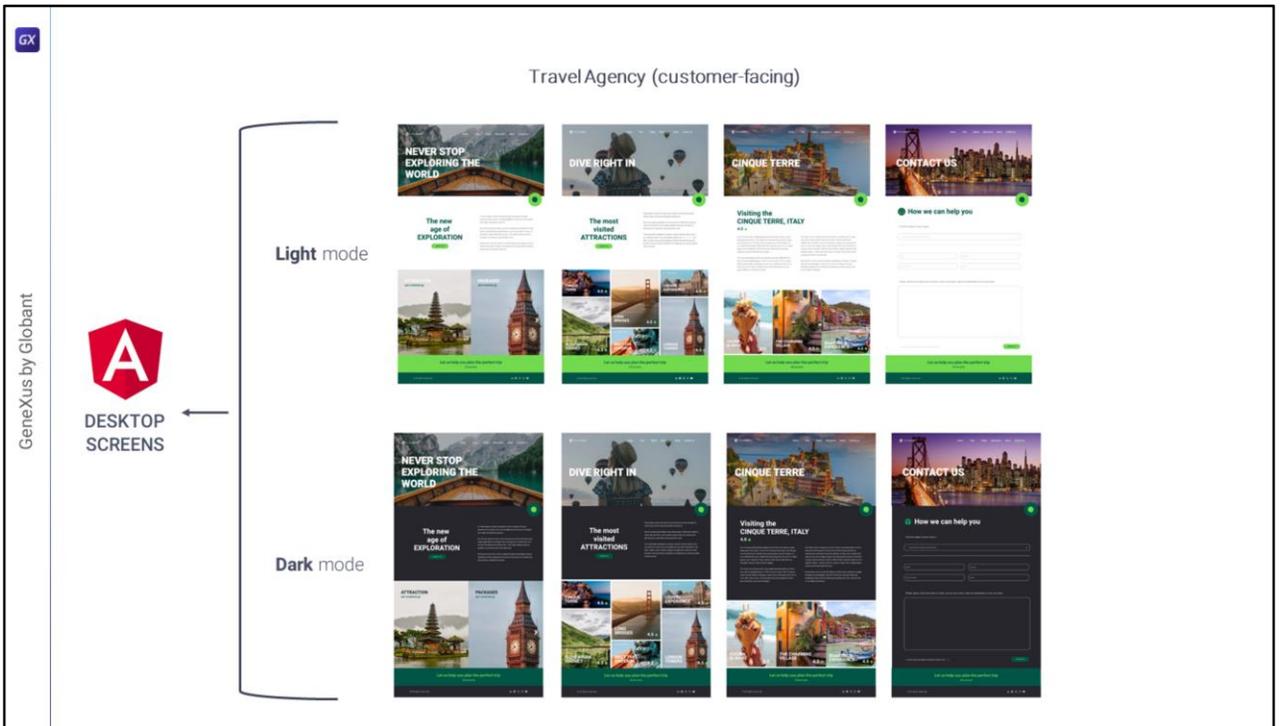


# Multiexperience. Summary and Final Notes



Cecilia Fernández



En los módulos anteriores a este nos dedicamos a interiorizarnos en cómo implementar desde cero una aplicación Angular para tamaño de pantalla desktop (tanto normal como big screen) y en sus modos light y dark.

# Angular



### Desktop screens



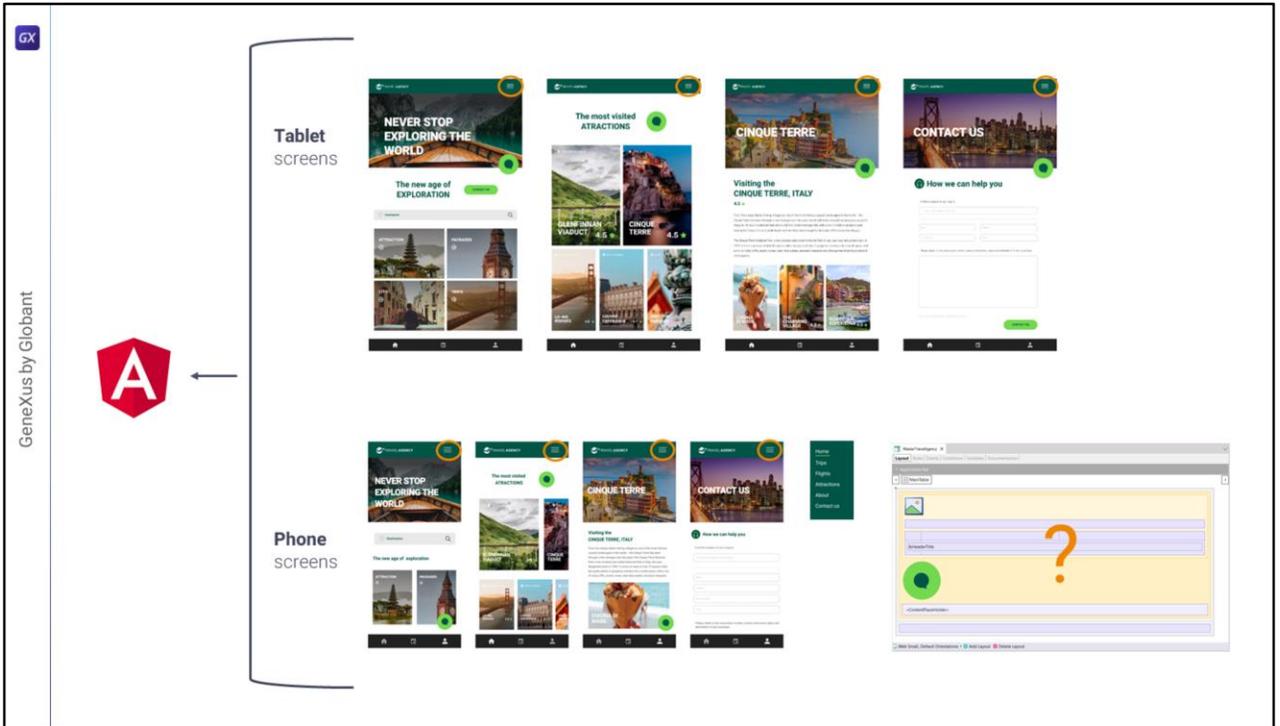
### Tablet screens



### Phone screens



Con lo visto allí (y sin meternos por ahora en el mundo nativo) es fácil de pensar la solución para los tamaños Tablet y Phone, siguiendo exactamente la misma lógica que veníamos usando.

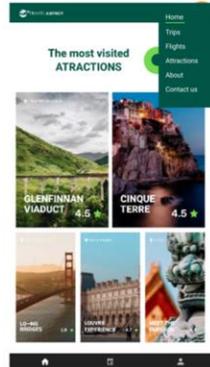
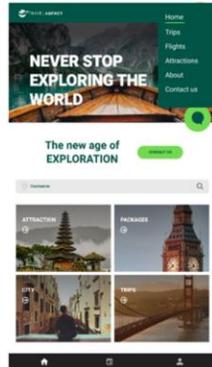


Lo que habría hecho falta pensar, en ese caso, es cómo implementaríamos el menú hamburguesa dentro del Master Panel.



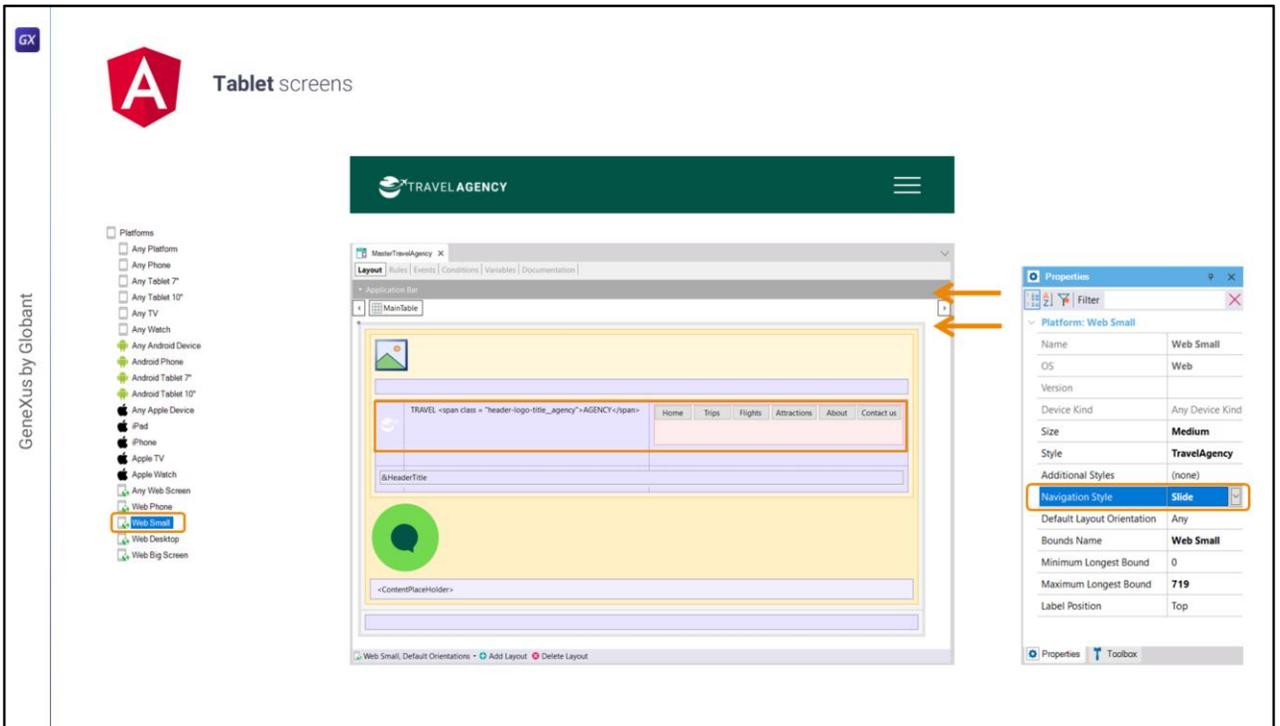
## Tablet screens

- Platforms
- Any Platform
  - Any Phone
  - Any Tablet 7"
  - Any Tablet 10"
  - Any TV
  - Any Watch
  - Any Android Device
  - Android Phone
  - Android Tablet 7"
  - Android Tablet 10"
  - Any Apple Device
  - iPad
  - iPhone
  - Apple TV
  - Apple Watch
  - Any Web Screen
  - Web Smart
  - Web Desktop
  - Web Big Screen



Por ejemplo, para el tamaño Tablet...

Seguramente el menú de tabs no irá (habría que hablar con la diseñadora pero lo más probable es que para web no vaya ese menú)... y lo único, entonces, que respecto a todo lo que estudiamos en los módulos del 1 al 4 no vimos sería cómo implementar el menú hamburguesa en el Master Panel.



Claramente tendríamos que quitar esta fila de la tabla, dado que eso ahora estará en la Application Bar, o en una fila 1 de la Main table, depende de cómo lo implementemos.

Podríamos pensar en 2 alternativas de implementación: una automática y una manual:

La **automática** sería análoga a la que vimos para Android, es decir, la provista por el estilo de navegación Slide. Lo que se vio allí vale para Angular, lo que vimos en los videos anteriores.

Pero no nos va a servir, porque en este caso tendríamos que cambiar dinámicamente el objeto main de la aplicación, lo que no es posible.

GeneXus by Globant

Tablet screens

Platforms

- Any Platform
- Any Phone
- Any Tablet 7"
- Any Tablet 10"
- Any TV
- Any Watch
- Any Android Device
- Android Phone
- Android Tablet 7"
- Android Tablet 10"
- Any Apple Device
- iPad
- iPhone
- Apple TV
- Apple Watch
- Any Web Screen
- Web Phone
- Web Desktop
- Web Big Screen

MasterTravelAgency X

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

TRAVEL <span class = "header-logo-title\_agency">AGENCY</span> Home Trips Flights Attractions About Contact us

&HeaderTitle

<ContentPlaceholder>

Any Platform, Default Orientations Add Layout Delete Layout

NEVER STOP EXPLORING THE WORLD

The new age of EXPLORATION

Attractions Packages

Let us help you plan the perfect trip.

Main: Home Panel

Recordemos que para el tamaño desktop (o mayor) el menú estaba integrado dentro del Master Panel como una parte indistinguible del resto, por lo que el objeto main era el objeto Home.

GeneXus by Globant

Navigation Style: Slide

Navigation Style: Default

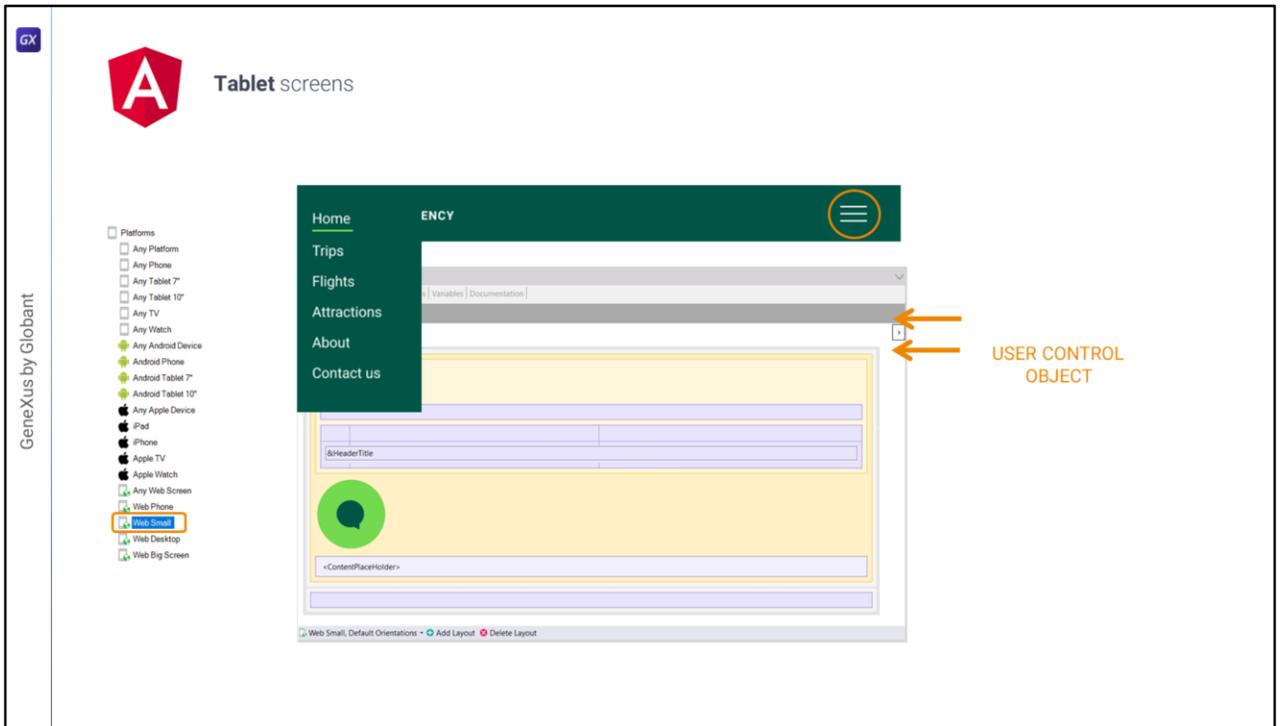
Platforms:

- Any Web Screen
- Web Phone
- Web Small
- Web Desktop
- Web Big Screen

Main: TravelAgency Panel

Main: Home Panel

Piensen en qué pasará cuando se ejecute la misma aplicación Angular en una Tablet o Laptop de ese tamaño. No sólo tendríamos que cambiar el estilo de navegación, sino que además tendría que ser el menú el objeto main. Y esto no es posible, puesto que la aplicación es la misma.



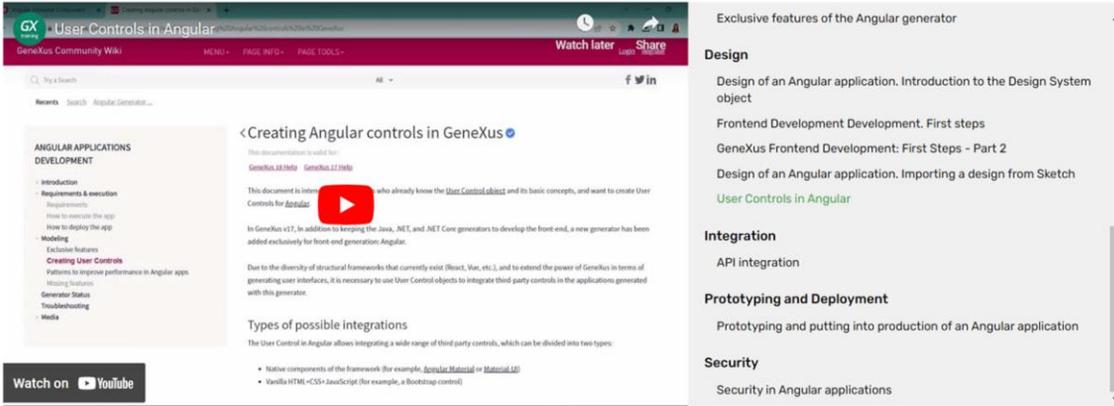
Entonces sólo nos queda la solución **manual**. Y allí tenemos varias alternativas. A las alternativas que pudimos ver para Android, aquí se nos suma la posibilidad de implementar el menú hamburguesa a través de un User Control, ya sea desarrollado por nosotros mismos, como por un tercero...

GeneXus for Angular Course

training.genexus.com/en/learning/courses/genexus/v18/angular/material/user-controls-in-angular-6105551

GeneXus DL Portal Issues

We have already seen several screen controls that help us build the user interface, we have also seen how to improve the design of the application by making definitions in a Design System object and how to import an early design made by a designer in Sketch. In this video we will see how, in addition to the predefined screen controls that we have available in the toolbar, we can create our own controls to further enrich the user experience.



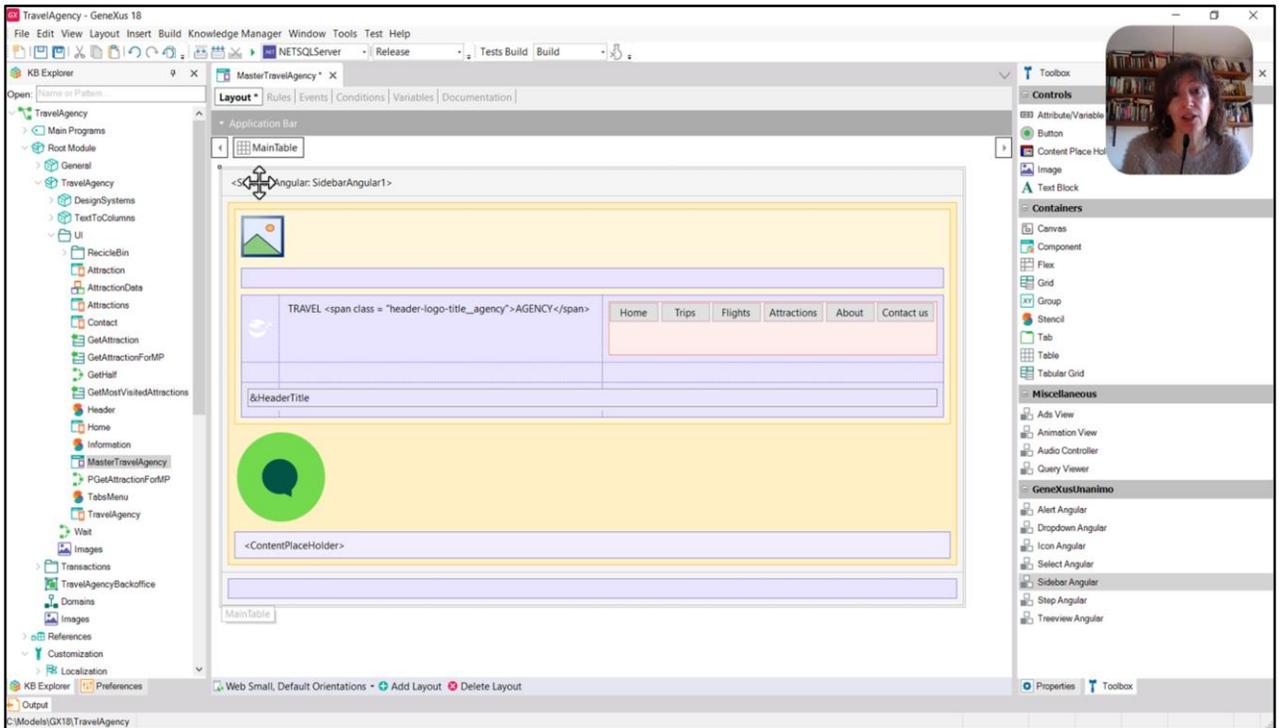
Exclusive features of the Angular generator

- 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
  - [User Controls in Angular](#)
- Integration
  - API integration
- Prototyping and Deployment
  - Prototyping and putting into production of an Angular application
- Security
  - Security in Angular applications

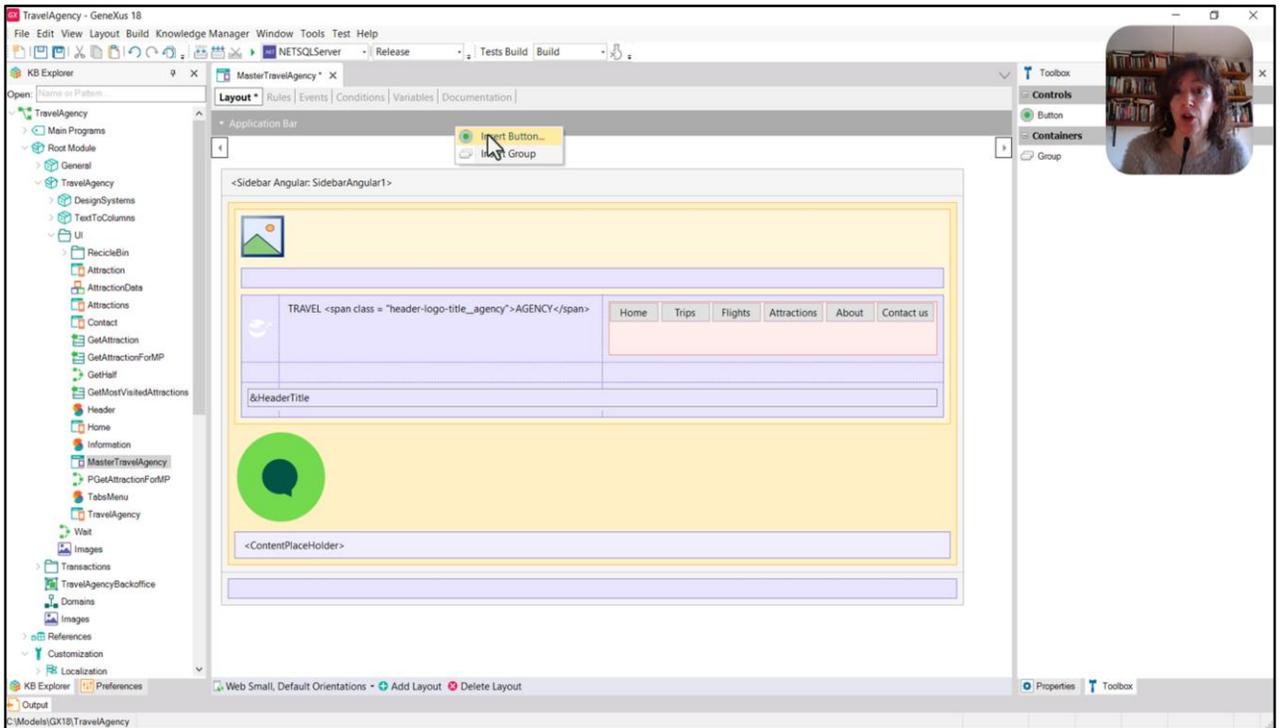
Video transcript

[User Controls in Angular - PDF](#)

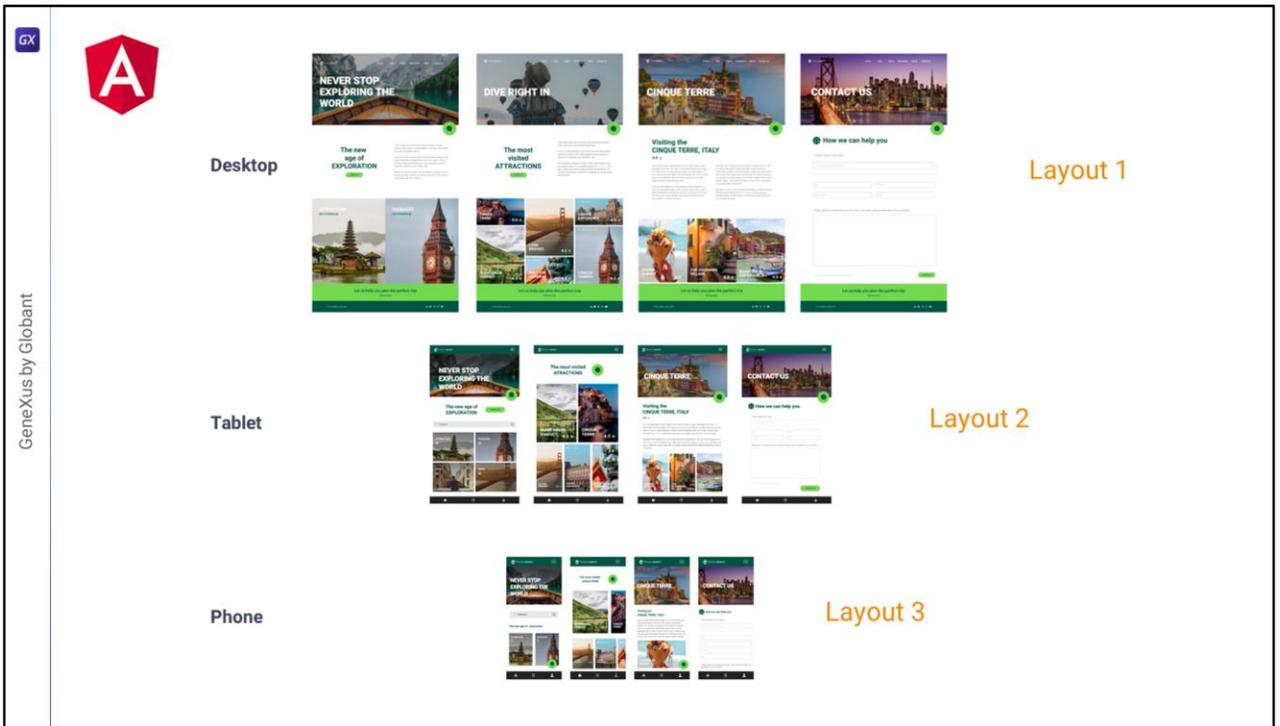
...para lo cual necesitaremos copiarnos de ese desarrollo el código html, colocarlo dentro del objeto User Control, hacer un par de modificaciones y ya podremos utilizarlo dentro de nuestra KB.



...o, incluso, podríamos tratar de utilizar el que ya nos provee el Design System Unanimo y que aparece a disposición en la toolbox.

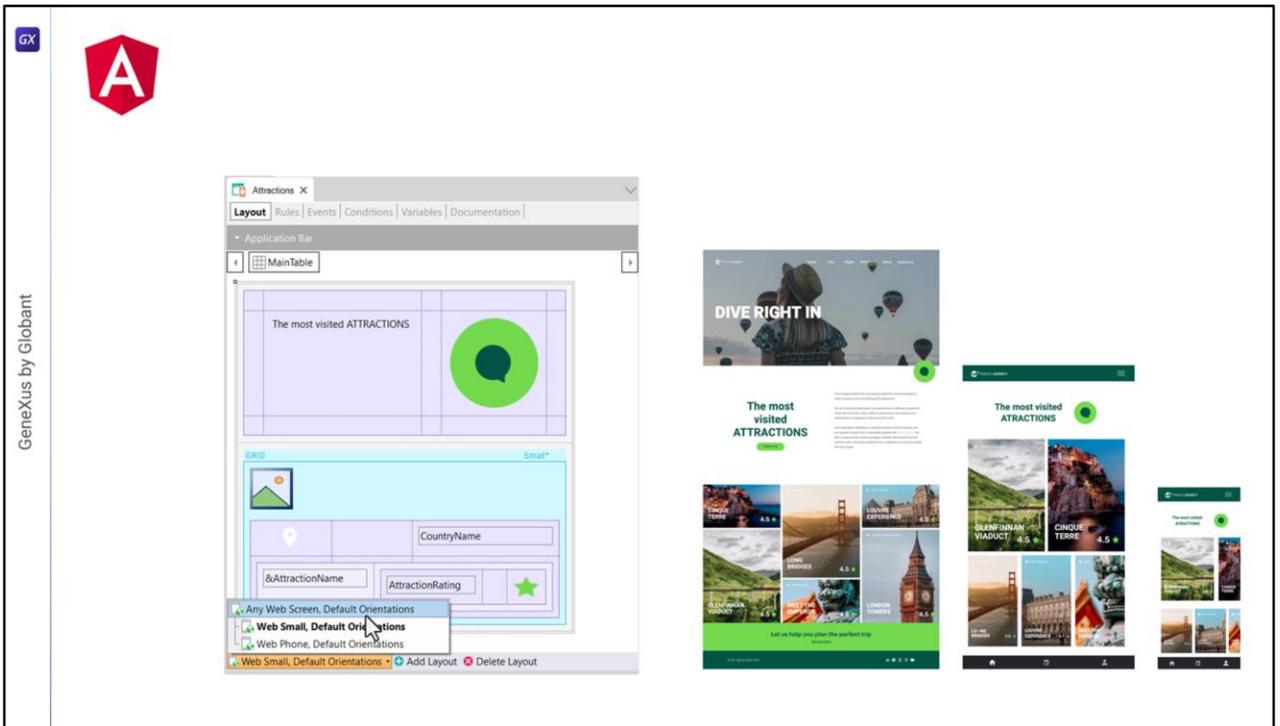


Para estos casos tendremos que emular la Application Bar en la fila 1 de la Main Table, porque en la Application Bar solamente se pueden colocar como controles, controles de tipo botón o Action Groups. No podemos colocar ninguna otra cosa, en particular no podemos colocar un User Control.



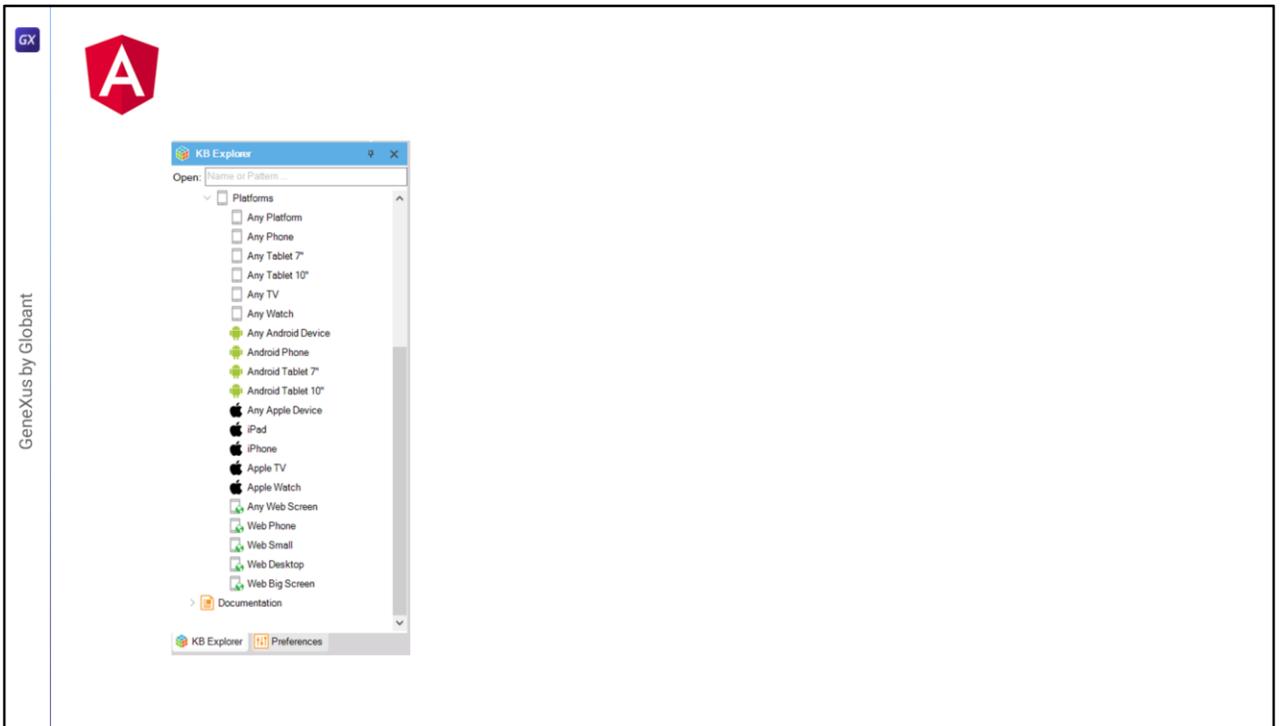
En definitiva, para tener la solución completa Angular, haríamos lo siguiente:

Teniendo las pantallas para Desktop, agregaremos un layout a cada panel (y al Master Panel) para los dos breakpoints restantes: Tablet y Phone.

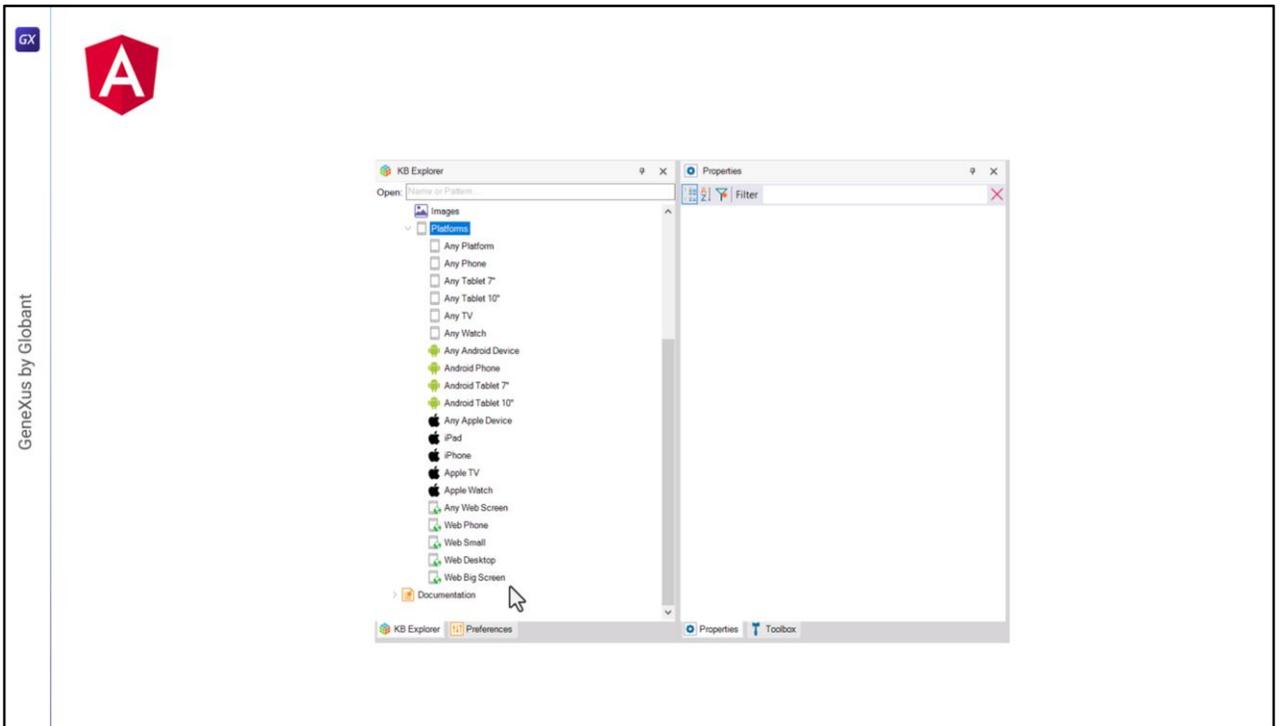


Por ejemplo, para el panel Attractions tendremos 3 layouts: el que es para tamaño Phone, el que es para tamaño Tablet que corresponde a la plataforma Web Small y el restante que es para toda pantalla Web que no sea ninguna de las otras dos, por lo que para tamaños Desktop y Big Screen será la utilizada.

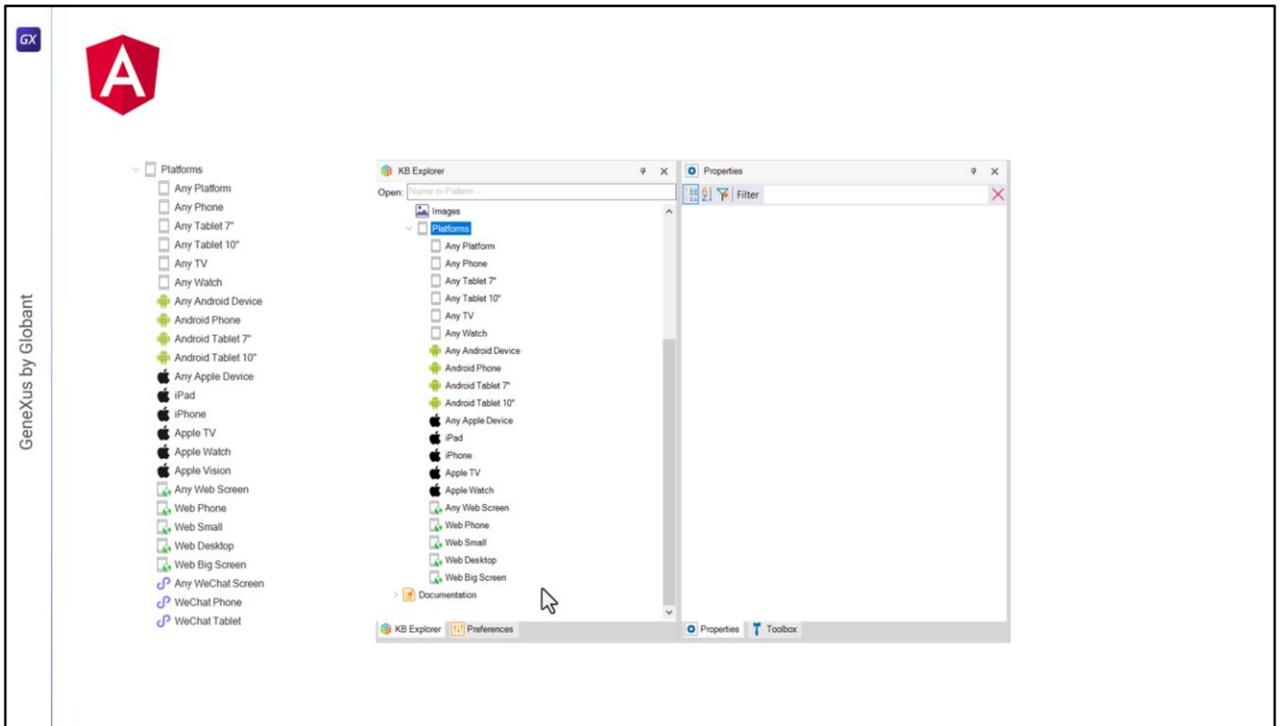
Pero esta no es la única forma de modelar los mismos 3 layouts.



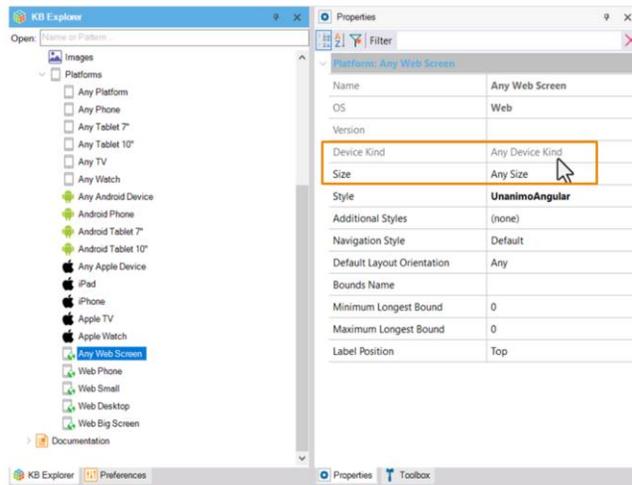
Es importante mencionar que esta definición del universo de plataformas es la default y establece las variantes que usualmente se necesitan en un desarrollo. Pero que podrán modificarse.



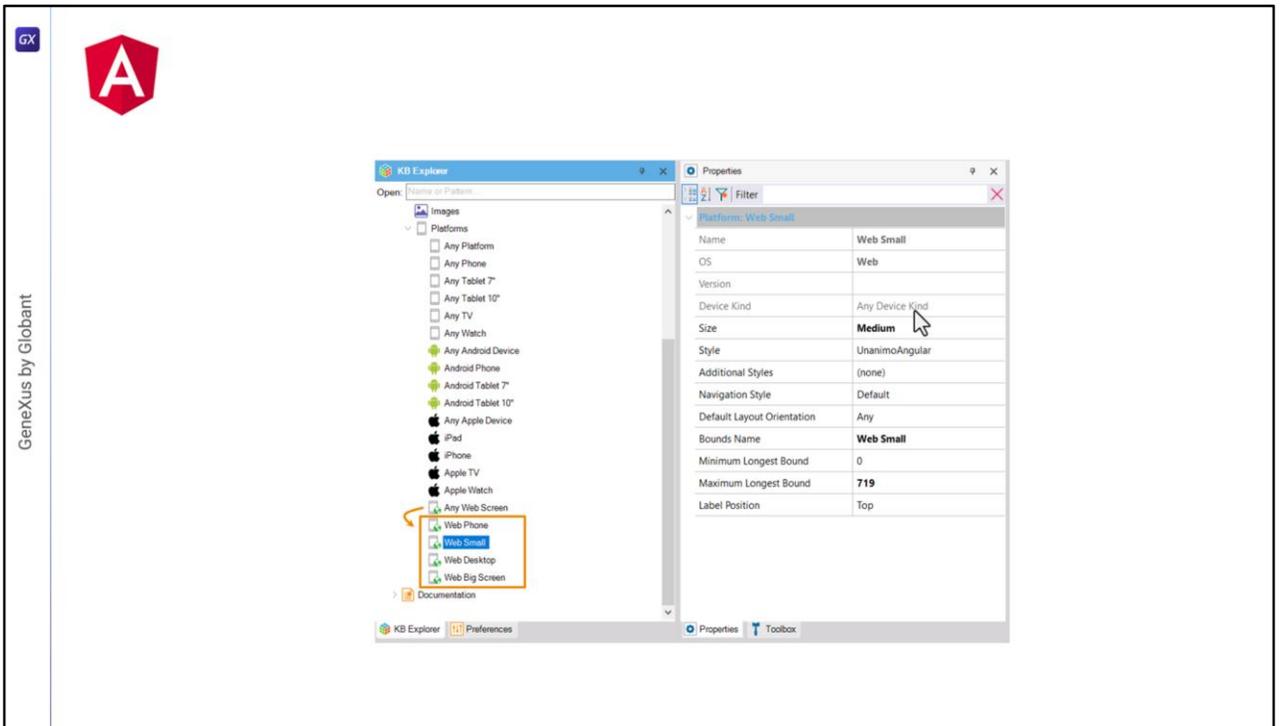
Aquí les estoy mostrando las definiciones default al crear una KB nueva (con el upgrade 8,...



... luego aparecieron más plataformas, como la WeChat, pero para lo que les quiero mostrar ahora no importan).

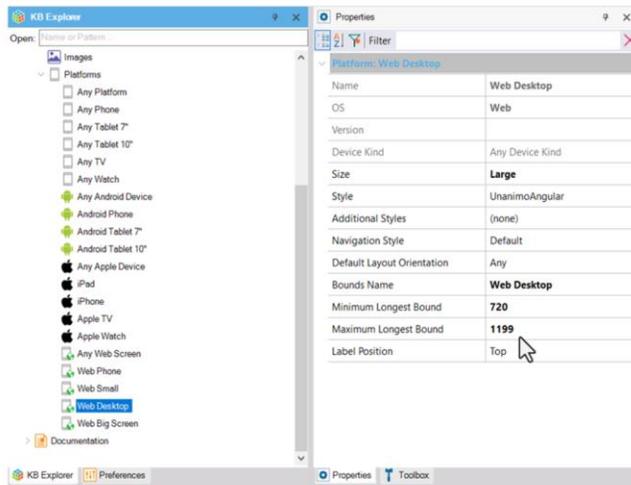


Vemos que tenemos una plataforma default que valdrá para todo caso de ejecución Web: y por eso vemos aquí “cualquier tipo de dispositivo” y aquí “cualquier tamaño”.

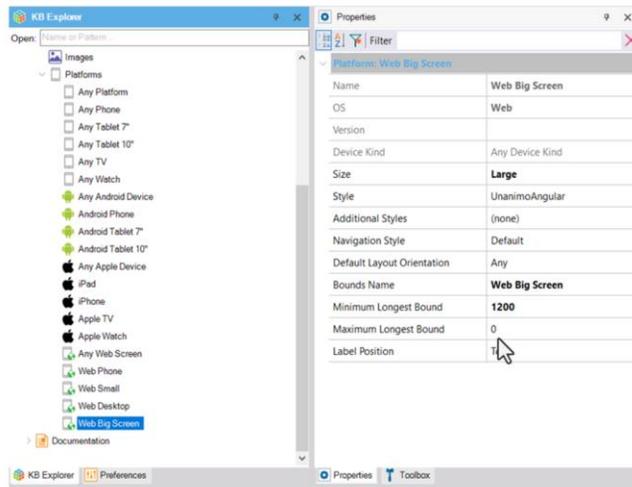


Pero además podemos especializar de acuerdo a estas 4 posibilidades, que dividen el universo de la plataforma Web de acuerdo a: tipo de dispositivo y tamaño de pantalla.

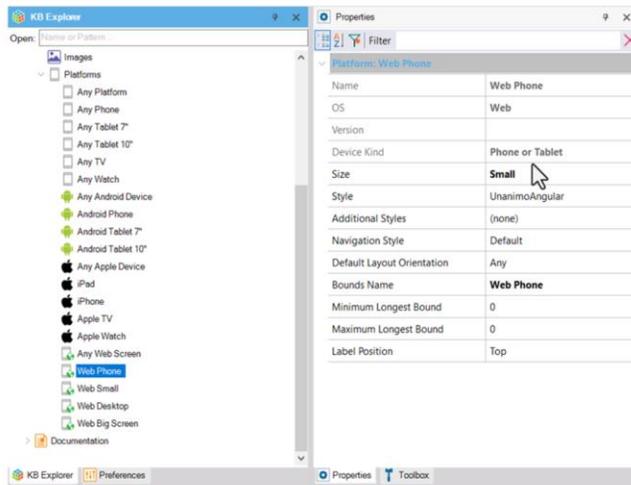
Así podemos ver que **Web Small** corresponderá a cualquier dispositivo (es decir, Phone or Tablet o PC o laptop) cuyo ancho sea **de hasta 719** dips. Dado que cuando un bound tiene 0 quiere decir que no se toma en cuenta, como si dijera Any.



Para cualquier dispositivo de tamaño Large con ancho de entre 720 y 1199 estará esta otra plataforma.

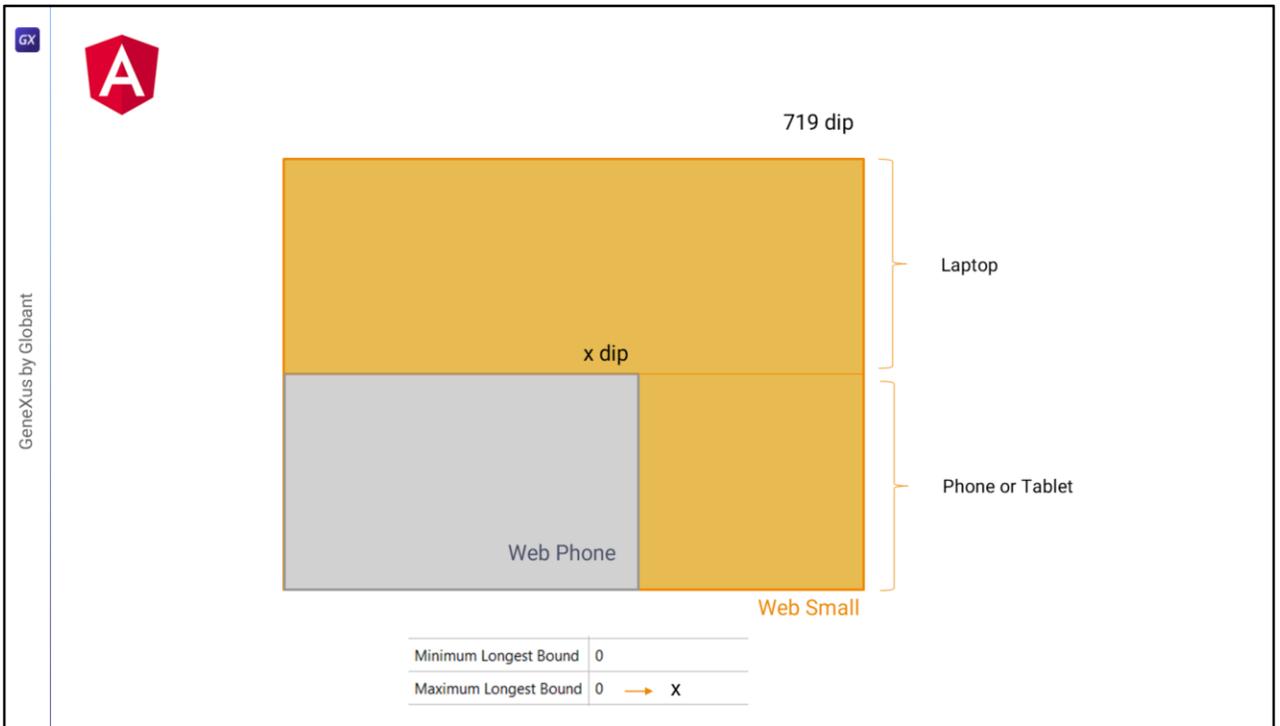


Y para los más anchos que eso, esta otra.



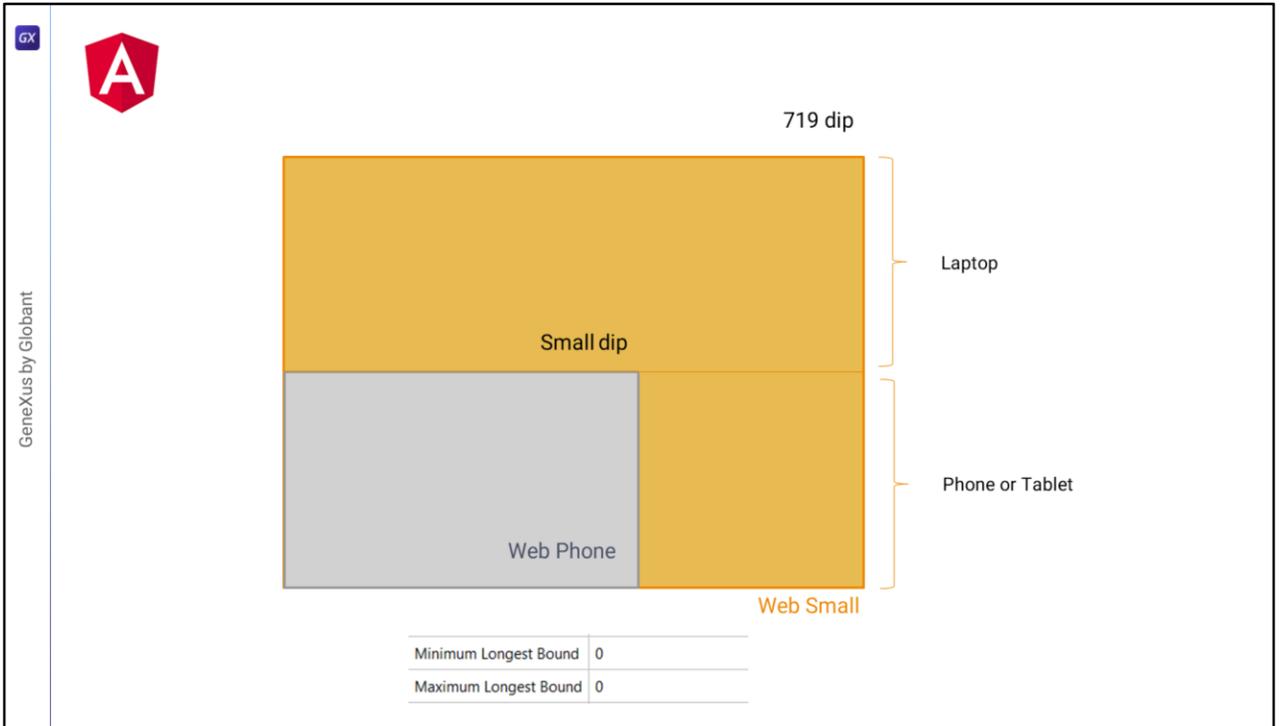
¿Y esta que nos queda? Ya es más específica, porque sólo vale para dispositivos móviles (no para laptops o PCs) de tamaño pequeño. Vemos que no tiene bounds definidos, así que es como si dijera "cualquiera", y lo que determina el tamaño es la propiedad Size, que en este caso es Small.

Pero... aquí se nos puede tornar confusa la cuestión, dado que el universo de la plataforma **Web Small** incluye a la **Web Phone**. Entonces, ¿a qué casos aplica una y la otra?

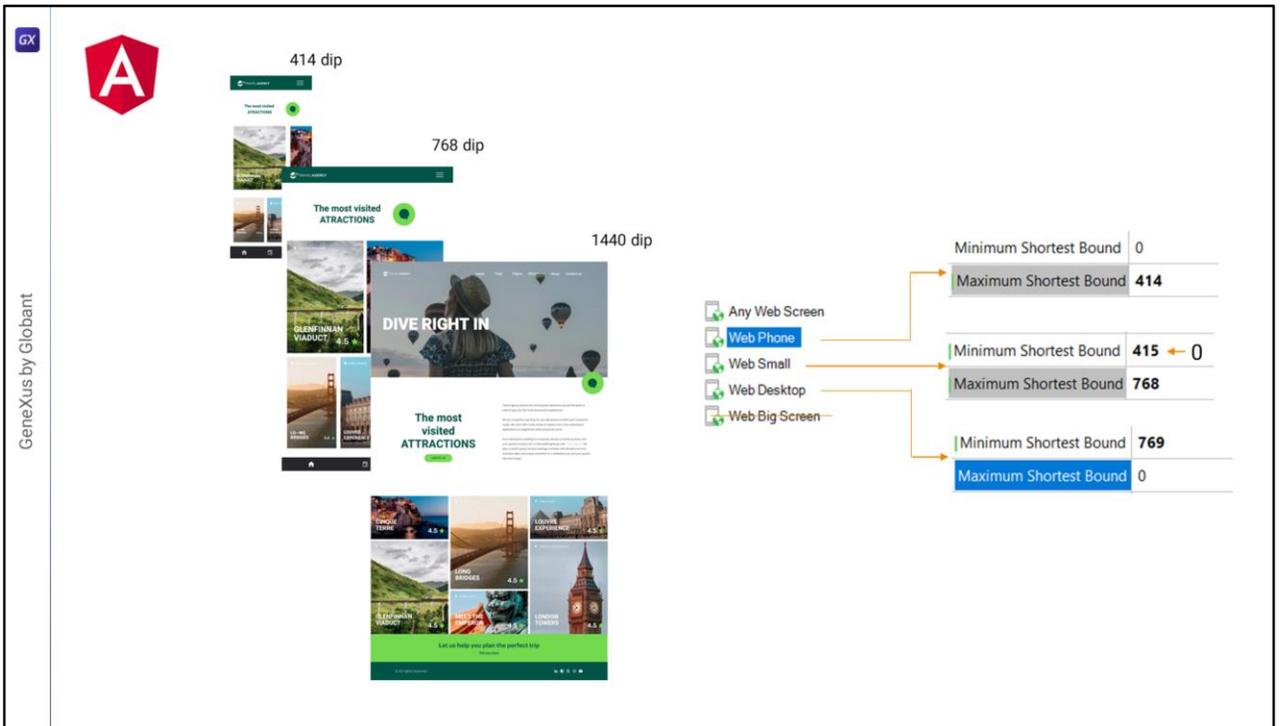


Si un caso de ejecución concreta cae dentro del universo representado por más de una plataforma, siempre matcheará con la más específica, con la que se ajuste más de cerca a sus características.

Así, si este es el universo que correspondería a la plataforma Web Small, con el ancho máximo de 719 dips, y que aplica tanto a la aplicación corriendo en una laptop de ese rango de ancho, como a un dispositivo móvil de hasta 719 dips, la Web Phone será un subconjunto, porque deja fuera todo lo que no sea dispositivo móvil, y hay que ver qué pasa con el ancho. Si le colocáramos a la propiedad del ancho máximo un valor de x dips, entonces corresponderá a la aplicación corriendo en un dispositivo móvil de hasta ese ancho. Y para cualquier otro caso (tanto de dispositivo móvil como de laptop) que llegue hasta 719 dips corresponderá a la plataforma Web Small.



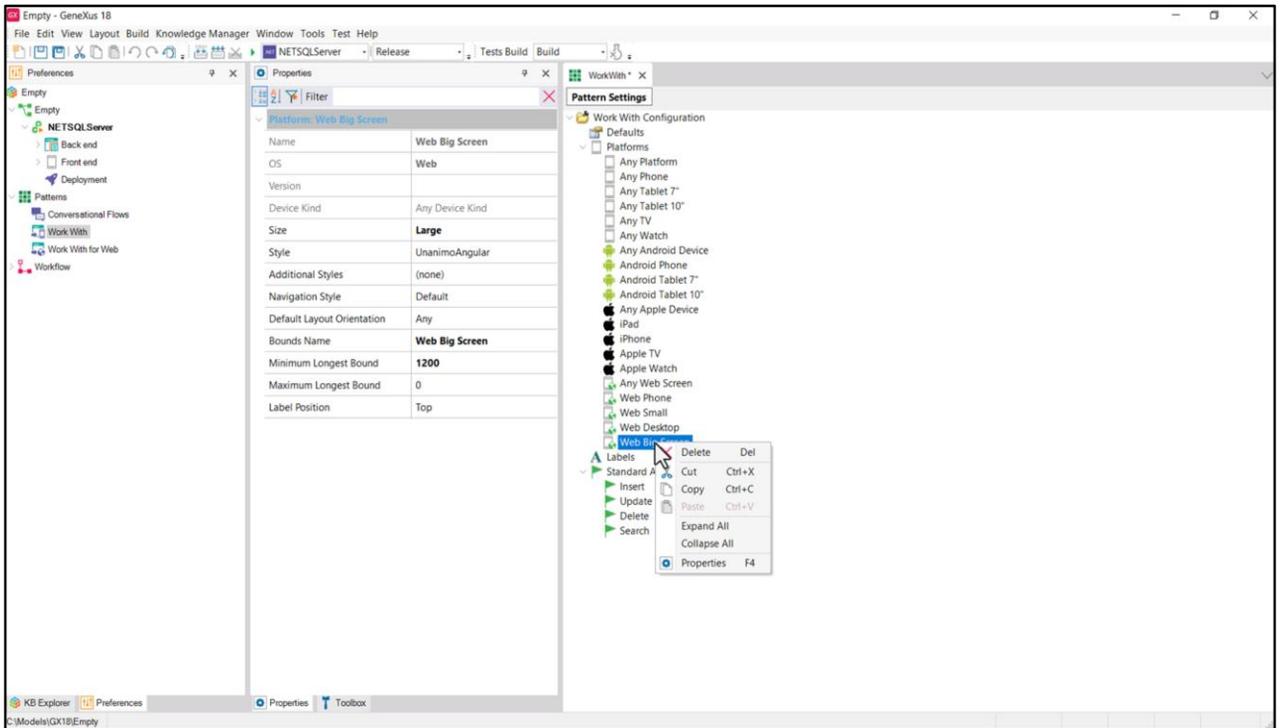
Si dejáramos el valor default de 0, entonces aplicará para lo que entiende como tamaño Small.



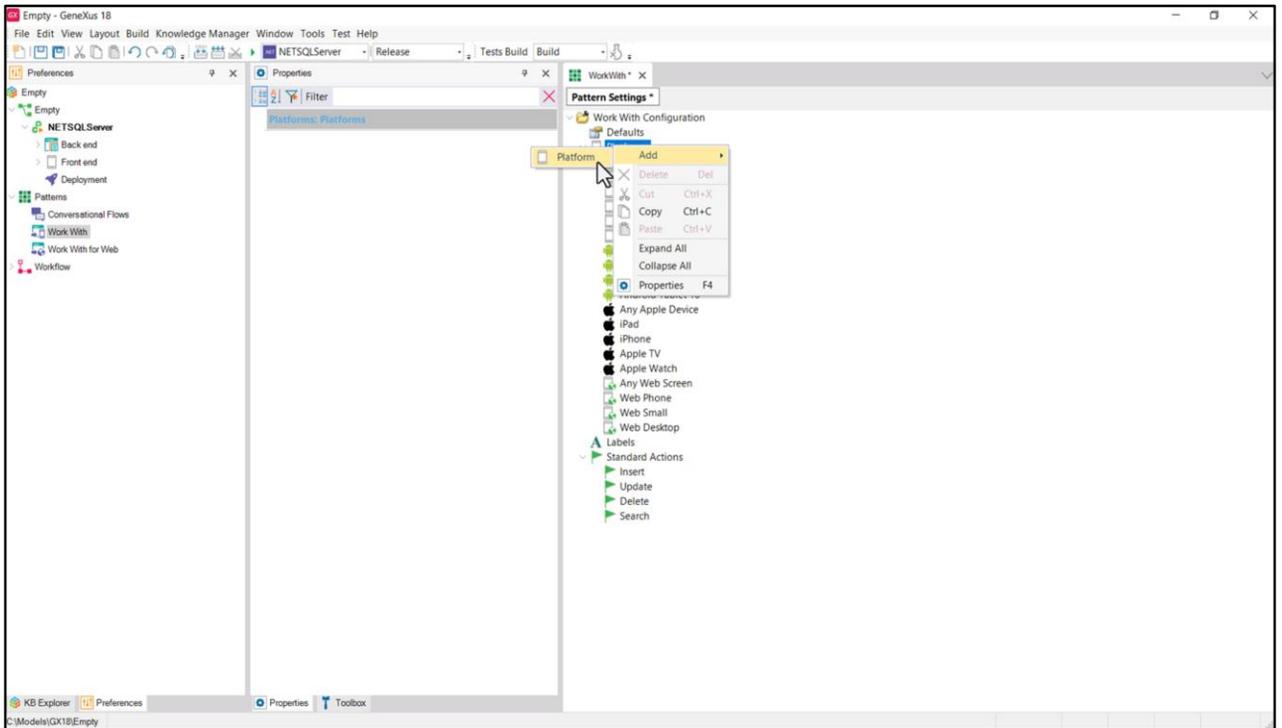
Para la realidad de nuestra aplicación, tomando de Figma los anchos de los diseños de Chechu, de las 4 necesitaríamos solamente 3 plataformas web, donde a la Web Phone le cambiaríamos el ancho máximo a 414, a la Web Small le definiríamos este rango (aunque bien podríamos dejar 0 aquí, sabiendo que la Phone sólo aplica a dispositivos móviles), y a la Web Desktop este.

No será necesario eliminar del nodo de plataformas la Web Big Screen, aunque sí sería más claro.

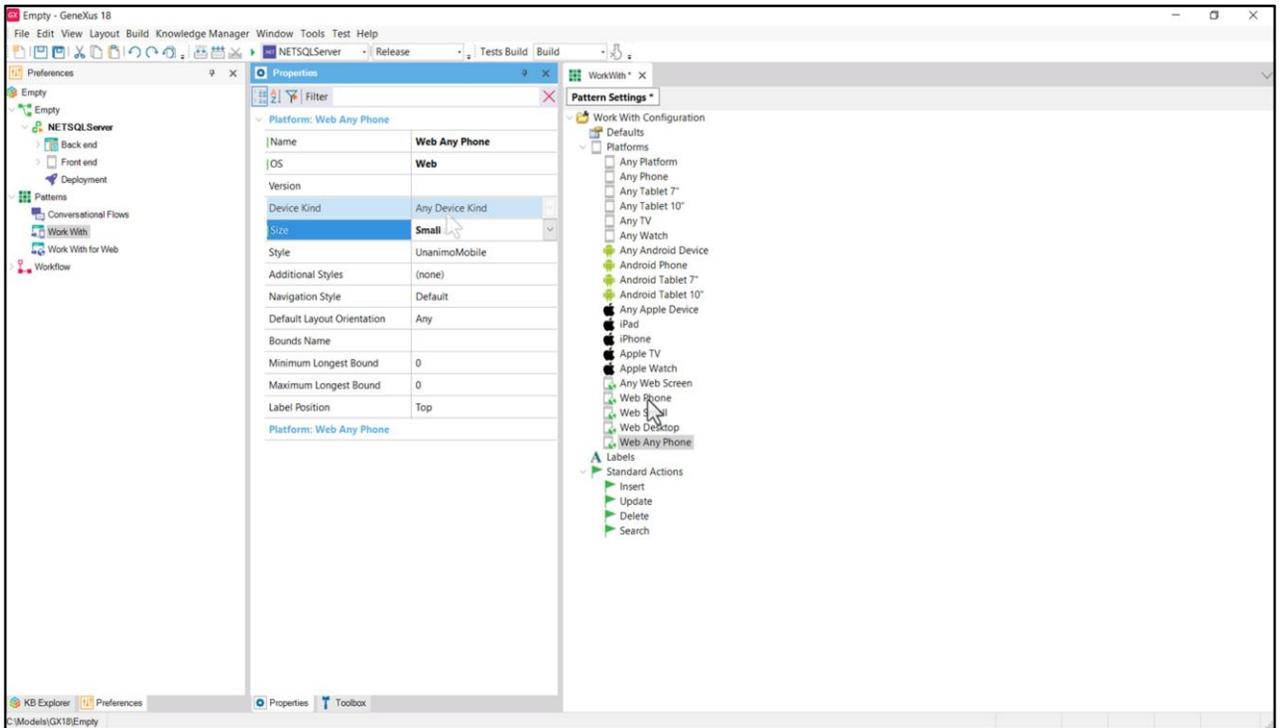
Ustedes podrán preguntarse cómo haríamos para eliminarla, o incluso para crear alguna otra plataforma si quisiéramos reorganizar el universo de posibilidades de acuerdo a otras diferenciaciones, y no a estas.



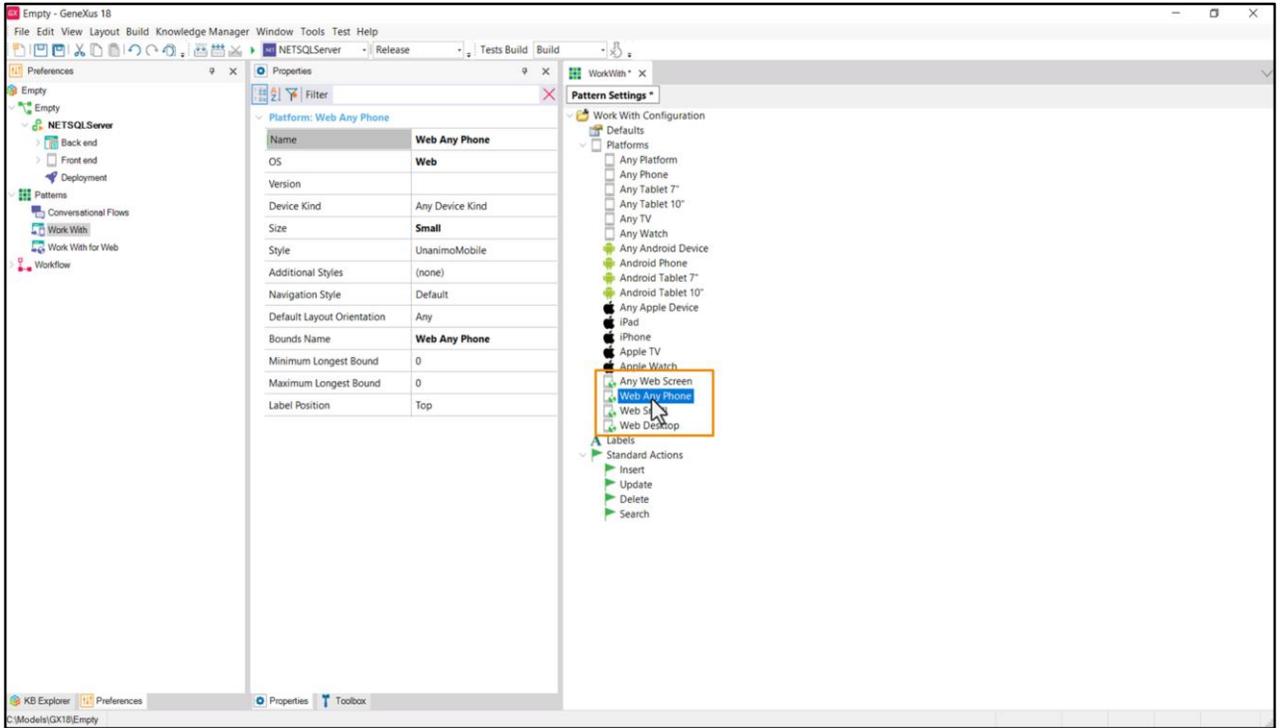
Es en la solapa de Preferences, nodo WorkWith, allí es donde podemos trabajar con las plataformas. Por ejemplo, eliminando esta.



Y si queremos agregar una que sustituya a la Web Phone, porque la queremos hacer valer para Any Device Kind y no sólo para dispositivos móviles... (aunque a priori no tenga sentido porque no hay laptops ni PCs tan pequeñas) así lo conseguimos.



Allí estamos agregando una nueva plataforma y tenemos que asignar valor a todas sus propiedades.



GeneXus by Globant

414 dip

768 dip

1440 dip

Any Web Screen

Web Phone

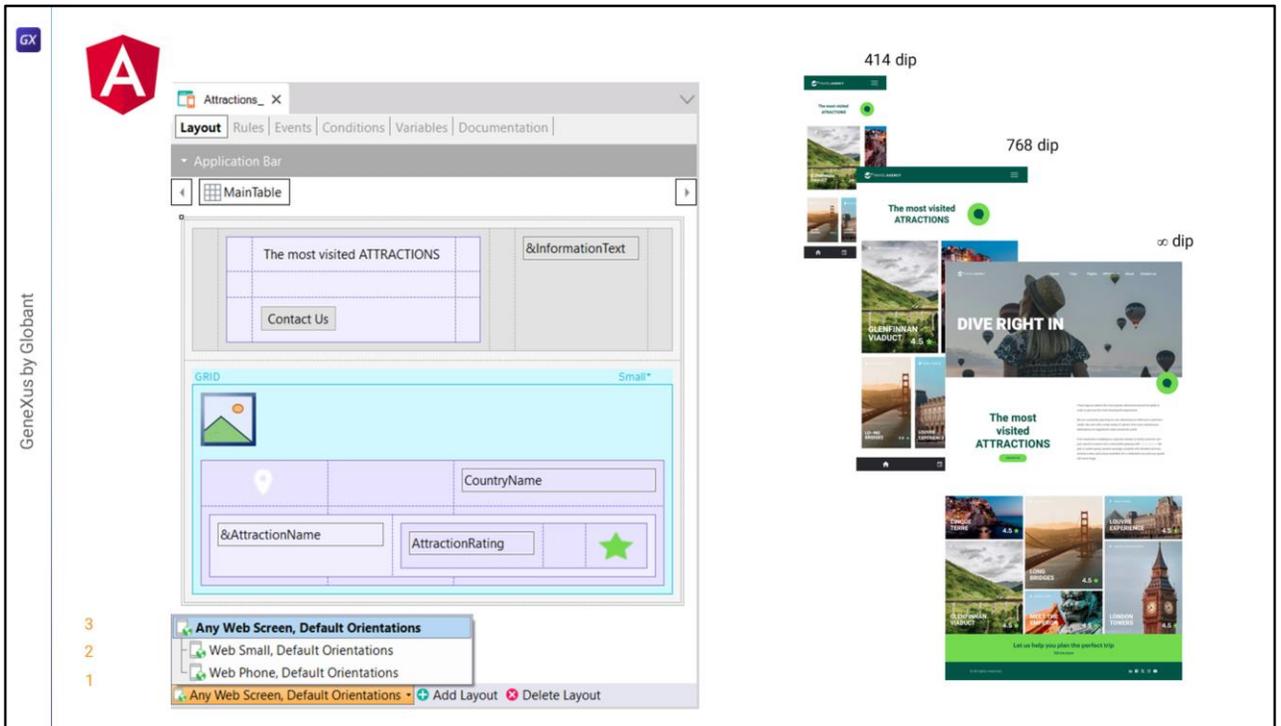
Web Small

Web Desktop

Web Big Screen

Minimum Shortest Bound	0
Maximum Shortest Bound	414
Minimum Shortest Bound	415
Maximum Shortest Bound	768
Minimum Shortest Bound	769
Maximum Shortest Bound	0

Teniendo entonces definidas las plataformas que necesitamos, supongamos que dejo estas...



Para cada objeto con UI en principio deberíamos crear tantos layouts como diseños distintos necesitemos, teniendo la precaución de definir esos layouts de manera que cada uno sea el correctamente elegido para cada plataforma para la que necesitaremos la aplicación.

Por ejemplo, si tenemos para el panel Attractions estos 3 layouts definidos, lo que se hace internamente para saber cuál elegir para cada caso es: primero extraer los parámetros de la plataforma de ejecución; segundo hacer una lista ordenada de los layouts definidos, del más específico al más general; y tres: de esa lista ordenada, el primer layout que matchee con los parámetros de ejecución será el elegido.



GeneXus by Globant

Attractions\_ X

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

The most visited ATTRACTIONS

&InformationText

Contact Us

GRID Small\*

CountryName

&AttractionName

AttractionRating

3

2

1

Any Web Screen, Default Orientations

Web Small, Default Orientations

Web Phone, Default Orientations

Any Web Screen, Default Orientations Add Layout Delete Layout

414 dip

768 dip

∞ dip

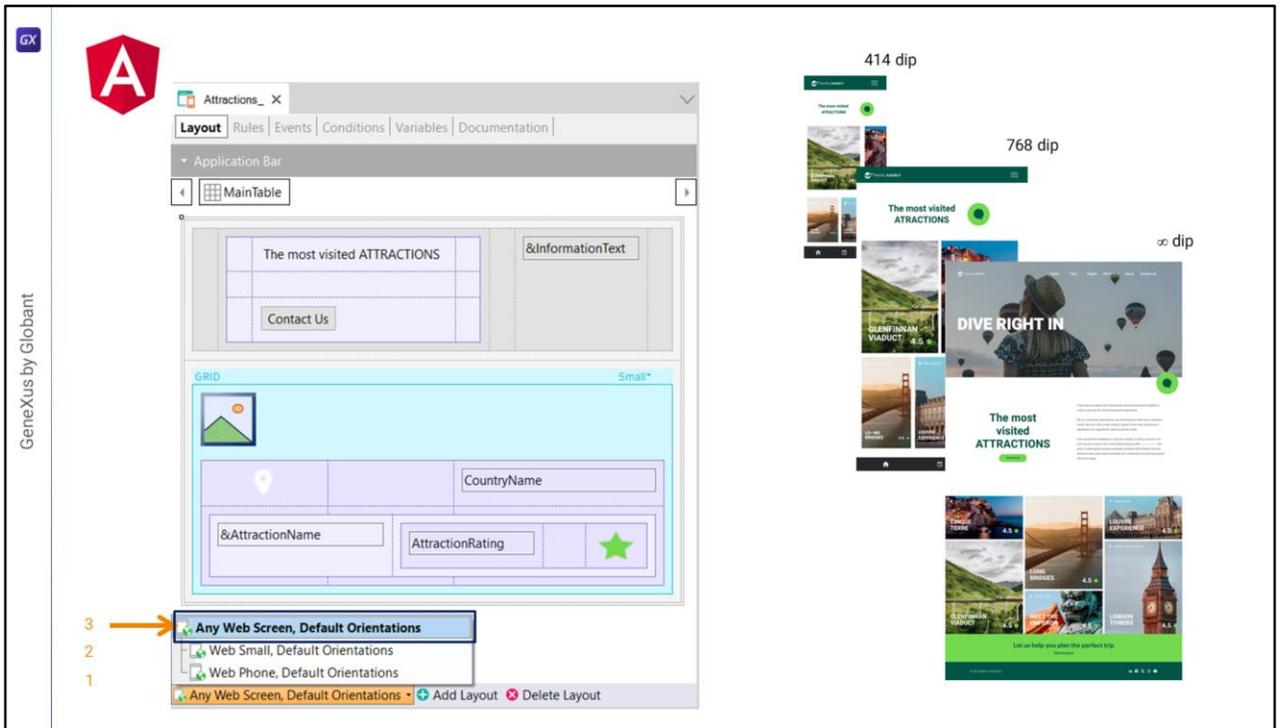
The most visited ATTRACTIONS

DIVE RIGHT IN

The most visited ATTRACTIONS

Let us help you plan the perfect trip

Si lo haremos en una Tablet de entre 415 y 768, o en una laptop de ancho de pantalla de hasta 768, elegirá este otro.



Y en cualquier otro caso web, elegirá este otro. Así que en browser de cualquier dispositivo de tamaño de pantalla mayor a 768 dips elegirá este. Vean que no indicamos exactamente un layout para Web Desktop. Podríamos haberlo hecho. Habrá que ver con cuidado qué opciones quedan fuera.

GeneXus by Globant

Minimum Shortest Bound	415
Maximum Shortest Bound	768

414 dip → Only Phone or Tablet

768 dip

∞ dip

Any Web Screen, Default Orientations

- Web Small, Default Orientations
- Web Phone, Default Orientations

Any Web Screen, Default Orientations Add Layout Delete Layout

Por ejemplo, como le colocamos este 415 aquí, si existiera una laptop de tamaño menor de 415, tendrá que elegir este layout, porque este es para Phone o Tablet nada más.

GeneXus by Globant

414 dip	Only Phone or Tablet	0
768 dip	Minimum Shortest Bound	415
	Maximum Shortest Bound	768

Para evitar ese caso es que le colocaríamos 0 a este bound y así lo que queda fuera de estas dos posibilidades es únicamente para cualquier dispositivo mayor de 768 dips.

GeneXus by Globant

Attractions\_ X

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

The most visited ATTRACTIONS

&InformationText

Contact Us

GRID Small\*

CountryName

&AttractionName

AttractionRating

Any Web Screen, Default Orientations

Web Small, Default Orientations

Web Phone, Default Orientations

Any Web Screen, Default Orientations Add Layout Delete Layout

414 dip

Only Phone or Tablet

768 dip

0

Minimum Shortest Bound 415

Maximum Shortest Bound 768

∞ dip

Claro, con esta definición de plataformas, si existiera una laptop de tamaño menor a 415 dips, para ese caso elegirá este layout. Allí necesitaríamos hacer lo que les mostré en el trabajar con Plataformas, para que elija este otro.

GeneXus by Globant

The image shows the GeneXus IDE interface for a project named 'Attractions'. The main window displays a layout editor with a 'MainTable' containing sections like 'The most visited ATTRACTIONS', '&InformationText', and 'Contact Us'. Below this is a 'GRID' section with a 'Small\*' orientation, containing a map icon, a location pin, and text fields for '&AttractionName', 'AttractionRating', and 'CountryName'. A star icon is also present. At the bottom, there is a list of styles: 'Any Web Screen, Default Orientations', 'Web Small, Default Orientations', 'Web Phone, Default Orientations', and 'Any Web Screen, Default Orientations'. To the right, a 'Style:' section lists 'Any Web Screen', 'Web Phone', 'Web Small', 'Web Desktop', and 'Web Big Screen'. Arrows point from 'Web Phone' to 'TravelAgencyPhone' and from 'Web Small' to 'TravelAgencyTablet'. Below this, two mobile device screens are shown: a smartphone displaying 'DIVE RIGHT IN' and a tablet displaying 'The most visited ATTRACTIONS' with a grid of attraction cards.

Por otro lado, no tenemos porqué definir los layouts de la misma manera para todos los paneles. Esto es panel a panel.

Por ejemplo, observen que entre estos dos diseños, en verdad la única diferencia parece ser en los tamaños de las fuentes y los altos y anchos, y en nada más. Si esos tamaños estuvieran definidos a nivel del DSO y no a nivel de los controles en el layout, entonces ambos layouts podrían ser el mismo, para lo que podríamos evitarnos definirlos dos veces. Podríamos agrupar estos dos en uno solo, y que la diferencia venga sólo dada por el DSO asociado a cada plataforma.

GeneXus by Globant

Attractions\_ X

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

The most visited ATTRACTIONS

&InformationText

Contact Us

GRID Small\*

CountryName

&AttractionName

AttractionRating

Any Web Screen, Default Orientations

Web Small, Default Orientations

Web Phone, Default Orientations

Any Web Screen, Default Orientations

Web Desktop, Default Orientations

Any Web Screen, Default Orientations

Add Layout Add Layout Delete Layout

Any Web Screen

Web Phone → TravelAgencyPhone

Web Small → TravelAgencyTablet

Web Desktop

Web Big Screen

DIVE RIGHT IN

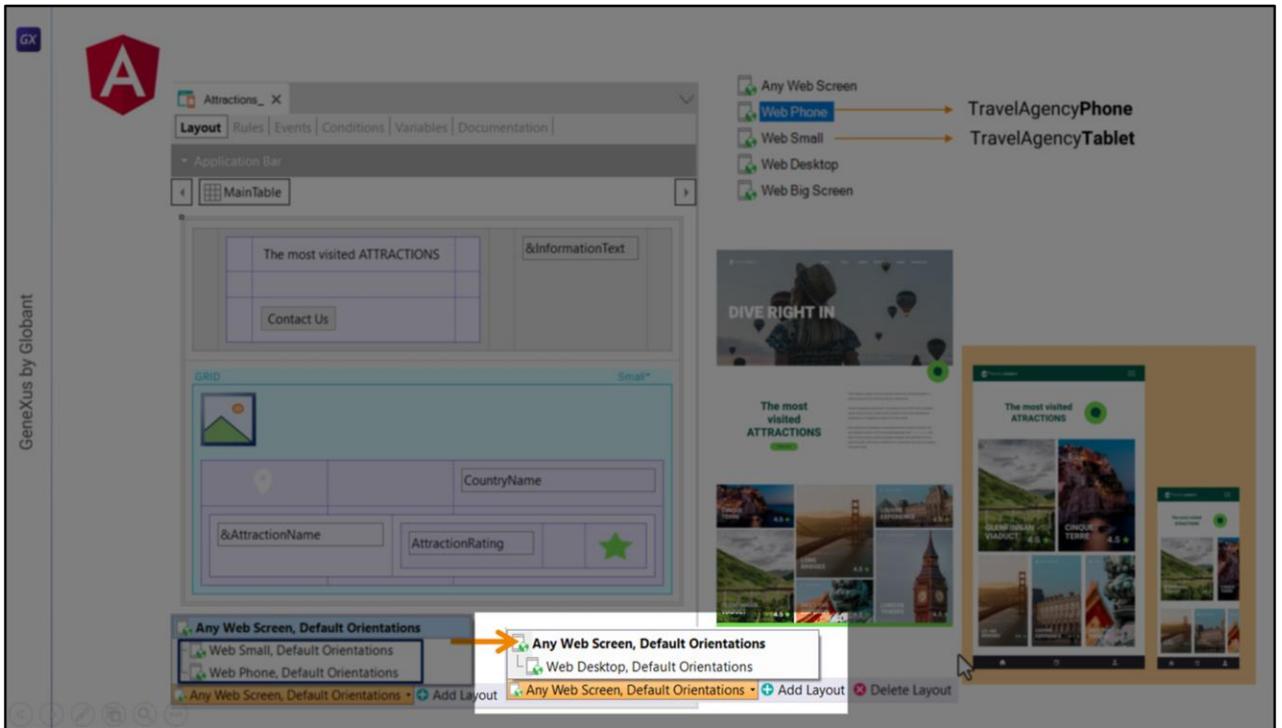
The most visited ATTRACTIONS

The most visited ATTRACTIONS

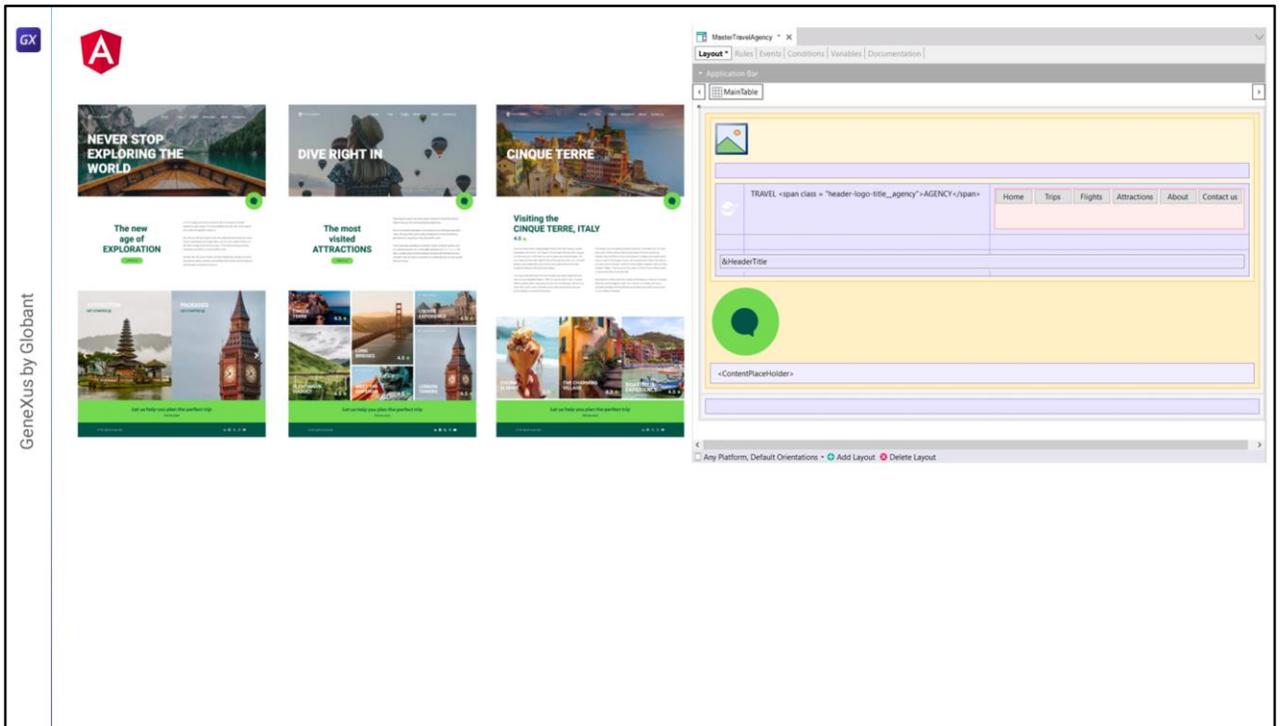
GIENNFINIAN VIADUCT 4.5

CINQUE TERRE 4.5

Pero para ello, tendríamos que indicar de otra forma estos layouts de aquí. Debemos a este decirle que es para Web Desktop y no para Any Web Screen; y dejar el Any Web Screen para aunar estos dos. ¿Se entiende?

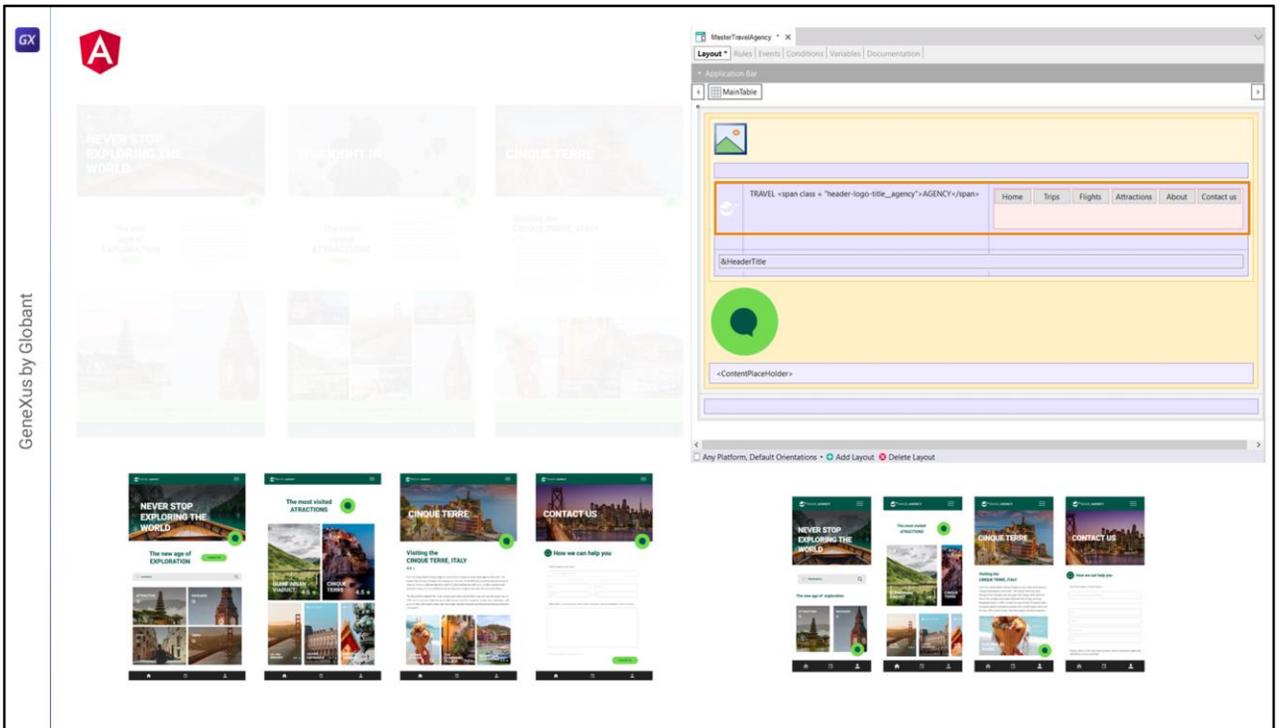


Porque en ese caso, si estamos ejecutando en el browser de un teléfono no matcheará con el Web Desktop, que es el más específico de los definidos, y entonces tendrá que quedarse con este otro. Y lo mismo sucederá para Laptop o Tablet de tamaño hasta 768 dips.

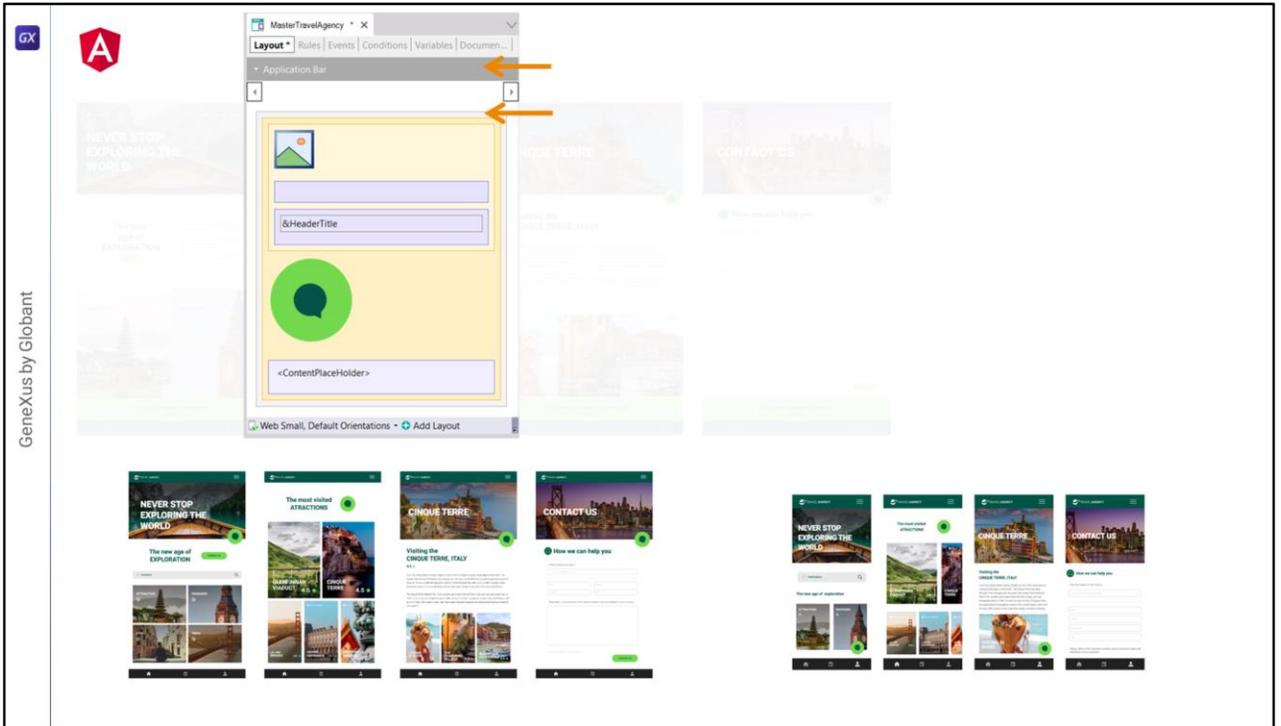


Bien, sabido todo esto...

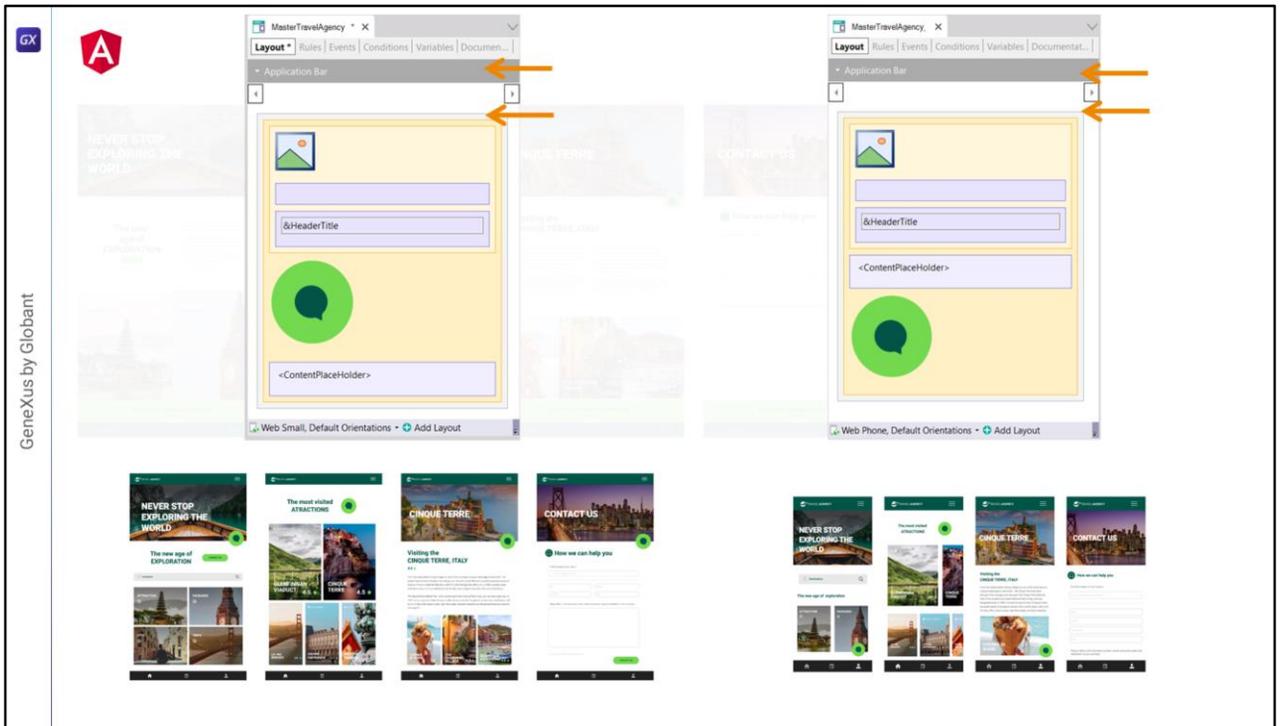
En lo que hace al Master Panel si aquí teníamos la implementación para Desktop (está faltando únicamente el footer)...



Los breakpoints para tamaño Tablet y para Phone, deberán quitar de aquí todo esto e implementar la funcionalidad del menú hamburguesa.



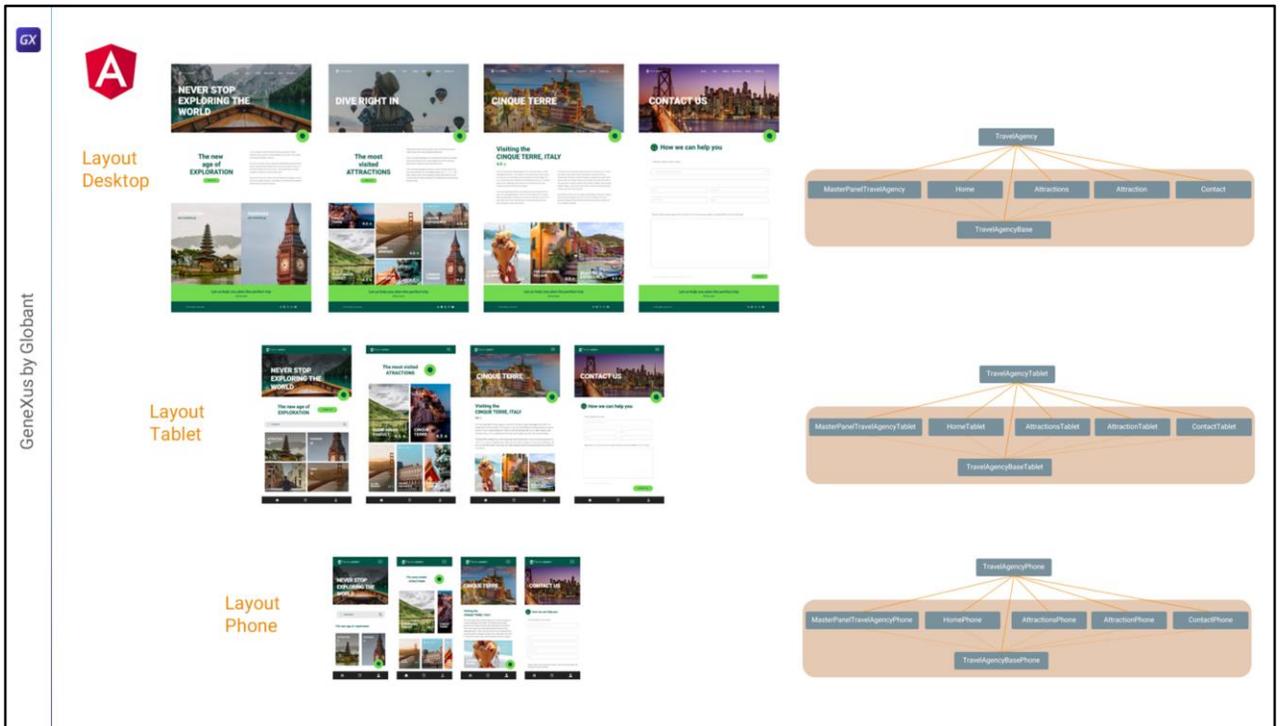
Ya sea aquí o aquí, pero además...



Y también necesitaremos modificar el Master Panel para Phone, dado que allí la imagen de Chatbot va abajo, aunque sólo en dos de las pantallas únicamente.

Entonces una opción sería a través de un layout particular, diferente.

Les dejo como tarea para ustedes pensar cómo harían para que para tamaño Tablet en el panel Attractions no se muestre el Header; y para el tamaño Phone no se muestre el chatbot abajo para el panel Attractions ni para el Contact.



Bueno, resumamos lo que hay que hacer para resolver la aplicación Angular adaptativa: crear layouts para los tamaños Tablet y Phone para los casos en los que varíen, y lo otro que hay que hacer es **especializar** el árbol de DSOs que habíamos desarrollado para Desktop, para modificar únicamente lo que cambia, en ese nuevo tamaño de pantalla, como los tamaños de fuente, espaciados y algunos detalles más. Esto, de hecho, lo habíamos iniciado tempranamente cuando empezamos a trabajar con las clases tipográficas para Desktop. Recuerden que allí, por ejemplo, habíamos visto que mientras que para Desktop las clases para los textos de las cards eran idénticas entre cards grandes y pequeñas, y entre las de Attractions y de Attraction, para Tablet y Phone, en cambio, variaban, y por eso habíamos especificado, ya en el módulo de preparación del proyecto, variaciones para esas clases.

Es posible, como en este caso, que además debamos modificar la implementación de alguna otra parte, como es el caso del menú hamburguesa.

Haciendo un balance, el diseño adaptativo de la solución Angular no resultará costoso. Habrá que adaptar a los breakpoints lo que ya se desarrolló para las pantallas del tamaño por el que se comenzó y este trabajo es bastante menos engorroso que diseñar cada breakpoint de cero.

GeneXus by Globant

Layout Desktop

Layout Tablet

Layout Phone

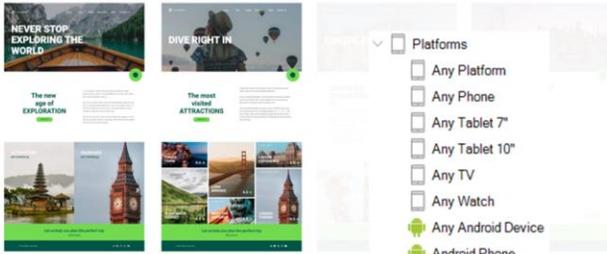
Platforms

- Any Platform
- Any Phone
- Any Tablet 7"
- Any Tablet 10"
- Any TV
- Any Watch
- Any Android Device
- Android Phone
- Android Tablet 7"
- Android Tablet 10"
- Any Apple Device
- iPad
- iPhone
- Apple TV
- Apple Watch
- Any Web Screen
- Angular Phone
- Web Small
- Web Desktop
- Web Big Screen

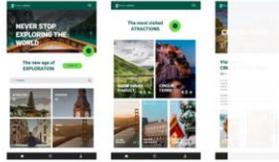
Por supuesto, debemos indicar a nivel de las Platforms la raíz del árbol de DSOs que corresponderá a cada breakpoint.



Layout Desktop



Layout Tablet

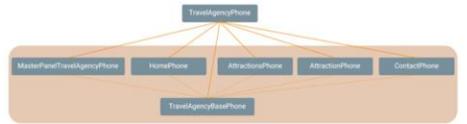
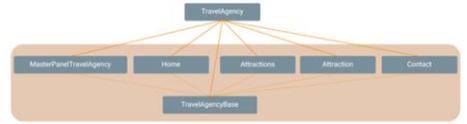


Layout Phone



Platforms

- Any Platform
- Any Phone
- Any Tablet 7"
- Any Tablet 10"
- Any TV
- Any Watch
- Any Android Device
- Android Phone
- Android Tablet 7"
- Android Tablet 10"
- Any Apple Device
- iPad
- iPhone
- Apple TV
- Apple Watch
- Any Web Screen
- Angular Phone
- Web Small
- Web Desktop
- Web Big Screen





Layout Desktop



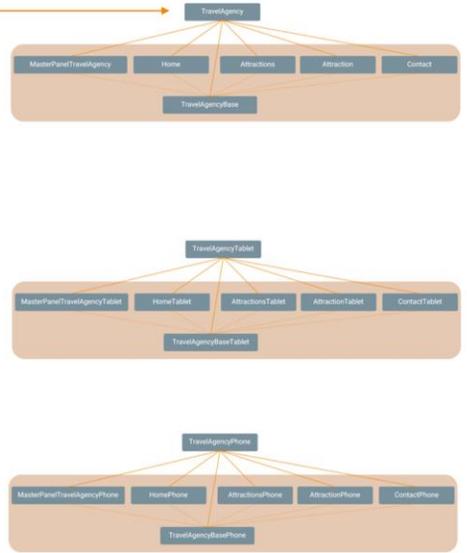
Layout Tablet



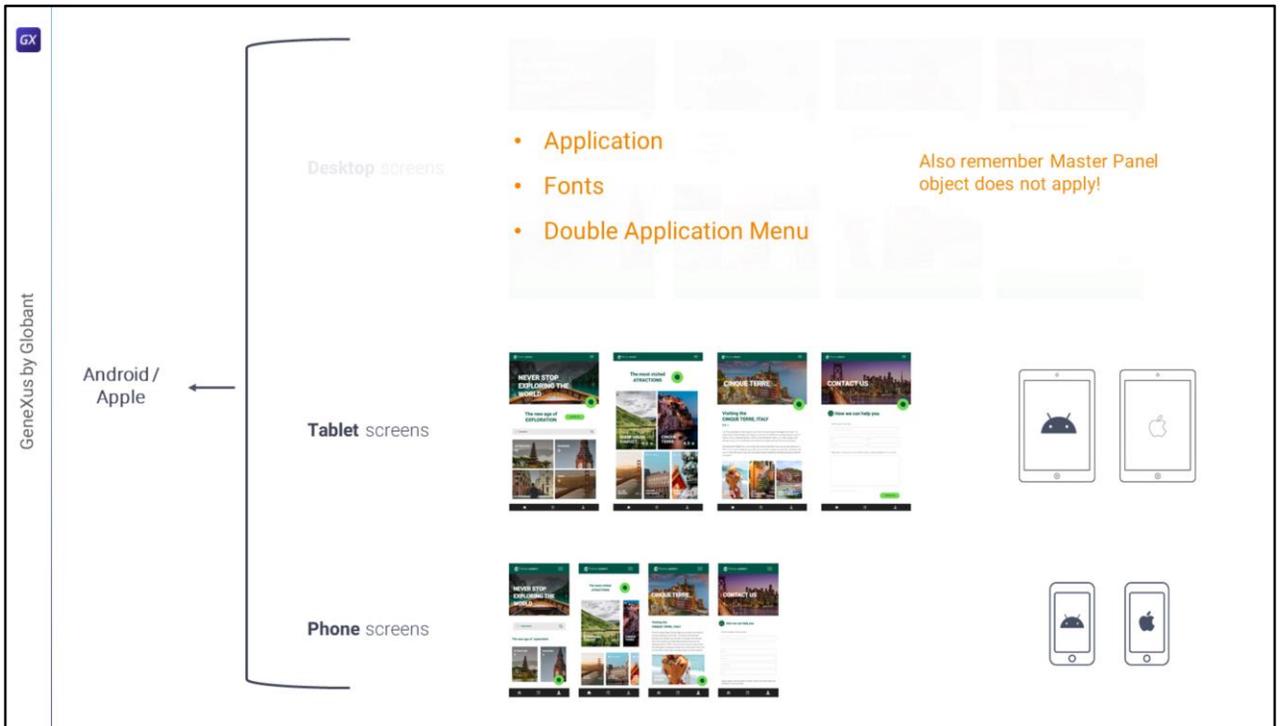
Layout Phone



- Platforms
- Any Platform
  - Any Phone
  - Any Tablet 7"
  - Any Tablet 10"
  - Any TV
  - Any Watch
  - Any Android Device
  - Android Phone
  - Android Tablet 7"
  - Android Tablet 10"
  - Any Apple Device
  - iPad
  - iPhone
  - Apple TV
  - Apple Watch
  - Any Web Screen
  - Angular Phone
  - Web Small
  - Web Desktop
  - Web Big Screen



# Native Mobile



¿Y si solamente quisiéramos la aplicación nativa, y no nos interesara desarrollarla para Angular?

No fue exactamente lo que hicimos en los videos anteriores, donde me concentré más bien en una especie de comparación entre Angular y lo nativo, aunque también les conté cuestiones que son particulares, especiales de lo nativo, como por ejemplo cuando les hablé de los colores que se definen a nivel de la clase Application, o el uso de las fuentes, o la imposibilidad de tener para la aplicación, al menos por el momento, un doble menú.

## For Native Mobile: a field with Format HTML can be styled through "style" html attribute

Desktop screens

Properties

General

Filter

TextBlock1

Control Name

TextBlock1

Caption

TRAVEL <span class = "header-logo-title\_agency">AGENCY</span

Appearance

Auto Grow

False

Class

header-logo-title\_travel

Visible

True

Invisible Mode

Keep Space

Enabled

True

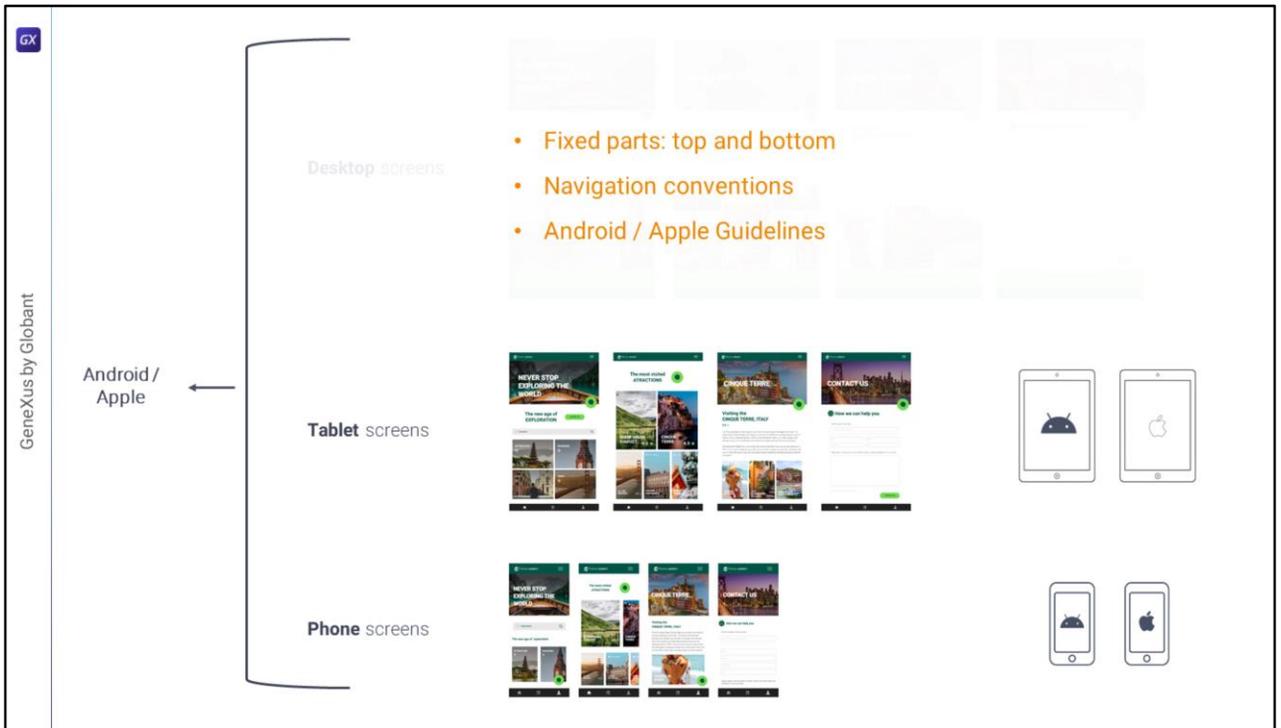
Format

HTML

Cell information

style="font-family:Heebo-700;"

Una diferencia, en la comparación que hicimos, que allí no mencioné pero que la vimos cuando implementamos el cambiarle el peso de la fuente a la palabra Agency para la aplicación Angular Desktop fue que podemos utilizar el **style** inline para hacer esto mismo en Android o Apple, si colocamos formato html para el textblock, ¿recuerdan? Aunque no parezca tener sentido hablar de html para lo nativo, si bien no será tomado en cuenta el atributo class, sí lo será el style, que nos permite hacer cosas como esta.



La otra característica que les mencionaba de las pantallas nativas es que en general se dejan fijos elementos en los bordes de arriba y de abajo de la pantalla, y la zona scrolleable suele ser la del medio. Pero además hay convenciones respecto a las navegaciones, y las maneras de volver a las pantallas anteriores, que son propias de los sistemas operativos, y que toda aplicación debe respetar. Seguir las guías de diseño de Android y de Apple es muy importante.

GeneXus for Mobile V18 Course

training.genexus.com/en/learning/courses/genexus-for-mobile/v18/course-genexus-for-mobile-genexus-18

GeneXus DL Portal Issues

GeneXus training Learning Certifications Universities Academic Partners Help Login

## GeneXus for Mobile course

Version: GeneXus 18

### Features of mobile applications

We will see the different types of applications for mobile devices: according to their purpose (oriented to the line of business, oriented to the consumer, FrontEnd applications, Backend applications), and according to their connectivity (connected, partially connected, disconnected)

Total length of videos: 5h

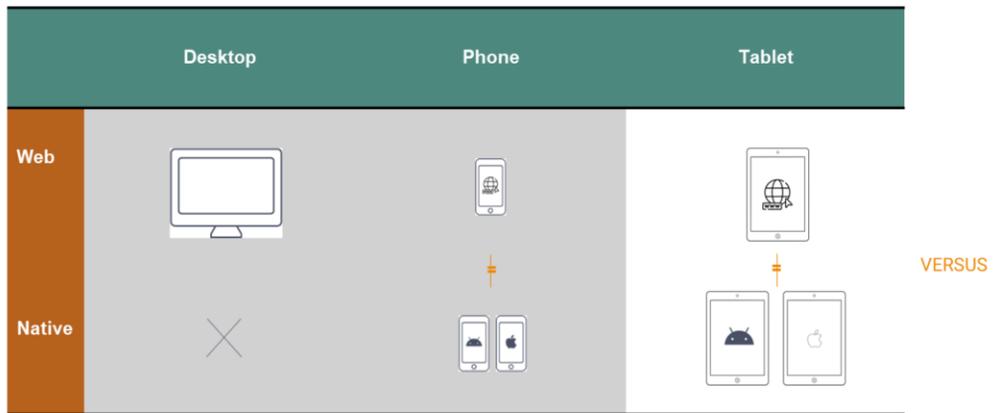
The video player displays a diagram titled "Features of mobile applications" with two main categories: "Web" and "Mobile".

- Web:** Includes logos for Java and .NET.
- Mobile:** Includes logos for iOS, Swift, Android, Java, Kotlin, and FrontEnd. A red "NEW" badge is next to Kotlin, with "Also:" written below it. An arrow points from the Java logo to the Kotlin logo.

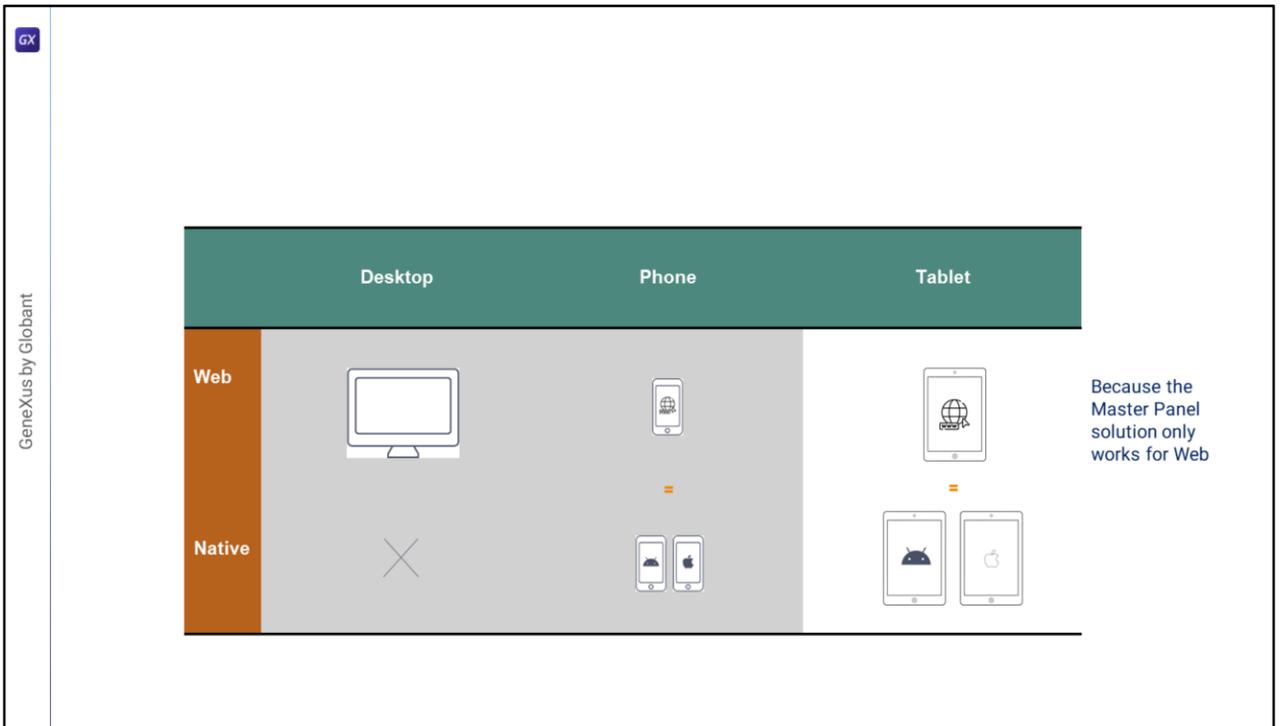
Below the diagram is a table of contents:

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

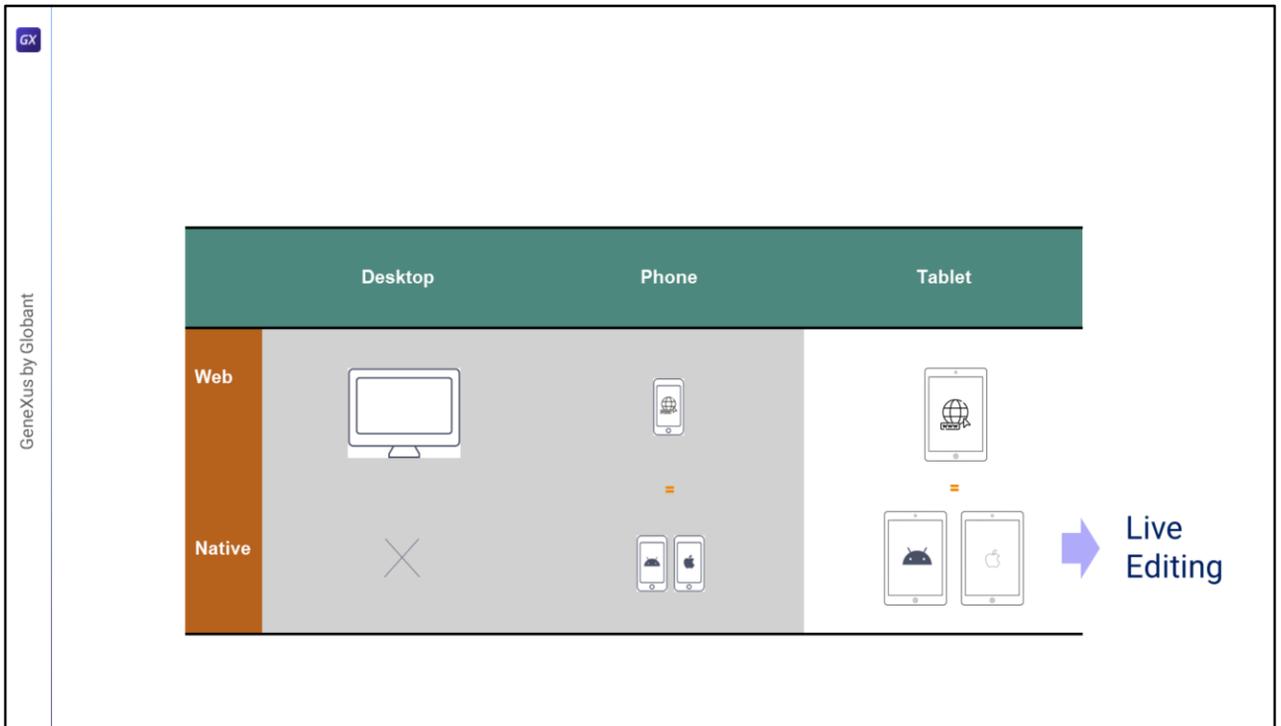
Todo esto ameritaría un curso en sí mismo, parte del cual ya existe, y es el de Mobile que ya les referí, junto con material adicional que podrán encontrar allí.



En los videos anteriores de este módulo que ahora estamos terminando me concentré, entonces, les decía, en mostrar las diferencias, pero me quedo con la sensación de que no destaque suficientemente todo lo que es igual, que es mucho. Por ejemplo los controles, básicamente son los mismos, con sus propiedades y sus valores, y también muchas de las propiedades que asociamos a las clases van a ser las mismas (las propiedades y sus valores).



Podríamos haber empezado el curso al revés. Podríamos haber empezado por diseñar la aplicación nativa para teléfono, y haber estudiado entonces todo lo que estudiamos en los módulos 1 al 4, pero para esta aplicación. Para teléfono y luego para Tablet. Y recién después pasar a lo Web. Habríamos pasado más o menos por las mismas estaciones, salvo en lo que hace a los eventos globales y al Master Panel.



Habríamos también visto allí la herramienta de Live Editing, que en los videos anteriores no me molesté en mostrarles, porque no estuvimos prototipando para lo nativo. Pero bueno, la herramienta Live Editing es muy importante para poder hacer cambios y verlos instantáneamente sin tener que compilar y ejecutar; y ni siquiera grabar, les digo más.

GeneXus for Mobile V18 Course

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

GeneXus DL Portal Issues

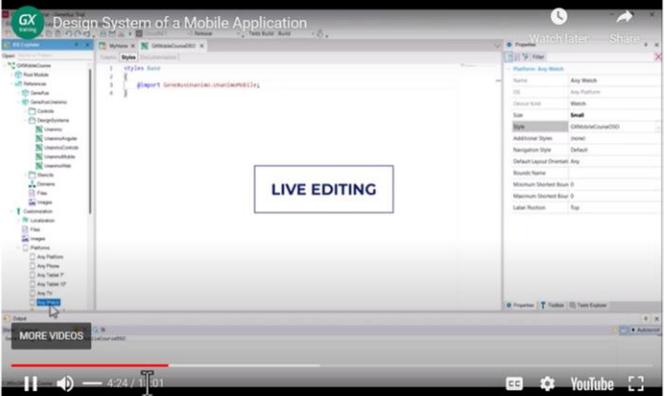
## GeneXus for Mobile course

Version: GeneXus 18

### Design System of a Mobile Application

In this video, we will study how to work with the Design System object to add a design to our application. We will see, among other things, how to use tokens, styles, and properties, as well as how to incorporate fonts and work with light and dark modes.

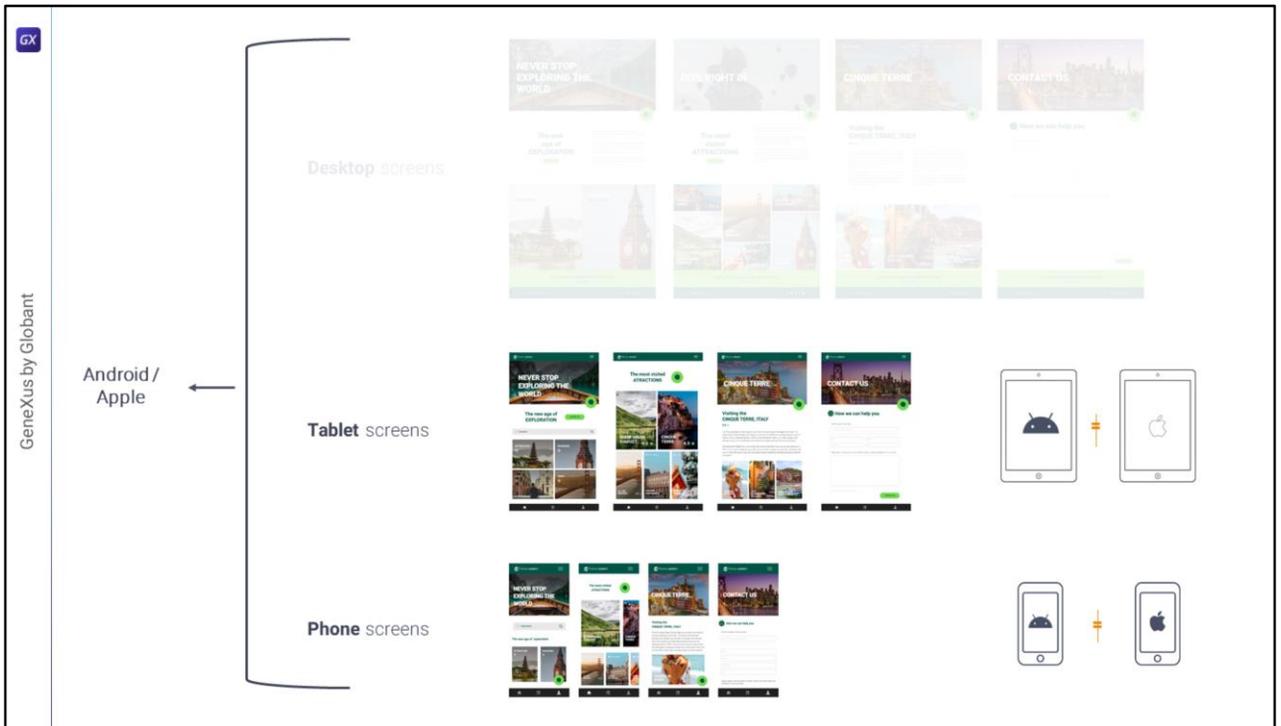
Total length of videos: 5h



The screenshot shows a video player displaying a GeneXus Design System interface. The interface includes a left-hand navigation pane with a tree view of design system components, a central workspace with a 'LIVE EDITING' overlay, and a right-hand properties panel. The video player controls at the bottom show a progress bar at 4:24 / 11:01.

- 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
  - Stencil object
- User Interface Components**

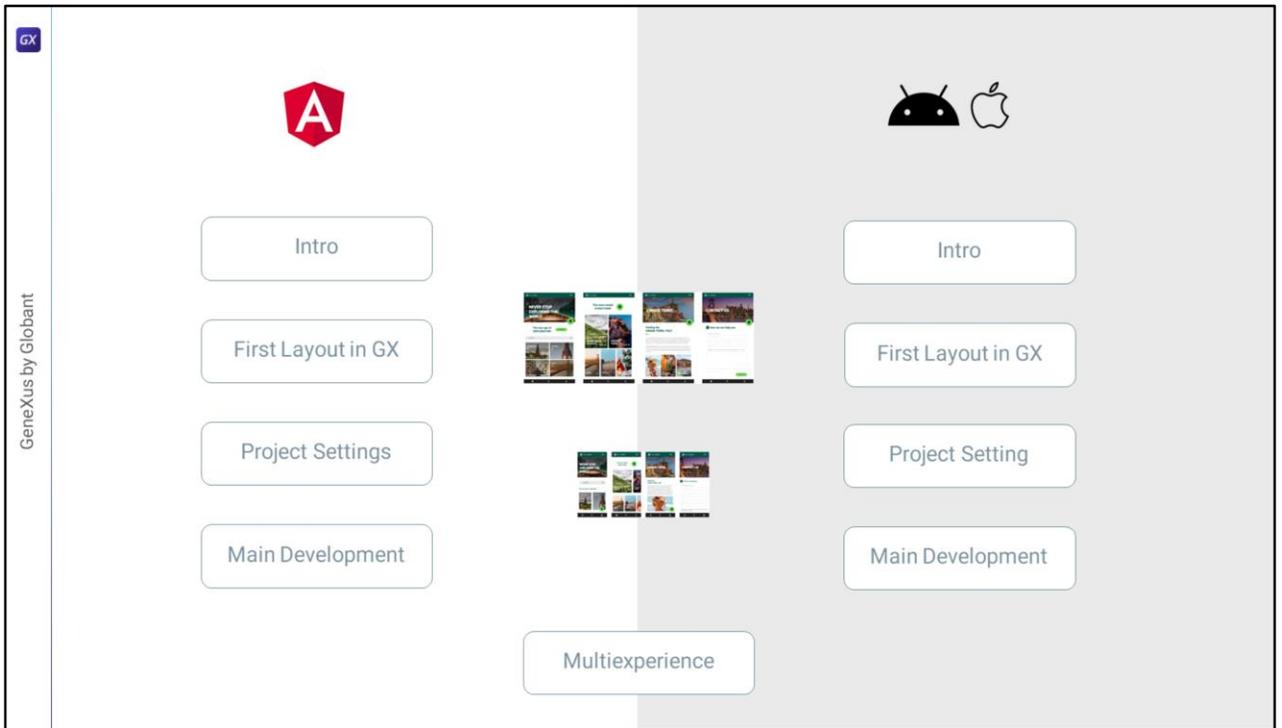
Si hacen el curso Mobile para GeneXus lo van a ver claramente.



Por otro lado en el primer video de este módulo fui poco precisa, digámoslo así, al igualar la aplicación para Android y la aplicación para Apple. ¿Por qué lo hice? Para simplificar, en verdad, porque es mucho más lo que tienen en común de lo que las diferencia. Pero claramente puede suceder que haya especificidades que debamos contemplar, que hacen a cada una de las plataformas, o incluso de las guías de diseño, que puede ser que ameriten diferenciar, entonces, los layouts por plataforma.

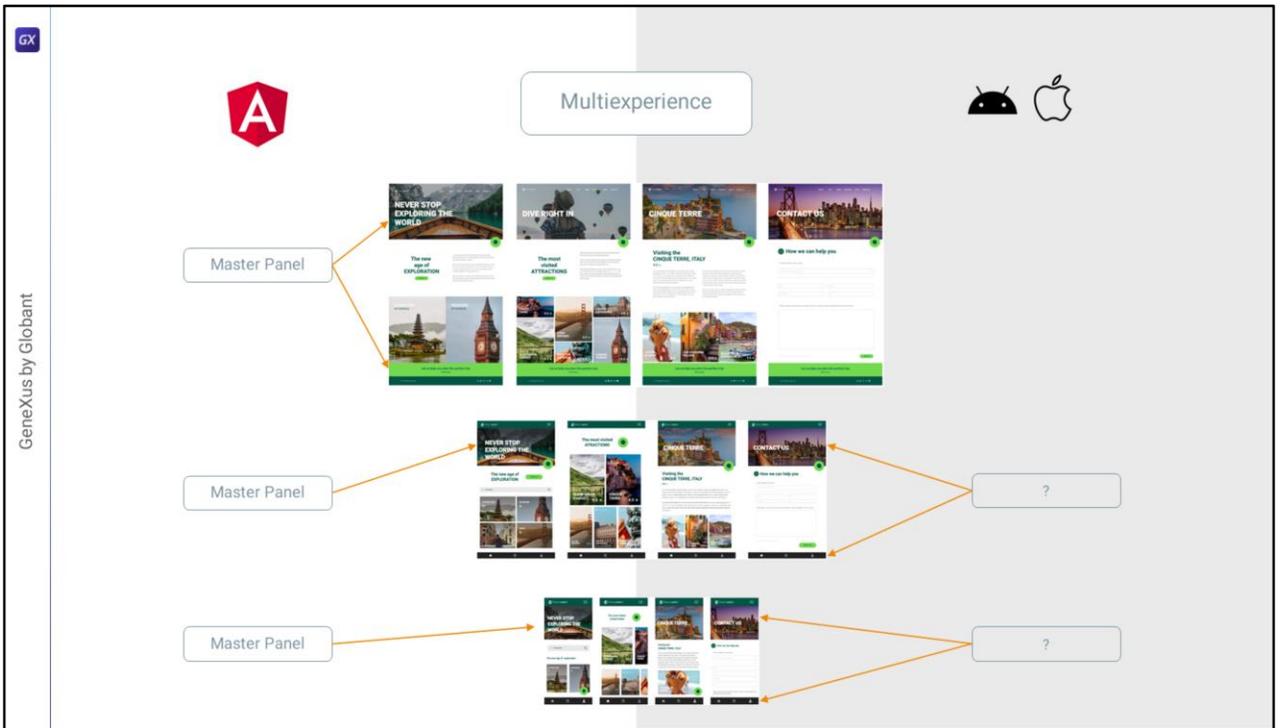
Y bueno, por supuesto esto es posible de hacer, lo que tenemos que tratar de lograr es concentrarnos primero en todo lo que es común, para recién, cuando debamos atender a las diferencias diferenciar justamente, para partir de un tronco común.

# MUX: Angular & Native Mobile

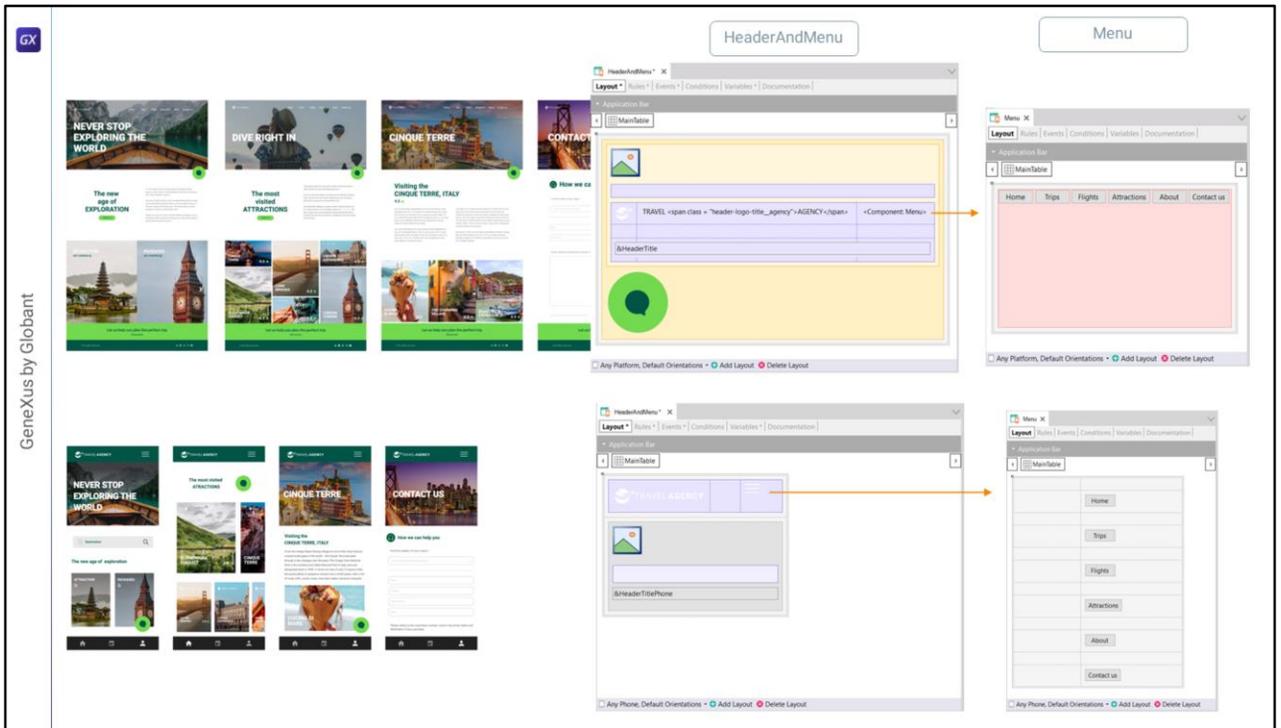


En definitiva, tendríamos que haber replicado buena parte de los módulos 1 a 4 para lo nativo, y luego agregar un último módulo que sea la síntesis de ambos paradigmas.

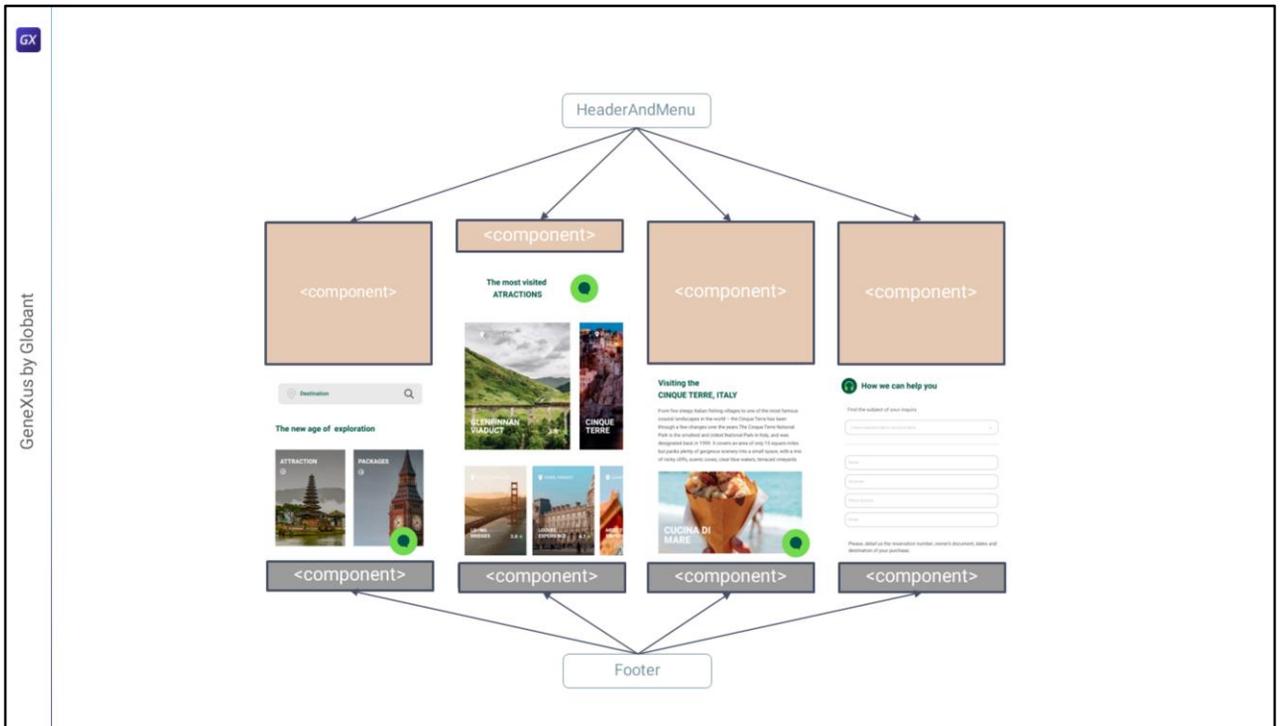
Entonces, si empezáramos ahora de cero la aplicación, considerando todo a la vez, se me ocurren algunas ideas de cómo organizaría las cosas, y se las quiero compartir, para que vayamos pensándolas más allá de este curso.



Para poder utilizar la solución más parecida posible para Angular y para lo nativo, sabiendo que la aplicación nativa no podrá utilizar Master Panel, se me ocurren varias ideas, pero les voy a compartir una.



Implementar Application Bar y Header en un panel independiente...



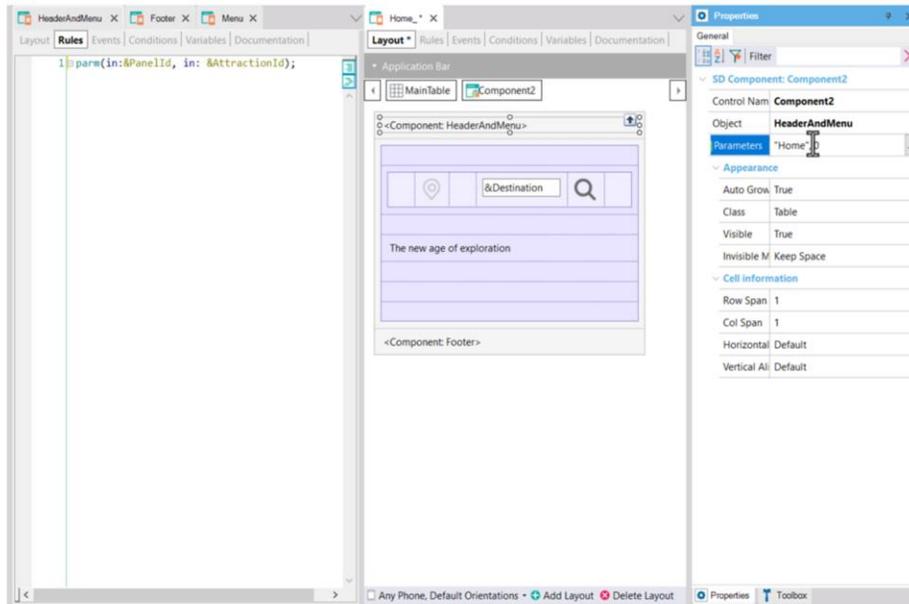
...e insertarlo como control **component** en los 4 paneles. Hacer lo mismo con la tab bar (convendrá utilizar componente en lugar de stencil para no tener que repetir la codificación de los eventos tap de cada imagen).

Todo esto para utilizar exactamente la misma solución que para la aplicación nativa que para la de Angular.

The screenshot displays the GeneXus development environment. The main workspace is divided into two panels. The left panel shows a layout editor with a grid of three icons: a home icon, a calendar icon, and a user icon. The right panel shows a component editor with a search bar containing the text "&Destination" and a magnifying glass icon, followed by the text "The new age of exploration". A right-hand sidebar contains a "Toolbox" with various controls and containers. The toolbox is organized into several categories: Controls (Attribute/Variable, Button, Image, Text Block), Containers (Canvas, Component, Flex, Grid, Group, Stencil, Tab, Table, Tabular Grid), Miscellaneous (Ads View, Animation View, Audio Controller, Query Viewer), and GeneXusTranimo (Alert Angular, Dropdown Angular, Icon Angular, Select Angular, Sidebar Angular, Step Angular, Treeview Angular). At the bottom of the screen, there are options to "Add Layout" and "Delete Layout" for the current component.

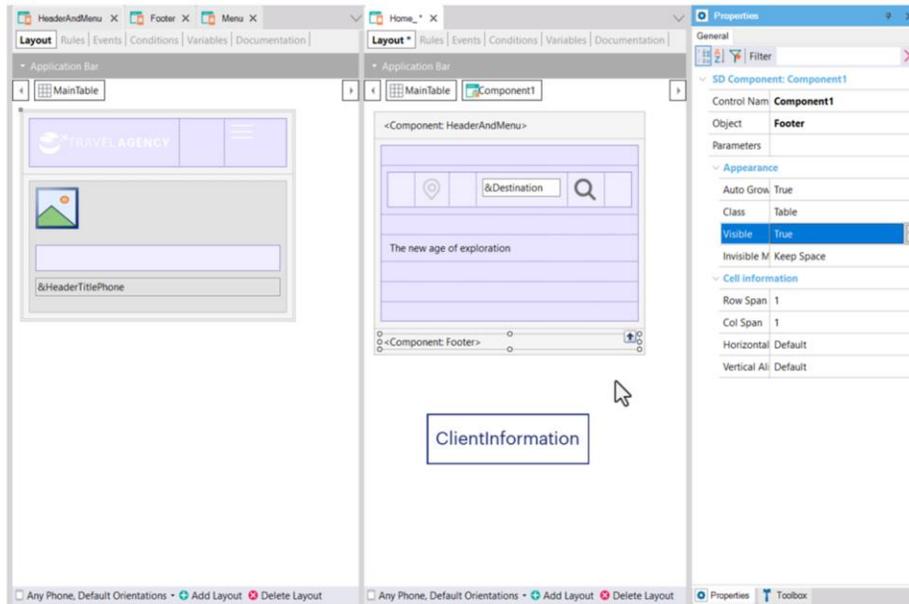


No lo vimos en este curso, pero al insertar un control componente...



...podemos pasarle parámetros (lo que no podíamos hacer con el Master Panel y por eso habíamos tenido que utilizar los eventos globales). Entonces para esta solución no necesitaremos los eventos globales, que sí son válidos, tanto para lo nativo como para Angular, por otra parte.

Cada panel que se esté cargando, cargará dinámicamente al componente HeaderAndMenu, pasándole un identificador de sí mismo. Y así el componente sabrá qué deberá cargar para la imagen y para el título. Y en el caso Attractions, que no deberá cargarlos siquiera.



Y en cuanto al menú de tabs, podemos cargar el componente únicamente si la aplicación se está ejecutando en un dispositivo Android o Apple (utilizando, como ya sabemos, el objeto externo ClientInformation, para saber, justamente, en qué plataforma se está ejecutando la aplicación).

Parece sencillo y parece que funcionaría. Habría que probar esta solución para ver si aparecen obstáculos, pero... lamentablemente este curso llegó a su fin. Chau.

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

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