items Layout Mode, Scroll Direction = Horizontal

Watch Later | Share

Record 1 | Record 2

Single

Record 1 | Record 2 | Record 3 | Record 4 | Record 5 | Record 6

Multiple by Quantity

Multiple by Size

Staggered by Quantity

Watch on YouTube

Design System of a Mobile Application

Stencil object

User Interface Components

- Basic UI Controls. Uses & Customization.
- Canvas control
- Multiple Layouts for a Panel
- Multiple Layouts in a Grid
- Grids. Types and Functionalities
- Navigation styles of a mobile application

Logic and behavior

- Work With Pattern in Mobile
- Data loading logic and base tables in a Panel
- Events in a Panel
- Invocations between Mobile objects

Advanced aspects of Design

Video transcript

Grids. Types and Functionalities - PDF

Para una primera aproximación les recomiendo este video del curso Mobile.

GeneXus for Angular Course

training.genexus.com/en/learning/courses/genexus/v18/angular/material/horizontal-grids-flexible-grids-and-control-flex-6105530

Multitaxperience | GeneXus | DL Portal | Issues

When we looked at the finished travel agency application at the beginning of this course, we could see the tourist attractions scrolling horizontally like a carousel. In this video we will see how to configure a grid to adopt this behavior, as well as other types of grids and flexible controls that we have available to develop our application.

Total length of videos: 5.5h

Horizontal Grids, Flexible Grids and Control Flex

Items Layout Mode property, Scroll direction = horizontal

Single

Multiple by Quantity
Multiple by Size
Staggered by Quantity

Watch on YouTube

Video transcript

[Horizontal Grids, Flexible Grids and Control Flex - PDF](#)

Components and Master Panel

User Interface and Experience

- Horizontal Grids, Flexible Grids and Control Flex
- Controls: Inputs, Rating, Switch
- Grouping controls and Control Tab
- More than one layout in a Panel
- Tables, Labels and Images

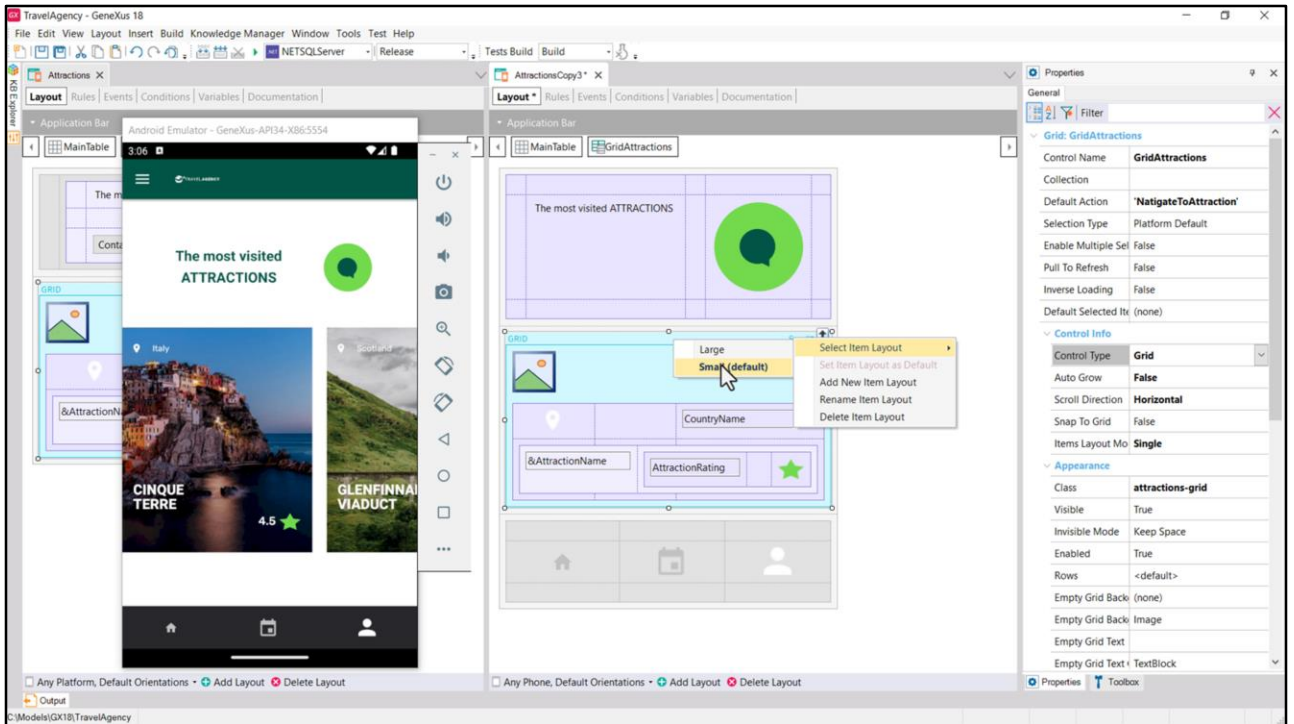
Exclusive functionalities

Exclusive features of the Angular generator

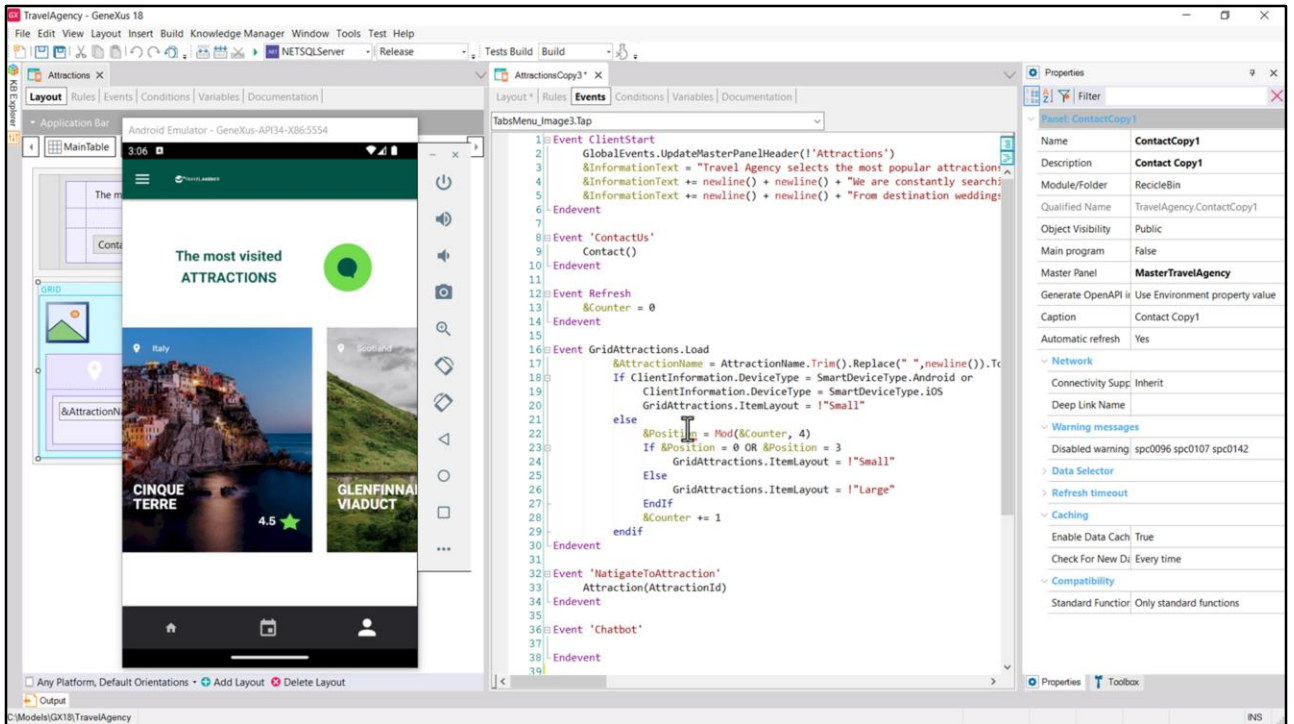
Design

- Design of an Angular application. Introduction to the Design System object
- Frontend Development Development. First steps
- GeneXus Frontend Development: First Steps - Part 2
- Design of an Angular application. Importing a design from Sketch

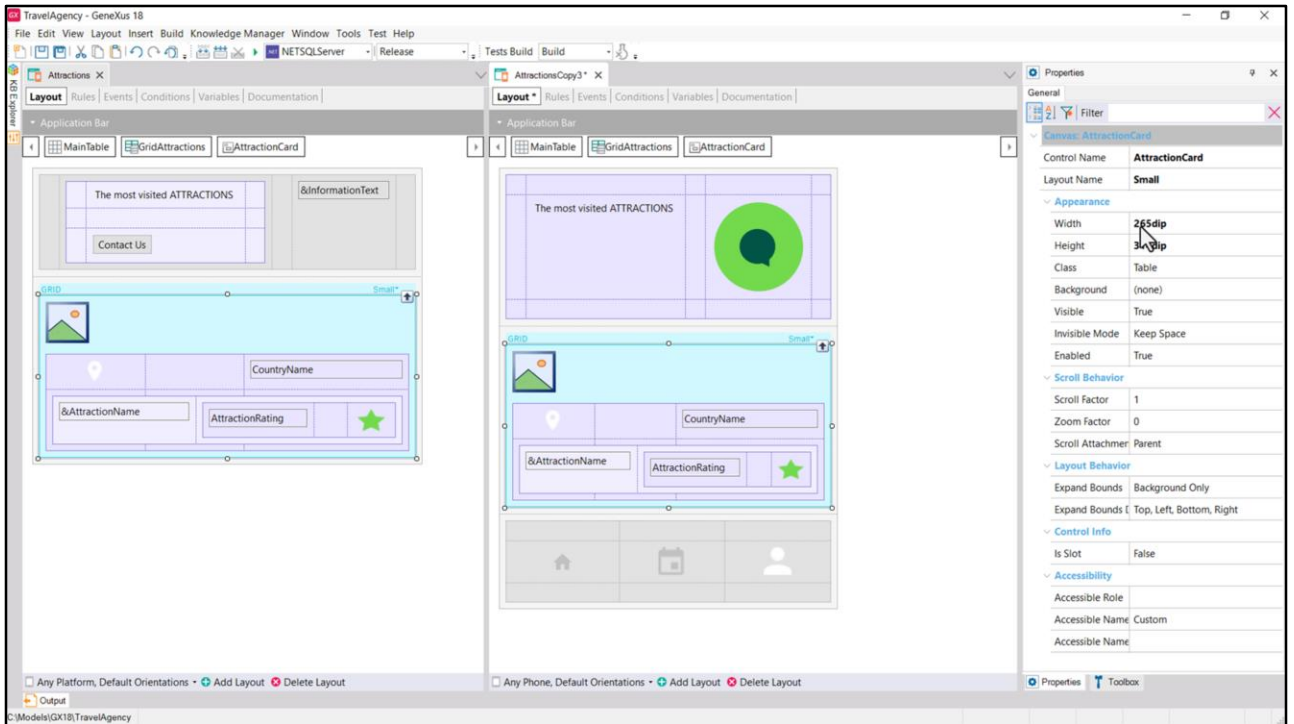
Tengan en cuenta que mucho de esto (si no todo) vale igual para Angular.



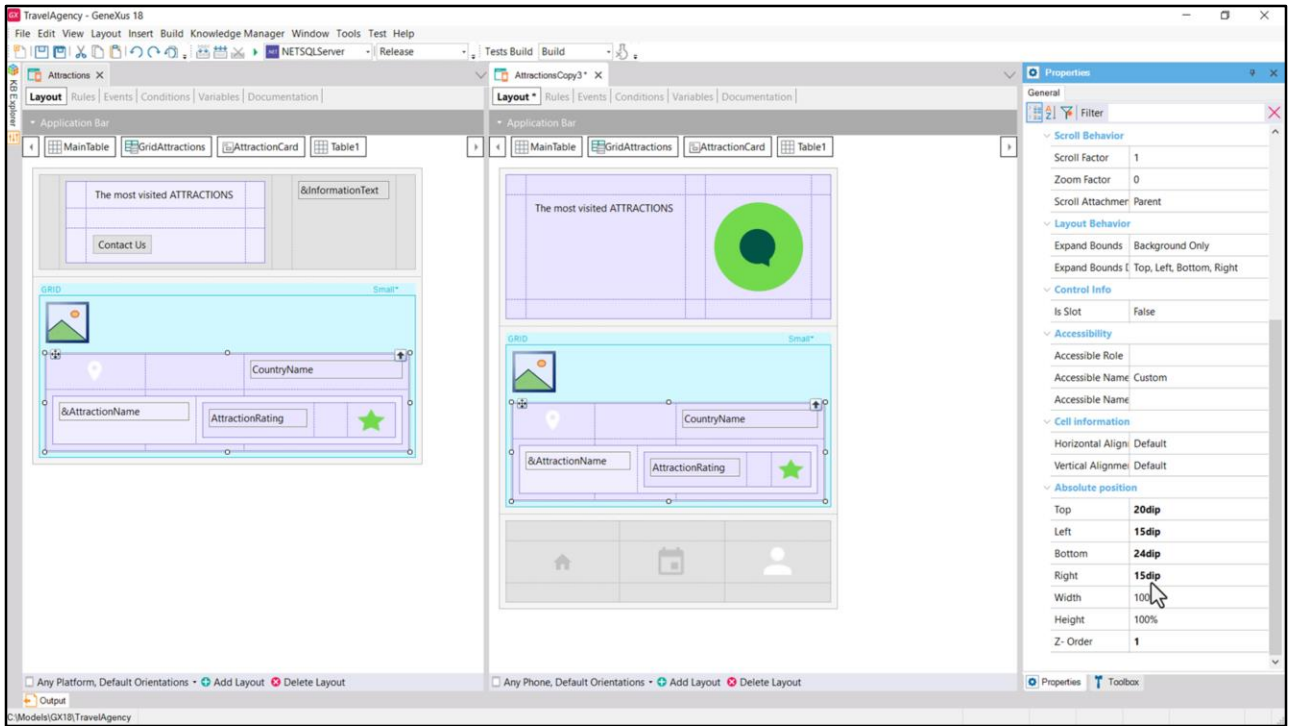
Volviendo a nuestro caso, para que la pantalla de Attractions se viera así, lo otro que hice fue utilizar únicamente el ítem layout Small...



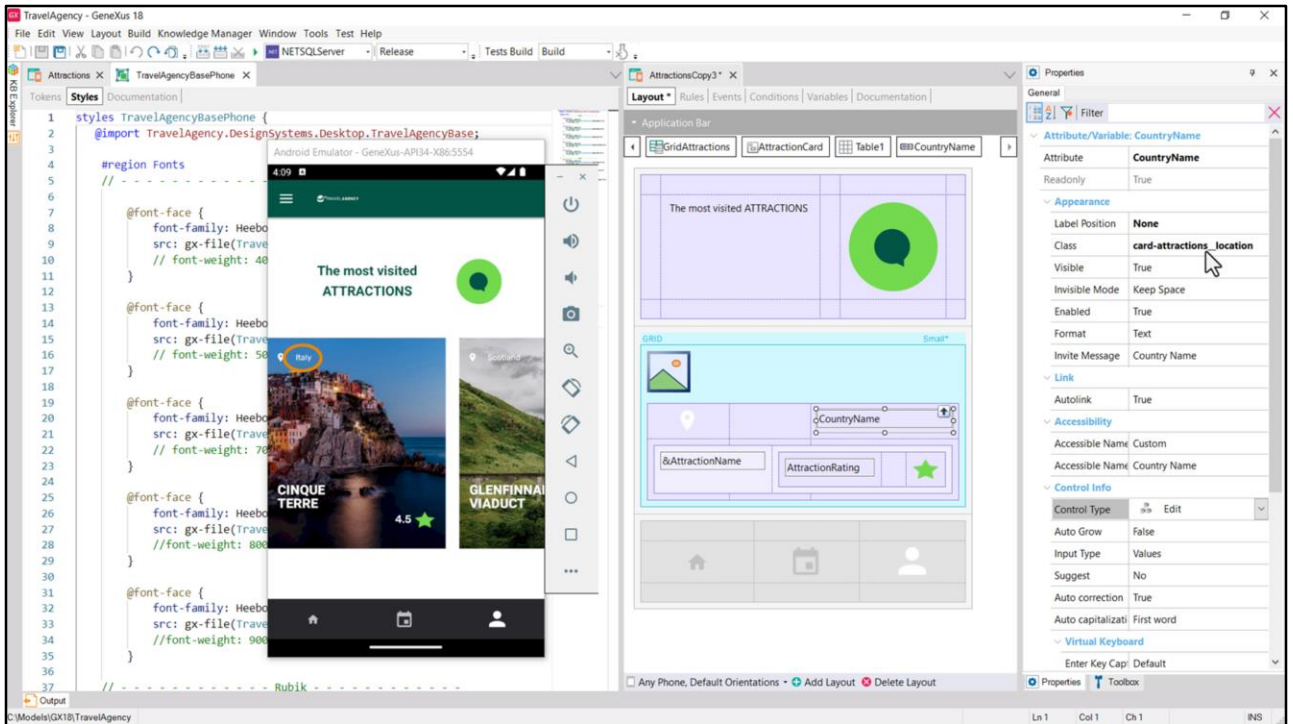
Vean que en el evento Load sólo si no es Android o Apple ejecutará el else que es el que carga ambos ítems layouts.



Y luego lo único que tuve que hacer fue ajustar el alto y ancho del canvas...



...y algunas de las distancias respecto a los bordes.

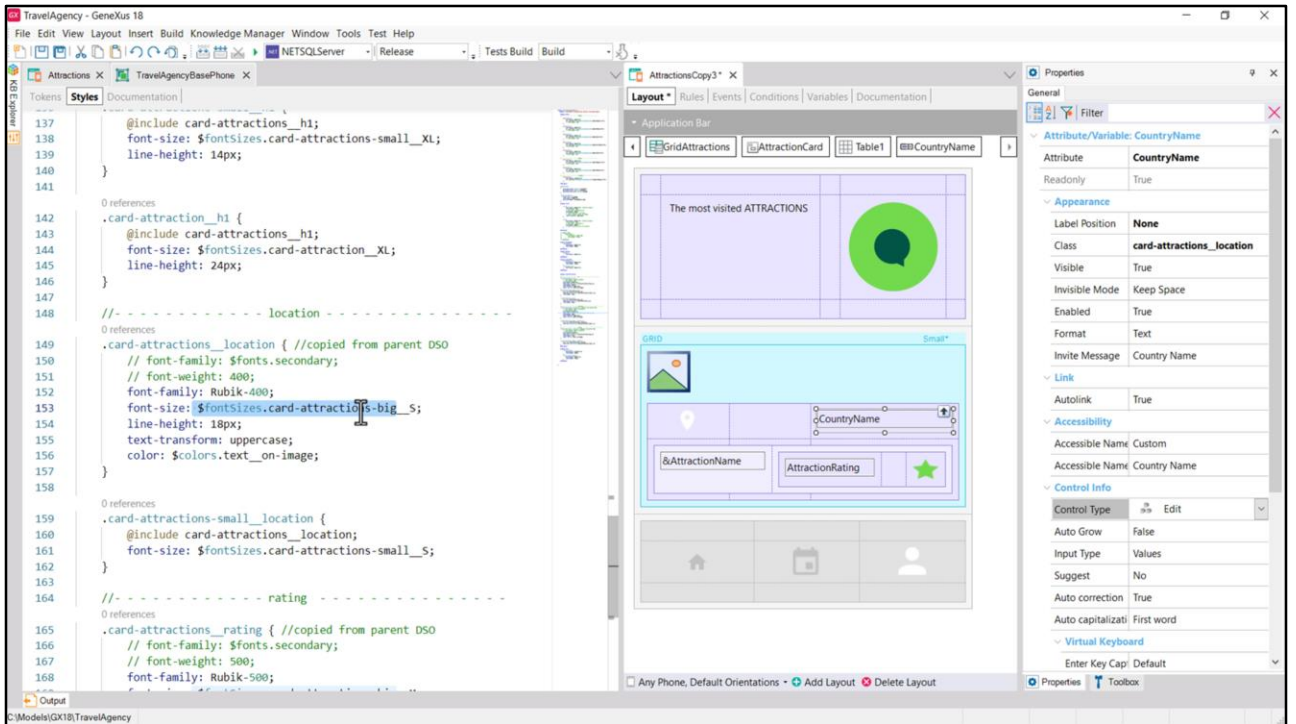


Respecto a las clases y los DSOs no tuve que hacer prácticamente más nada para que todo se viera así.

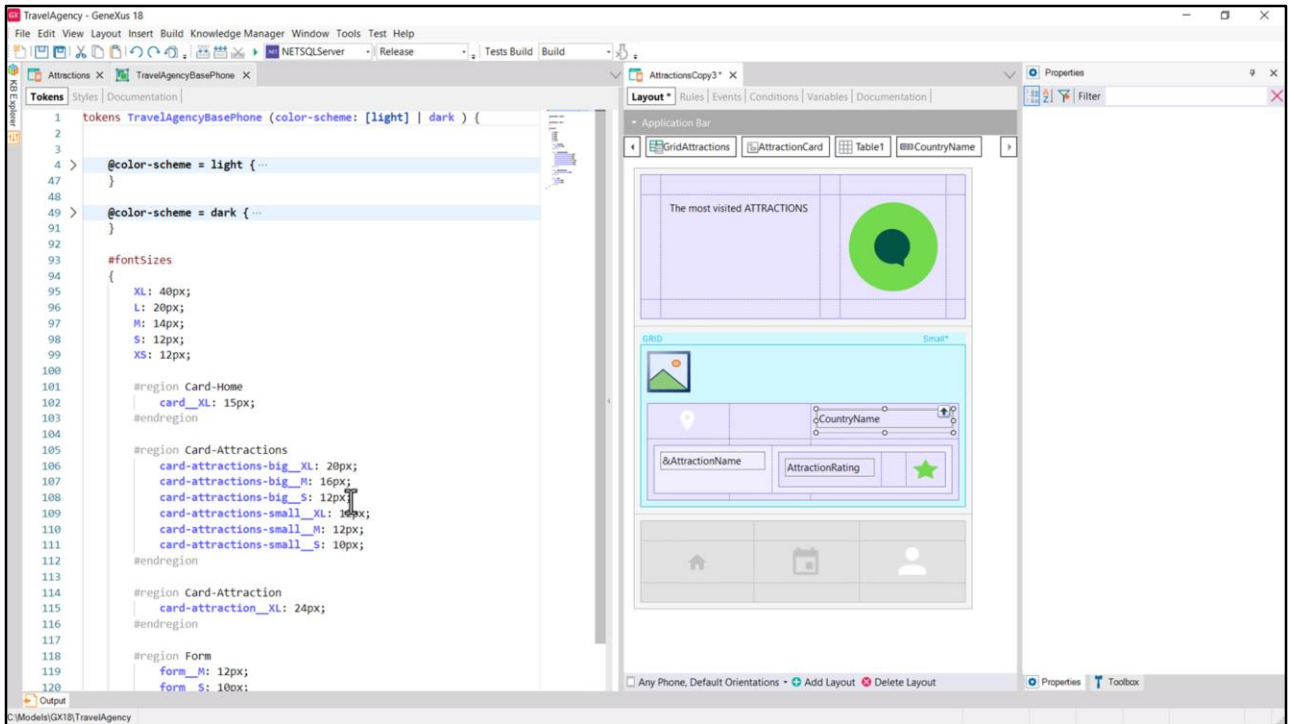
Si al principio de este módulo pudo parecer que la multiexperiencia era trabajosa, aquí resultará todo lo contrario: abro el DSO base para Phone que estuvimos trabajando, que se construyó sobre el DSO base para Angular Desktop, actualizando los tokens de tamaño de fuentes y las clases tipográficas.

Y vean cómo no tuve que prácticamente hacer más trabajo para dar estilo a la pantalla para Android.

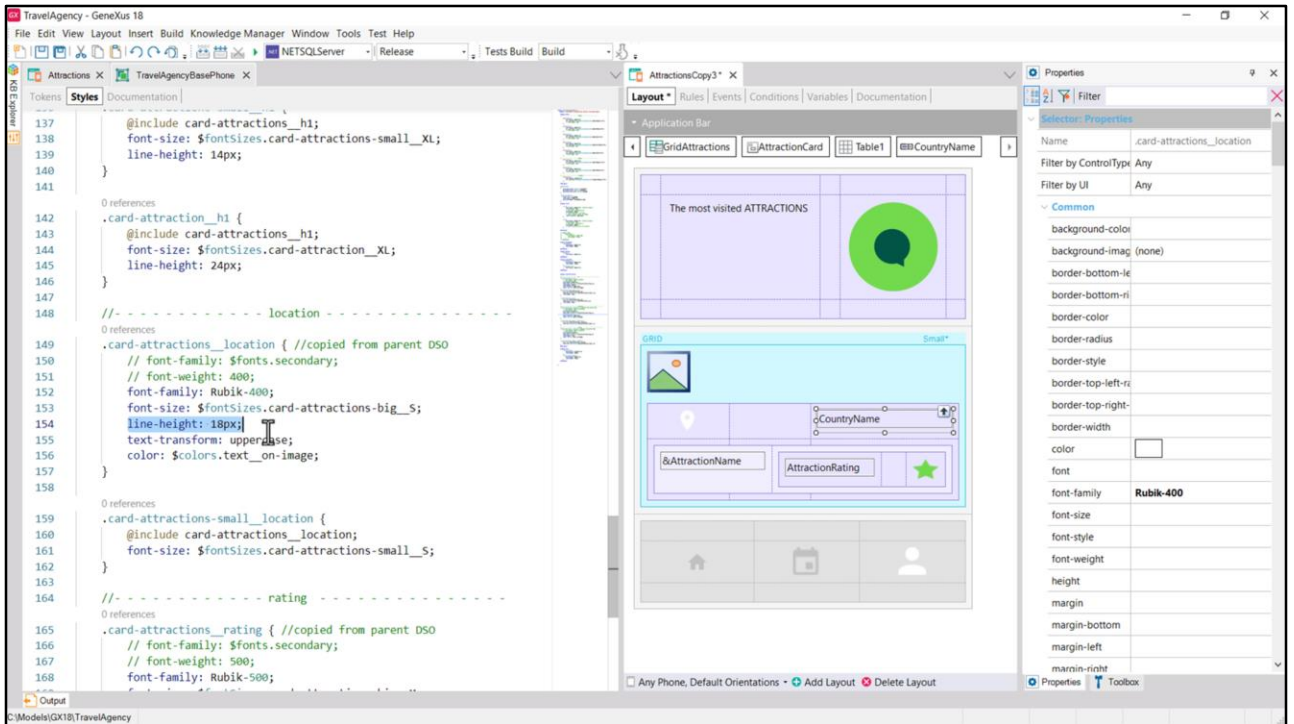
Por ejemplo, el nombre del país se está viendo así y es debido a esta clase que es la que tenía el mismo control para el layout Any Platform (el que usamos para Angular Desktop)... que no toqué para nada al crear el nuevo layout a partir de este otro...



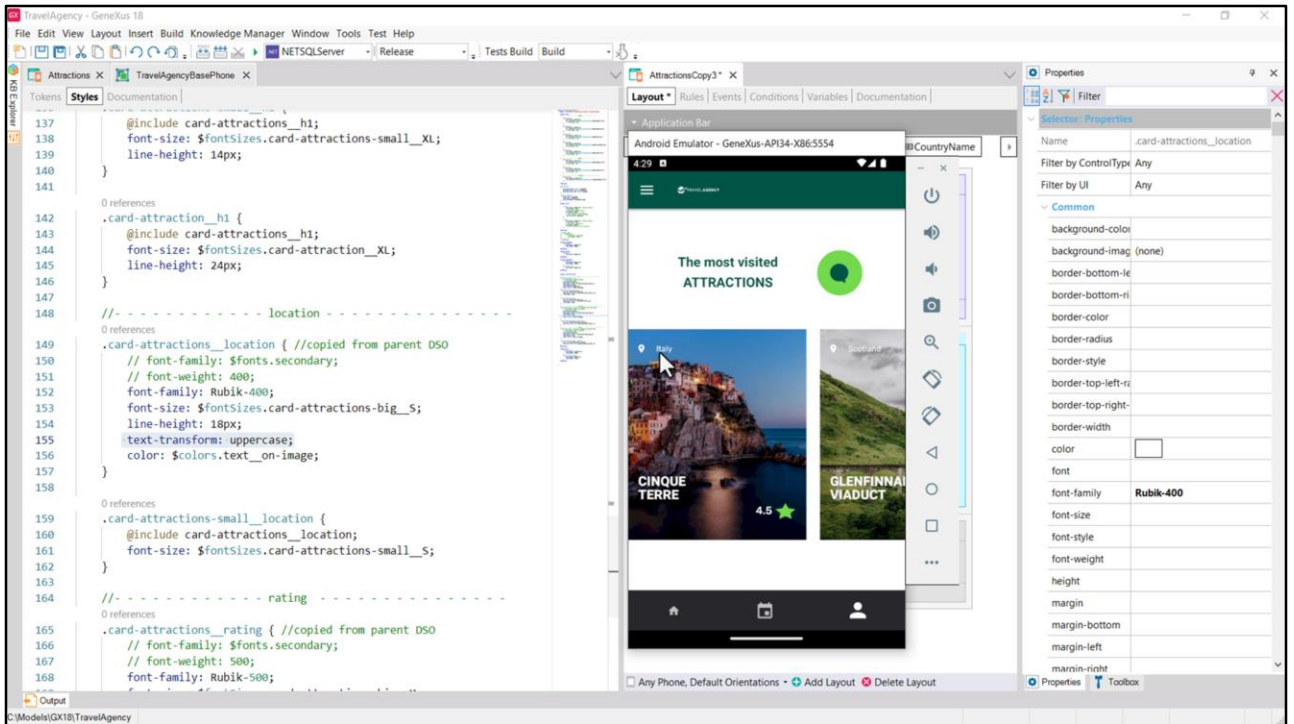
Y a nivel de la clase, recuerden que la habíamos copiado cuando iniciábamos el proyecto, muchos videos atrás, para poder incluirla luego en esta otra variación. Ahora lo único que hicimos fue cambiarle la referencia a la fuente y nada más.



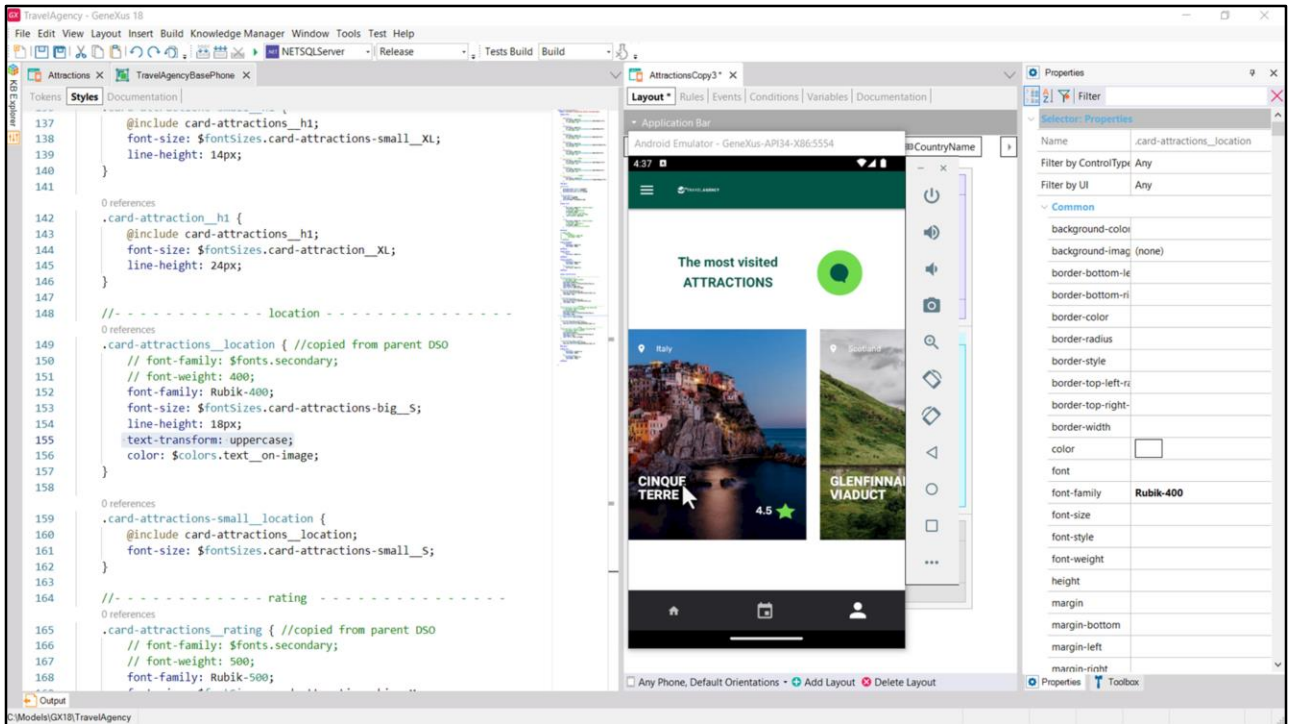
Sabemos que el font-size será el correcto, porque lo habíamos redefinido en este DSO, en aquella oportunidad.



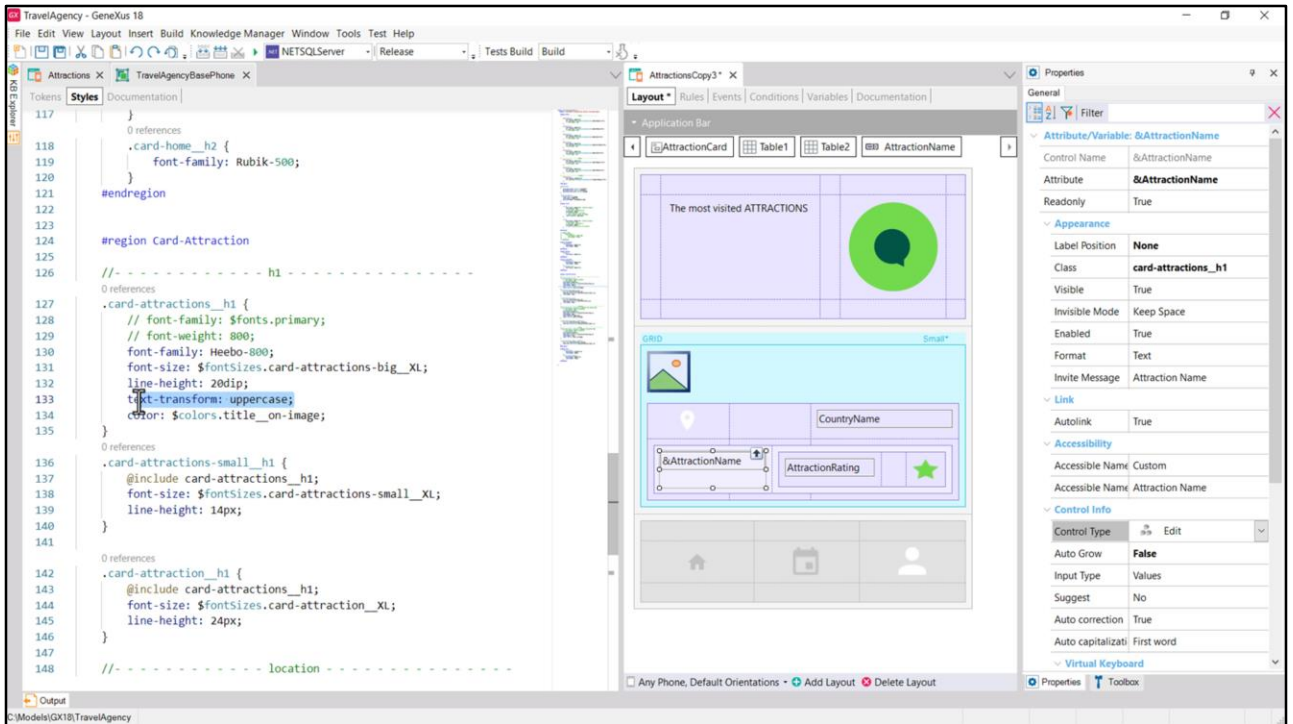
Pero, por ejemplo, no revisamos si este line-height es el correcto para este tamaño...



...y tampoco tomamos en cuenta que esta propiedad no va a tener efecto y es por eso que estamos viendo así el texto.

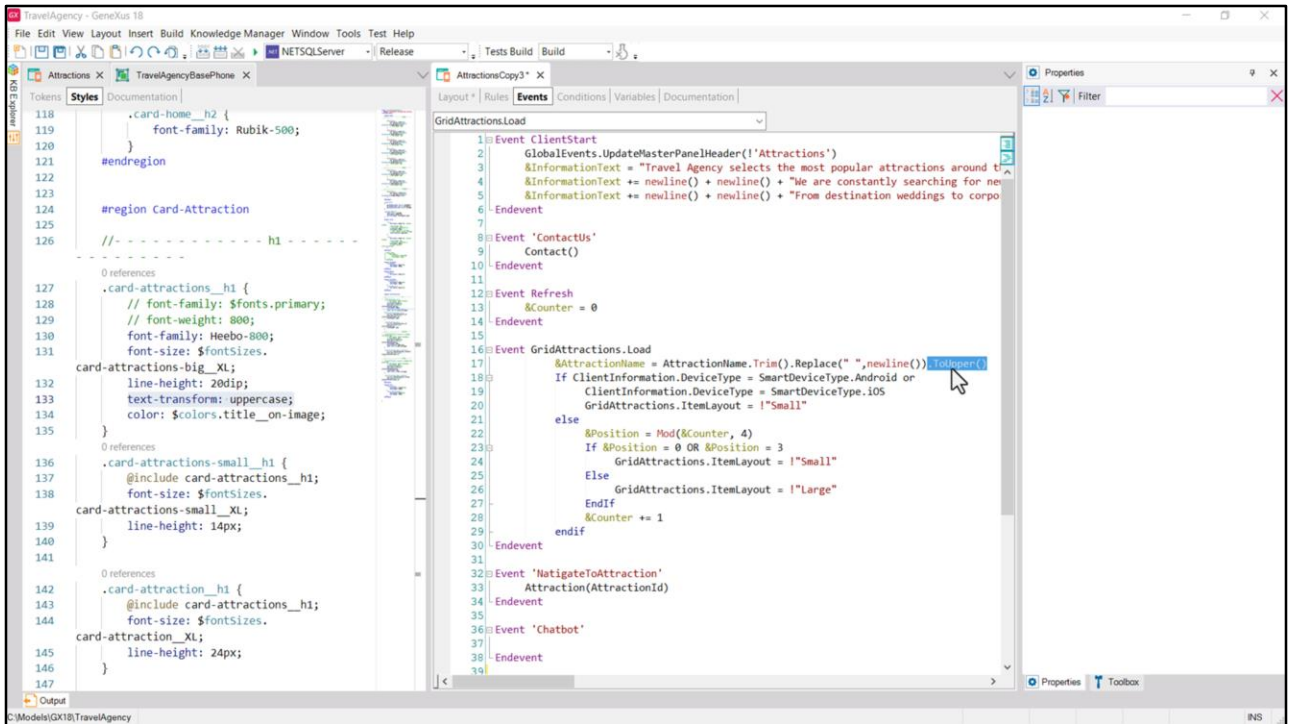


Este texto también debería estarse mostrando en minúsculas.

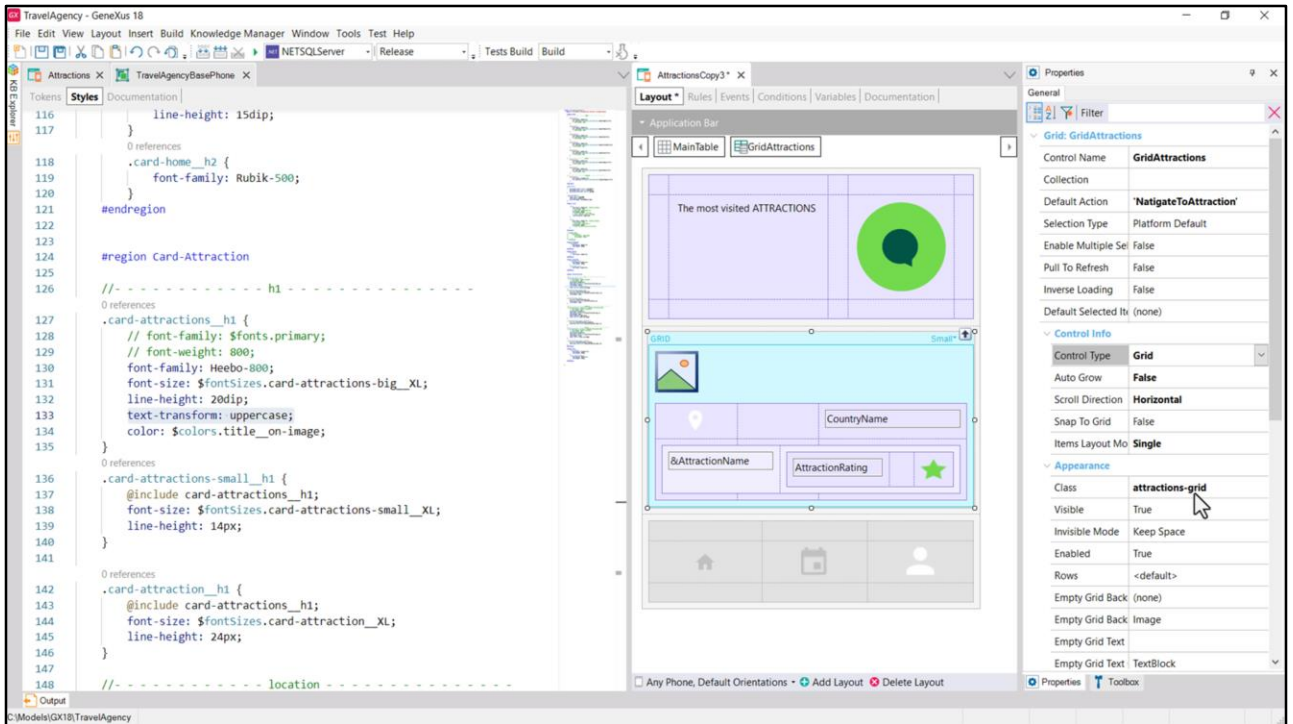


Vean que la clase es esta... a la que, como a la otra, sólo me preocupé de cambiarle la definición de la familia de fuentes. Todo lo demás está tal cual lo habíamos especificado en aquella etapa temprana.

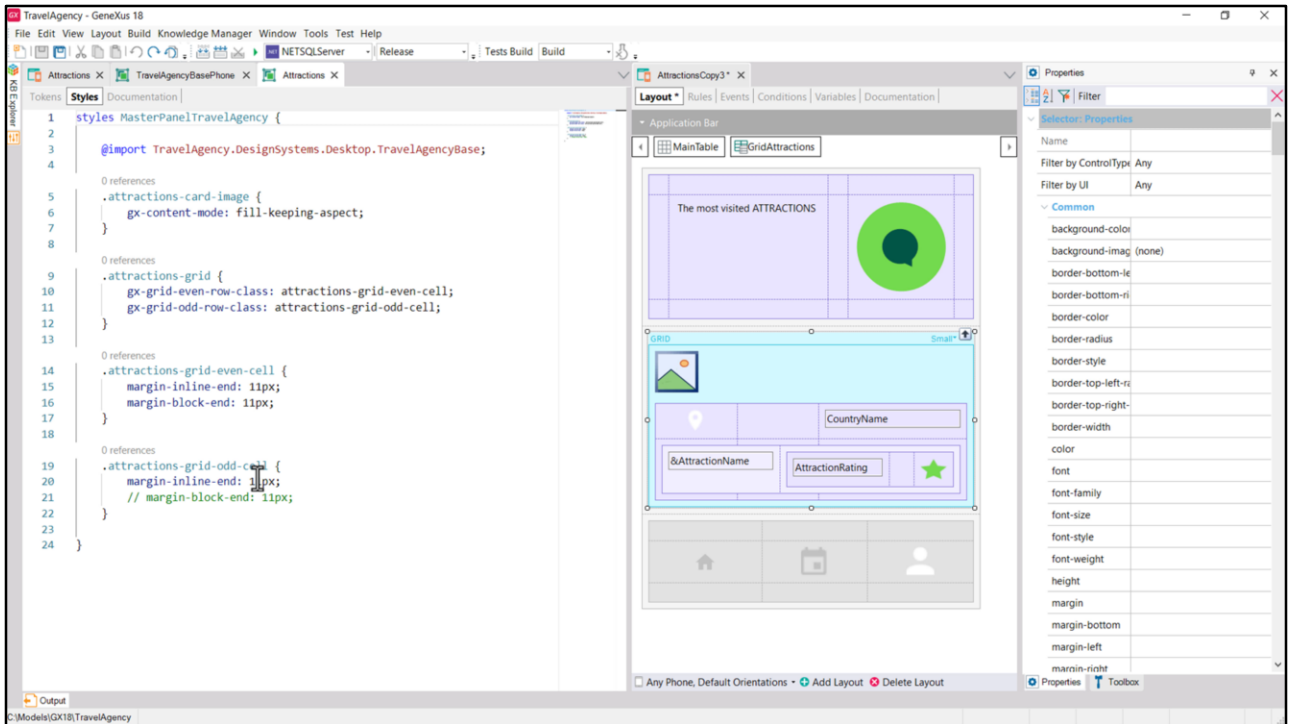
Pero... me di cuenta cuando ejecuté (antes de mostrarles nada de este módulo) que no aplicaba el valor uppercase...



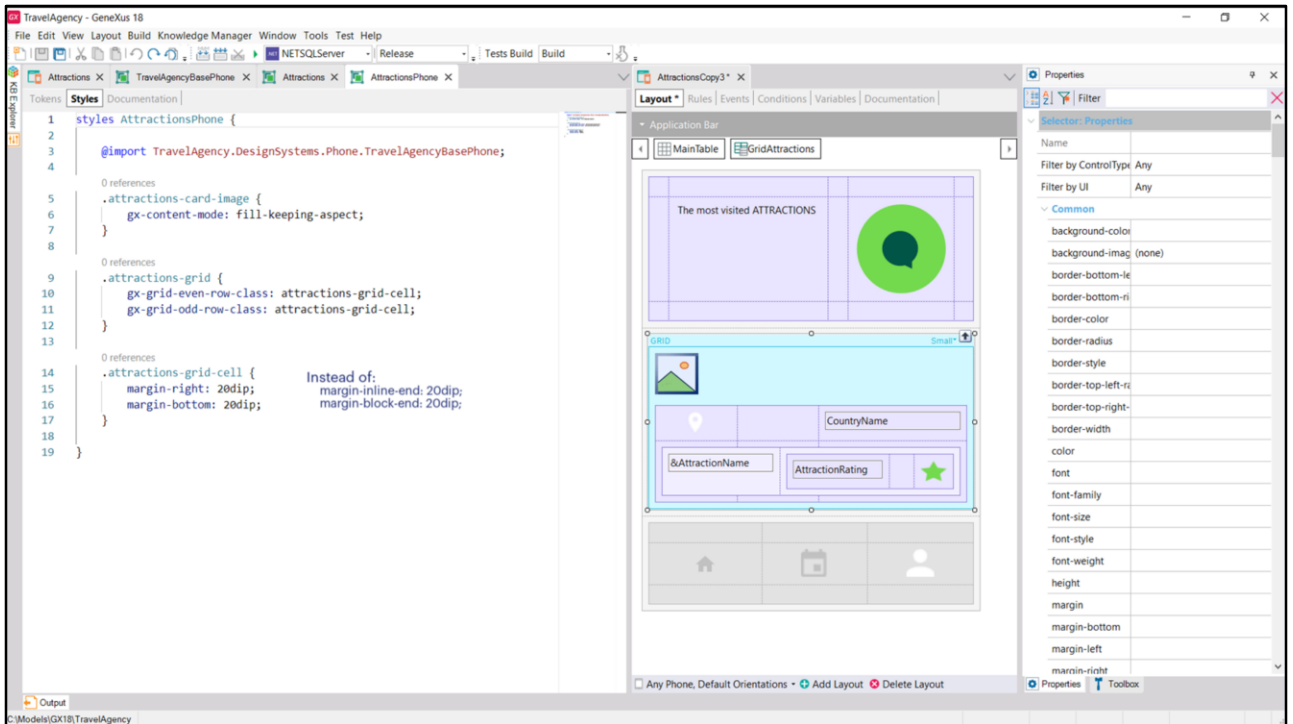
...y entonces cambié programáticamente aquí.



Para el rating es la misma cosa. Por último, el grid tenía como clase esta...



...que era a través de la cual para la aplicación Desktop dábamos los márgenes de cada ítem dependiendo de si eran los de arriba o los de abajo.



Aquí tuve que redefinirla, no sólo porque tenemos por ahora un solo tipo de ítem, sino porque las propiedades lógicas del margen no funcionan aún para el mundo nativo.

En el próximo video vamos a empezar por hacer un resumen de todo esto que estuvimos viendo en este módulo, dar un par de ideas más y concluir. Los espero ahí.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com