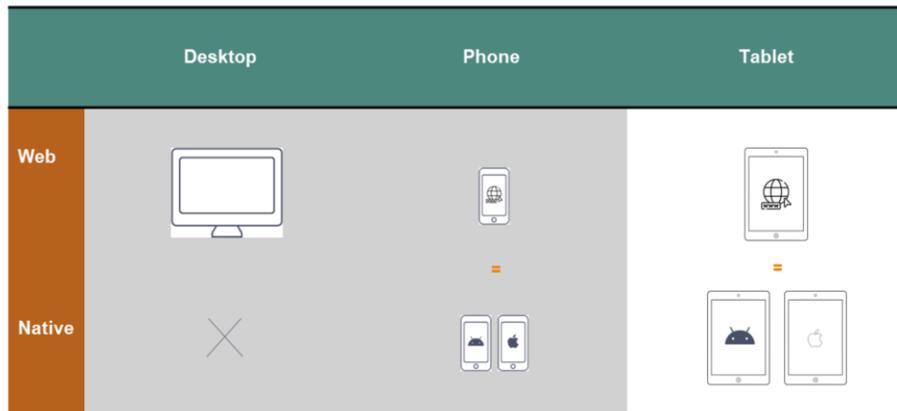


# From Travel Agency Web to Native Mobile (part 2)

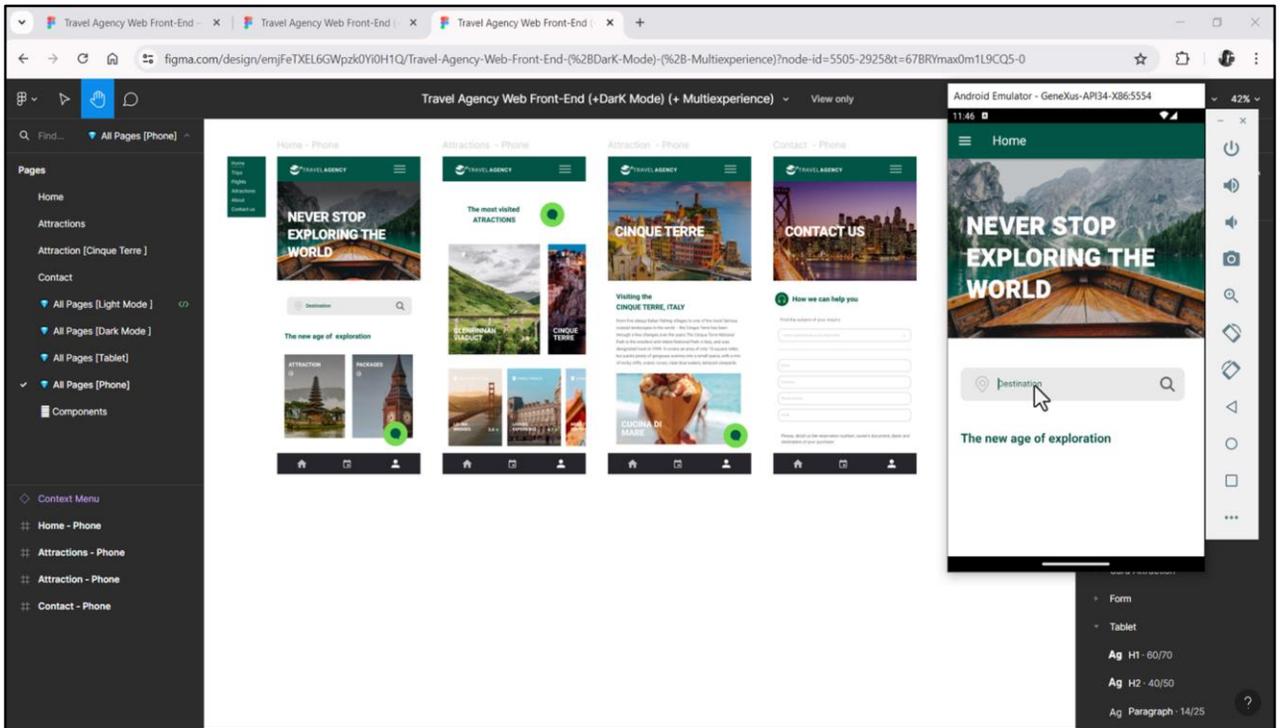


Cecilia Fernández

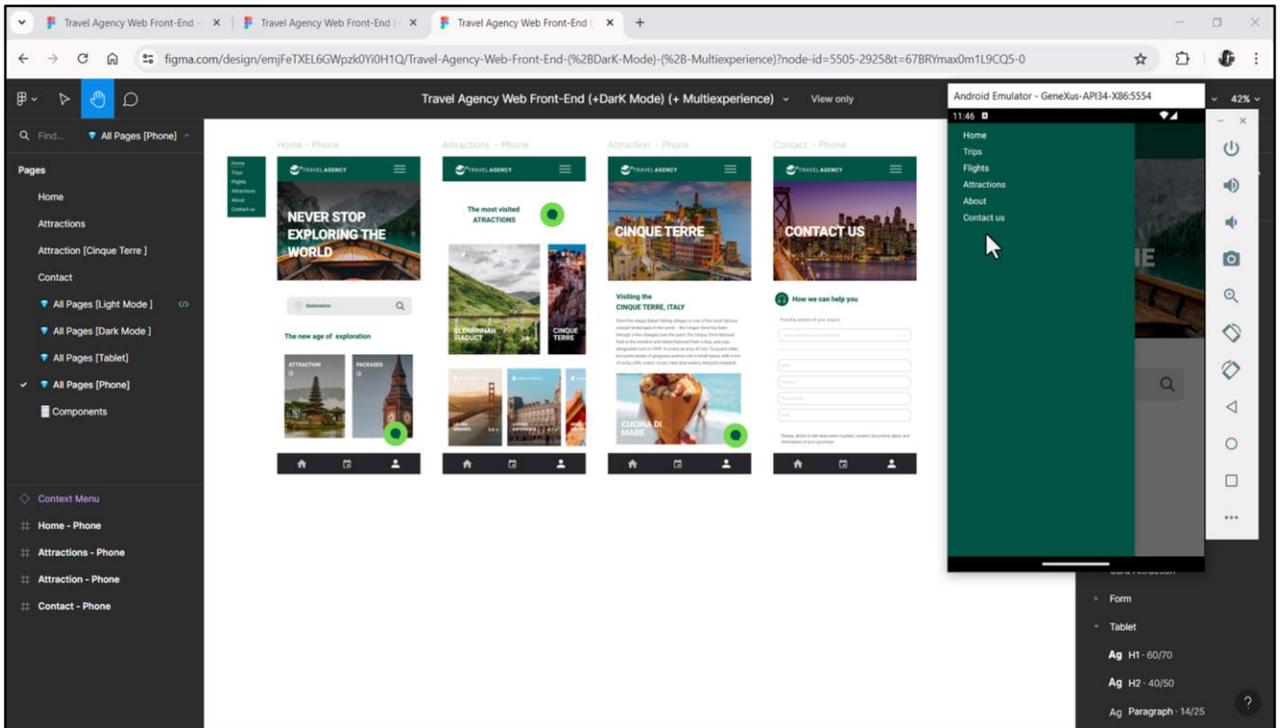


Bueno, voy a retomar lo que interrumpí sobre el final del video anterior, donde veníamos hablando de los efectos que tiene que las aplicaciones nativas no estén construidas sobre el paradigma CSS.

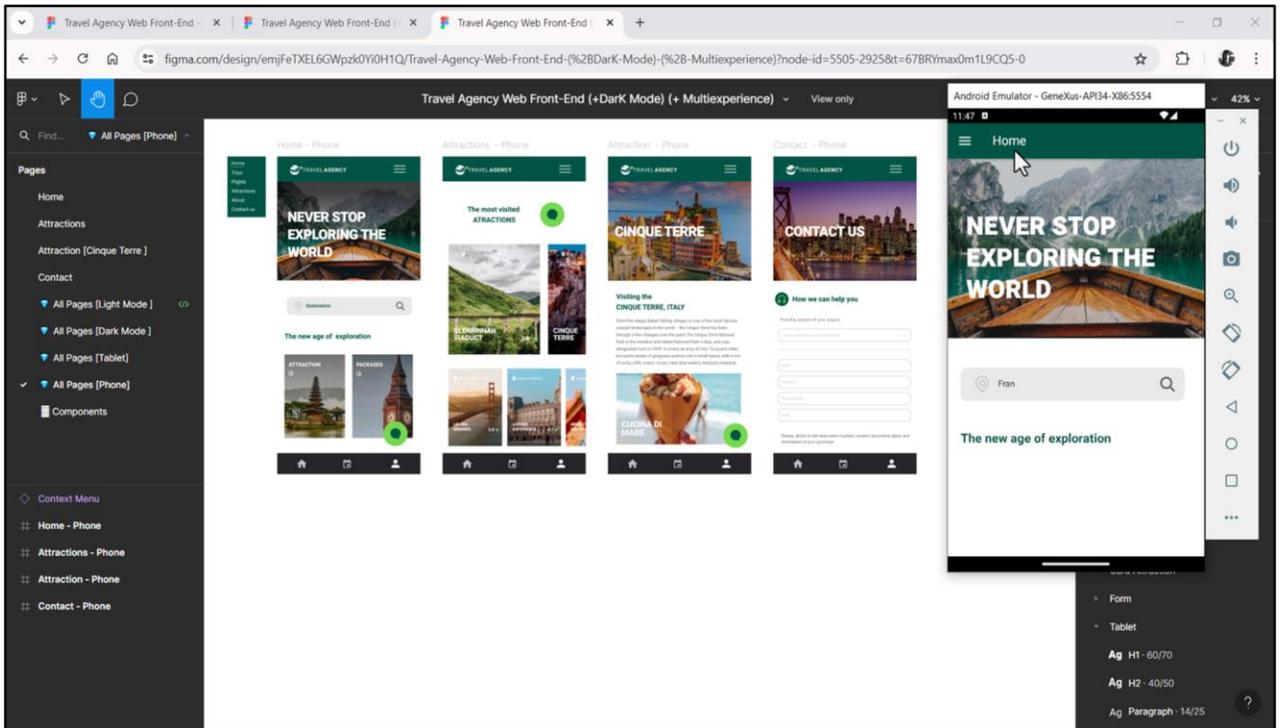
Allí les decía lo siguiente:



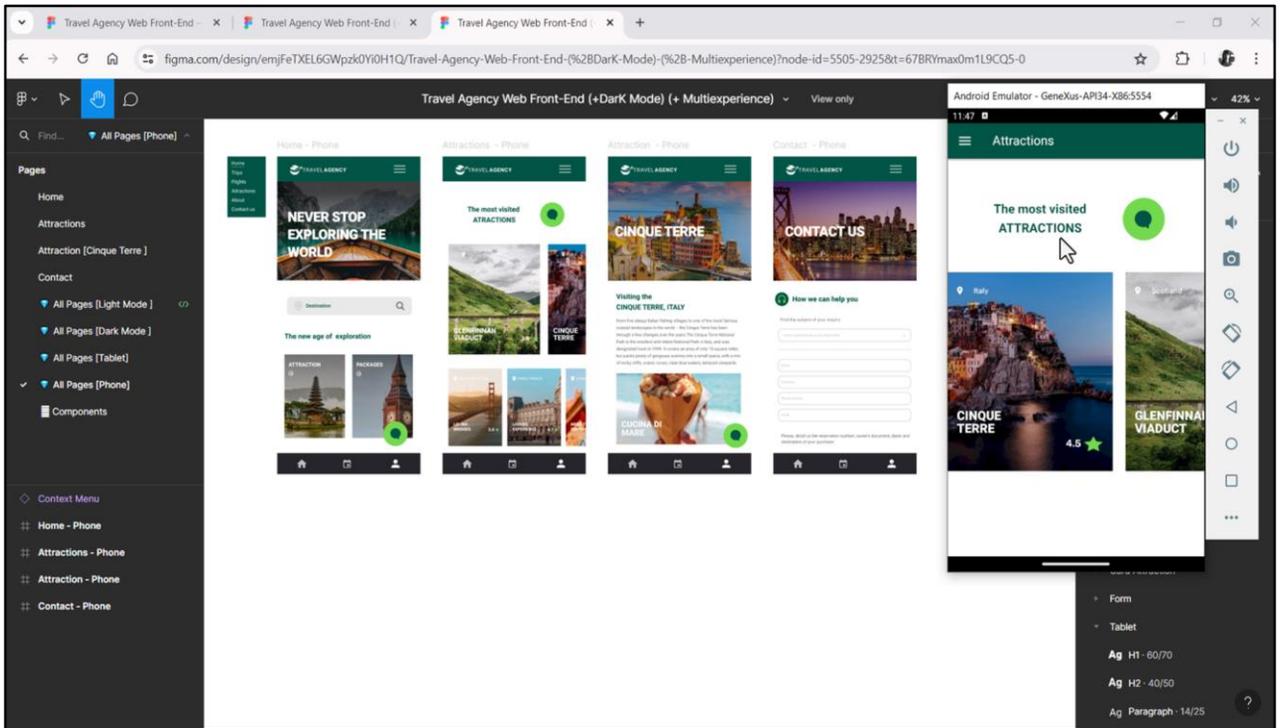
Para entusiasmarlos avancé rápidamente en mi KB para que vayan viendo en este emulador cómo me van quedando las pantallas de la aplicación Android para Phone.



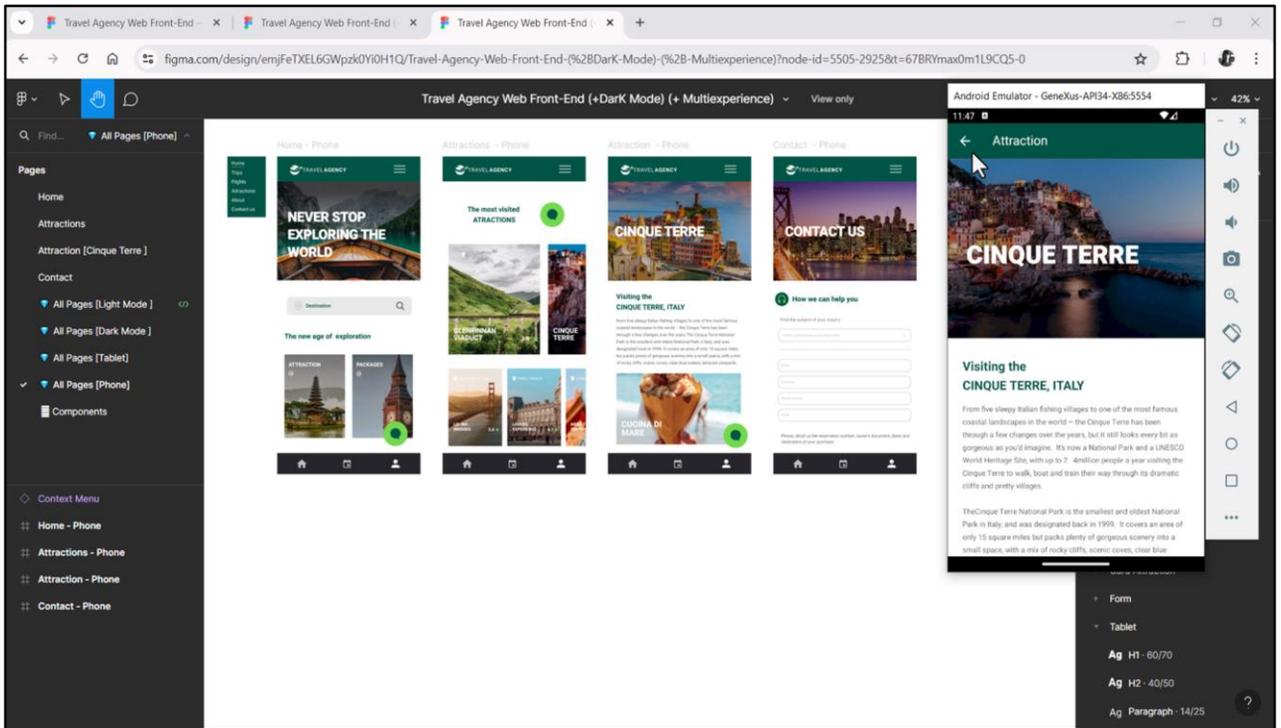
Por ejemplo, vemos el menú hamburguesa (a la izquierda, como es el estándar en su sistema operativo, y no a la derecha, como nos propone el diseño de Chechu)...



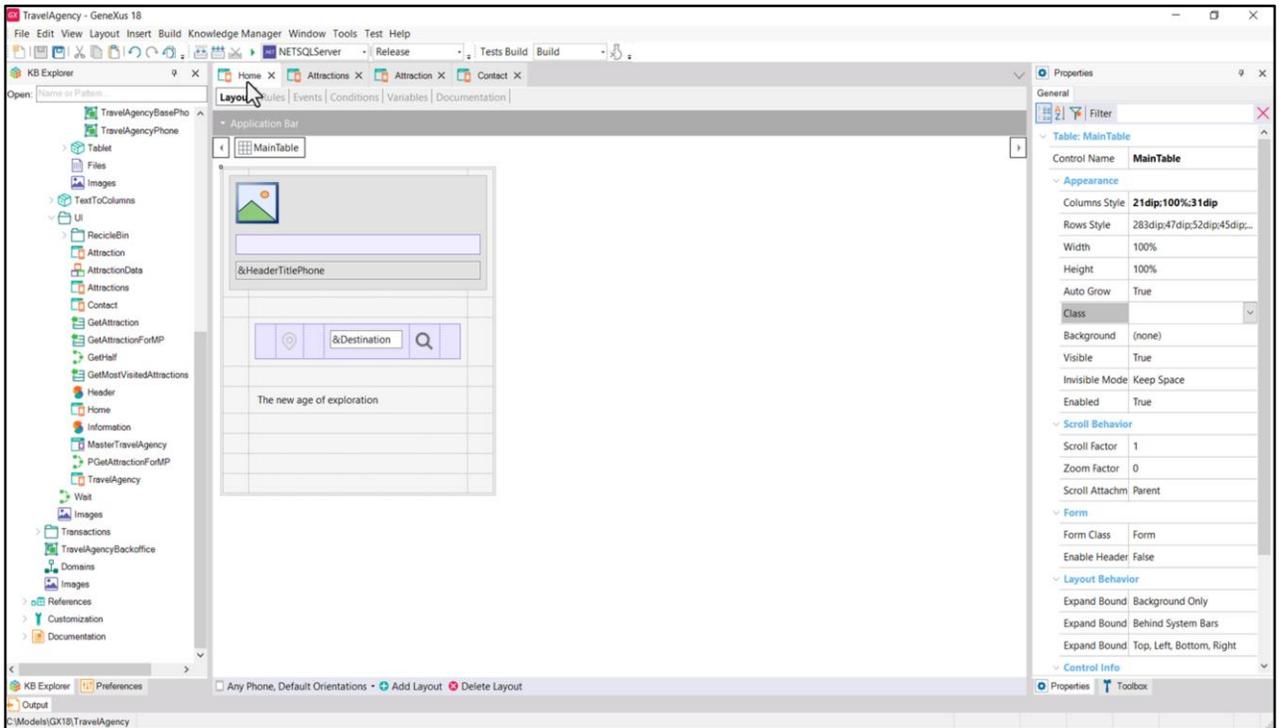
Vemos como título en la Application Bar el nombre del objeto que se está cargando, y no este ícono y texto que luego les mostraré cómo cambiar de la manera más simple.



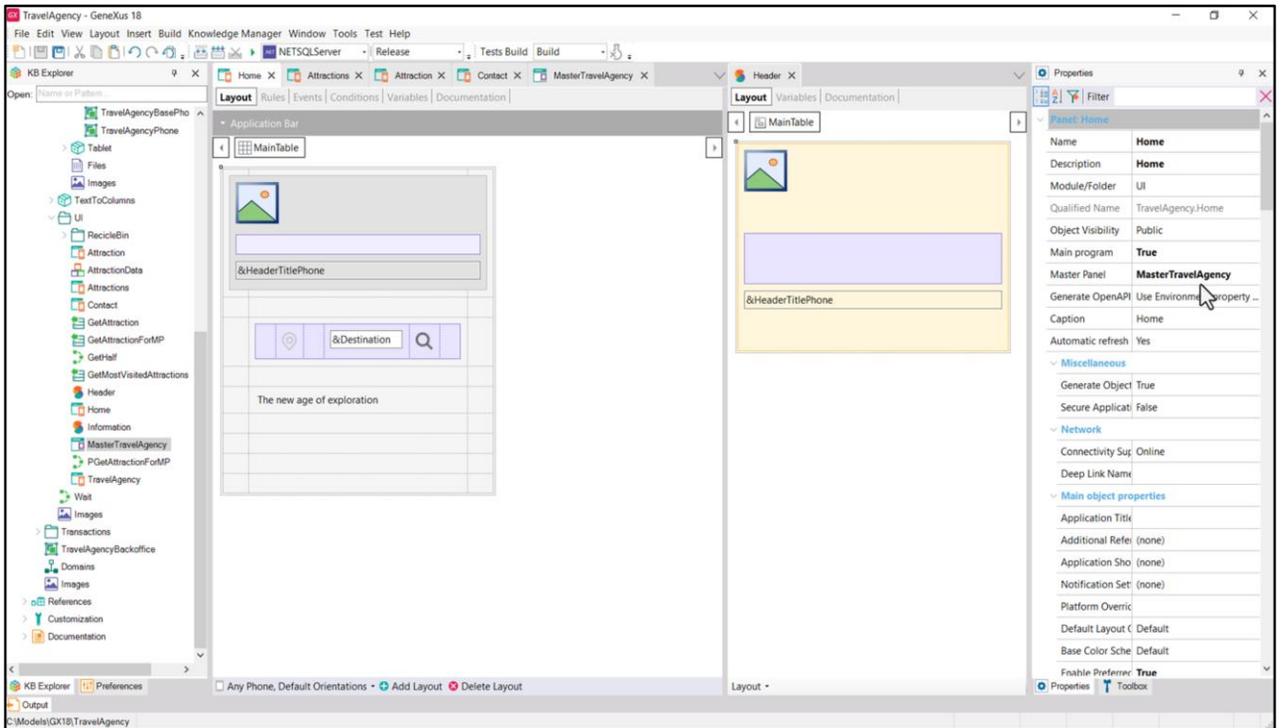
Vemos que hice que el panel de Attractions no tenga el Header, tal como lo indica el diseño; y vemos que por ahora implementé el carrusel como grid horizontal (para hacerlo rápido y sencillo)...



...y que si hago tap sobre una atracción nos lleva al panel Attraction; y automáticamente, y también según los mecanismos propios del sistema operativo y las guías de diseño de Android, nos aparece el botón de back para volver al objeto que lo llamó.



Los paneles son los mismos, exactamente, que los que utilizamos para la aplicación Angular para Desktop. Pero, claro, les definí a cada uno otro layout, esta vez para Any Phone, para que valga también para iPhone. Ahí tenemos los 4 layouts a la vista.

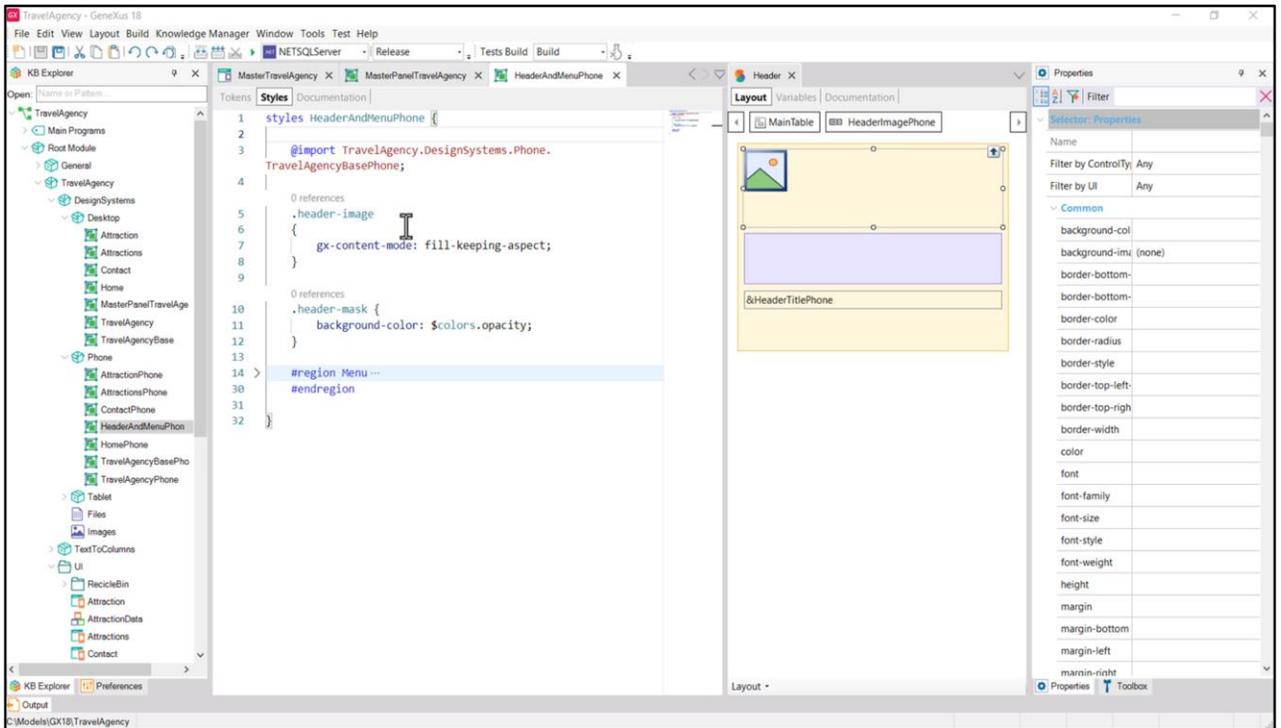


Sólo he tenido que agregar dos objetos a la UI: este, que será el menú y punto de entrada de la aplicación, y este Stencil que compite con el Master Panel.

¿Por qué digo que compite? Porque las aplicaciones nativas ignoran la propiedad Master Panel y el objeto referido, claro. No lo toman en cuenta. Y es por eso que tuve que insertar en cada uno de los paneles que llevaban este Header el stencil que lo diseña.

Así, mientras el menú y el Header estaban incluidos para Angular Desktop en el objeto Master Panel, para la aplicaciones nativas al Header no tenemos más remedio, si queremos reutilizar, que incluirlo como stencil en cada uno de los layouts. O si no, como component, pero en principio para nuestro caso no sería necesario.

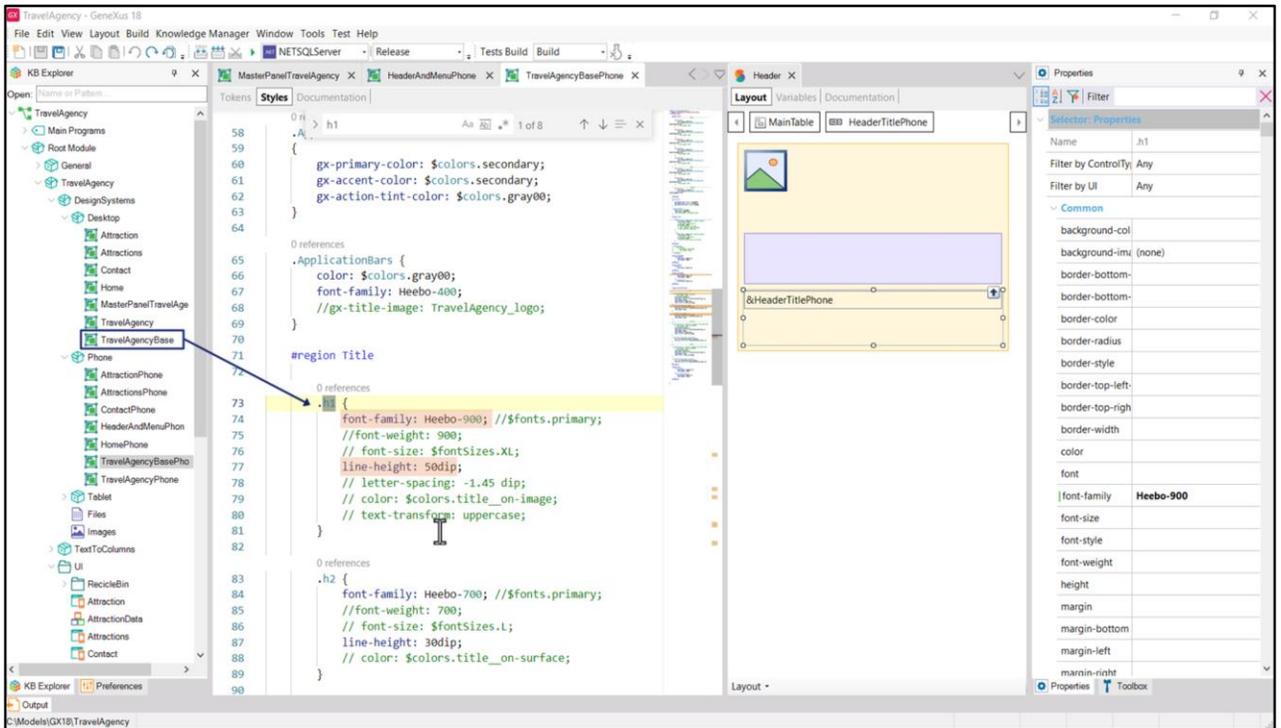
Luego veremos qué pasa con el menú.



En el Stencil tengo al canvas con la variable para la imagen, la tabla que hace de máscara y la variable para el título.

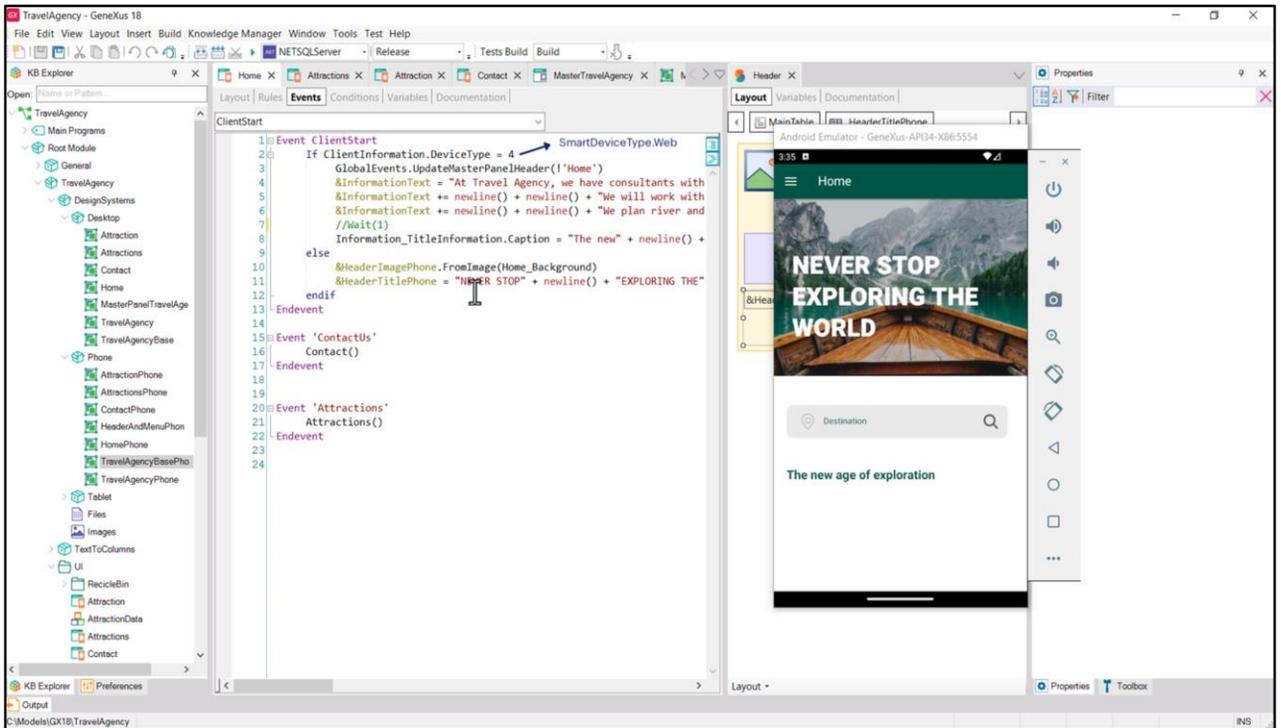
¿Dónde definí el estilo de sus clases?

Si para Angular Desktop lo había hecho en el DSO de igual nombre que el Master Panel, aquí me creé uno paralelo al que le llamé así. Y aquí defino las propiedades del Header y también las del menú que luego veremos.



La h1 está en el DSO base para Phone, y tendrá estas características (todas las que provienen del DSO TravelAgencyBase con estos cambios de aquí).

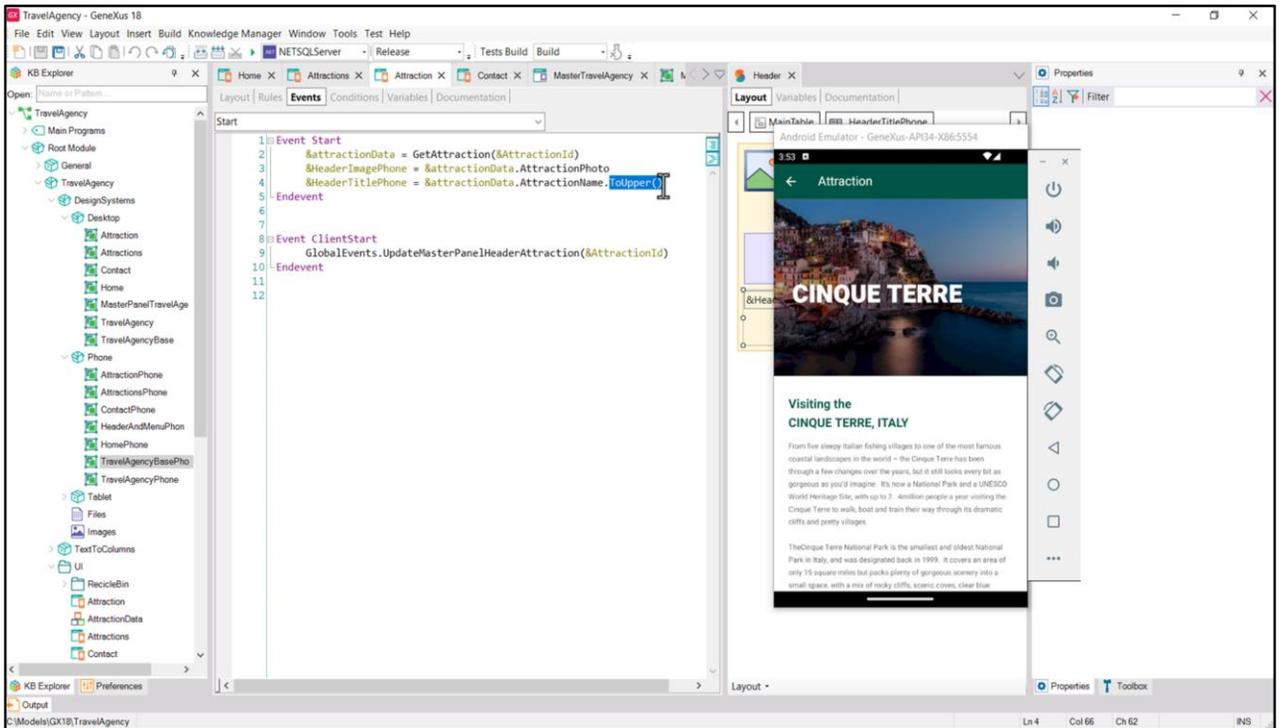
Sin embargo habrá que ver si todas estas propiedades aplican al mundo nativo. Como les dije antes, no todas las propiedades CSS lo harán. De hecho, por el momento, la text-transform con valor uppercase no funcionará.



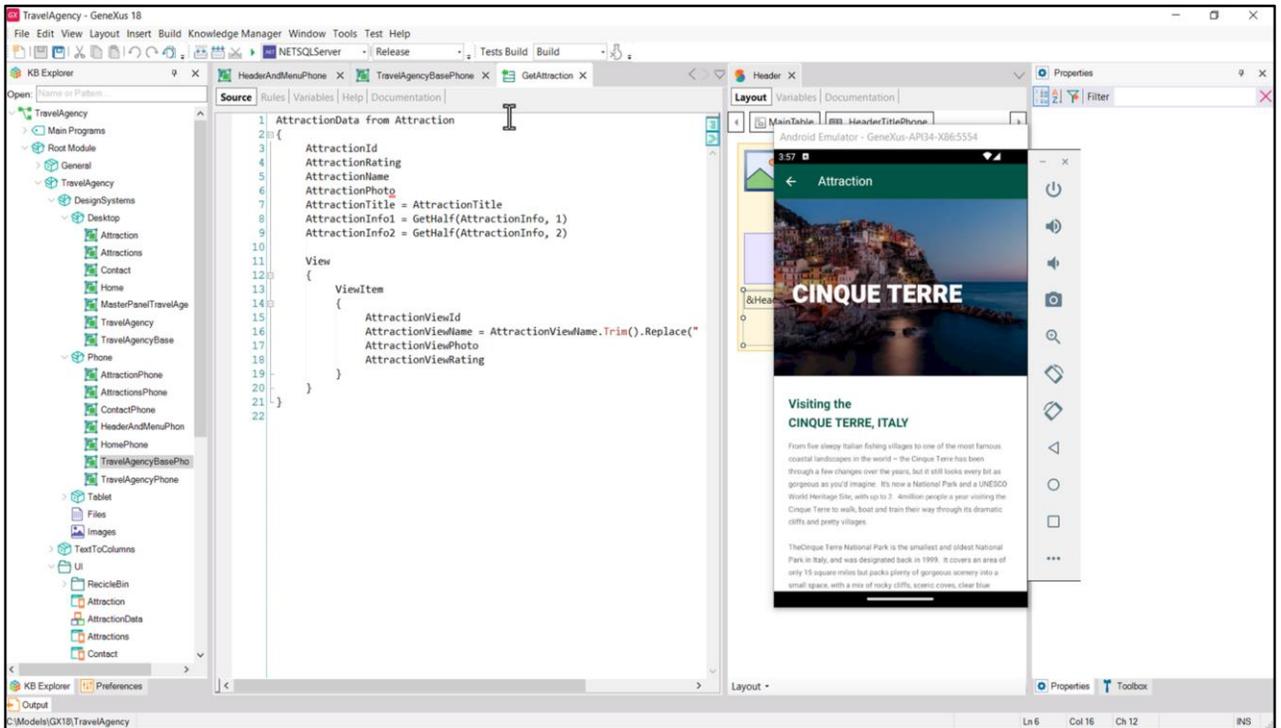
Observen que logro que salga en mayúsculas porque el contenido de la variable está en mayúsculas en todas las pantallas.

Aquí vemos cómo cargué en el ClientStart estas dos variables del Header sólo si no se está ejecutando Angular. En verdad no es el corte que vamos a querer realizar, porque, justamente, en aras de la transversalidad, queremos utilizar la misma solución para Angular de este tamaño de ventana. Pero por ahora no importa.

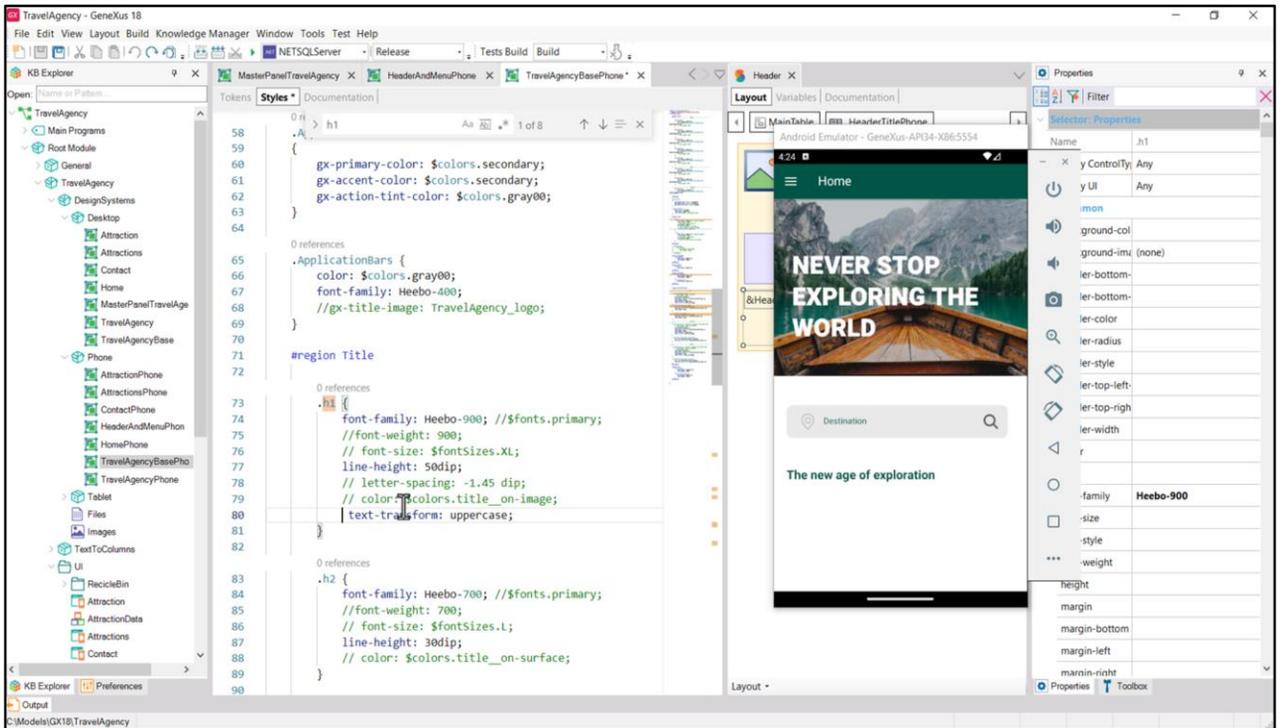
Vemos que el título está ingresado en mayúsculas.



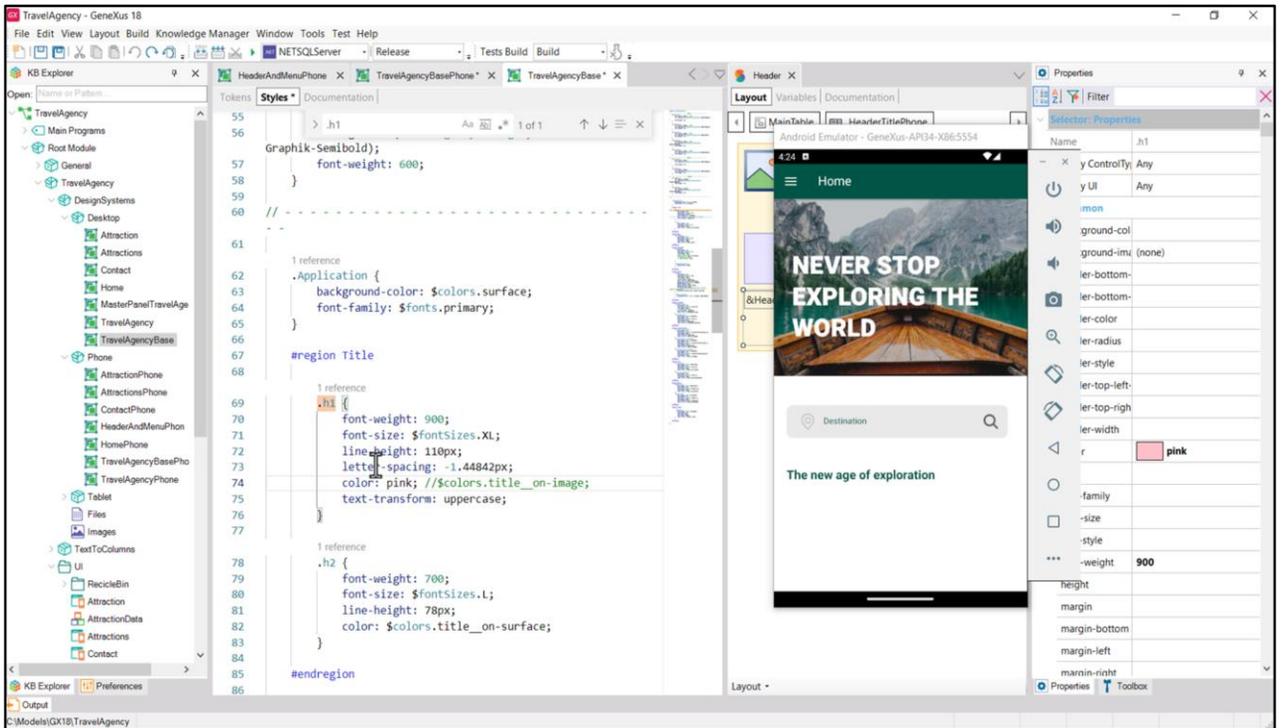
Y si ahora voy a ver el título para Attraction, que es tomado de la base de datos... por lo que no sé si allí está en mayúsculas... lo obligo a estar en mayúsculas con este método.



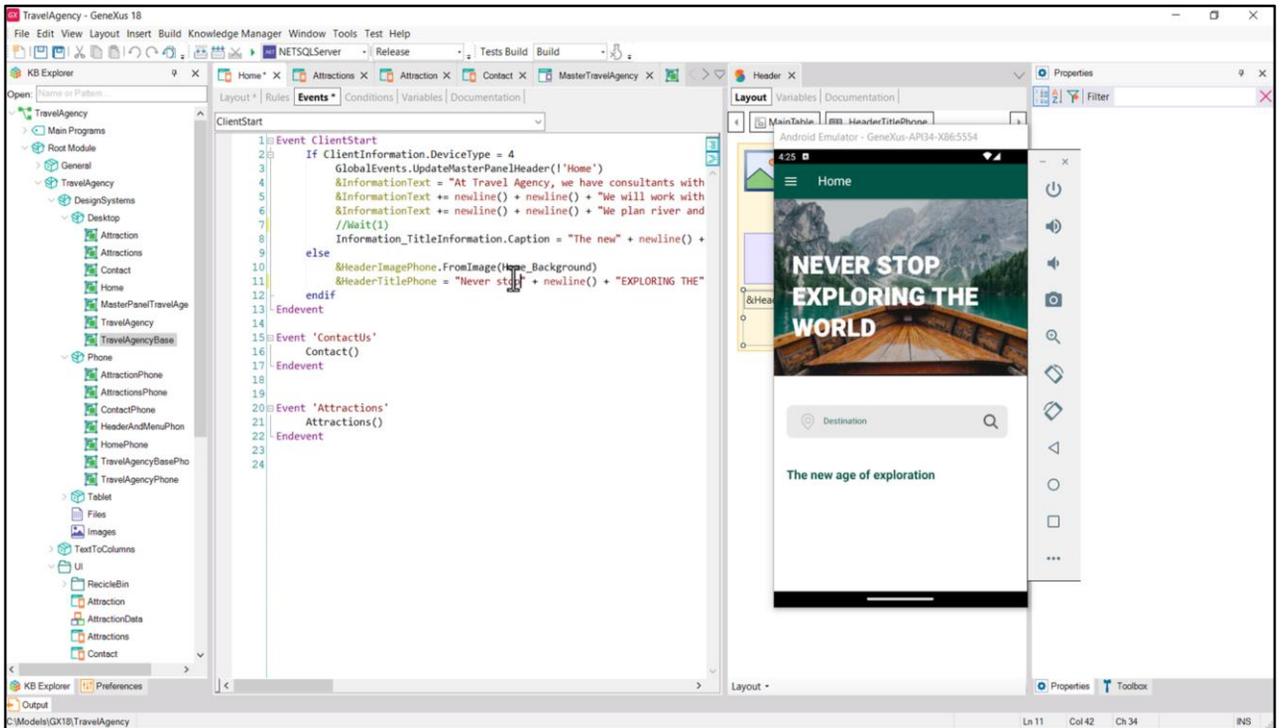
Observen que agregué al Data Provider que habíamos utilizado para la aplicación Desktop, y que me devolvía los datos de la atracción, los valores de AttractionPhoto y AttractionName, que no los cargábamos en aquella oportunidad.



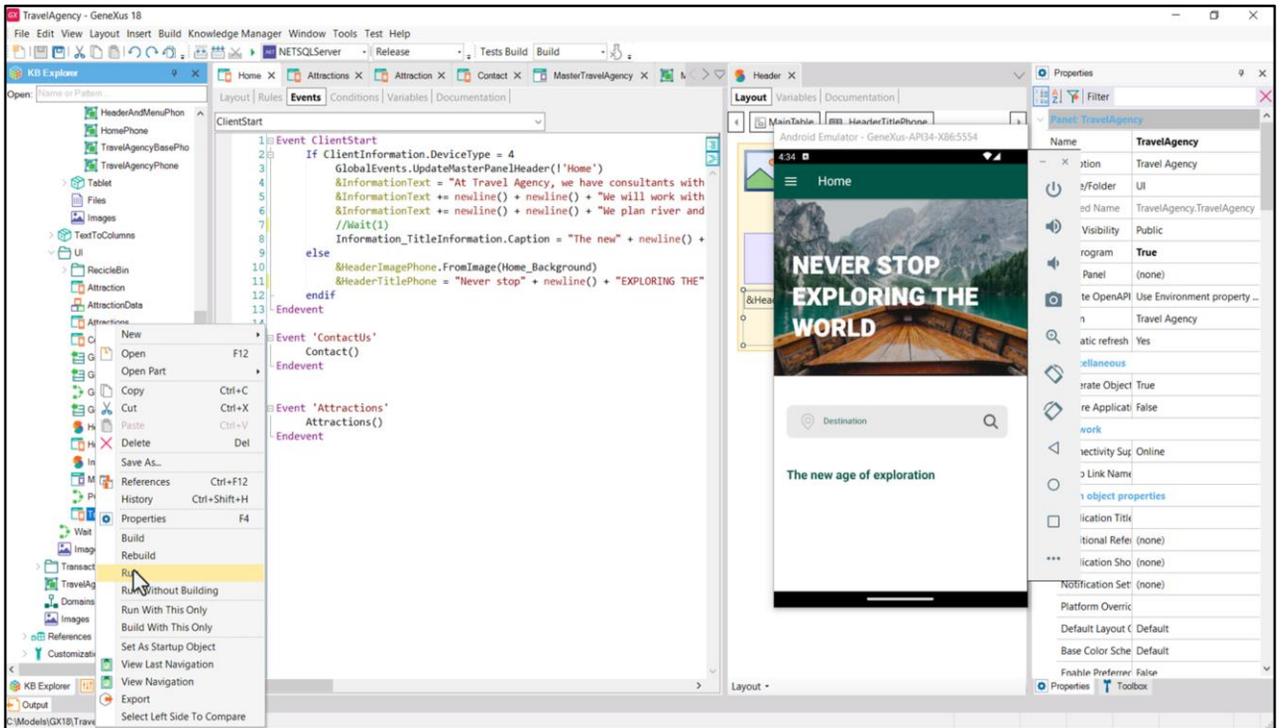
Les voy a mostrar explícitamente para que no quepan dudas de que no está aplicando esta propiedad CSS. La voy a descomentar, por si dudan de que se esté tomando correctamente del TravelAgencyBase...



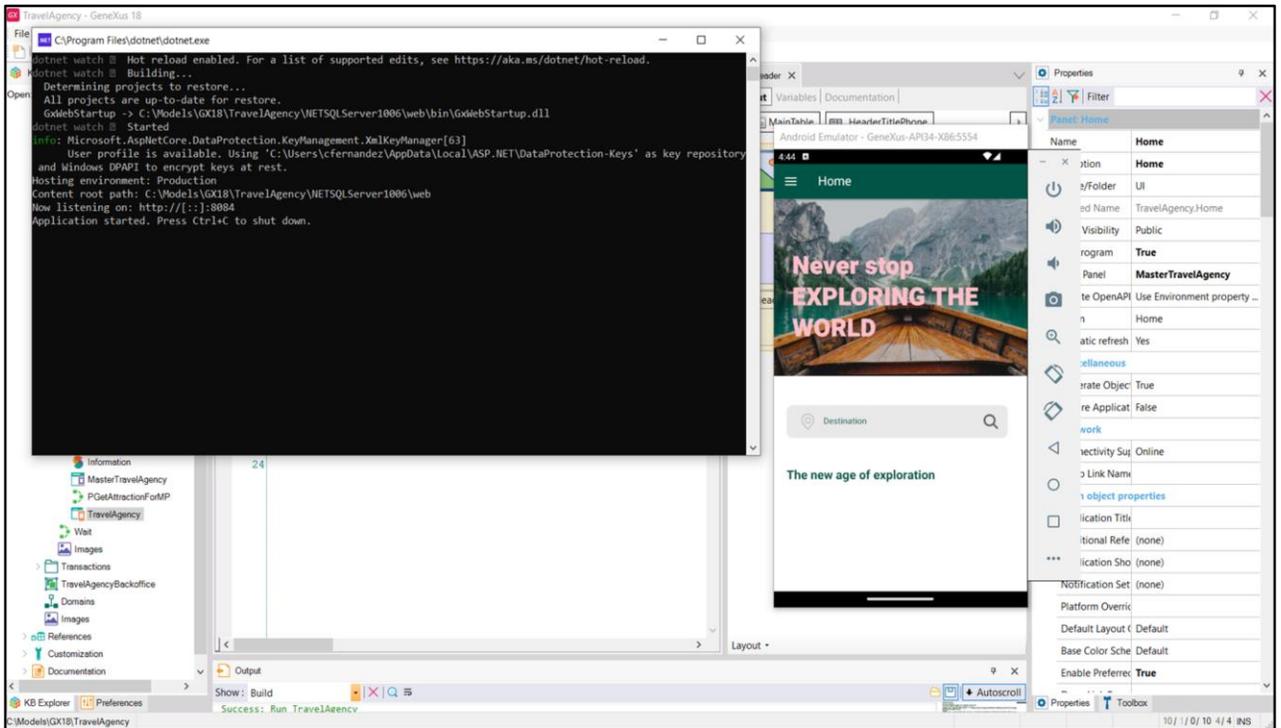
Y a la vez en ese DSO voy a cambiarla de color, para que sea rosado, para que vean que efectivamente no es ese el problema, porque sí toma las propiedades que provienen de allí.



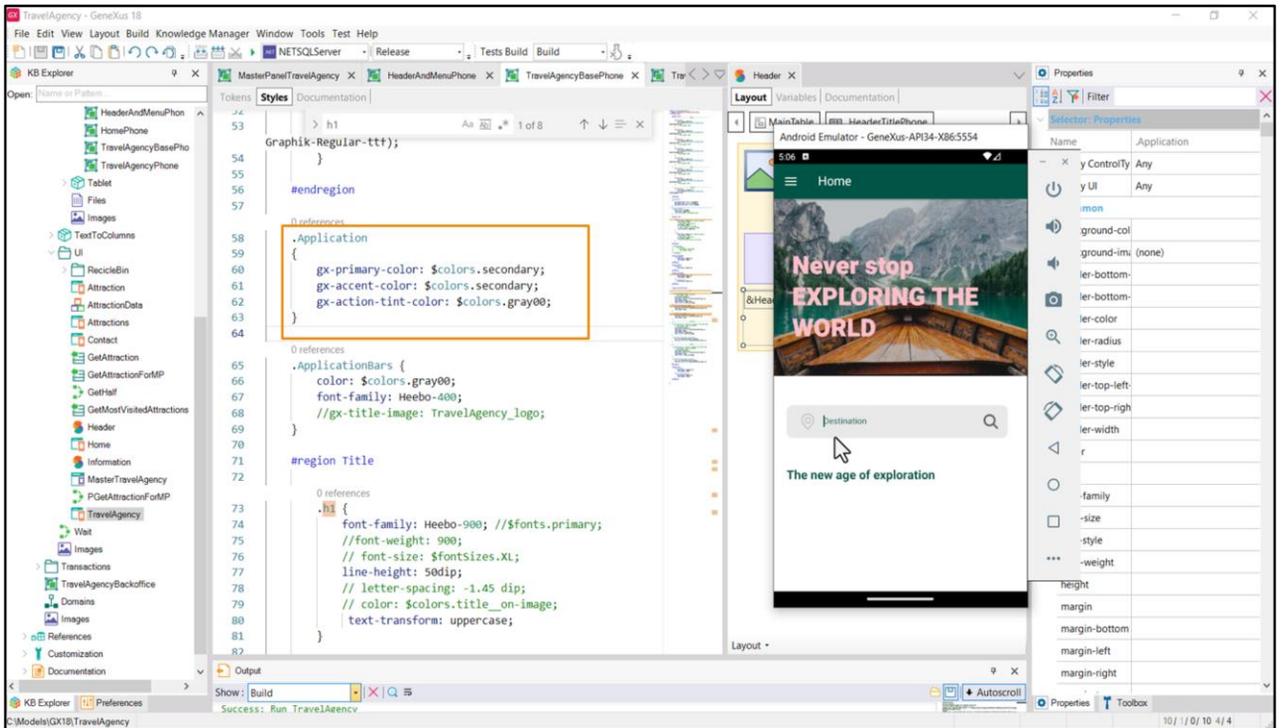
Y lo otro que haré será modificar aquí esto para que esté en minúsculas.



Mi objeto main es el menú, así que es el que voy a ejecutar.



Bueno, con esto habrá quedado claro el punto, ¿no?



Antes de continuar aprovecho a mostrarles que la manera de especificar que este color sea el de la application bar, y que este blanco sea el color del los textos y del ícono hamburguesa, y que para campos editables este sea el color del cursor... es a través de estas propiedades especificadas a nivel de la clase Application. Sólo esta clase las entenderá.

Application Class for Smart Devices

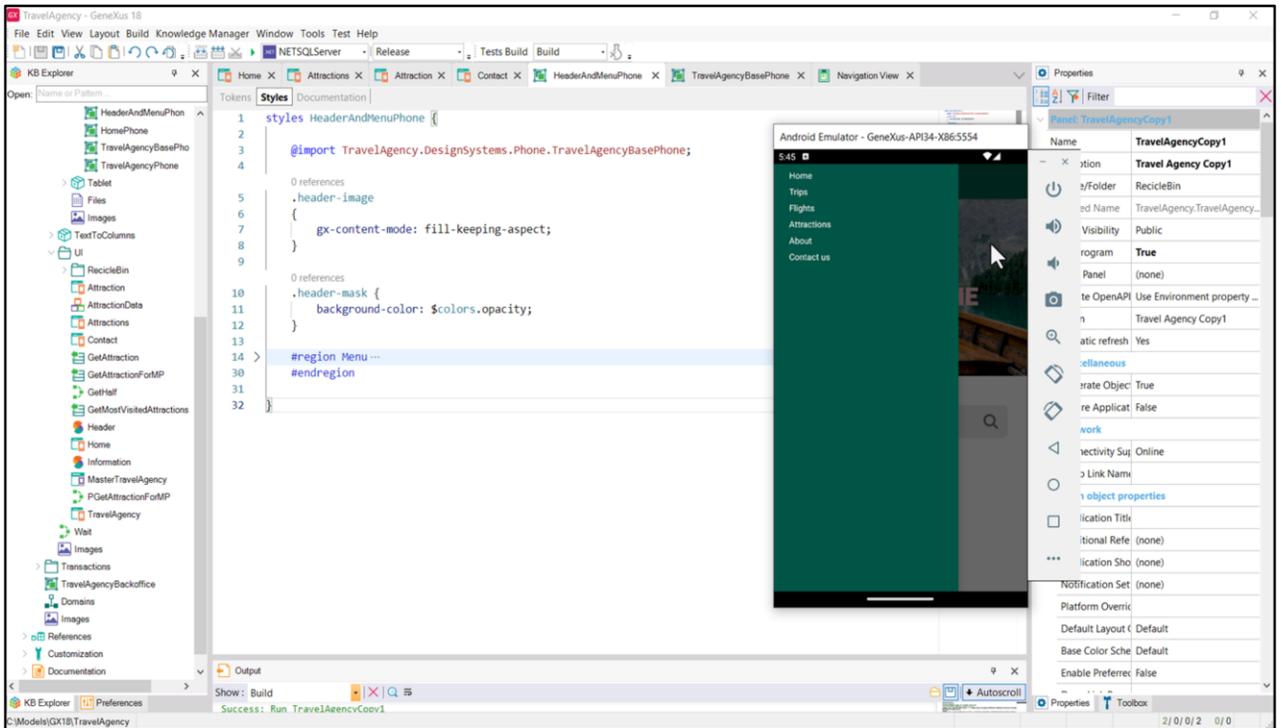
This documentation is valid for: [GeneXus 18 Help](#) [GeneXus 17 Help](#) [GeneXus 16 Help](#) [GeneXus 15 Help](#)

Allows the developer to set some properties to the whole application. GeneXus takes this theme class automatically when runs the application.

Class: Application	
Name	Application
Description	
Background Color	<input type="checkbox"/> Background
Background Image	(none)
Background Image Fill Mode	Scale To Fill
Placeholder Image	(none)
Image Loading Indicator	False
Date Picker Image	(none)
Prompt Image	(none)
Launch Screen Animation Class	
Application Colors	
Primary Color	Primary
Primary Color Dark	PrimaryDark
Accent Color	Accent
Control Normal Color	AttributeLabel
Control Activated Color	Accent
Control Highlight Color	Primary
Action Tint Color	ActionTint

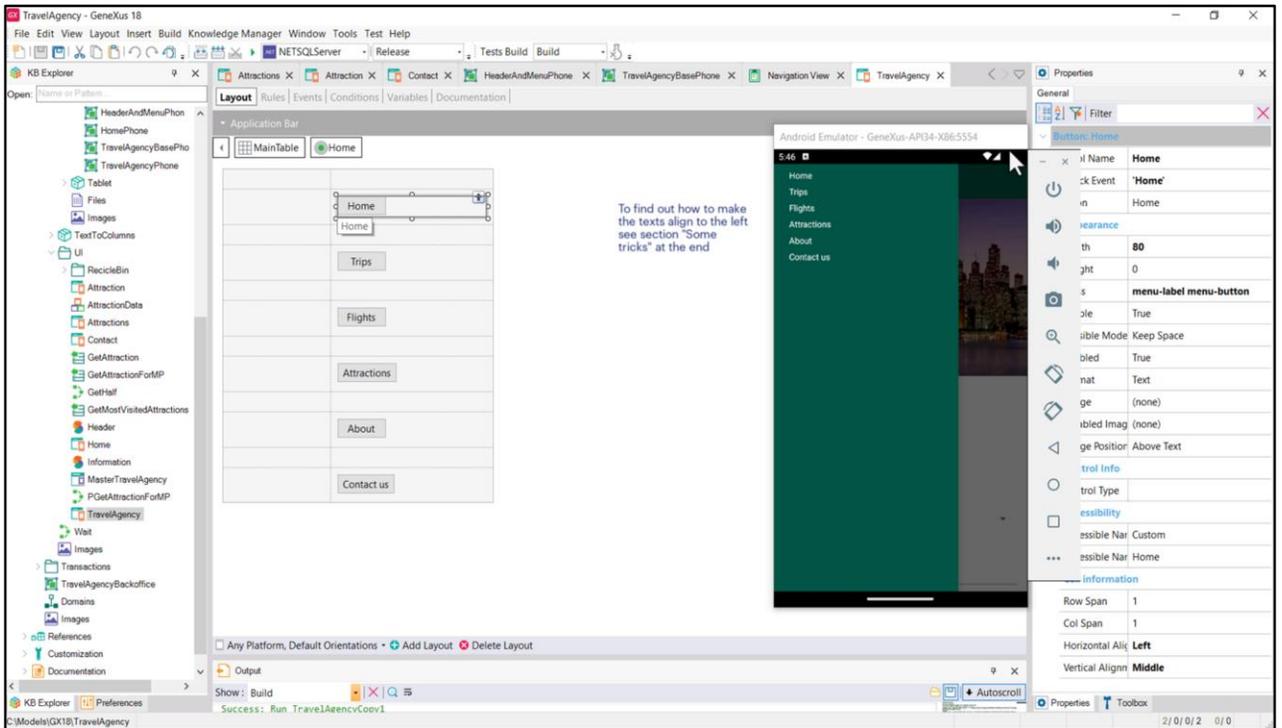
[Ask here!](#)

Pueden ver en el wiki de GeneXus la información completa. Corresponde a definiciones propias de las guías de diseño de Android y Apple.

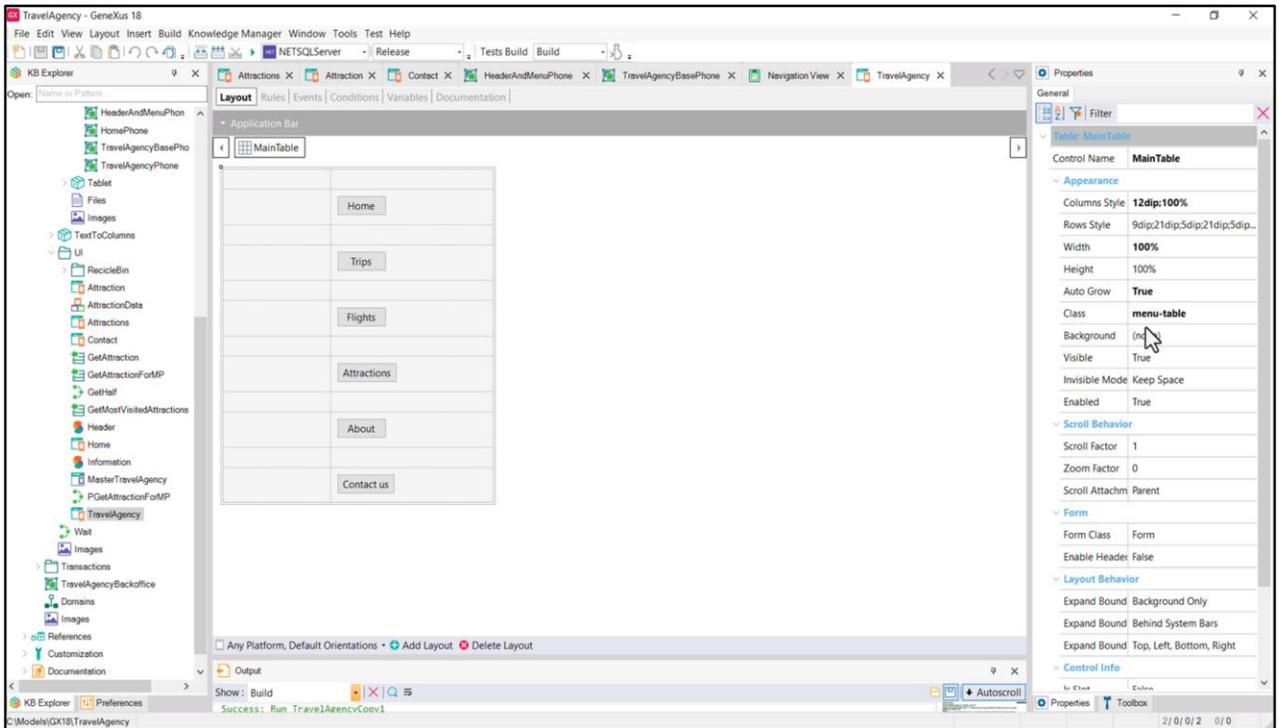


¿Y cómo es que implementé el menú hamburguesa? Tenemos dos cuestiones en juego:

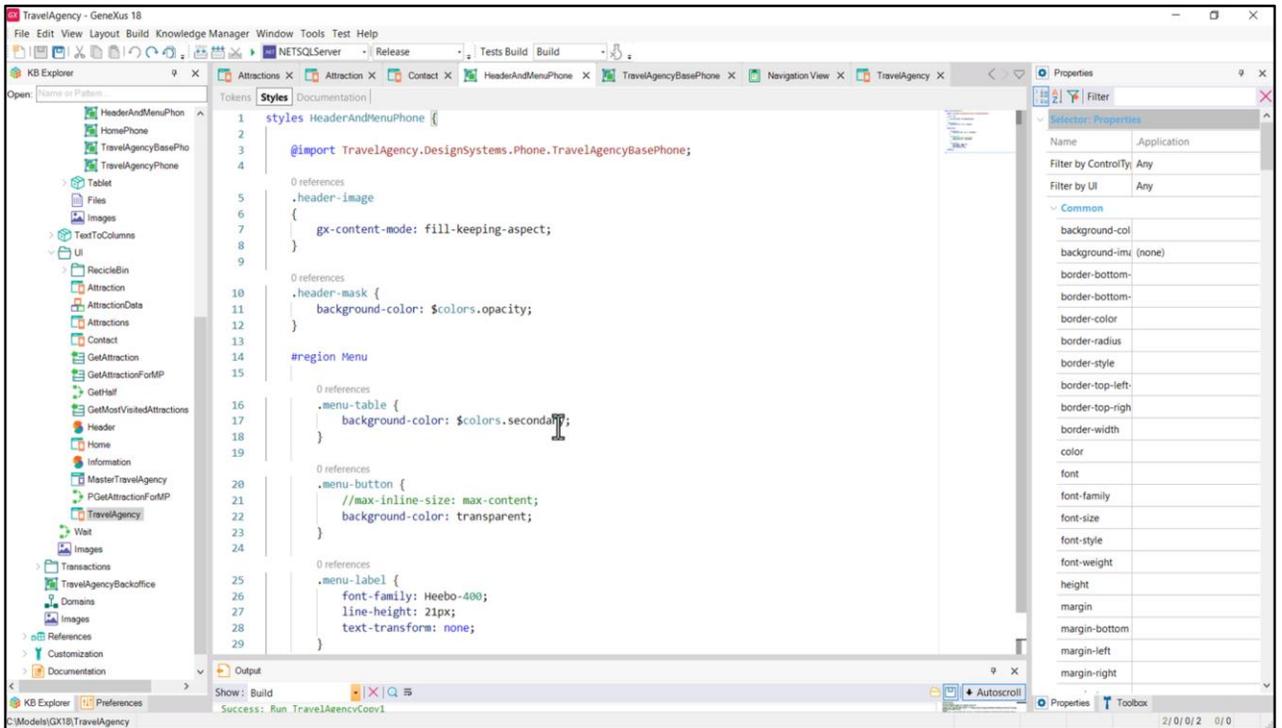
1. Implementar el menú propiamente dicho y,
2. Lograr que tenga el comportamiento típico del menú hamburguesa.



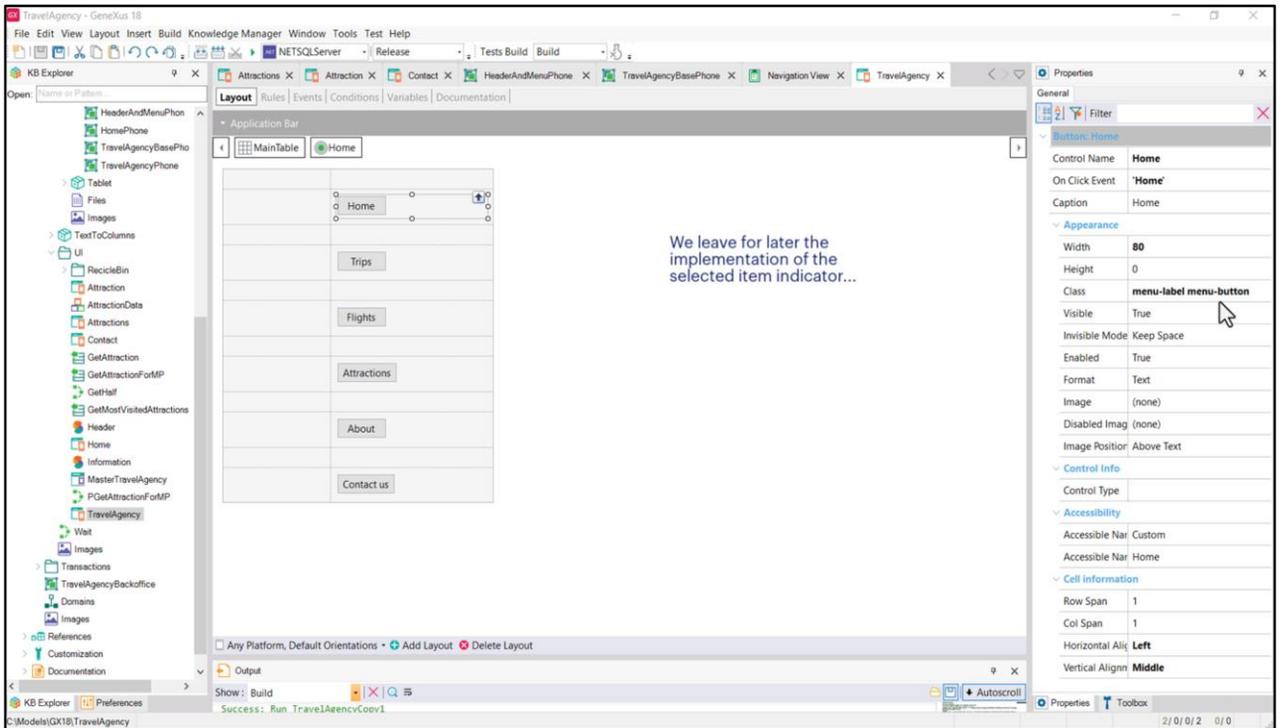
Lo primero lo hice con un objeto panel con una tabla con los 6 botones.



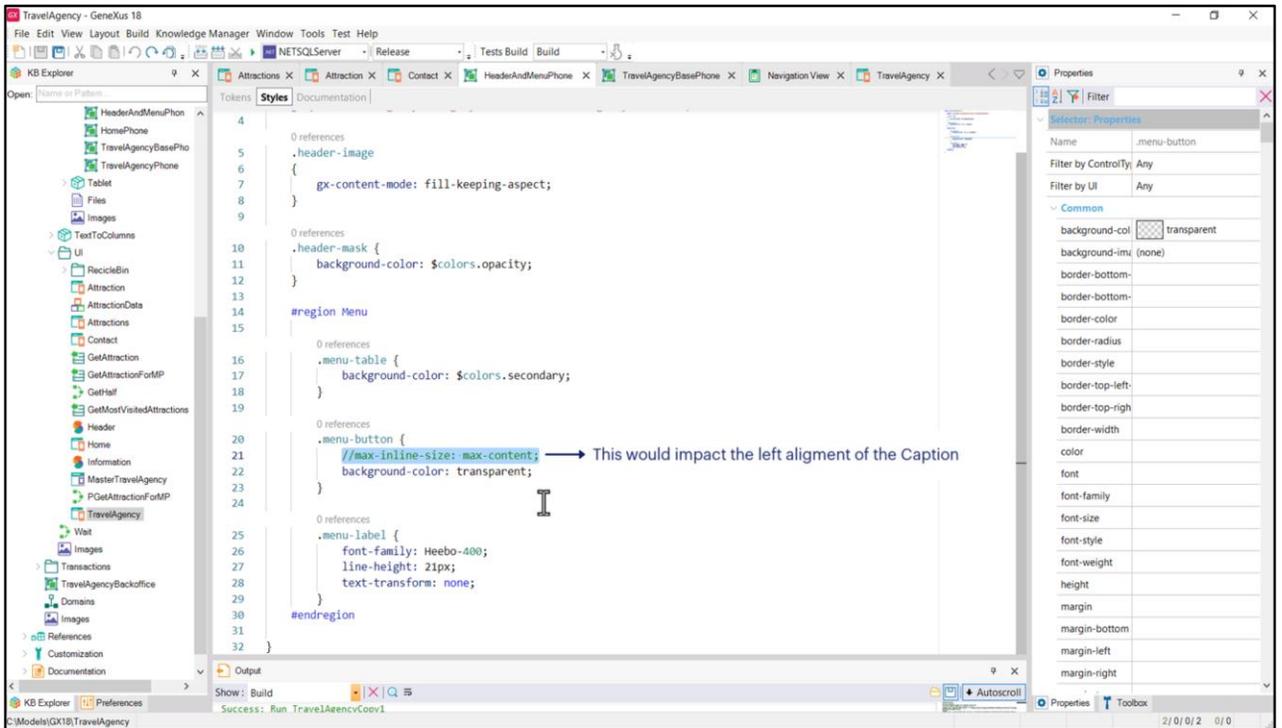
A esta tabla le asocié una clase...



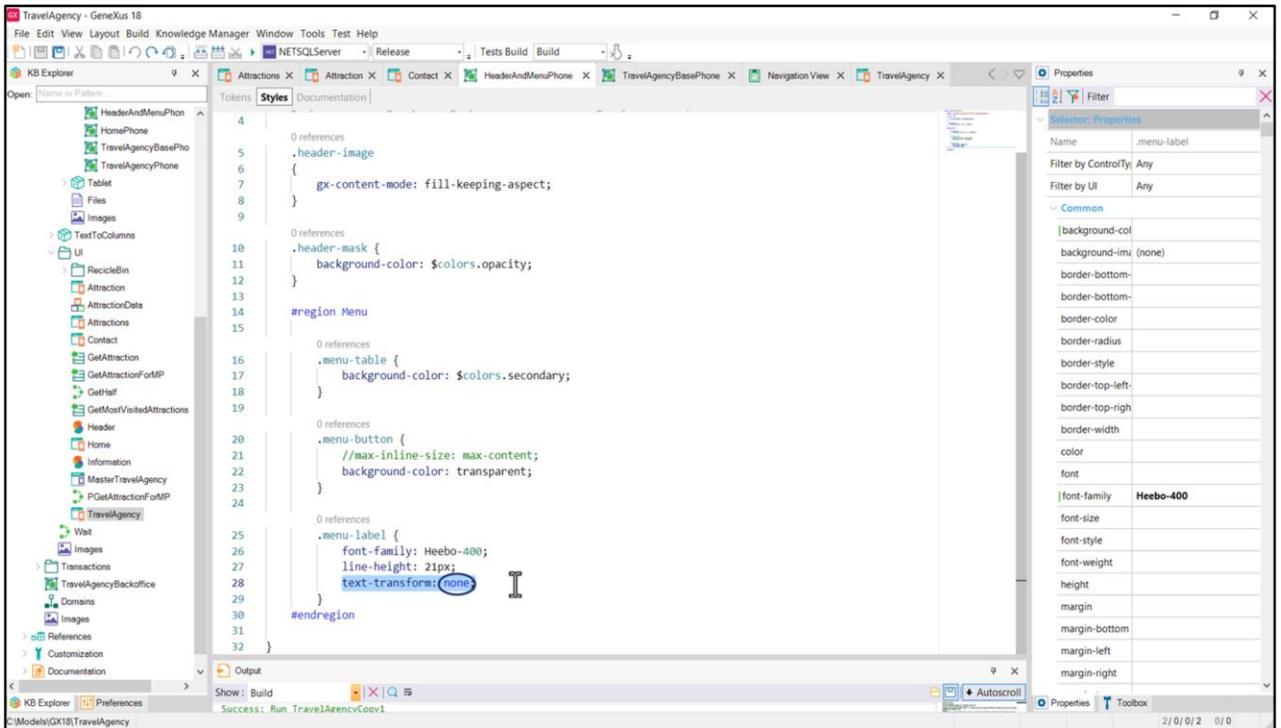
...con el color de fondo que quiero.



Y a los botones les asocié las mismas dos clases que usábamos en la aplicación Angular Desktop (me copié la menu-label para aquí, la quité de TravleAgencyBasePhone).

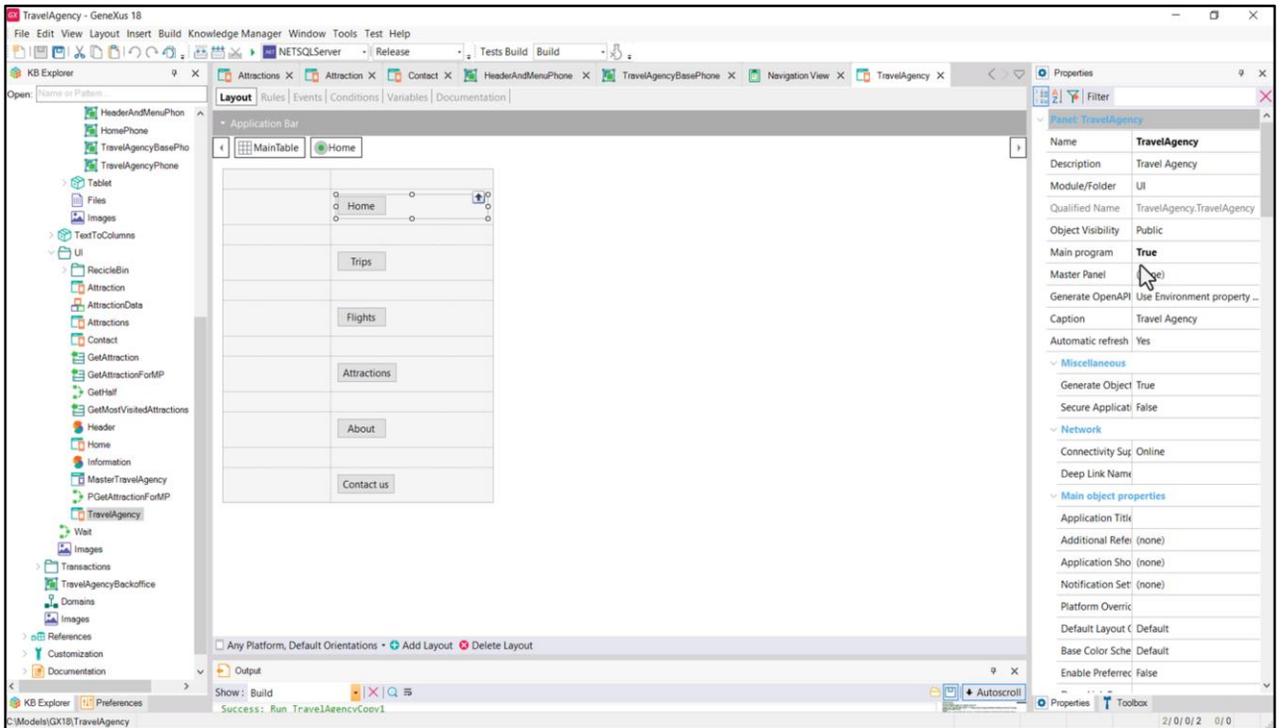


Pero aquí aparecen también algunas diferencias. Por ejemplo, esta propiedad CSS que usábamos para que el botón tuviera como ancho el de su contenido, aquí no funciona.

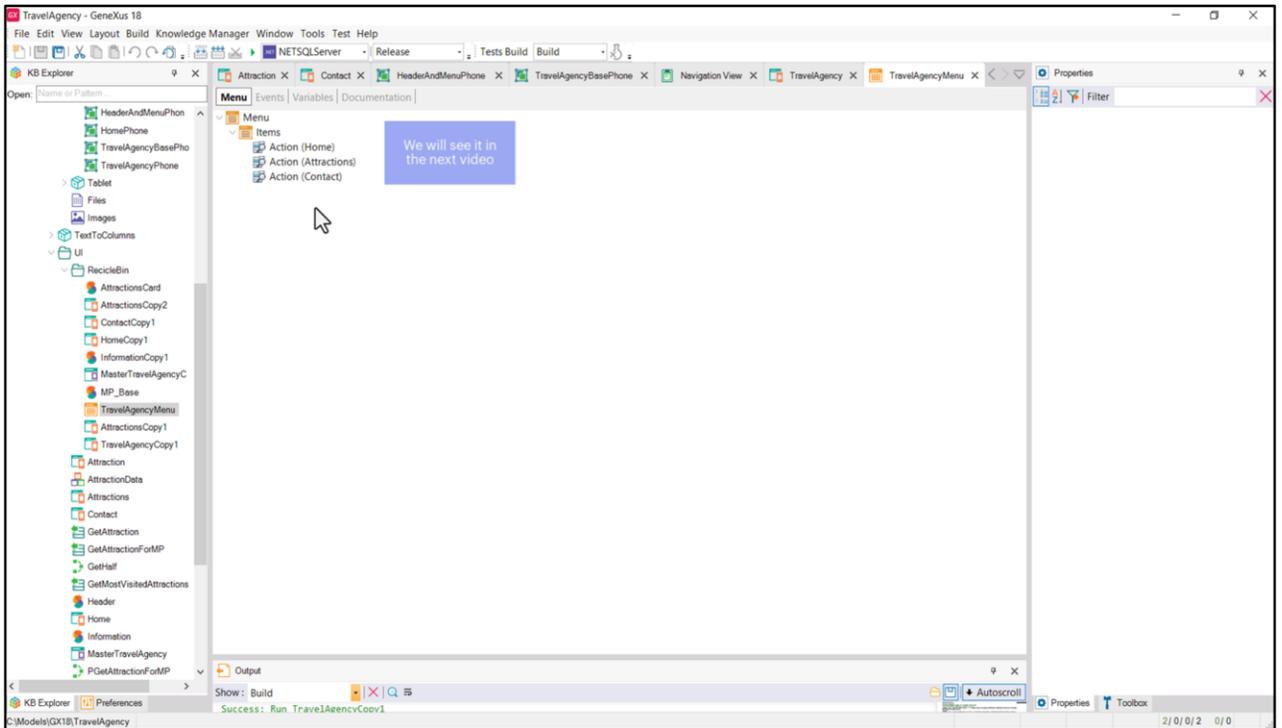


Y por defecto, además, todo botón saldrá con su caption en mayúsculas, así que utilizo la propiedad `text-transform` para que no lo haga. Sí, aquí tendrá efecto la propiedad `text-transform`, pero con este valor.

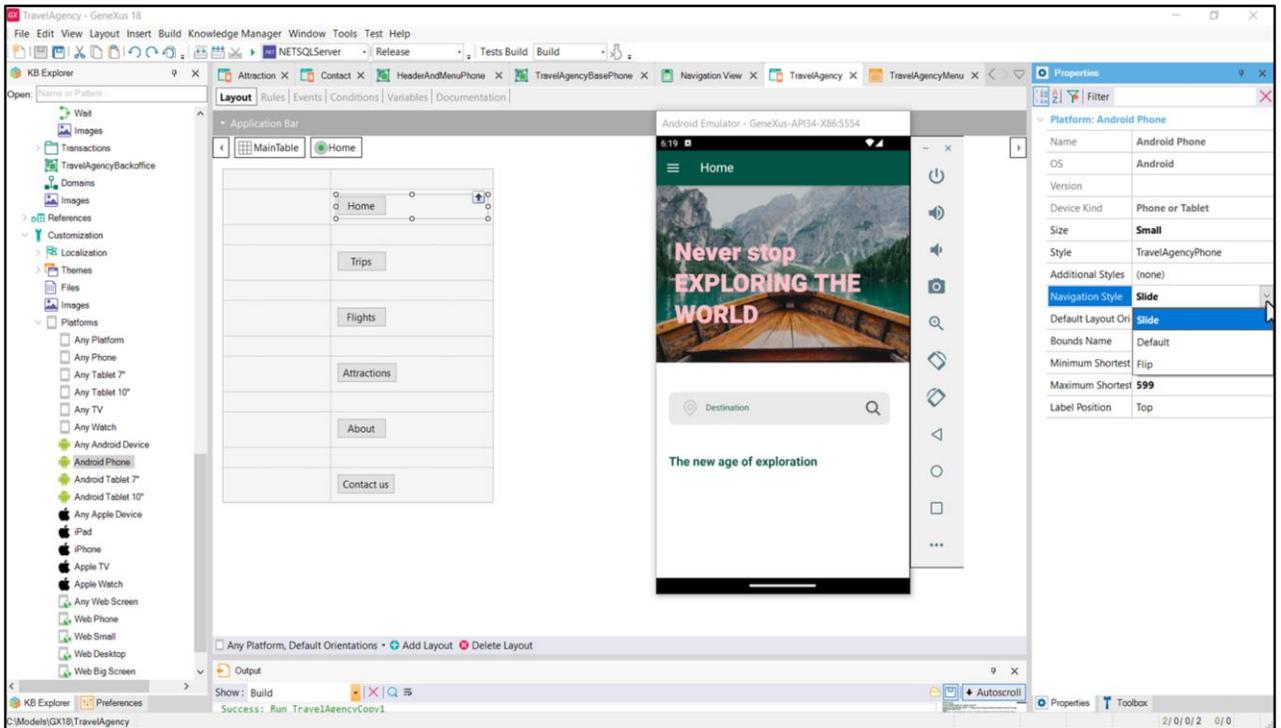
En conclusión: algunas propiedades de CSS valen para lo nativo exactamente igual, otras valen pero para otros valores y otras no valen en absoluto. La idea de GeneXus es ir eliminando todas estas diferencias para que el uso de las propiedades sea lo más transversal posible.



Bueno, tengo entonces este objeto que hará las veces de menú. Por eso mismo tengo que especificarlo como objeto main. Será el punto de entrada de la aplicación. Este será el que se compilará y se subirá al store cuando toda la aplicación esté lista.

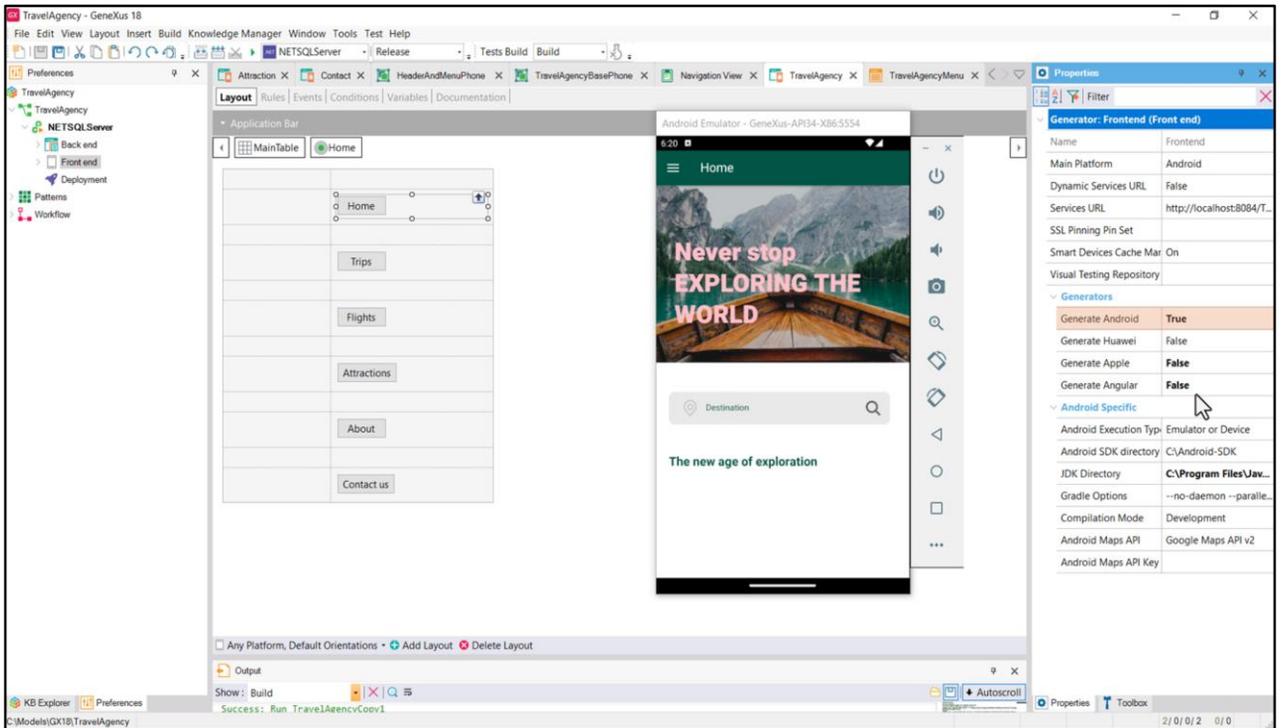


Podría haber utilizado, en lugar de un objeto Panel común y corriente, un objeto de tipo Menu, que ya tiene esta semántica. Como es menos flexible, opté por un panel, que me da toda la flexibilidad.

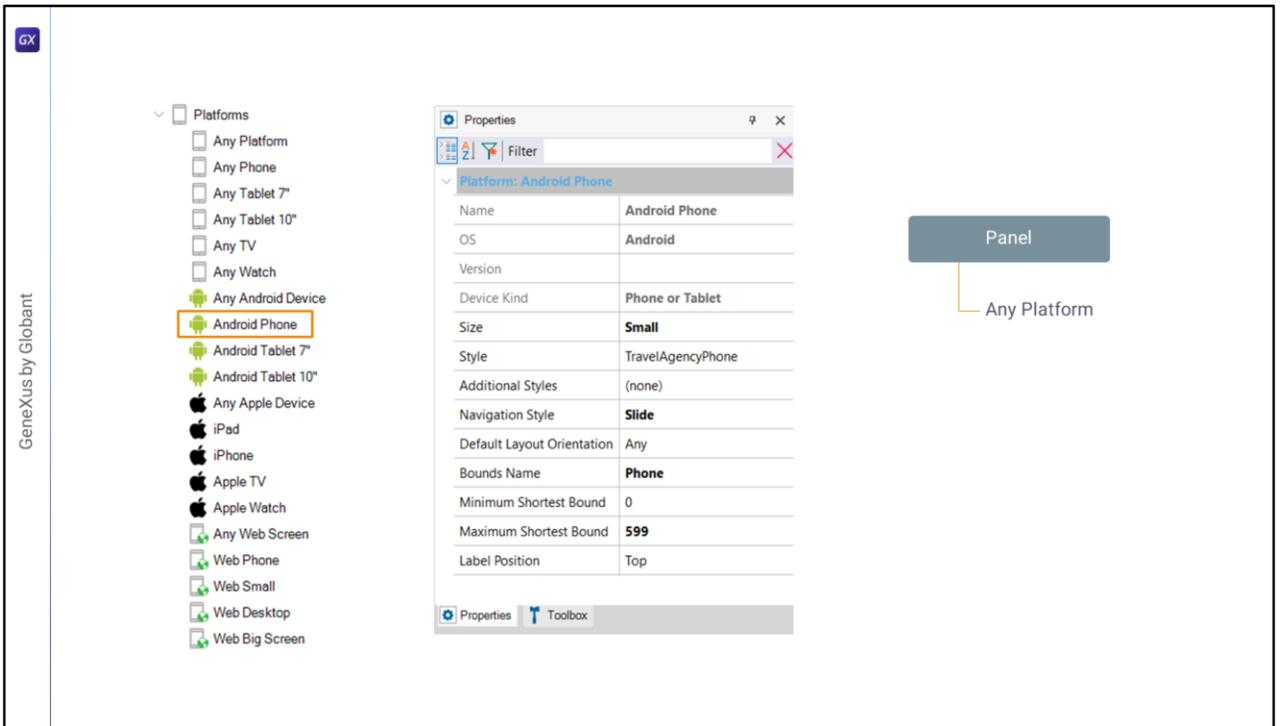


Bien, pero, ¿cómo conseguimos que este objeto haga las veces de menú hamburguesa? Si utilizamos la solución nativa propia del sistema operativo del dispositivo móvil (que para el caso de Android es con menú a la izquierda), será sencillísimo. Alcanza con indicar que el estilo de navegación para la plataforma es Slide.

Fue lo que indiqué para Android Phone. El valor por defecto es Flip.



Por supuesto para probar todo como se los estoy mostrando tuve que prender para el Frontend la generación para Android (y apagar provisoriamente la generación para Angular). Haciendo esto al ejecutar se abre automáticamente el emulador.



Tal vez antes de seguir valga la pena aclarar una cuestión en lo relativo a las plataformas. Si estoy ejecutando en un teléfono Android la aplicación nativa, se tomarán en cuenta las propiedades que estén aquí indicadas. Por ejemplo el DSO y el estilo de navegación. Pero ¿qué layout se tomará de los que cada panel tenga especificados?

El que se acerque más en sus características a las del dispositivo en el que estoy ejecutando. Así, si estoy ejecutando en un teléfono Android un panel para el que tuviéramos estos 4 layouts, ¿cuál utilizará? (esto no se elige en runtime, sino cuando se compila la aplicación). Claramente utilizará el de Any Phone. Si, en cambio, la aplicación se ejecutara en un iPhone, sería este layout el elegido. Y si se ejecuta en cualquiera de los dos, Android Phone o iPhone, pero la aplicación web, será este otro. Si para un panel sólo tuviéramos el layout Any Platform, en cualquier caso se ejecutará ese.

The screenshot shows a web browser window with the URL [training.genexus.com/en/learning/courses/genexus-for-mobile/v18/course-genexus-for-mobile-genexus-18/~26080/first-steps-with-a-native-mobile-app](https://training.genexus.com/en/learning/courses/genexus-for-mobile/v18/course-genexus-for-mobile-genexus-18/~26080/first-steps-with-a-native-mobile-app). The page features the GeneXus training logo and navigation links for Learning, Certifications, Universities, Academic Partners, and Help. A search icon and a 'Try GeneXus' button are also visible.

## GeneXus for Mobile course

Version: GeneXus 18

### First steps with a native mobile app

During the video you will be asked to import a file that will allow you to obtain all the objects from the travel agency's Back-Office. You can find the file [here](#).

Total length of videos: 5h

The main content area displays a video player with the title 'First steps with a native mobile app'. The video interface includes a play button, a progress bar, and a list of video chapters:

- Introduction
  - Features of mobile applications
- Architecture
  - Online Applications Architecture
  - Architecture of offline applications
- First steps with a Mobile application
  - First steps with a native mobile app
  - Prototyping a native mobile app
- UI Design/UX
  - Design of a mobile application
  - Design System of a Mobile Application

The video player also shows a menu with options like 'GeneXus Trial' and 'GeneXus Full', and a list of tools including 'Extensions Manager', 'Application Integration', 'Workflow', 'Options', 'Advanced', 'Explore Knowledgebased Directory', 'Explore Target Environment Directory', 'CMD Environment Directory', 'GeneXus Access Manager', 'Update Android SDK', and 'GeneXus Account'.

A todos aquellos que no hayan prototipado nunca desde GeneXus un desarrollo nativo, les recomiendo este video de nuestro curso GeneXus for Mobile. De hecho les recomiendo todo el curso, que es bastante breve.

GeneXus for Mobile V18 Course

training.geneXus.com/en/learning/courses/geneXus-for-mobile/v18/course-geneXus-for-mobile-geneXus-18/~26080/navigation-styles-of-a-mobile-application

Multitexperience | GeneXus | DL Portal | Issues

## Navigation styles of a mobile application

Mobile applications developed with GeneXus can be navigated according to four styles: Flip, Split, Slide and Cascade. The style is configured per platform. Start events associated with navigation styles are also studied.

Total length of videos: 5h

Slide  
Split  
Cascade  
Flip  
Tabs

.Start()

Any Apple Device  
iPad  
iPhone  
Apple TV  
Apple Watch

Slide.Start()

Watch on YouTube

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

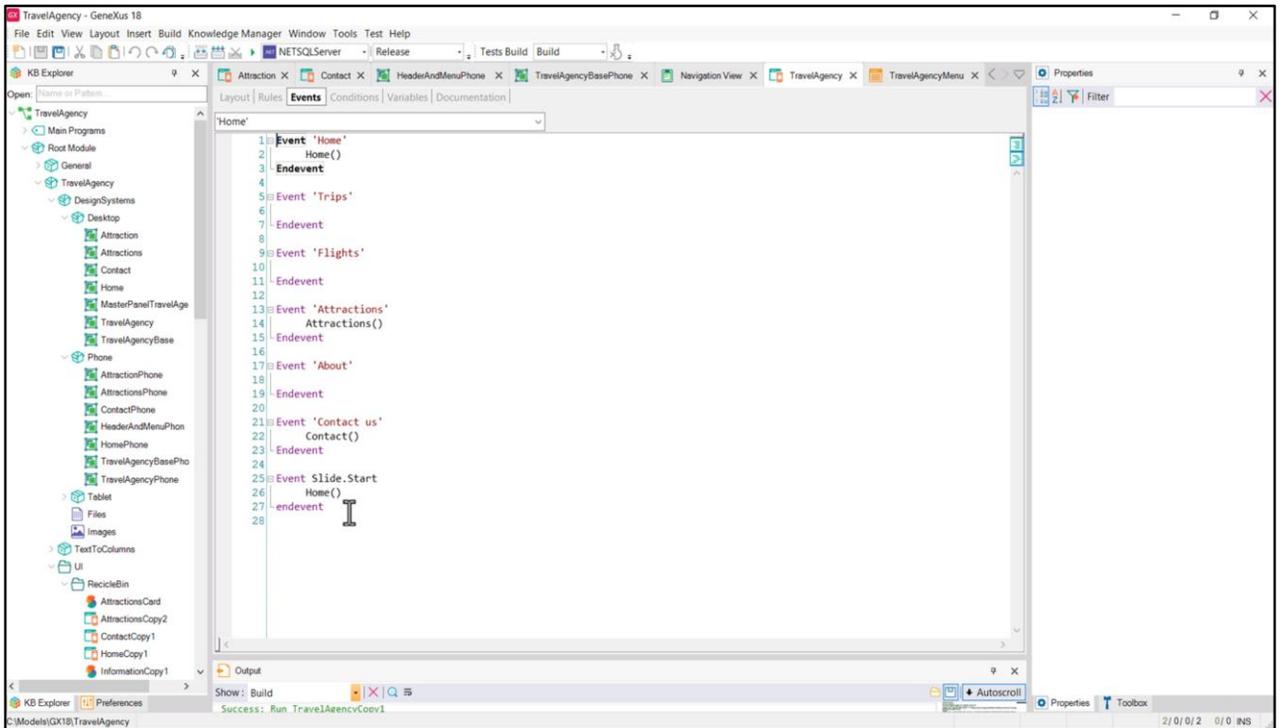
- Introduction to Offline Applications
- Generation of the Offline Database object
- Offline Database Synchronization

Integration

Video transcript

Navigation styles of a mobile application - PDF

En particular aquí podrán ampliar esto de los estilos de navegación. Justamente, como allí podrán ver, necesité utilizar el evento Slide.Start...



...para poder indicar cuál objeto abrir cuando se ejecute el objeto main que hará de menú. Cuando el estilo de navegación es Slide, justamente.

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

## 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.

Total length of videos: 5h

```
Event 'NewEvent'
...
B CallOptions.Type = CallType.Push
B(&param)
...
EndEvent
```

Watch on YouTube

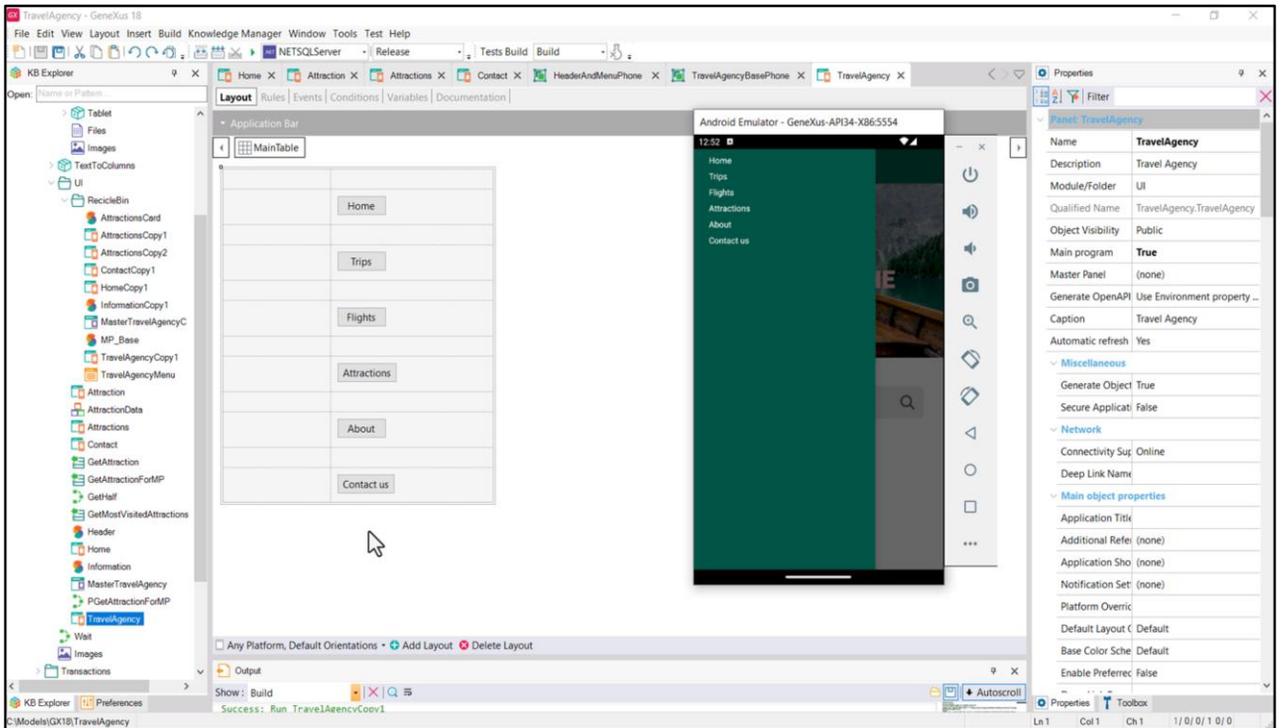
Video transcript

Invocations between Mobile objects - PDF

- Multiple Layouts in a und
- 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

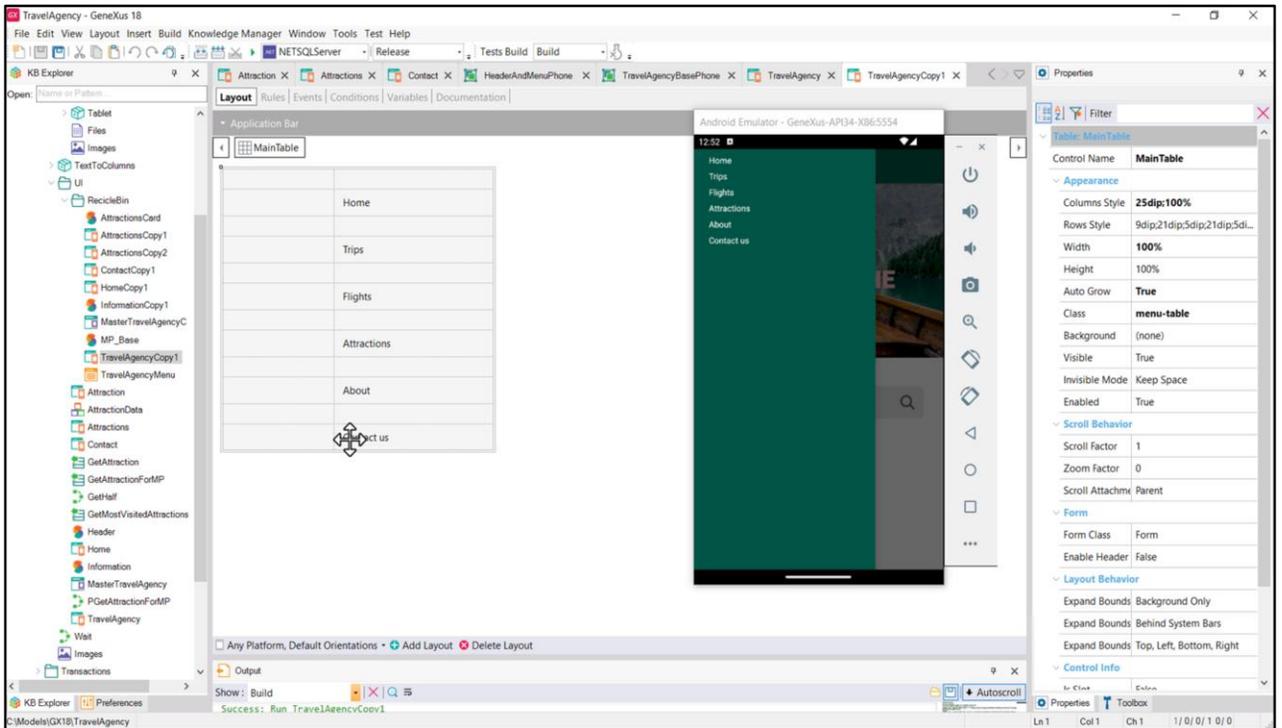
También les recomiendo ver este otro video que será de gran utilidad para ver las distintas posibilidades para las invocaciones. Por ejemplo, vamos a ver en el próximo video una posible implementación manual del menú hamburguesa por la derecha. No nos terminará sirviendo para esta funcionalidad en particular, pero valdrá en general para abrir ventanas por la derecha cuando el estilo de navegación es a pantalla partida.

# Some Tricks

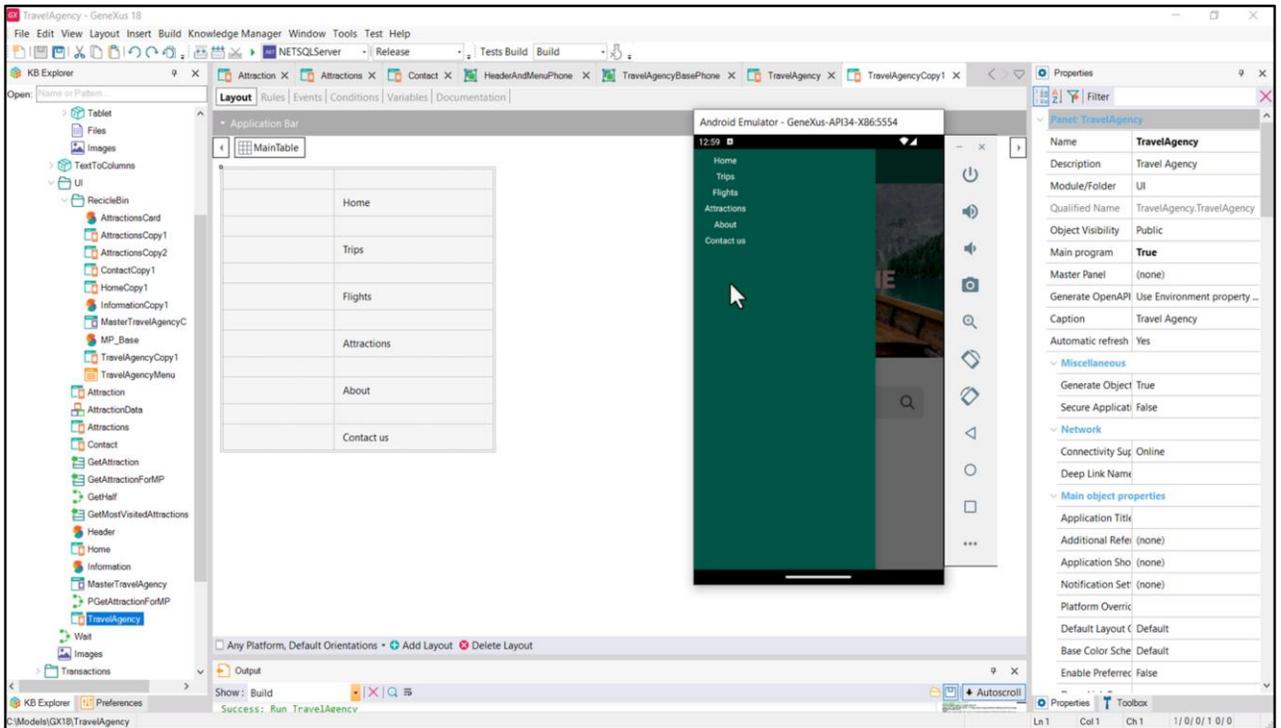


A lo largo del curso me escucharon repetir que por accesibilidad todo lo que sean acciones debería implementarse como botones.

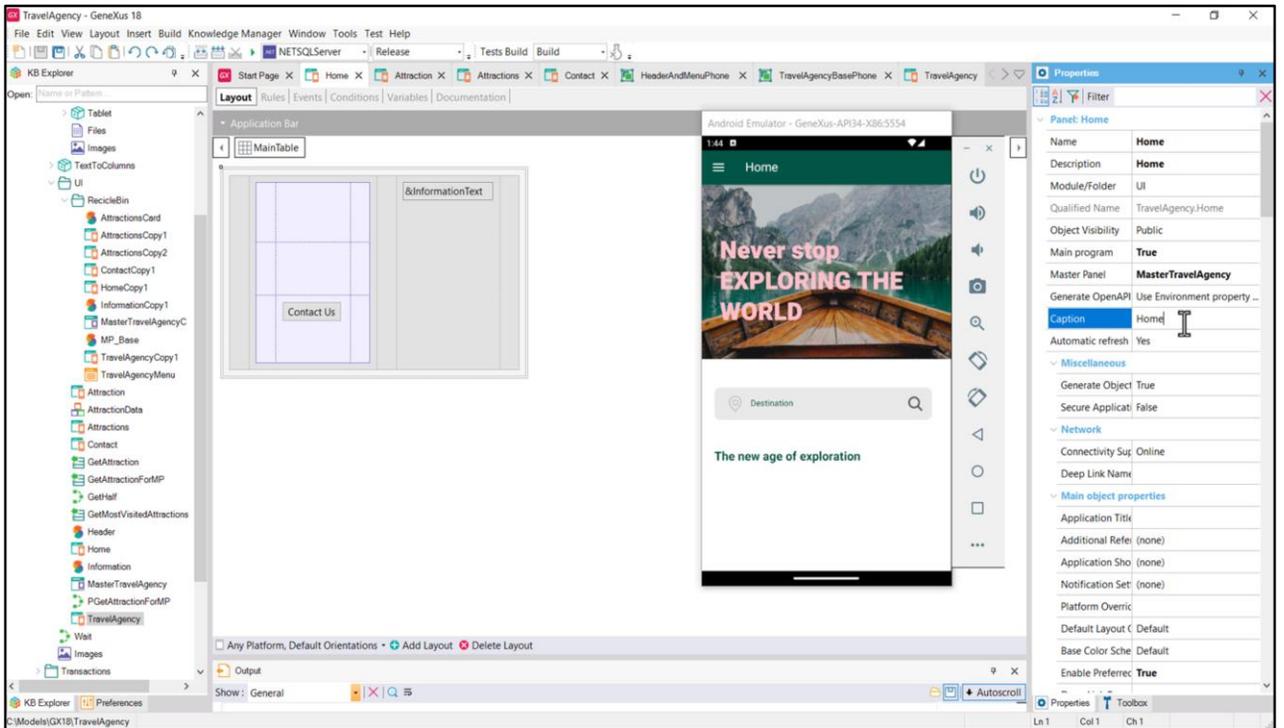
Por eso les mostré este menú con botones.



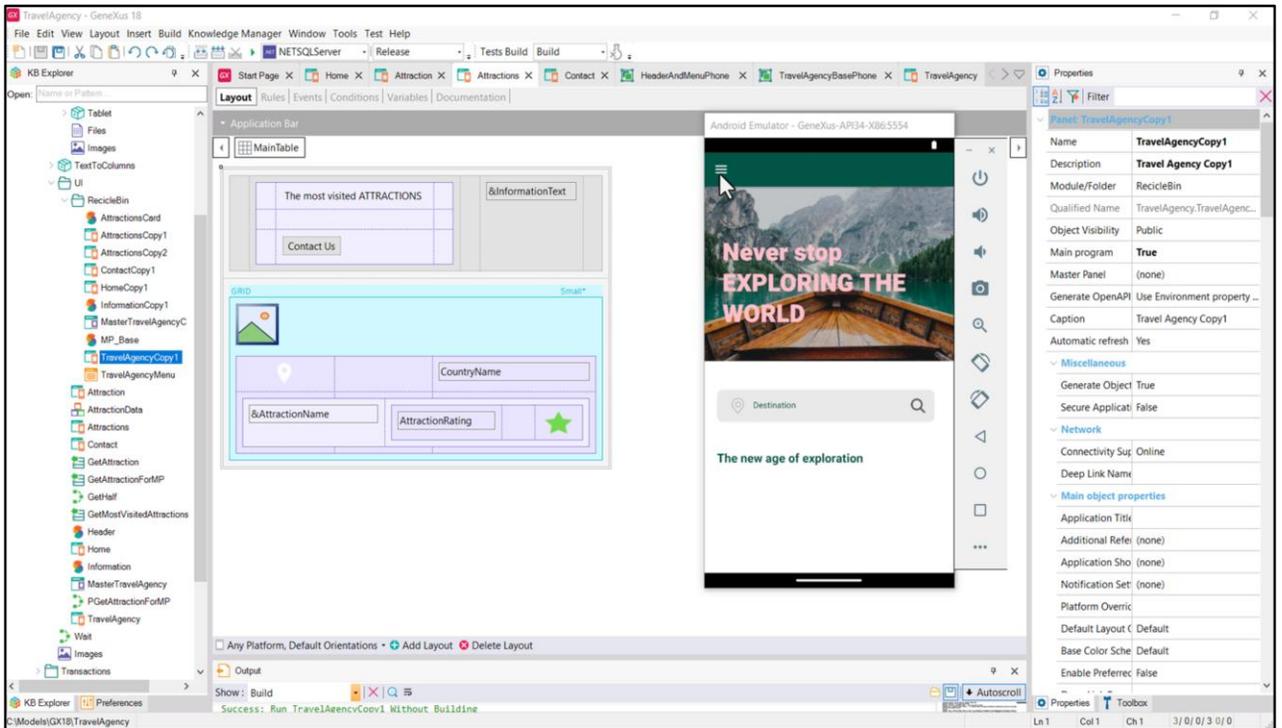
Pero en verdad el que vieron ejecutarse no es este, sino este otro con Textblocks en lugar de botones.



Es que la solución con botones me queda así: es decir, como no conseguí hacer que los botones tuvieran como ancho el del caption del botón, y tampoco puedo decirles que el caption se alinee a la izquierda, no tengo manera de alinear sus textos a la izquierda. Esto se corregirá a futuro.

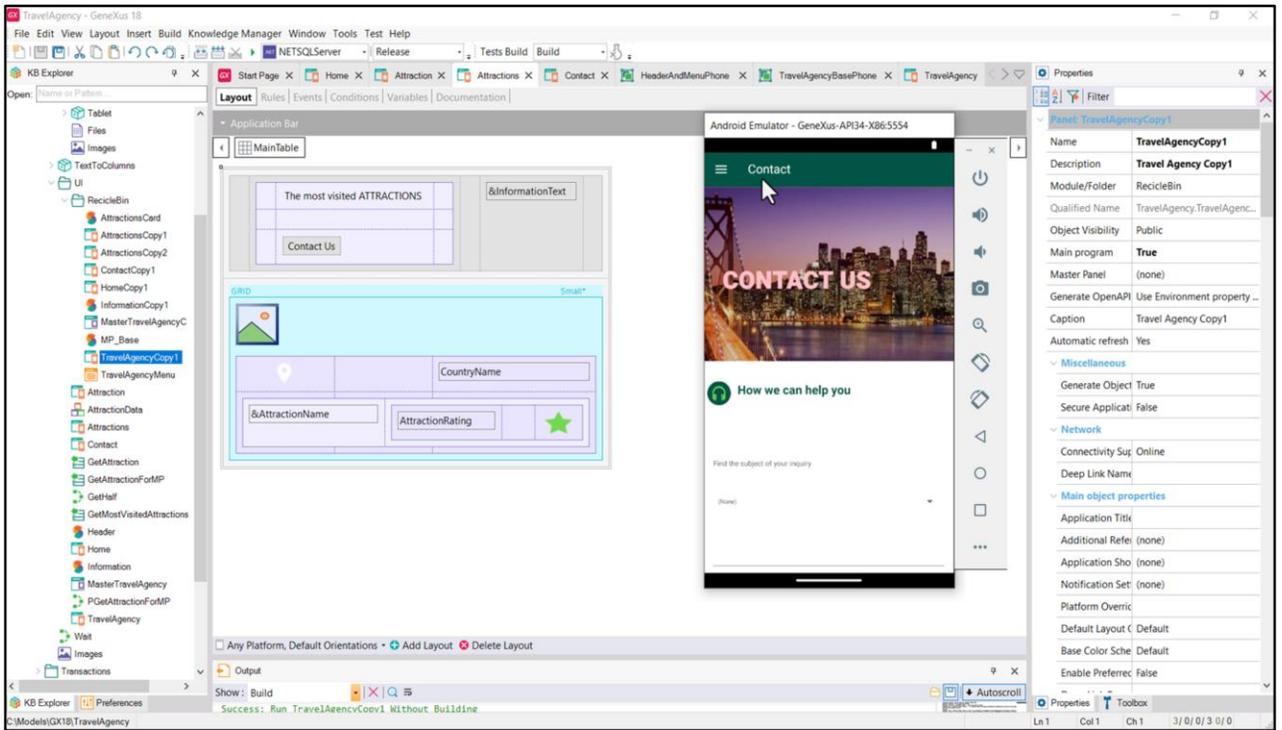


Vean ahora este texto que está cargándose por defecto en la Application Bar. Se toma de la propiedad Caption del panel.

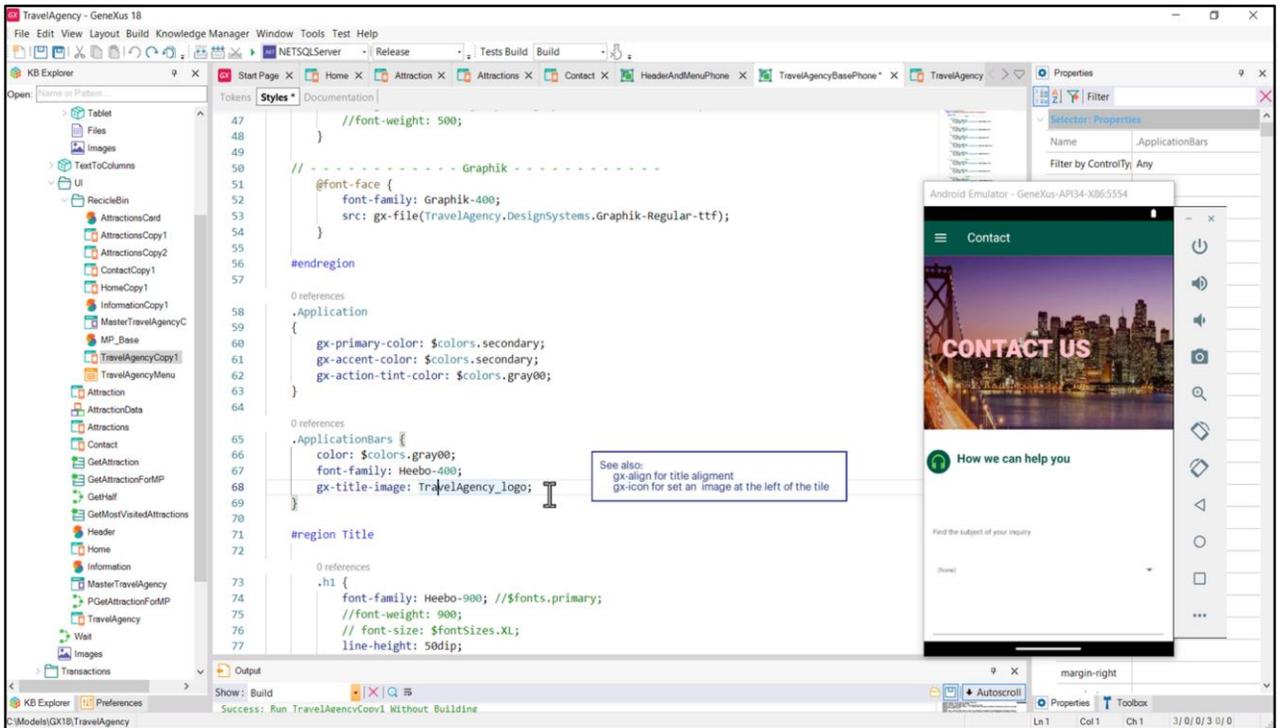


Voy a quitarla para que vean... Y voy a quitarla también para el de Attractions, pero no para los otros dos.

Ejecutamos y lo vemos...

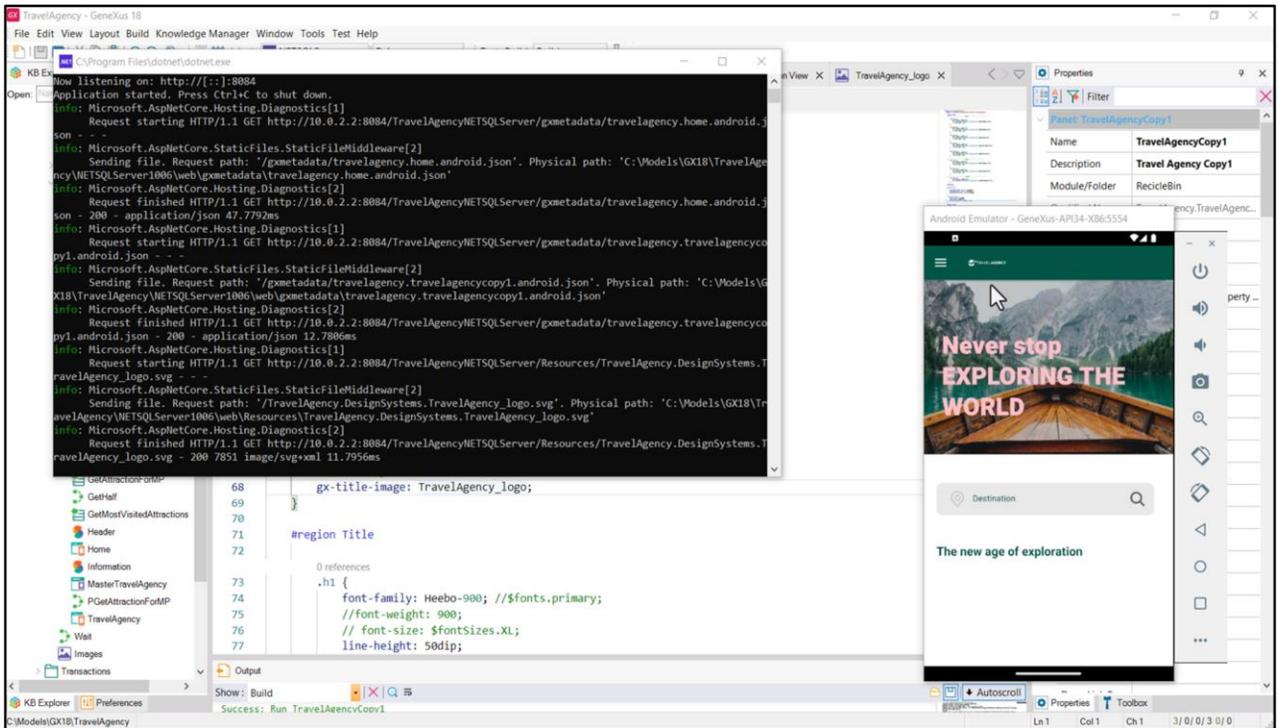


En cambio para Contact... bien.

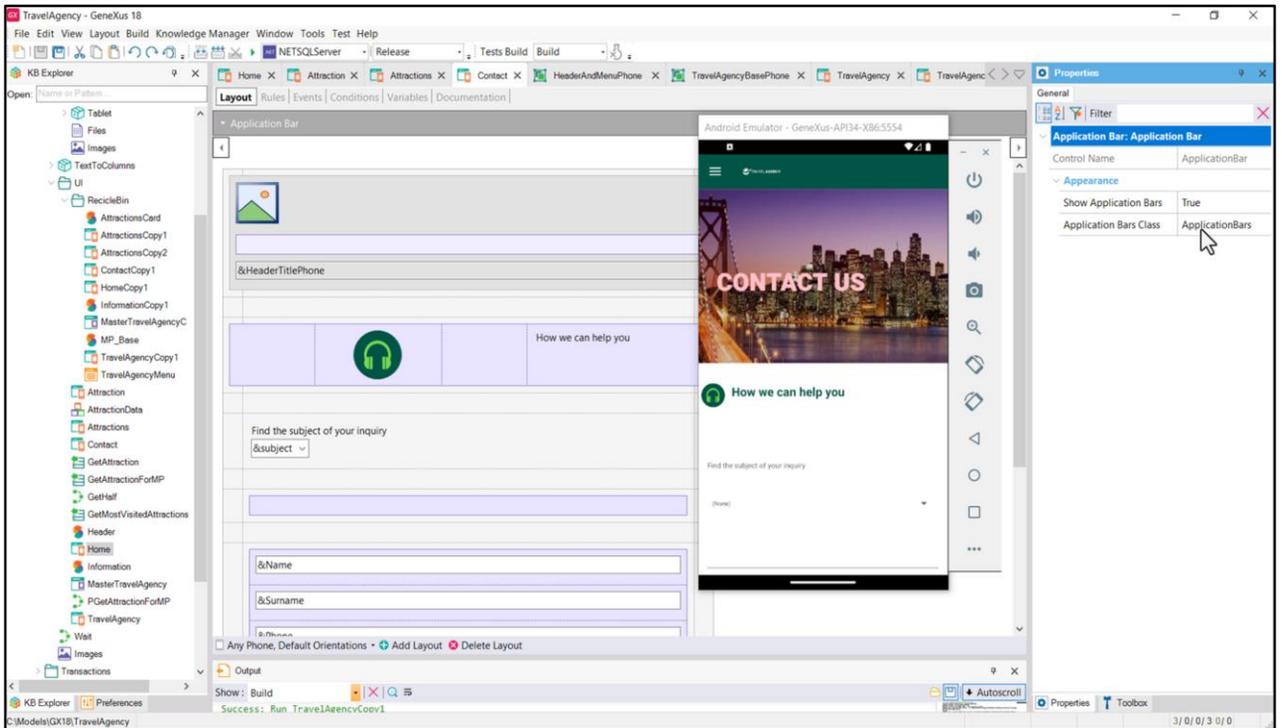


Si ahora queremos que en la Application Bar se muestre una imagen, entonces podemos utilizar en la clase ApplicationBars esta propiedad. Este es el nombre de la imagen que tomé de Figma e inserté en la KB.

Como es blanca no la vemos, pero lo que hice, como se imaginarán, fue exportarla de Figma...



Si ahora ejecutamos... la vemos en todos los paneles.



Algo que omití mostrarles es que la clase ApplicationBars es la que podemos ver asociada a los paneles, a través de estas propiedades.

Application Bar control in Native+ Mobile+ Applications

Application Bar.

Layout Rules Events Conditions Variables

+ Application Bar Update Delete

Insert Button...  
Insert Group

Application Bar section

Form section

Id CompanyId

Name CompanyName

Country Id CountryId

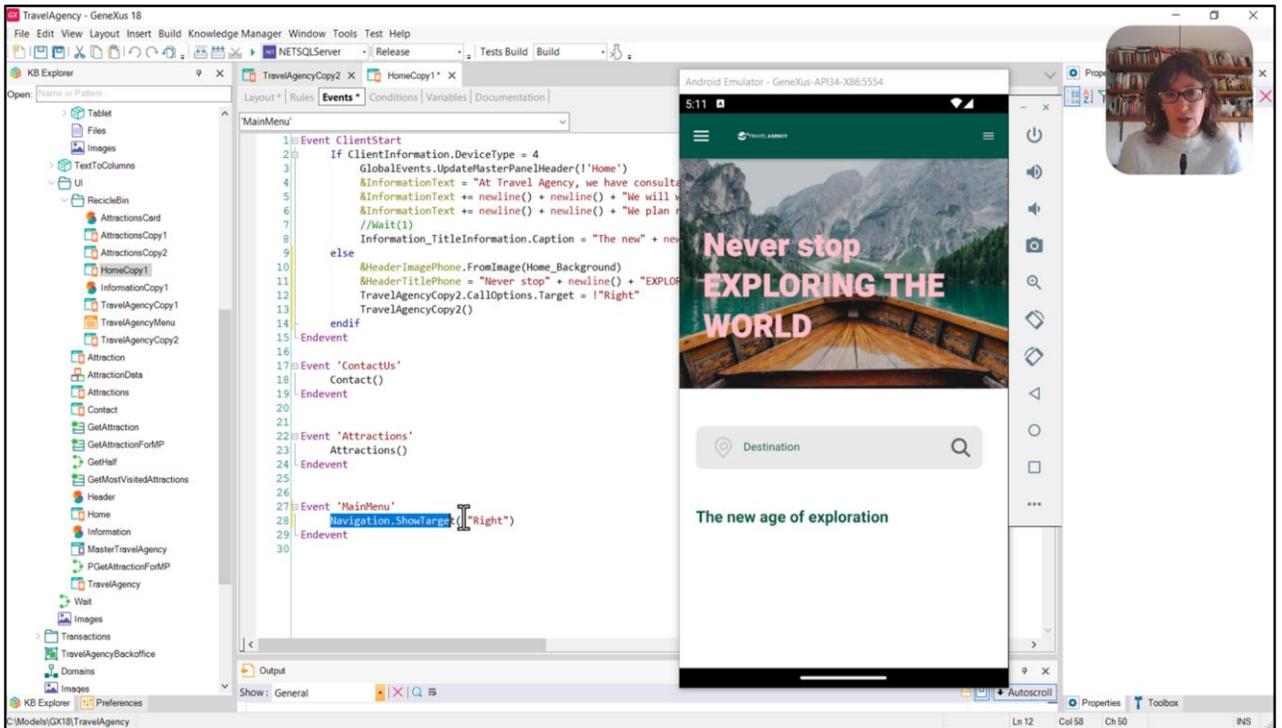
Country Name CountryName

Country Flag

Note that the Application Bar section can contain buttons ([actions](#)) and by right-clicking on them you can add more [Button controls](#) or [Action group controls](#). Also, they can be dragged from the toolbox once the Application Bar section is selected.

Ask here!

Por supuesto allí podrán colocarse botones para que se muestren allí mismo o como menús contextuales.



En el próximo video retomaremos esto y veremos cómo agregar un botón a la Application Bar con la imagen del menú hamburguesa, y cómo haremos para que el menú se despliegue a la derecha.

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

[training.genexus.com](https://training.genexus.com)