

FontSizes system in GeneXus



Cecilia Fernández

GeneXus by Globant

GX

The new age of EXPLORATION

Tell me more

Token	Desktop	Tablet	Phone
XL	100	60	40
L	67	40	20
M	20	16	14
S	16	14	12
XS	14	14	12
card_XL	42	20	15
card_L	23	-	-
card-attractions-Big_XL	36	36	20
card-attractions-Big_M	38	38	16
card-attractions-Big_S	14	14	12
card-attractions-Small_XL	36	20	12
card-attractions-Small_M	38	16	12
card-attractions-Small_S	14	12	10
card-attraction_XL	36	23	24
card-attraction_M	38	21	-
form_M	20	12	12
form_S	16	10	10
banner_XL	36	-	-
banner_L	20	-	-

STOP BRING THE

plan the perfect trip

the subject of your inquiry

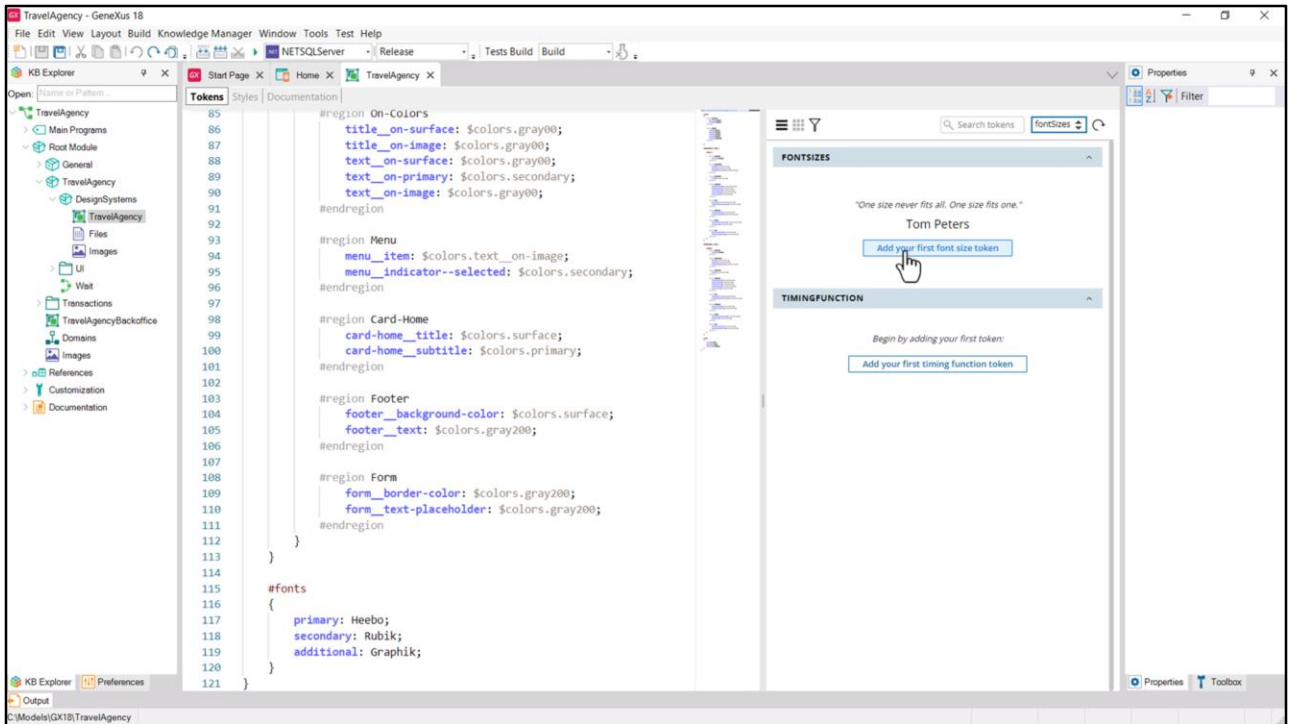
At Travel Agency, we have consultants with an average of 10 years of experience and a passion for travel available who will help you plan and create unforgettable vacations.

We will work with you to plan a worry free adventure that meets your needs, expectations and budget. When you plan your trip, our consultants are there throughout the entire process. This means you can have all the information available to you before, during and after travel.

We plan river and ocean cruises, romantic honeymoon destinations, family, adventure and wellness travel, all tailored to your and domestic customized itineraries.

Nos quedó pendiente del video anterior llevar los tokens de tamaño de fuentes a GeneXus.

Vamos a hacerlo solo con los tokens para Desktop. Esto significa que no vamos a agregarle la opción breakpoint al conjunto de tokens, como habíamos visto un par de videos atrás, no sé si se acuerdan. Si me tienen un poquito de paciencia ya van a entender por qué.



Si uno no tiene muy clara la sintaxis para agregar tokens para fontSizes, conviene el primero agregarlo desde el editor gráfico. Así...

The image shows a development environment with two main windows. The left window is a code editor showing a list of tokens and their values. The right window is a spreadsheet showing a table of token values for different device types.

Code Editor Tokens:

```

90 text_on-image: $colors.gray00;
91 #endregion
92
93 #region Menu
94 menu_item: $colors.text_on-image;
95 menu_indicator--selected: $colors.secondary;
96 #endregion
97
98 #region Card-Home
99 card-home_title: $colors.surface;
100 card-home_subtitle: $colors.primary;
101 #endregion
102
103 #region Footer
104 footer_background-color: $colors.surface;
105 footer_text: $colors.gray200;
106 #endregion
107
108 #region Form
109 form_border-color: $colors.gray200;
110 form_text-placeholder: $colors.gray200;
111 #endregion
112 }
113
114 #fonts
115 {
116 primary: Heebo;
117 secondary: Rubik;
118 additional: Graphik;
119 }
120
121 #fontSizes
122 {
123 XL: 100px;
124 }
125
126 }

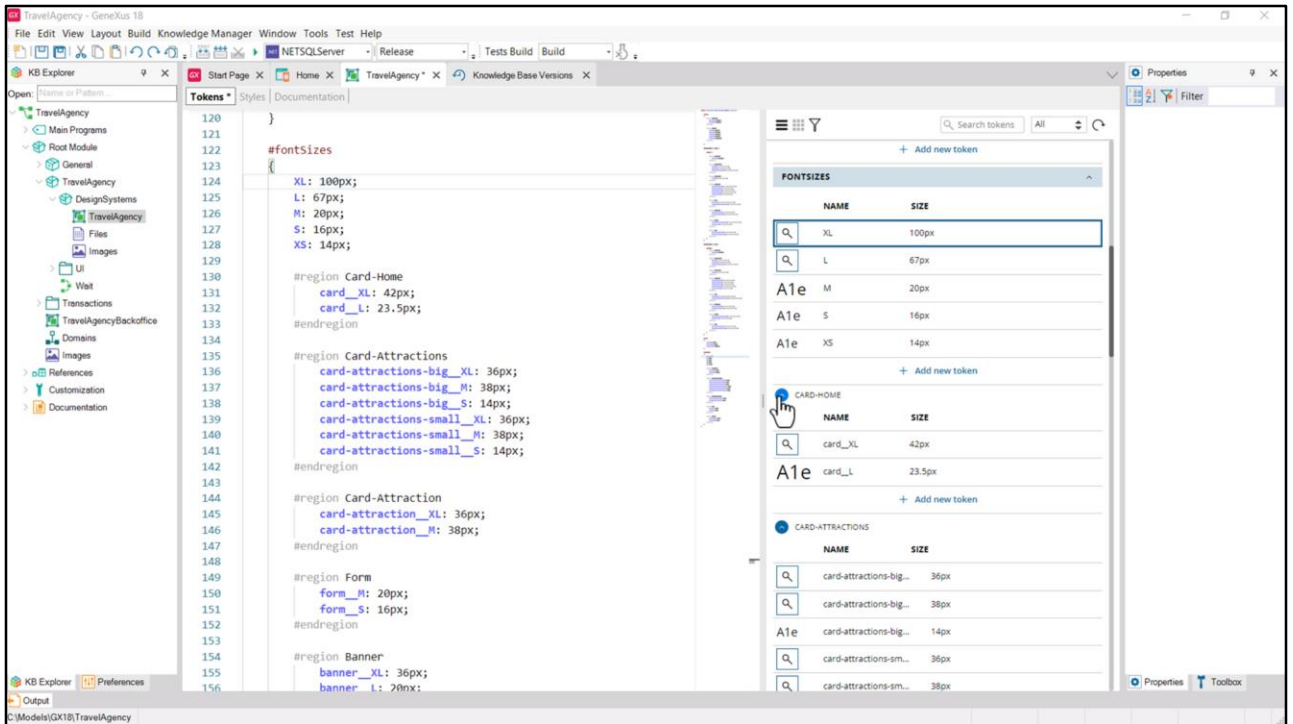
```

Spreadsheet Table:

	A	B	C	D	E	F
	Token	Desktop	Tablet	Phone		
2	XL		100	60	40	
3	L		67	40	20	
4	M		20	16	14	
5	S		16	14	12	
6	XS		14	14	12	
7						
8	card_XL		42	20	15	
9	card_L		23.5	-	-	
10						
11	card-attractions-Big_XL		36	36	20	
12	card-attractions-Big_M		38	38	16	
13	card-attractions-Big_S		14	14	12	
14						
15	card-attractions-Small_XL			20	12	
16	card-attractions-Small_M		38	16	12	
17	card-attractions-Small_S		14	12	10	
18						
19	card-attraction_XL		36	23	24	
20	card-attraction_M		38	21	-	
21						
22	form_M		20	12	12	
23	form_S		16	10	10	
24						
25	banner_XL		36	-	-	
26	banner_L		20	-	-	

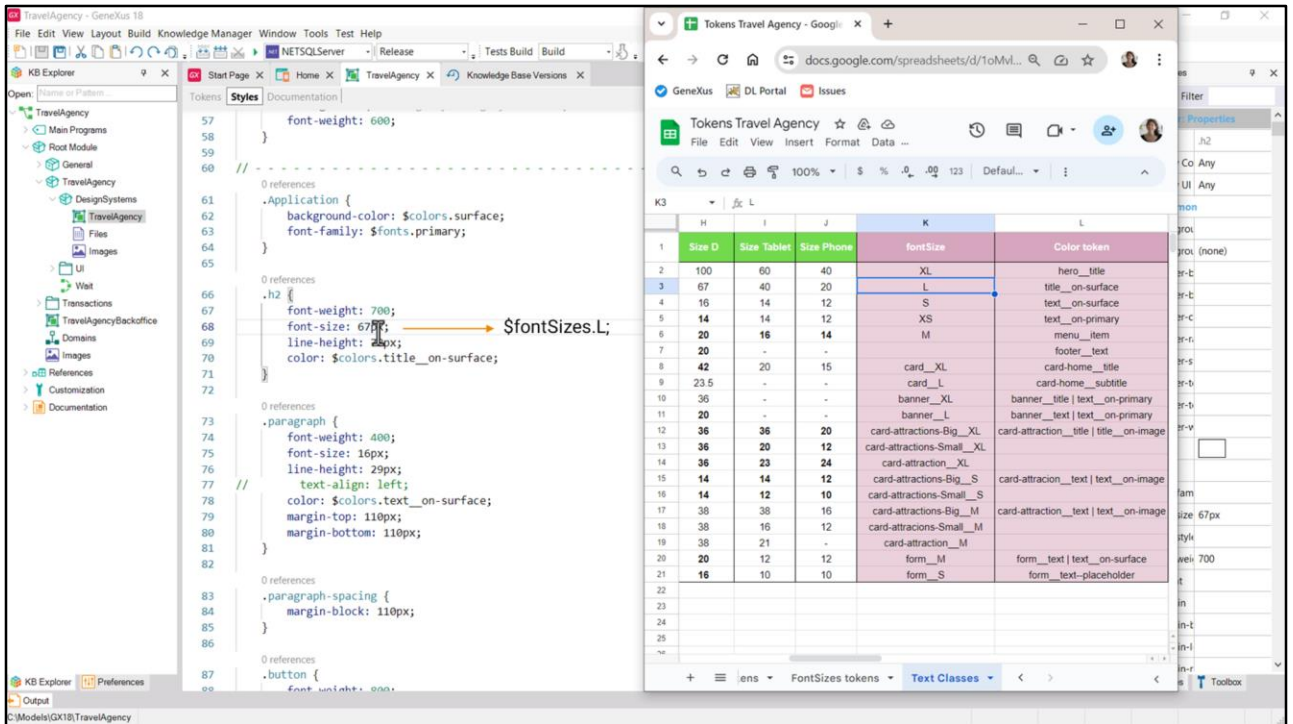
Y allí le da el nombre que quiere... y el valor.

A partir, entonces, del resultado final del análisis que habíamos efectuado en el video anterior y que se expresaba en esta planilla, podemos ingresar estos tokens. Como les decía, sólo utilizaré los valores para Desktop en este DSO.



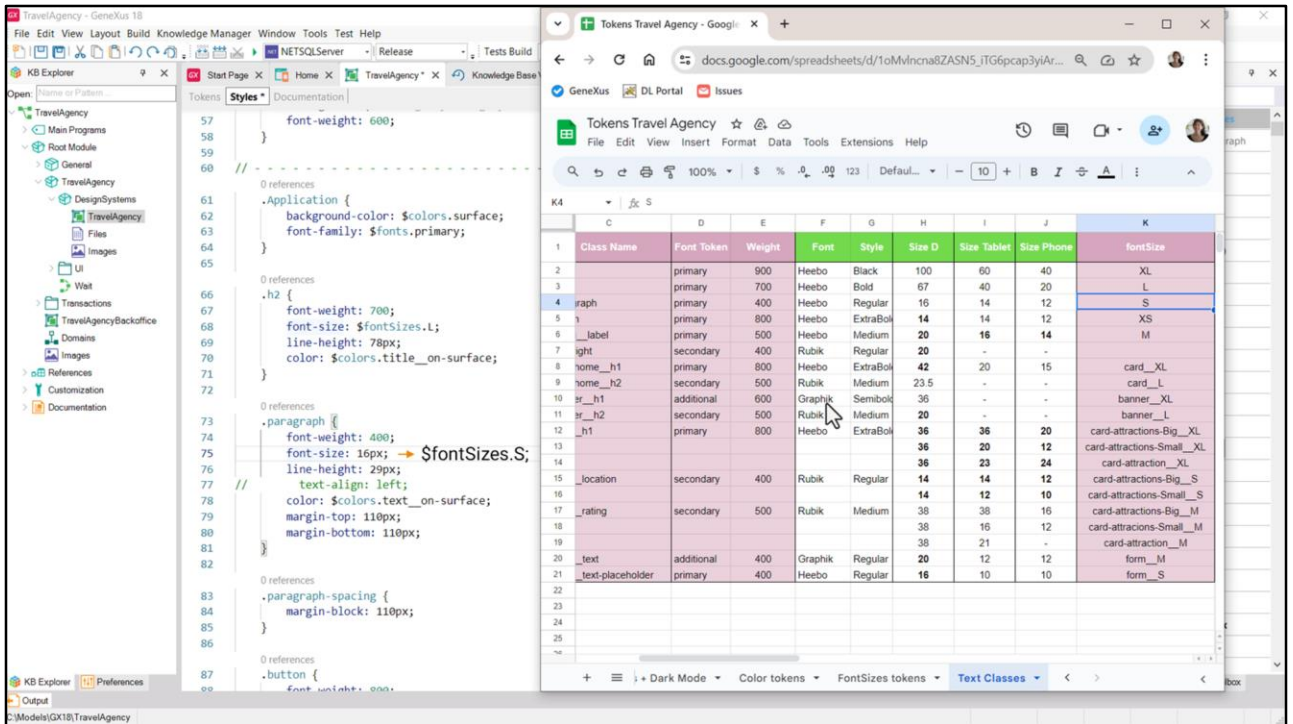
Aquí los copié todos, colocando regiones, como antes, con fines organizativos. Podemos ver, en el editor gráfico, así representada la información.

A continuación deberíamos aprovechar a cambiar, en las clases que tenemos ya definidas, todas las propiedades font-size para que utilicen ahora el token de tamaño de fuente apropiado.



Así, si buscamos en nuestra planilla, en la solapa donde teníamos las clases de texto el token para h2... era el de nombre L.

Entonces aquí escribimos pesos para referir a un token, de tipo fontSizes y de nombre L.



Luego buscamos el token para la clase paragraph... es el de nombre S.

Por lo que sustituimos por... fontSizes.S.

The image shows a development environment with two windows. The left window is a code editor (likely Visual Studio Code) showing CSS code for a 'TravelAgency' project. The right window is a Google Sheet titled 'Tokens Travel Agency' containing a table of design tokens.

Code Editor (Left Window):

```

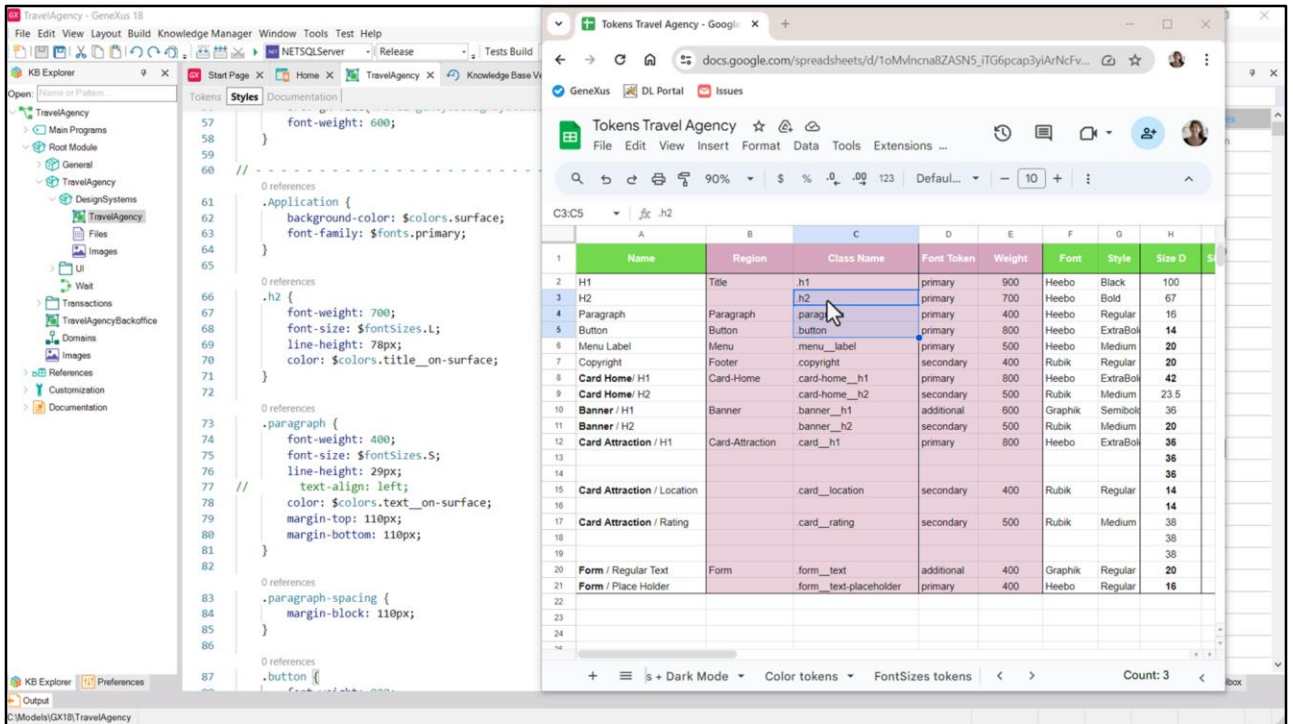
66 0 references
67  .h2 {
68    font-weight: 700;
69    font-size: $fontSizes.L;
70    line-height: 78px;
71    color: $colors.title_on-surface;
72  }
73 0 references
74  .paragraph {
75    font-weight: 400;
76    font-size: $fontSizes.S;
77    line-height: 29px;
78    text-align: left;
79    color: $colors.text_on-surface;
80    margin-top: 110px;
81    margin-bottom: 110px;
82  }
83 0 references
84  .paragraph-spacing {
85    margin-block: 110px;
86  }
87 0 references
88  .button {
89    font-weight: 800;
90    font-size: 14px;
91    line-height: 18px;
92    text-align: center;
93    text-transform: uppercase;
94    color: $colors.text_on-primary;
95    border-radius: 30px;
96    max-inline-size: max-content;
97    padding-inline: 45px;

```

Google Sheet (Right Window):

	E	F	G	H	I	J	K	L
1	Weight	Font	Style	Size D	Size Tablet	Size Phone	fontSize	Color token
2	900	Heebo	Black	100	60	40	XL	hero_title
3	700	Heebo	Bold	67	40	20	L	title_on-surface
4	400	Heebo	Regular	16	14	12	S	text_on-surface
5	800	Heebo	ExtraBoli	14	14	12	XS	text_on-primary
6	500	Heebo	Medium	20	16	14	M	menu_item
7	400	Rubik	Regular	20	-	-	-	footer_text
8	800	Heebo	ExtraBoli	42	20	15	card_XL	card-home_title
9	500	Rubik	Medium	23.5	-	-	card_L	card-home_subtitle
10	600	Graphik	Semibold	36	-	-	banner_L	banner_title text_on-primary
11	500	Rubik	Medium	20	-	-	banner_L	banner_text text_on-primary
12	800	Heebo	ExtraBoli	36	36	20	card-attractions-Big_XL	card-attraction_title title_on-image
13				36	20	12	card-attractions-Small_XL	
14				36	23	24	card-attraction_XL	
15	400	Rubik	Regular	14	14	12	card-attractions-Big_S	card-attraction_text text_on-image
16				14	12	10	card-attractions-Small_S	
17	500	Rubik	Medium	38	38	16	card-attractions-Big_M	card-attraction_text text_on-image
18				38	16	12	card-attractions-Small_M	
19				38	21	-	card-attraction_M	
20	400	Graphik	Regular	20	12	12	form_M	form_text text_on-surface
21	400	Heebo	Regular	16	10	10	form_S	form_text-placeholder

Y sólo nos resta la clase button... que será XS.



Recordemos que tenemos estas tres clases nada más porque, con fines didácticos, habíamos estado trabajando ya en ellas en el módulo anterior, con bastante detalle, para que ustedes pudieran paladear las fibras de todo esto y entender un poco de qué va la cosa.

Pero cuando todo eso ya es sabido y hay que empezar a trabajar en un proyecto, en general se hace lo que estuvimos haciendo en este módulo: ingresar los assets primero, es decir, los recursos (imágenes, fuentes... y en este caso también incluirlas en el DSO con las reglas font-face) todo lo que estuvimos haciendo al principio de este módulo...

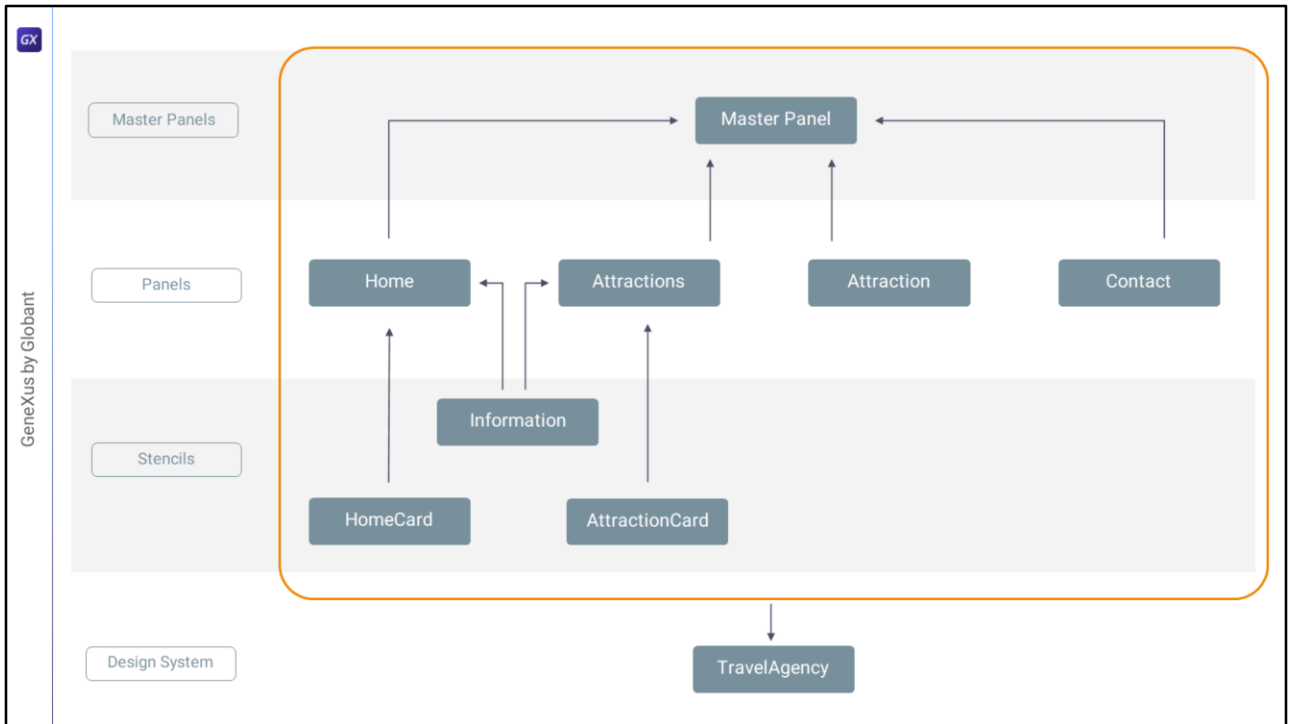
...armar el sistema de colores, de tamaño de fuentes. Y, también de espaciados, radios, bordes (no lo haremos aquí, para no aburrir, porque es el mismo tipo de análisis que ya hicimos para los colores y para los tamaños de fuentes). Agregaríamos, entonces, también esos tokens para utilizarlos en las propiedades correspondientes...

Y luego de todo eso, ingresaríamos todas estas clases de texto que ya tenemos identificadas (no sólo las 3 que ya tenemos ingresadas sino todas) para que cuando empecemos a trabajar objeto por objeto, no tengamos que detenernos en cosas más evidentes, como estas, y podamos concentrarnos en el problema concreto. Lo haremos en el próximo video.

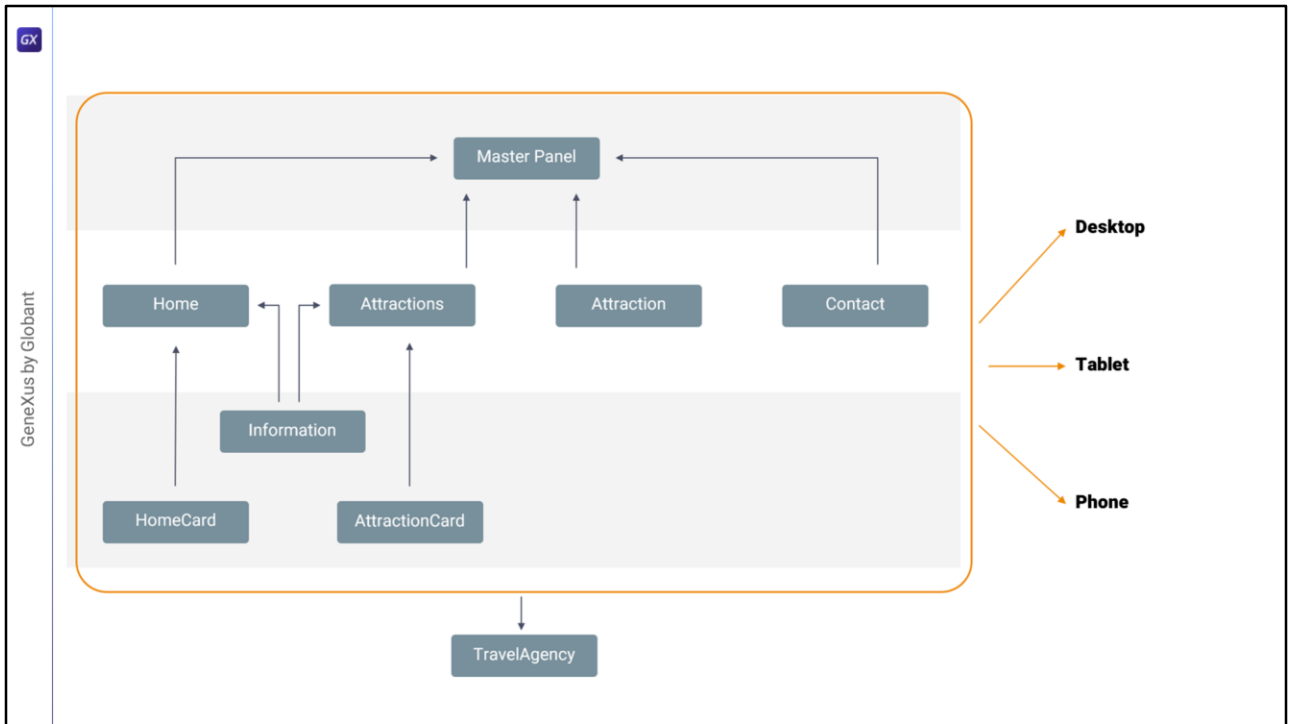


Ahora para terminar este video sólo quiero mencionar, para que no quede como un misterio, por qué no agregué una opción breakpoint para hacer variar los tokens por tamaño de pantalla.

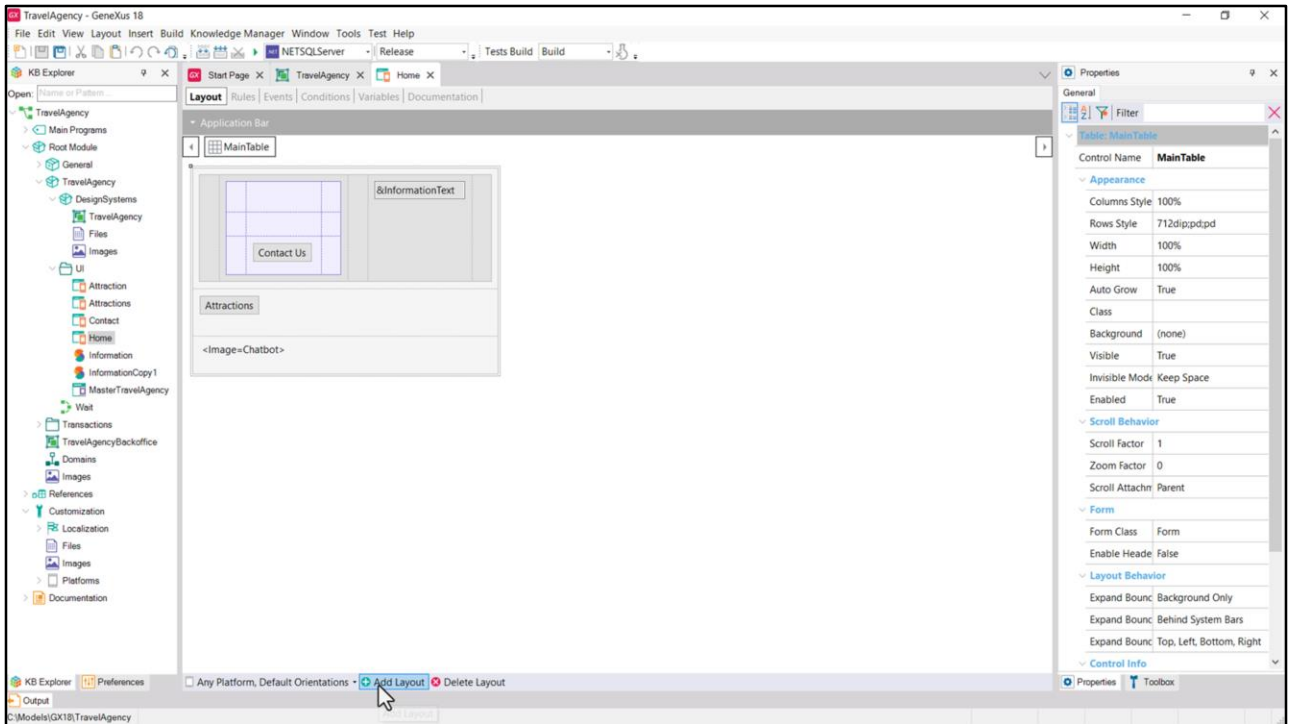
Esta que voy a utilizar yo es una alternativa posible entre varias. No es que sea la única. Voy a pensar los distintos tamaños de pantalla como distintas versiones de un mismo concepto: el Design System para la aplicación.



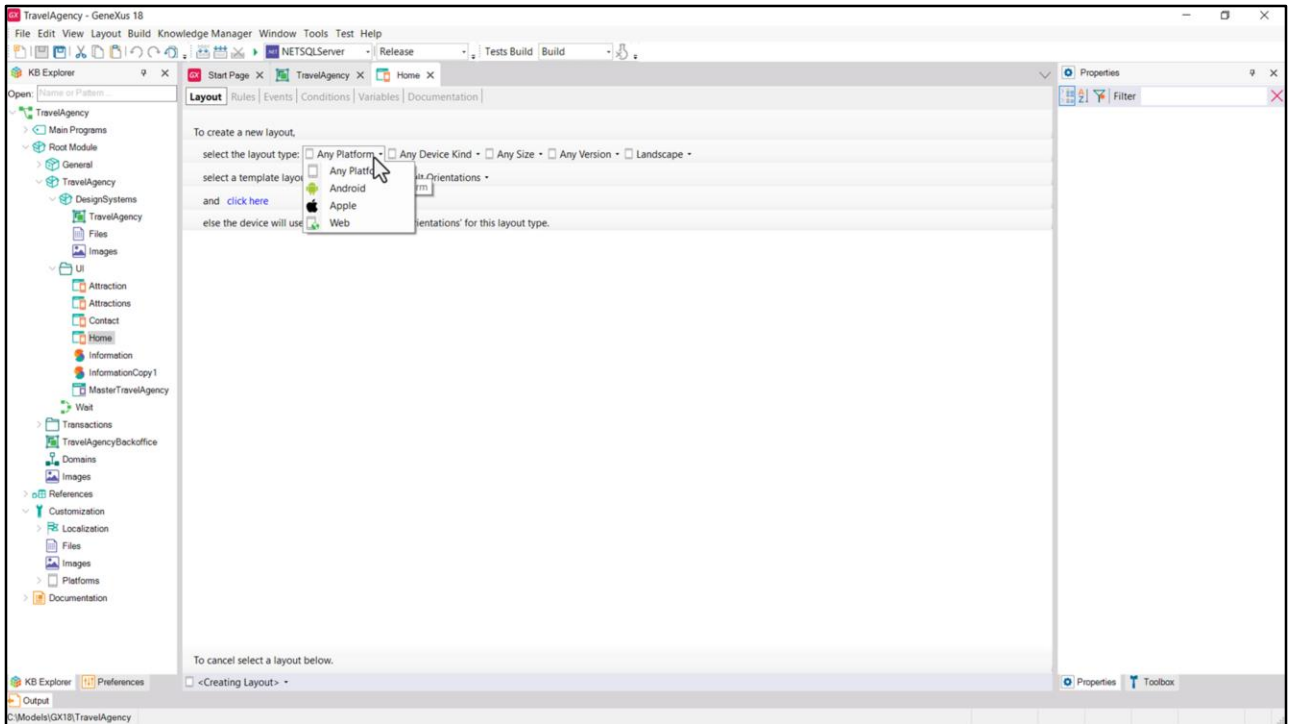
¿Se acuerdan que en una clase muy del principio donde habíamos analizado todos los jugadores, habíamos separado así las cosas? Aquí todos los objetos que iban a implementar la User Interface... y aquí el objeto Design System.



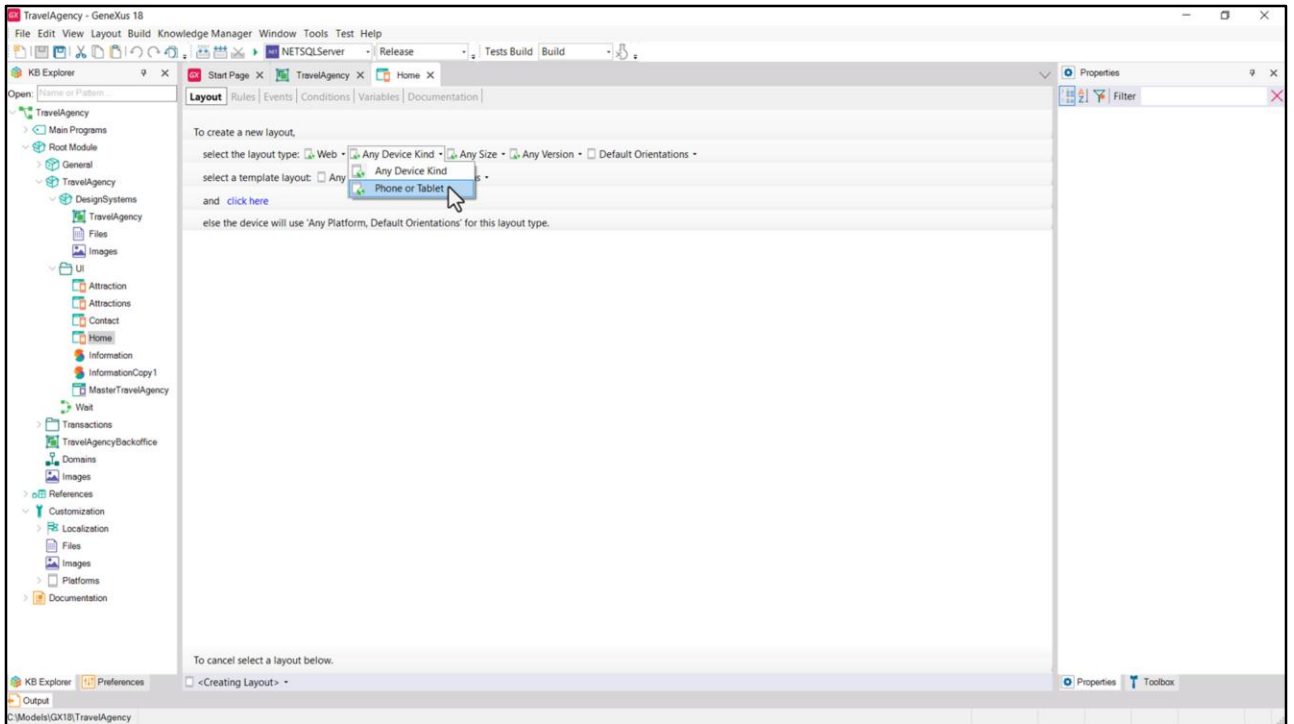
Bueno, pues resulta que todos estos objetos con layout podrán definir distintas versiones del layout de acuerdo a la plataforma...



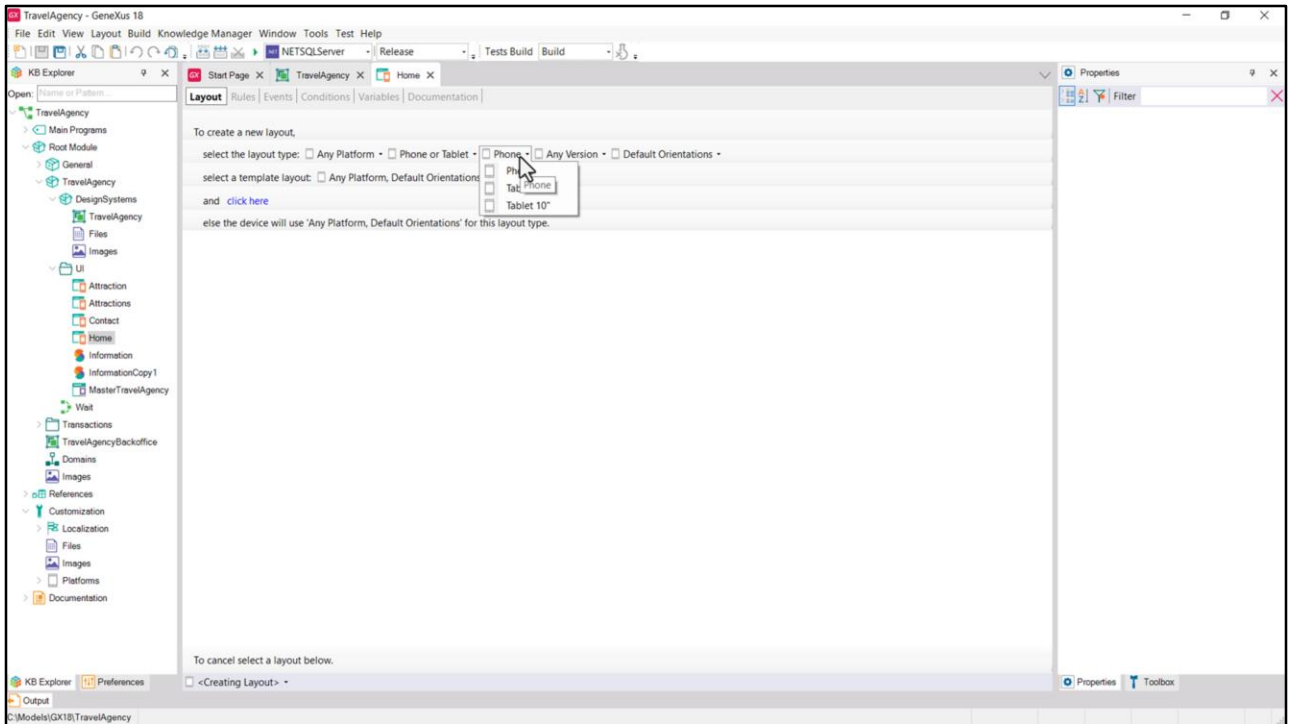
Veamos, por ejemplo, el panel Home. Por el momento solo tiene un layout definido, que vale para toda plataforma en toda orientación. Podríamos agregarle otros layouts, específicos.



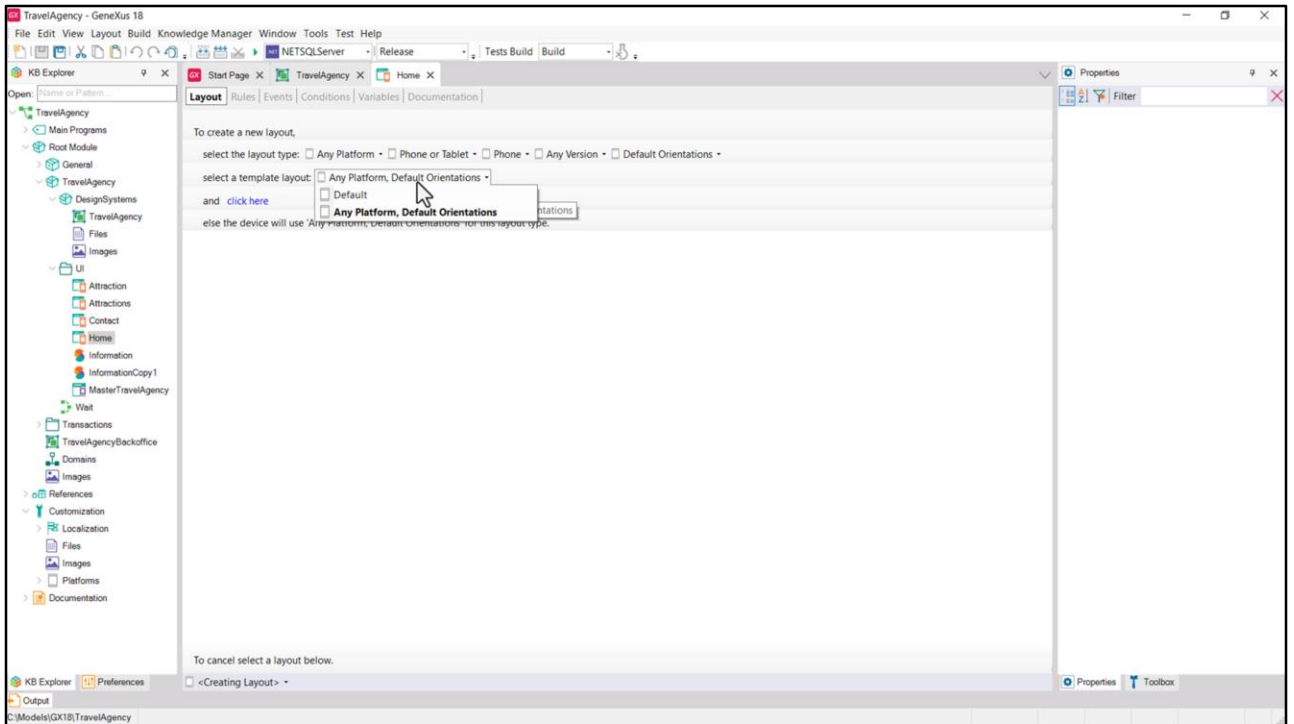
Por ejemplo, para discriminar por plataforma: Web, Android o Apple.



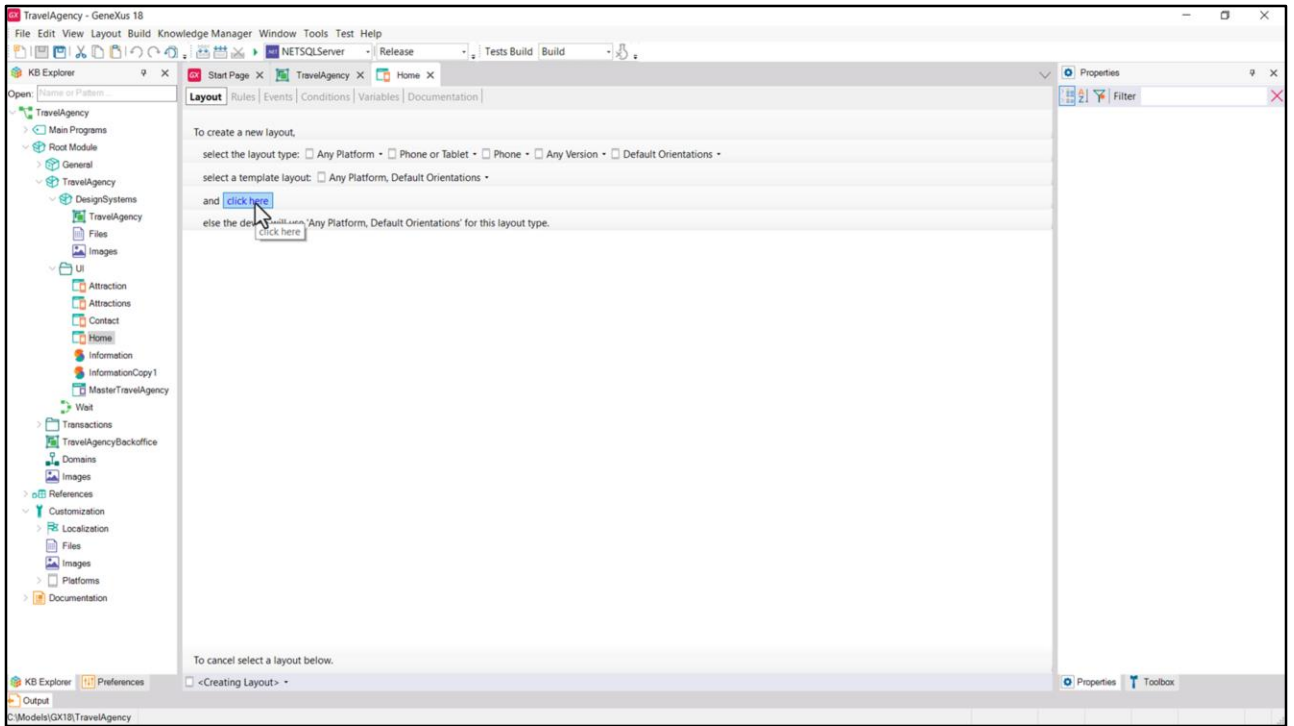
Por ejemplo, elijo Web, y también puedo discriminar si es Web para cualquier tipo de dispositivo, o para Phone o Tablet.



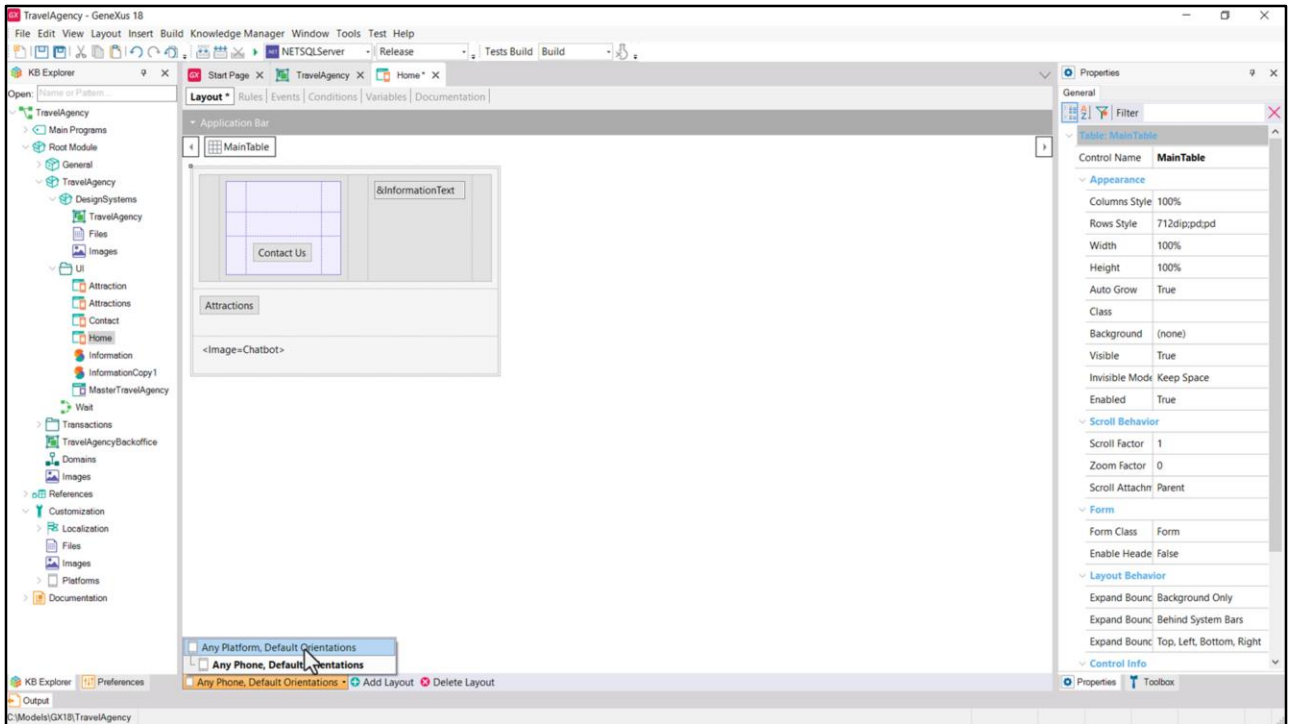
O, por ejemplo, puedo agregar un nuevo layout que valga para cualquier plataforma (nativa o web) pero para tamaño Phone o Tablet. Y aquí elegir si será Phone o será Tablet (de 7 o de 10 pulgadas)... y así.



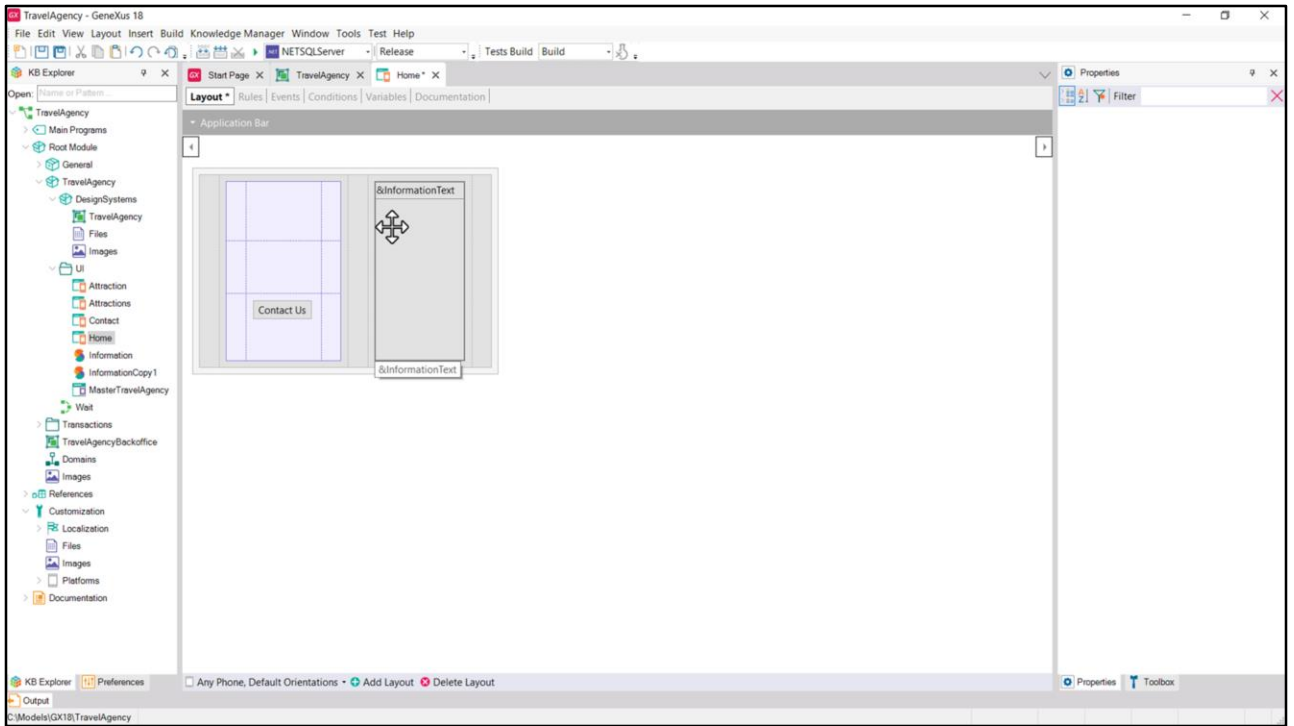
Por ejemplo, si elijo Phone y quiero inicializar el layout con este, que es el único que tenemos hasta ahora...



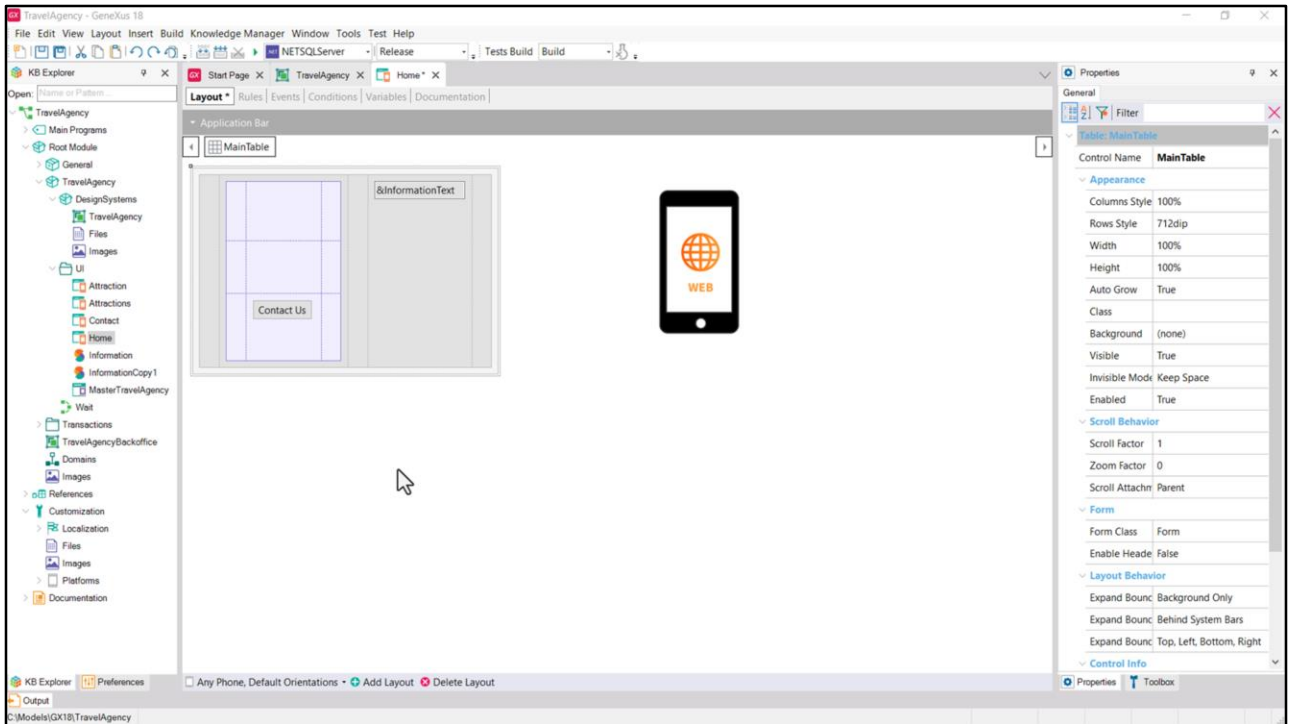
...hago click aquí.



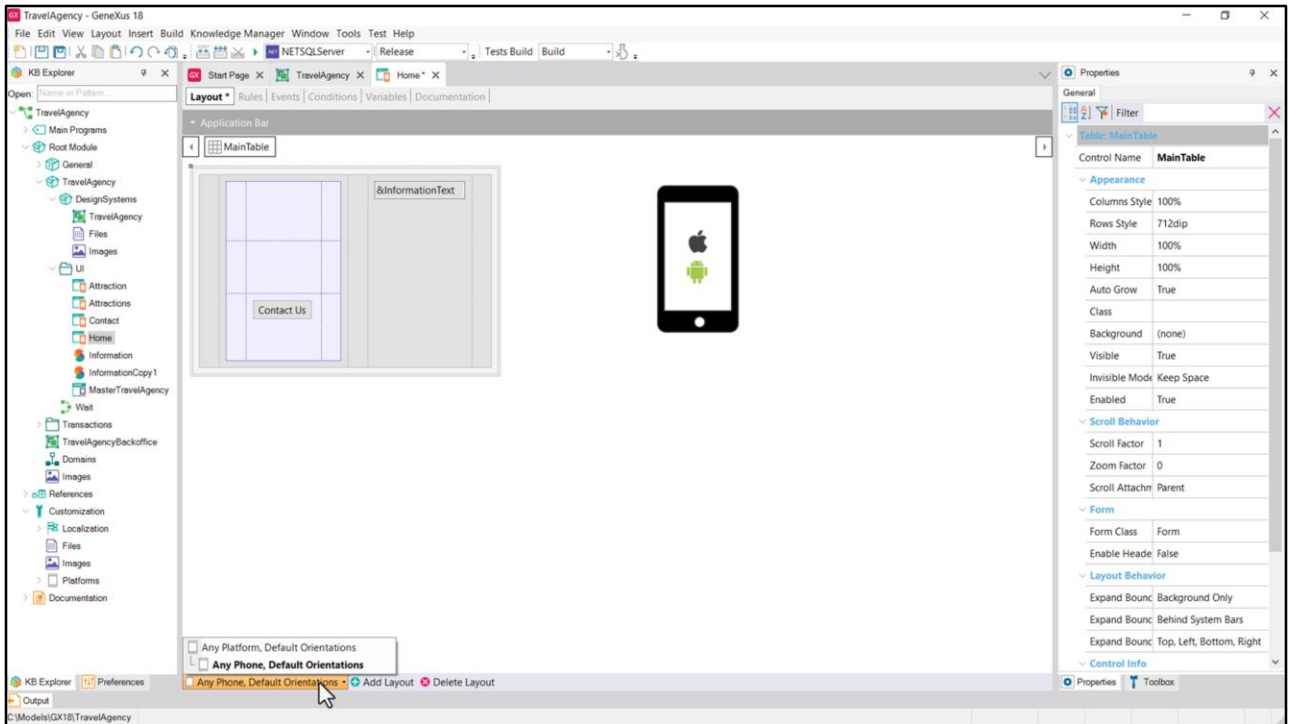
Y ahí vemos que ahora tenemos dos layouts... este es una especialización de este otro. Y entonces puedo manipular este aparte del otro.



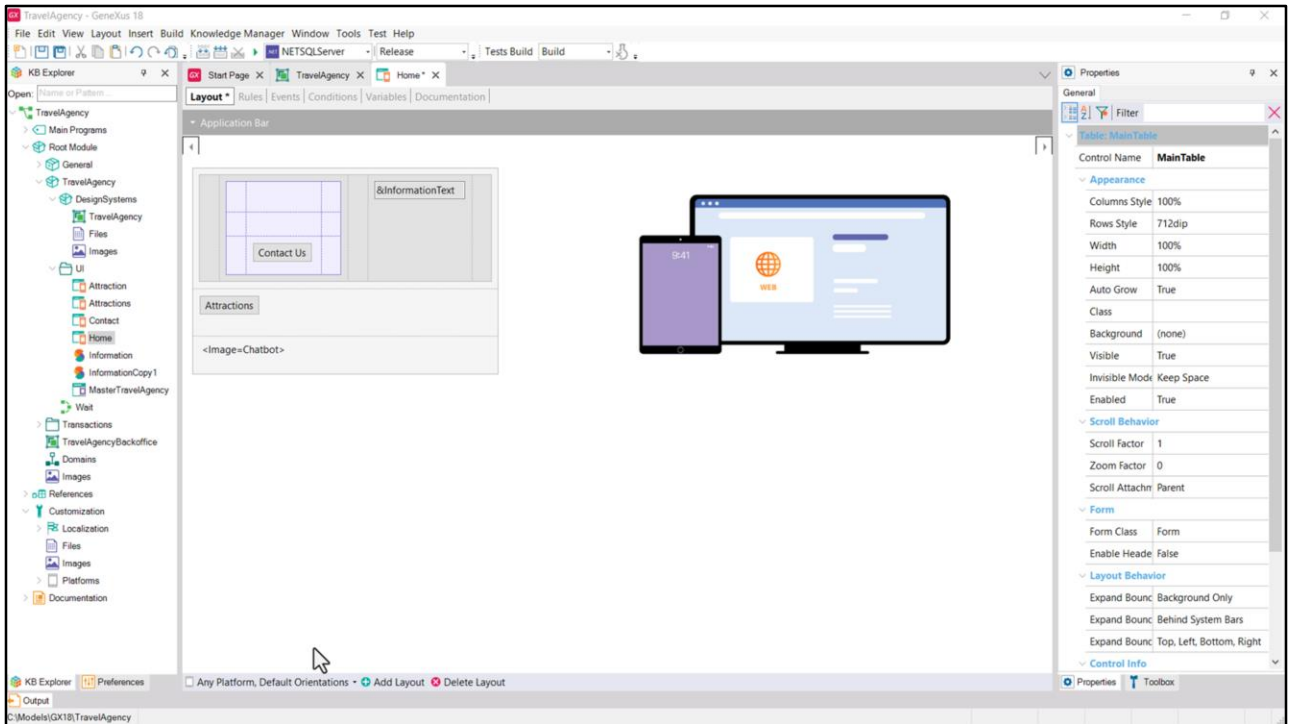
Por ejemplo...



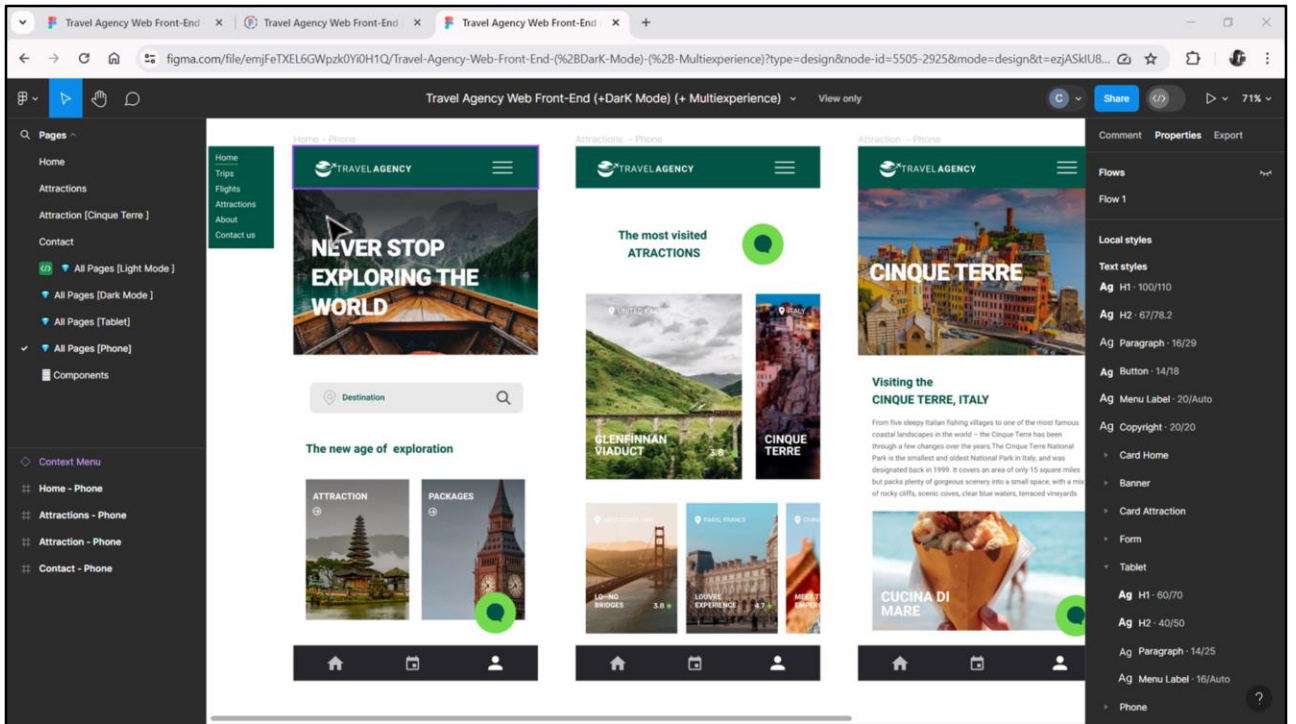
Si ejecuto luego este panel en mi aplicación web Angular en un teléfono, utilizará este layout porque entre los dos que hay, es el que más se aproxima al dispositivo y plataforma de ejecución.



Si ejecutara la aplicación en ese mismo teléfono pero la aplicación nativa, Android o Apple, también utilizará este layout.

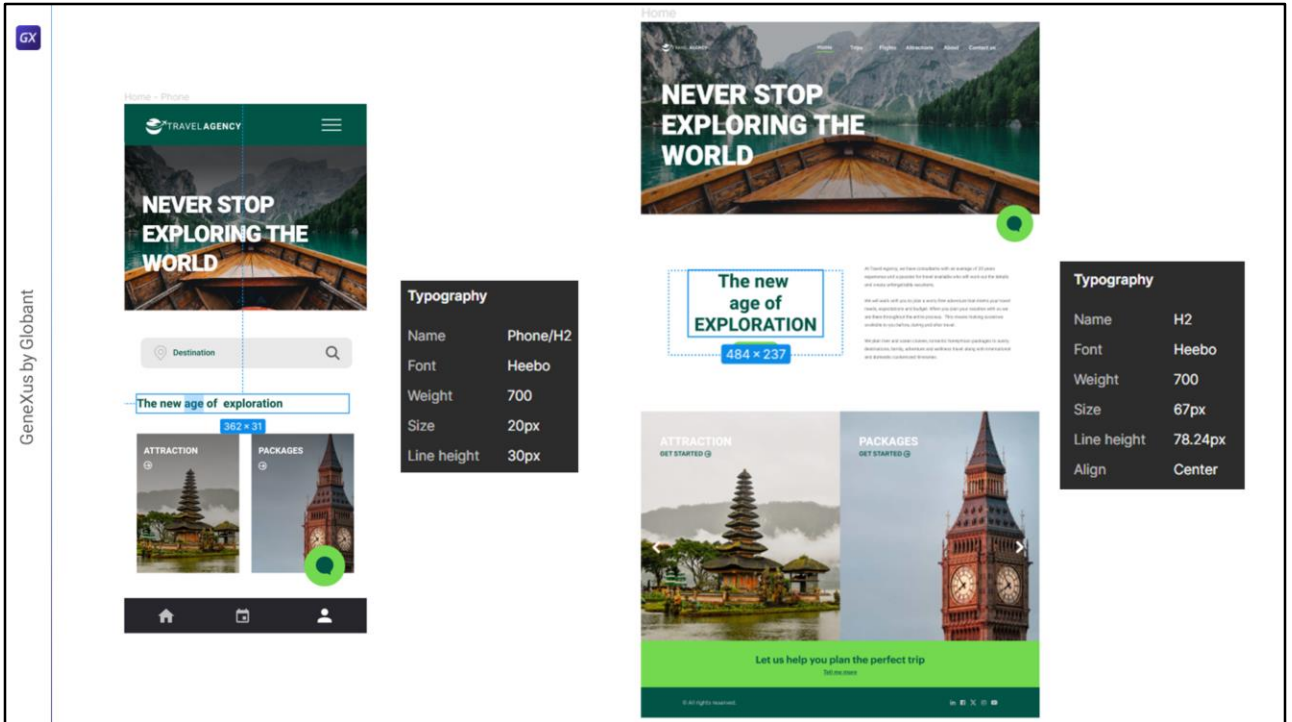


En cambio, si ejecuto la aplicación en cualquier otro dispositivo y plataforma, utilizará este otro, que es el más general. Así, si ejecuto la aplicación web Angular tamaño desktop, será este el layout.

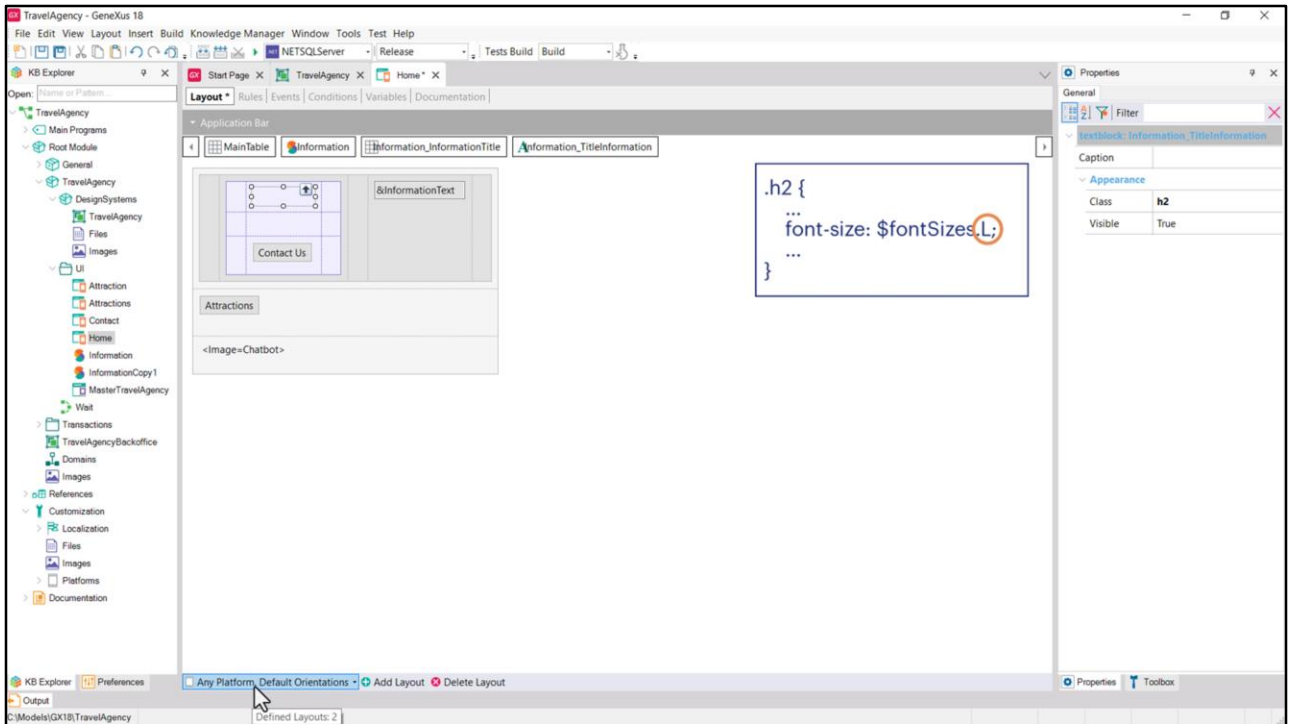


De hecho tendremos que rediseñar para tamaño Phone casi todo el layout porque será bastante diferente en su estructura.

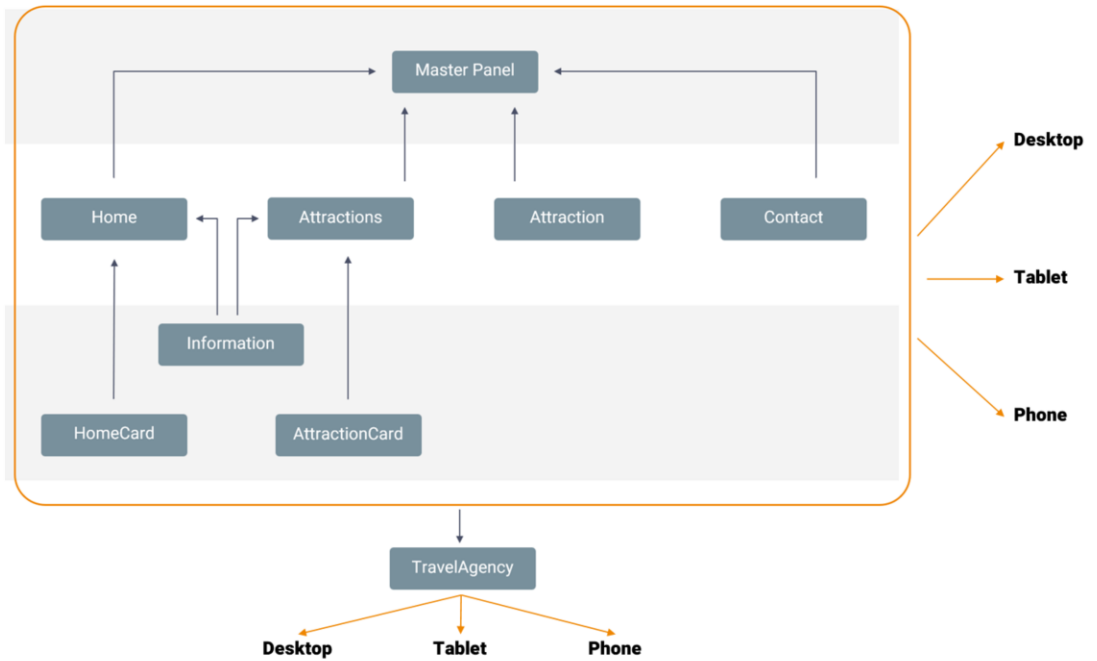
Sobre este asunto de los layouts volveremos sobre el final del curso. Ahora no me quiero concentrar eso. Pero sí en los estilos de los elementos...



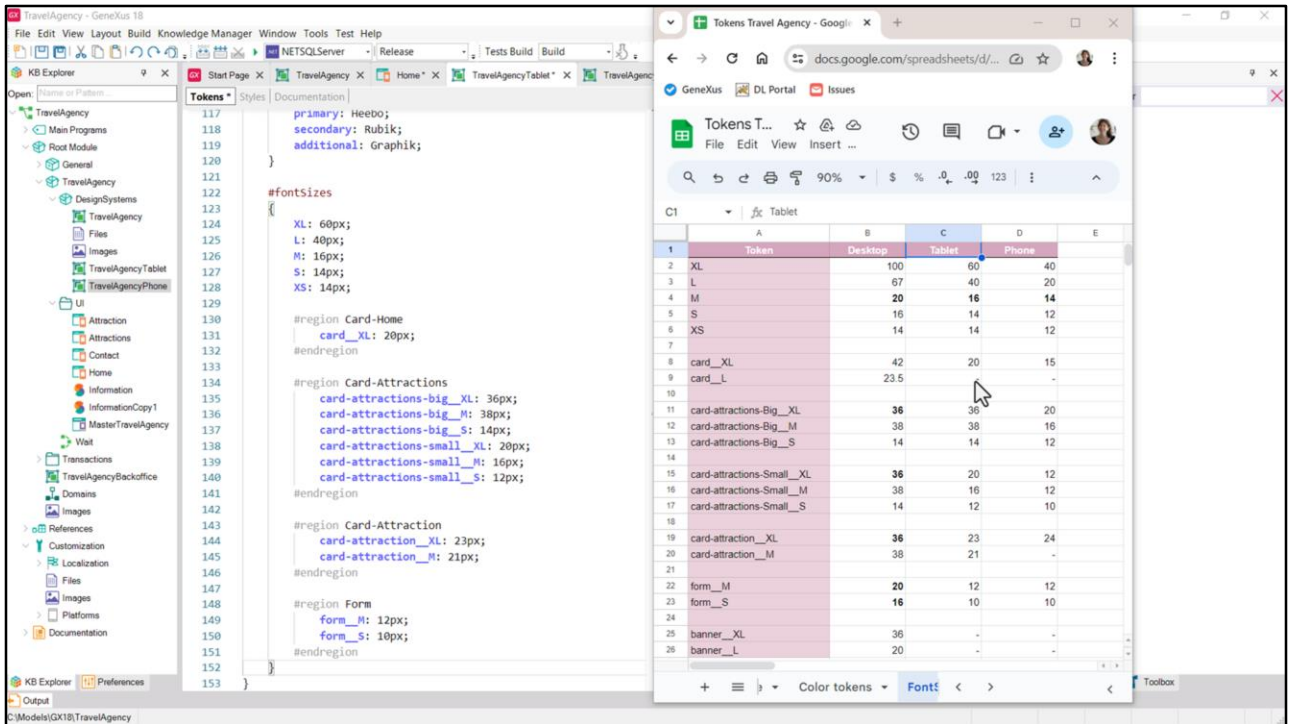
Por ejemplo, ambos layouts compartirán este texto, que corresponde al mismo estilo tipográfico... el h2.



Ahora bien, ¿qué pasa con la propiedad font-size de esa clase? Le podemos decir que corresponderá al token \$fontSizes.L, sí, perfecto. Pero no debería tener el mismo tamaño para la aplicación Desktop que para la que se ejecute en el teléfono. En un caso debería ser de 67 píxeles y en el otro de 20.



Una forma de conseguir esto es pensar también versiones del DSO.



Por ejemplo, al DSO para Desktop dejarle el nombre TravelAgency, y hacer un save as de ese con nombre TravelAgencyTablet.

Y otro con nombre TravelAgencyPhone.

Y a cada uno cambiarle los valores de los tokens que varían por tamaño de pantalla. En este caso les asignaríamos los valores de esta columna... vean que los que no se usan para este tamaño no los especifico...

The image shows a development environment with a code editor on the left and a spreadsheet on the right. The code editor displays CSS tokens for font sizes and regions. The spreadsheet on the right lists font sizes for various tokens across different device types: Desktop, Tablet, and Phone.

Code Editor Content:

```

primary: Heebo;
secondary: Rubik;
additional: Graphik;
}

#fontSizes
{
  XL: 40px;
  L: 20px;
  M: 14px;
  S: 12px;
  XS: 12px;
}

#region Card-Home
  card_XL: 15px;
#endregion

#region Card-Attractions
  card-attractions-big_XL: 20px;
  card-attractions-big_M: 16px;
  card-attractions-big_S: 12px;
  card-attractions-small_XL: 12px;
  card-attractions-small_M: 12px;
  card-attractions-small_S: 10px;
#endregion

#region Card-Attraction
  card-attraction_XL: 24px;
#endregion

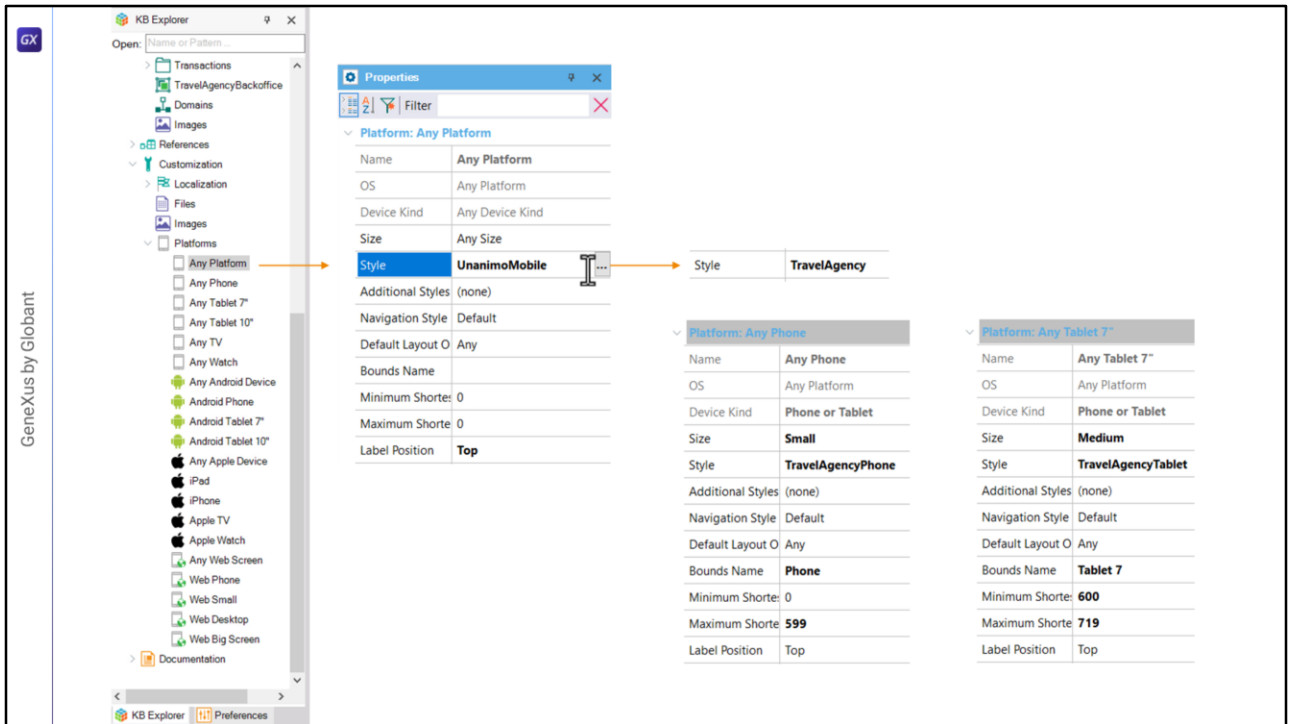
#region Form
  form_M: 12px;
  form_S: 10px;
#endregion

```

Spreadsheet Content:

	A	B	C	D	E
	Token	Desktop	Tablet	Phone	
1					
2	XL	100	60	40	
3	L	67	40	20	
4	M	20	16	14	
5	S	16	14	12	
6	XS	14	14	12	
7					
8	card_XL	42	20	15	
9	card_L	23.5	-	-	
10					
11	card-attractions-Big_XL	36	36	20	
12	card-attractions-Big_M	38	38	16	
13	card-attractions-Big_S	14	14	12	
14					
15	card-attractions-Small_XL	36	20	12	
16	card-attractions-Small_M	38	16	12	
17	card-attractions-Small_S	14	12	10	
18					
19	card-attraction_XL	36	23	24	
20	card-attraction_M	38	21	-	
21					
22	form_M	20	12	12	
23	form_S	16	10	10	
24					
25	banner_XL	36	-	-	
26	banner_L	20	-	-	

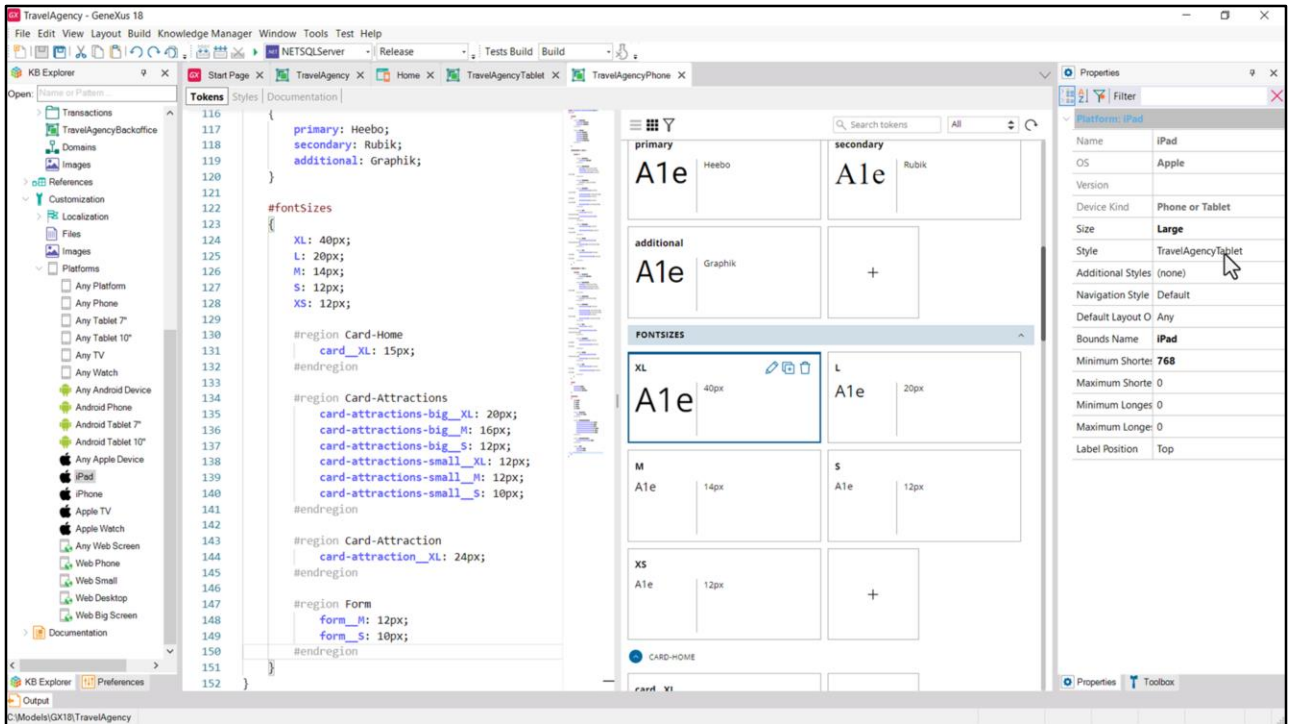
Y en este otro caso coloco los valores de la columna Phone de la planilla.



¿Bien, y dónde indico qué DSO debe utilizar en cada caso?

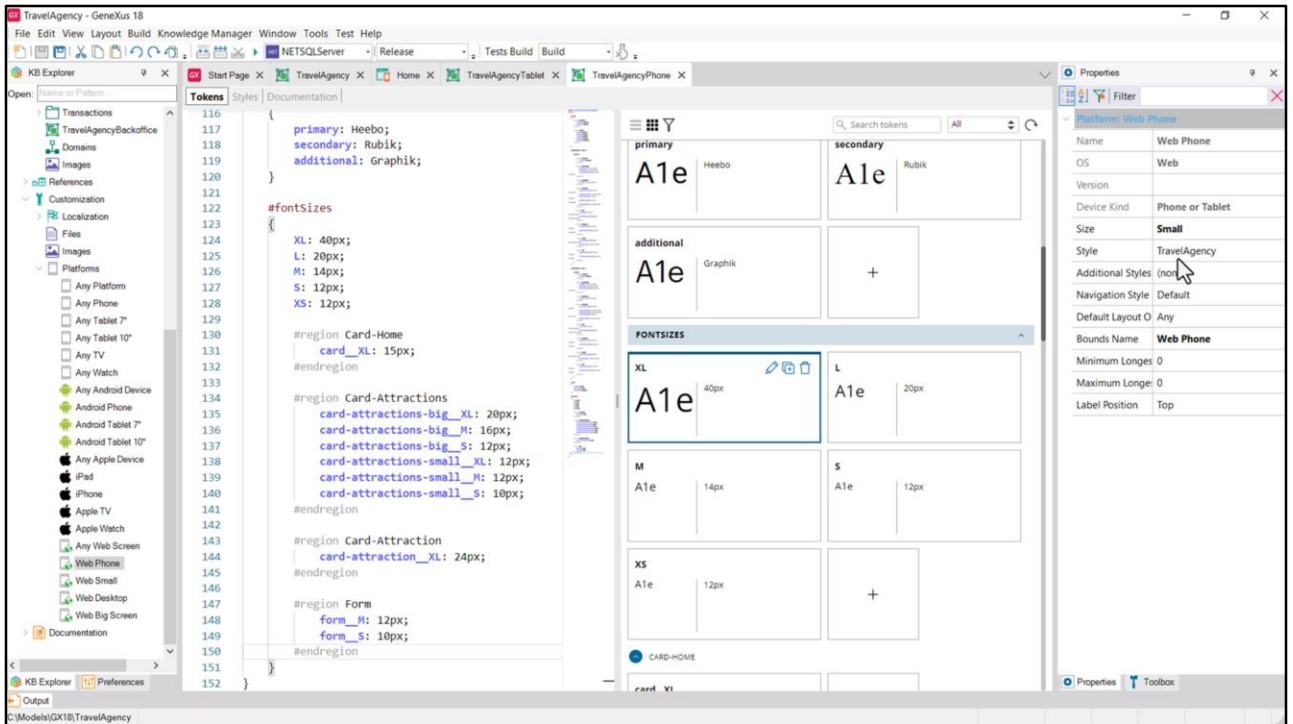
En las plataformas...

Si no voy a discriminar para los tamaños Phone y Tablet entre aplicación web y nativa, entonces puedo especificarlo a nivel del Any... Para Any Platform específico como Style el del DSO TravelAgency, mientras que para Any Phone coloco el TravelAgencyPhone y para Any Tablet de 7 el TravelAgencyTablet, y lo mismo para el de 10.

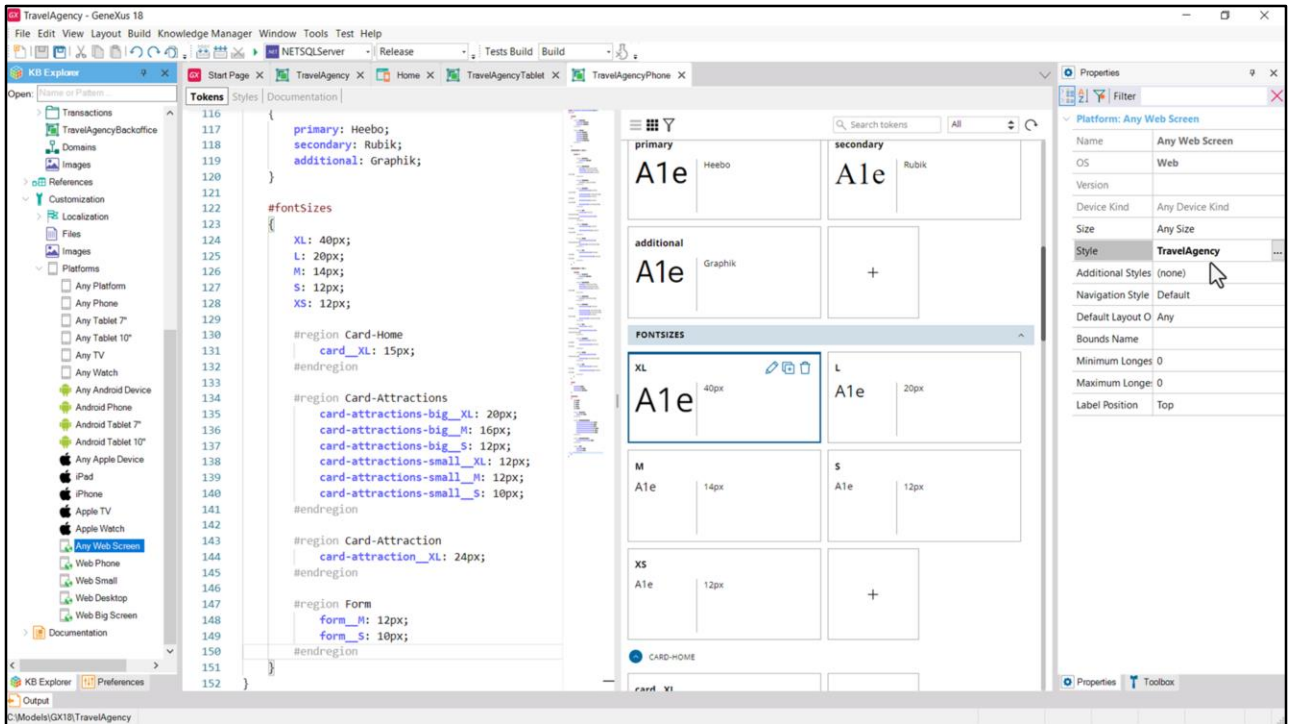


Haber hecho esto a nivel de los Any impacta después en las plataformas específicas. Así, observemos que ahora el valor para Any Android será el mismo que el de Any Platform... y el de Android Phone? El de Any Phone. No nos extrañará ver para Android Tablet 7 y Android Tablet 10 el DSO que indicamos para Any Tablet de 7 y de 10.

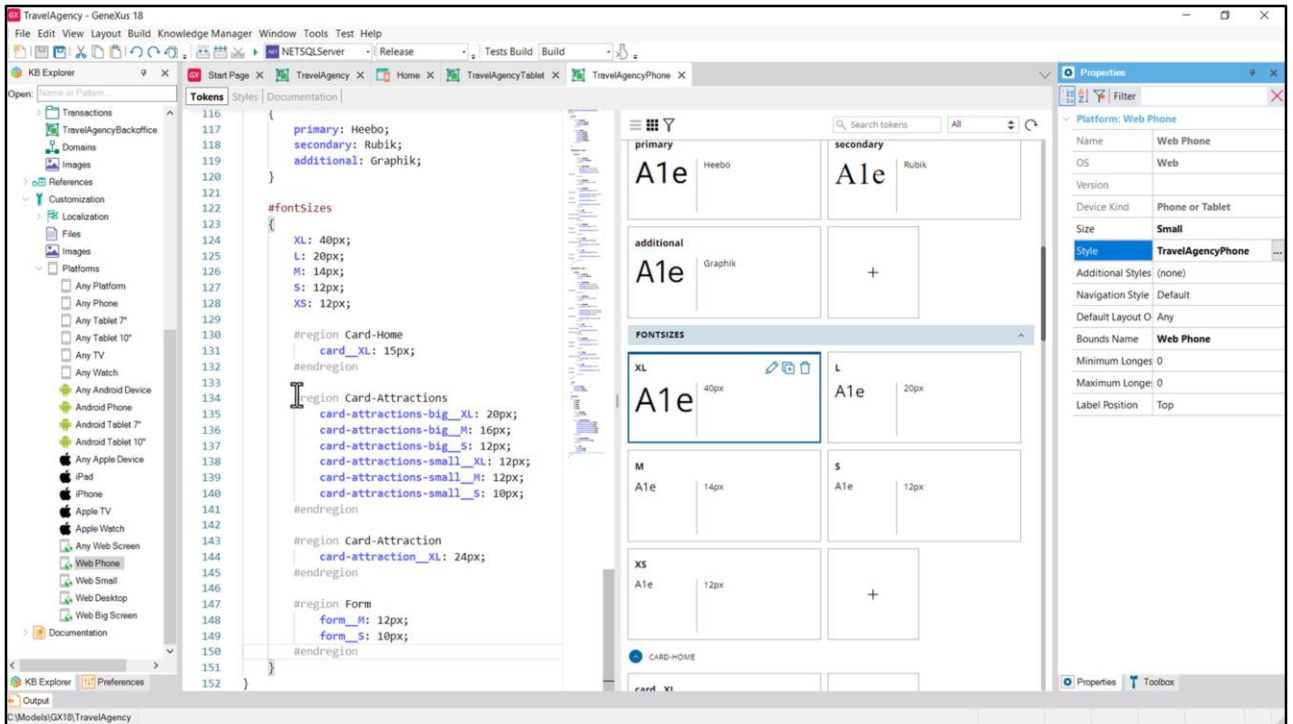
Veamos Any Apple...
iPad...



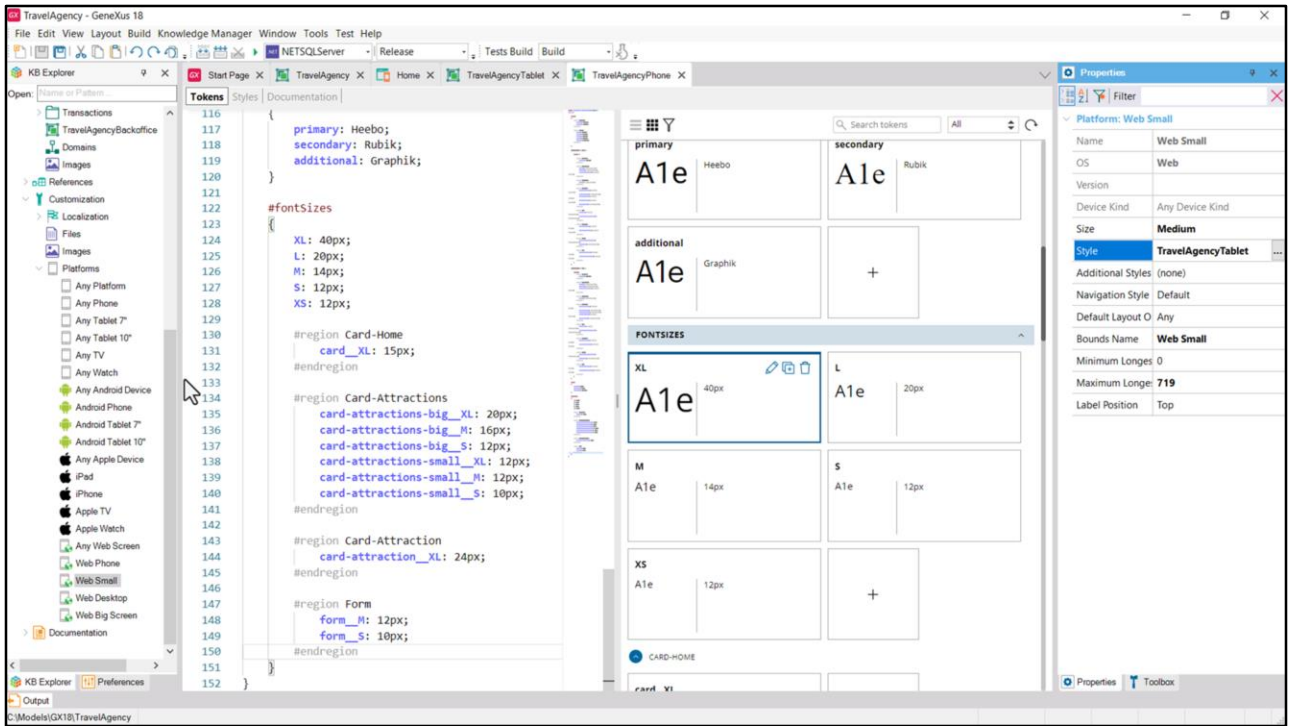
Bueno, y pasemos a lo web... ¿por qué acá no se está respetando? ¿Por qué siempre está TravelAgency? Es que, no sé si se acuerdan, habíamos cambiado el valor default, el que se heredaba de Any Platform, al principio del curso. El default aquel, el de Any Platform era uno de los Unanimos.



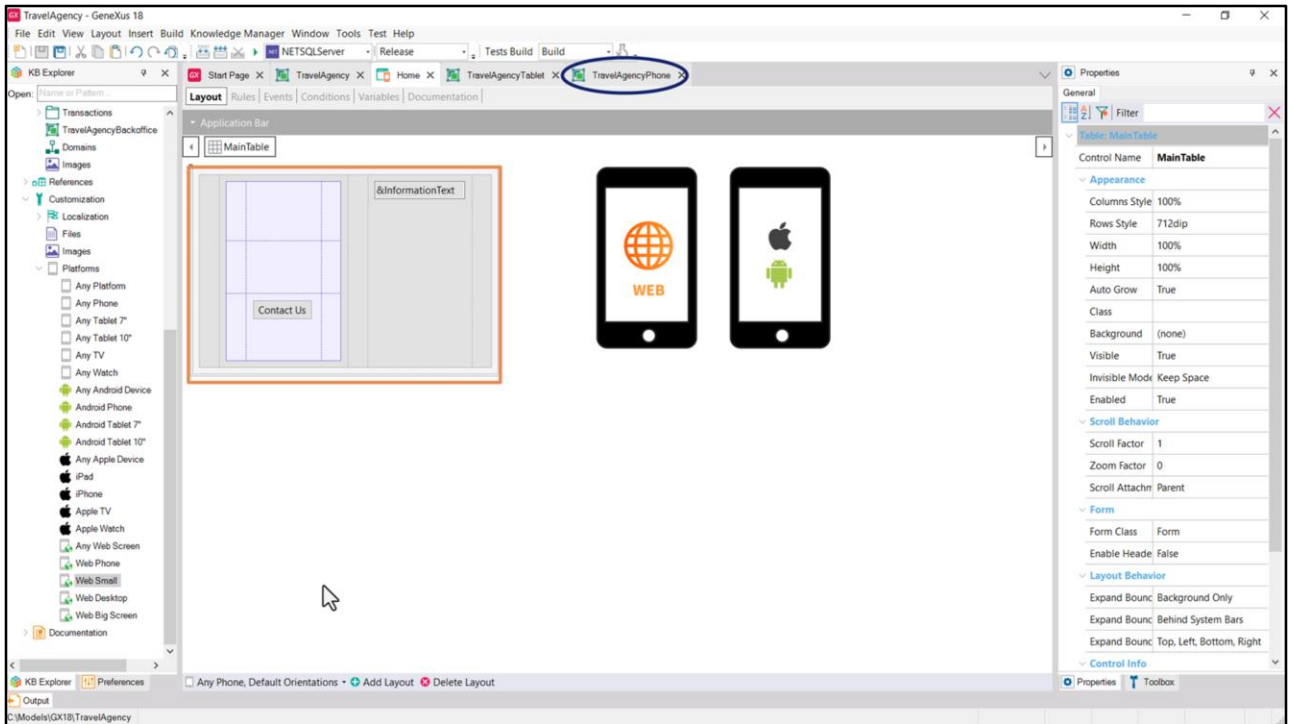
Y nosotros cambiamos aquí (habíamos cambiado) por nuestro TravelAgency, ¡por eso aparece en negrita! Y lo que sucede a partir de allí es que todos estos otros heredan el valor de ese.



Así que tendríamos que cambiar aquí por este...

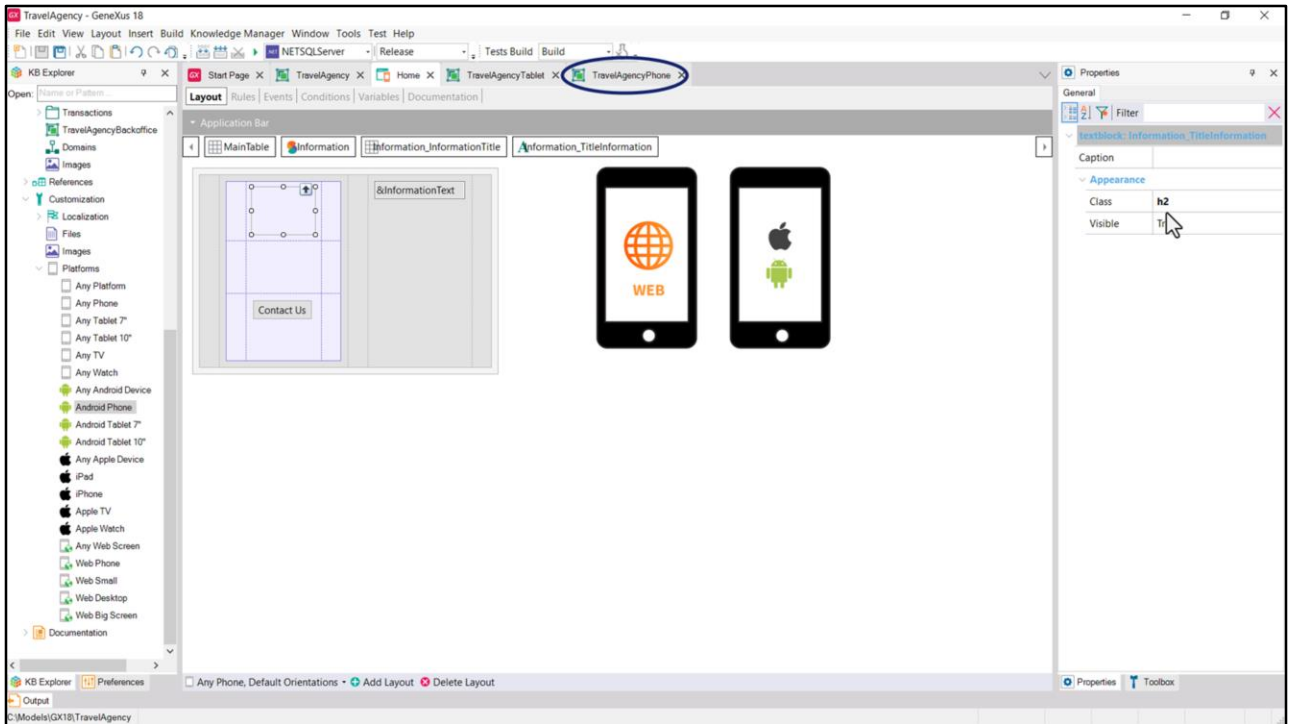


...y aquí por este otro.

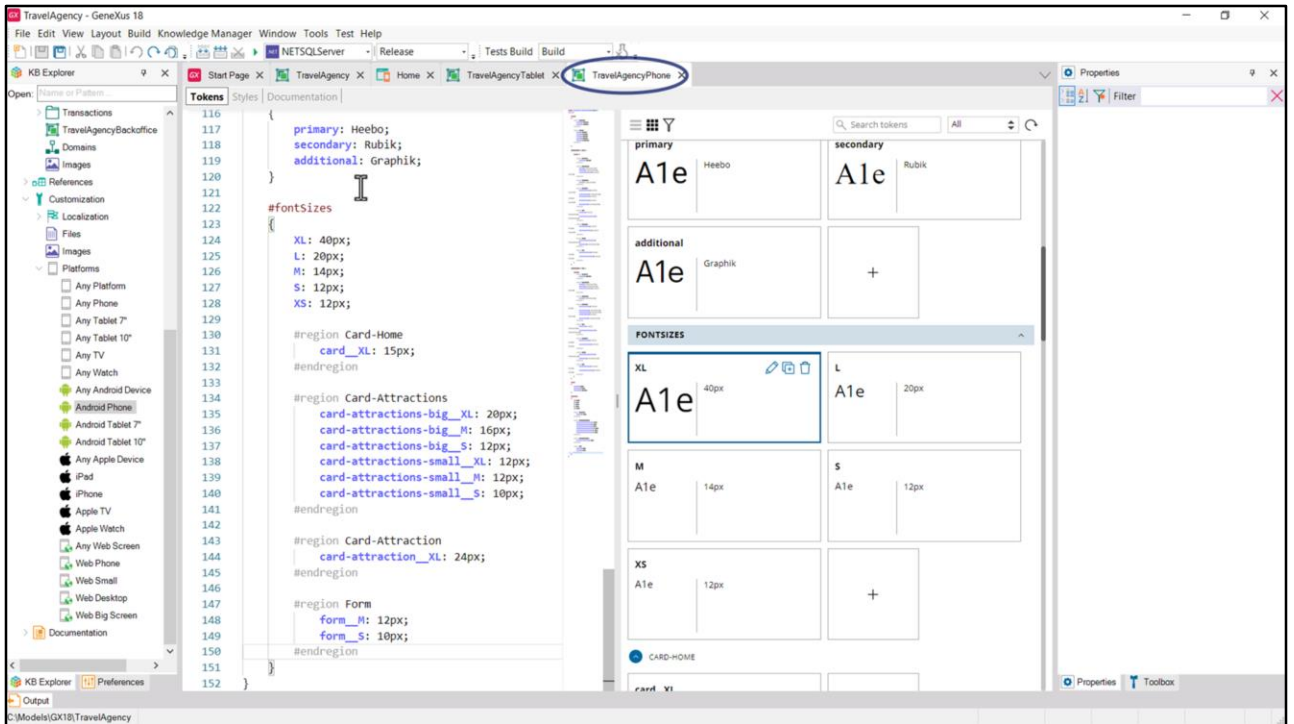


Si pensamos con esta configuración qué pasará cuando ejecutemos nuestro panel Home en un teléfono... el layout que va a elegir, de los dos que hay, va a ser este. No importa si la aplicación es la nativa o es la web, en cualquiera de los dos casos, como no hay opciones más específicas, este será el layout...

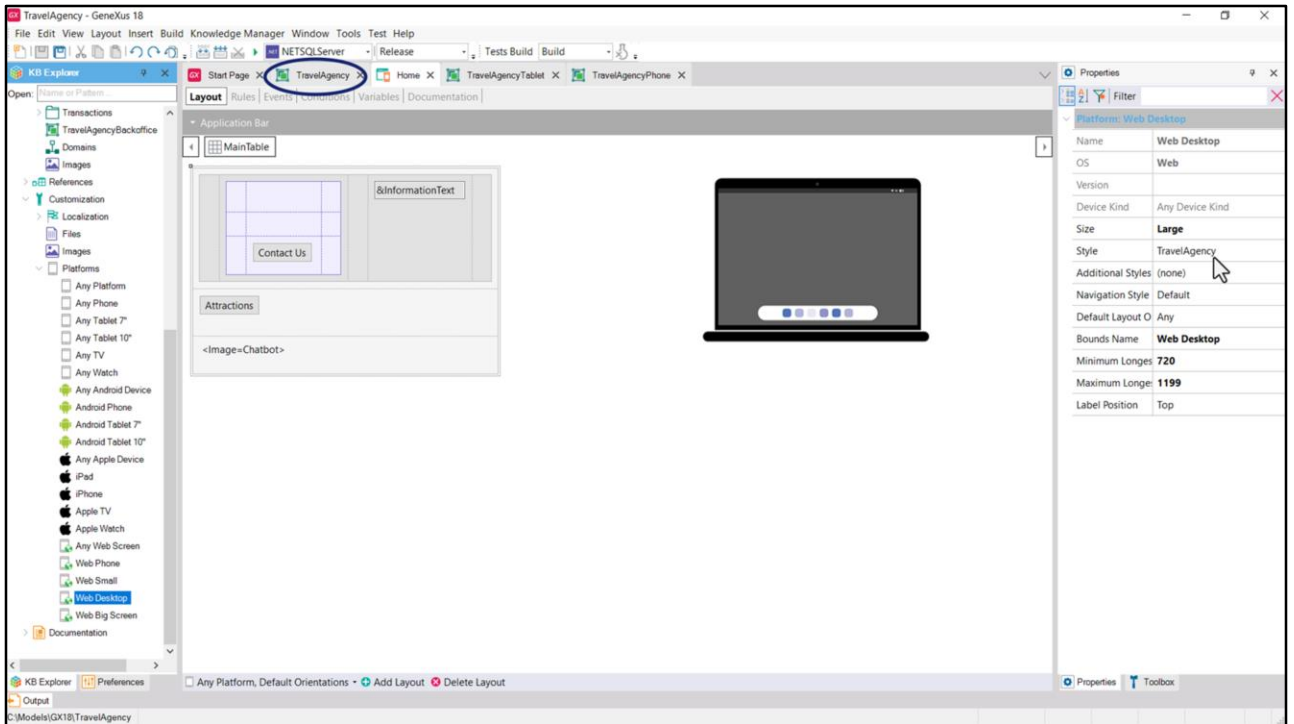
¿Y el DSO que utilizará? Si la aplicación es web, será este...
Si la aplicación es para Android... este, es decir, el mismo...



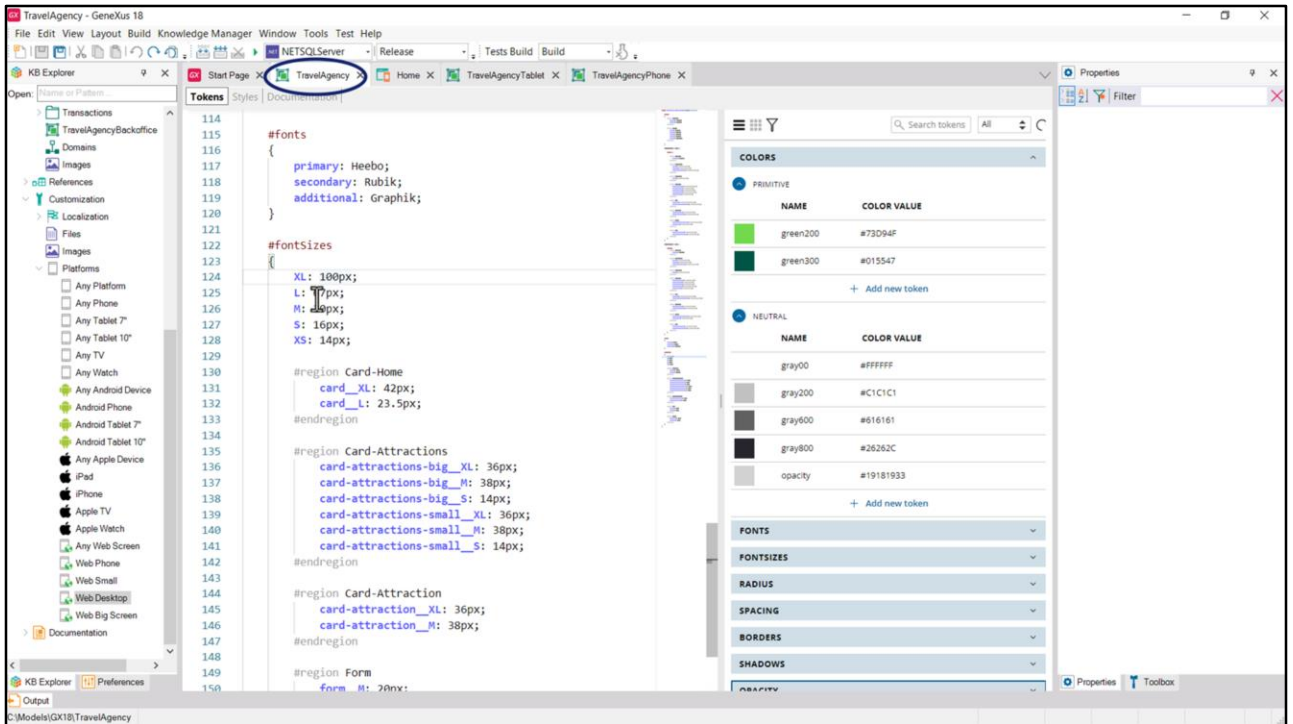
Entonces ¿qué estilo tendrá este texto? El que indique la clase h2... del DSO correspondiente, este...



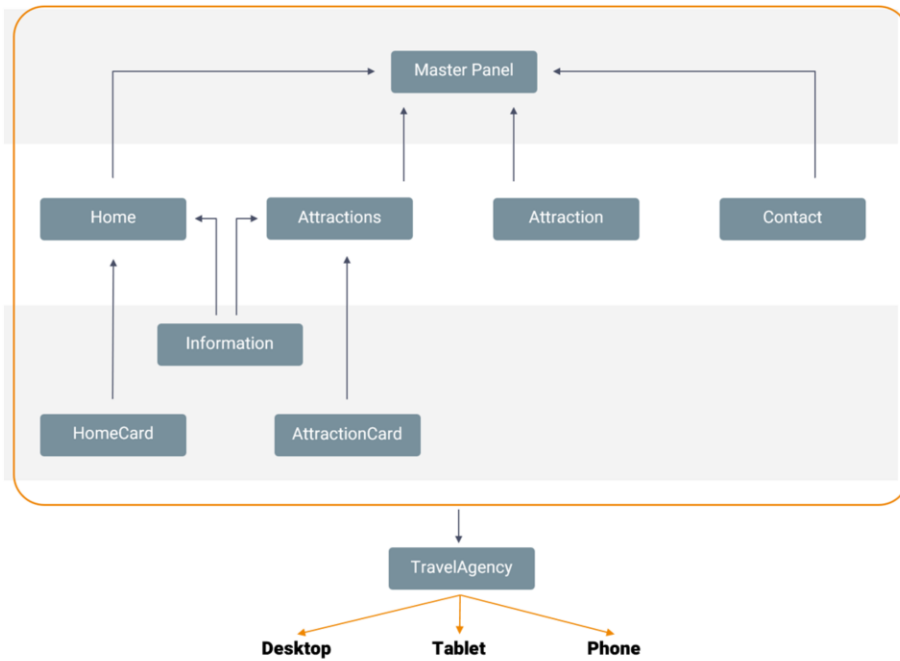
Entonces el tamaño de fuente de ese texto será el que indica este token... que en este DSO vale... 20 píxeles.



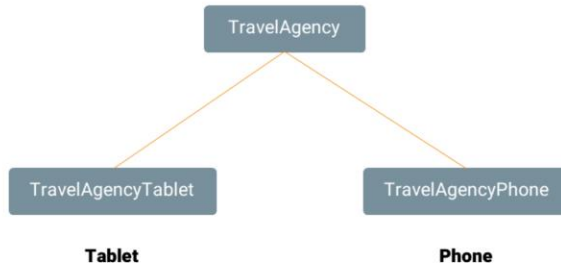
En cambio, si estuviéramos ejecutando en el navegador de una laptop la aplicación web, este será el layout elegido, cuyo estilo corresponderá al de este DSO...



Entonces este texto se estilará de acuerdo a esta clase, que en ese DSO tiene esta definición, idéntica por el momento a la del otro. Lo único que varía es el valor de este token, que aquí será, en cambio, de 67 píxeles.



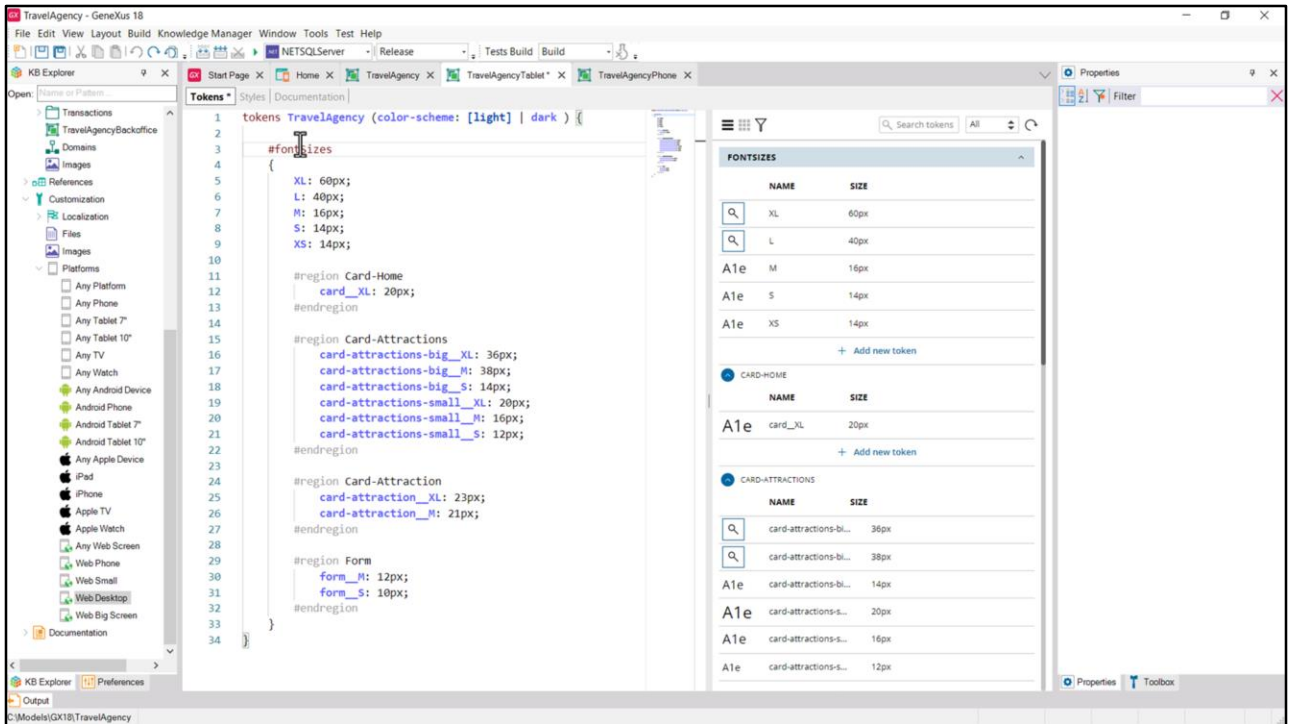
Ustedes ya estarán pensando que si tantas cosas son idénticas entre los 3 DSOs y solo algunas varían...



... ¿no convendría reutilizar lo que es idéntico y especializar solo lo que es diferente? La respuesta a eso siempre es Sí.

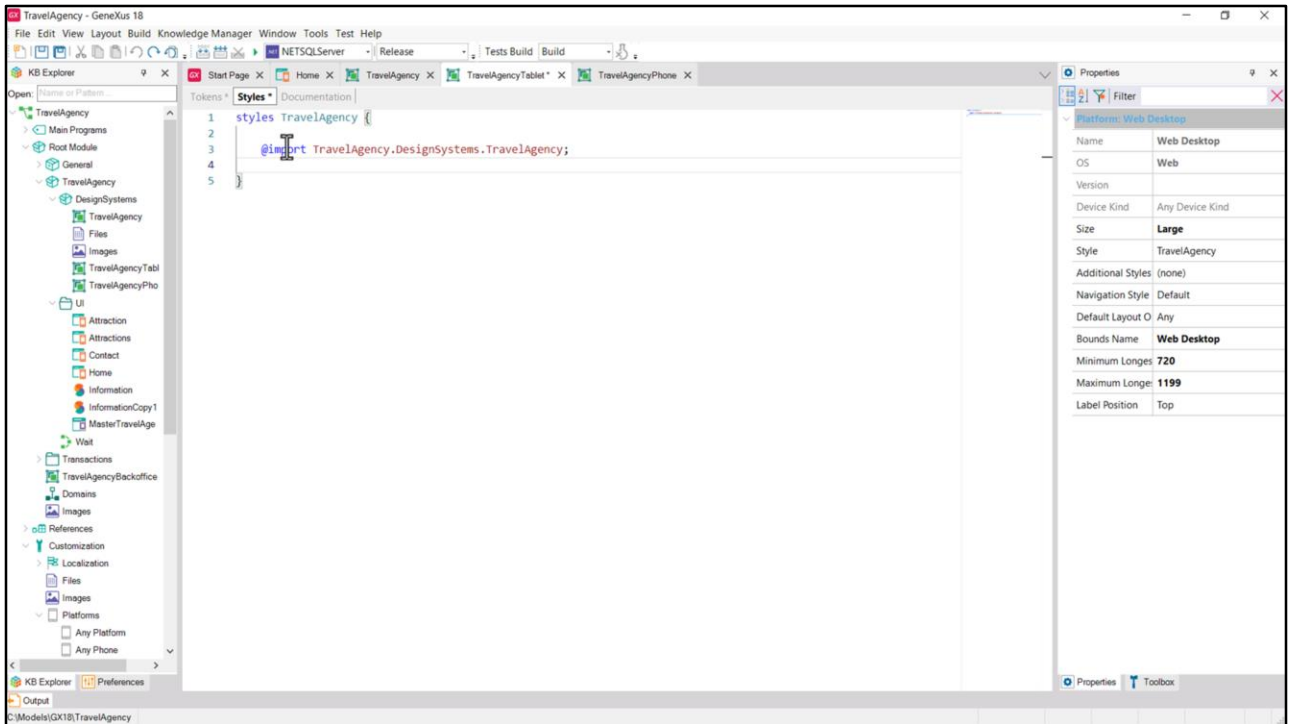
Como pensar a la vez todas las variaciones es un poco enloquecedor, en general uno se dedica a una de las versiones, que en nuestro caso es la Desktop y recién cuando termina trabaja con las otras.

Con el estado actual de las cosas, entonces, yo pensaría que el default será nuestro DSO TravelAgency, que es en el que venimos trabajando y que contiene todas las definiciones, y luego en el DSO de Tablet y de Phone importaría todo lo del DSO default y luego especializaría lo que es distinto.

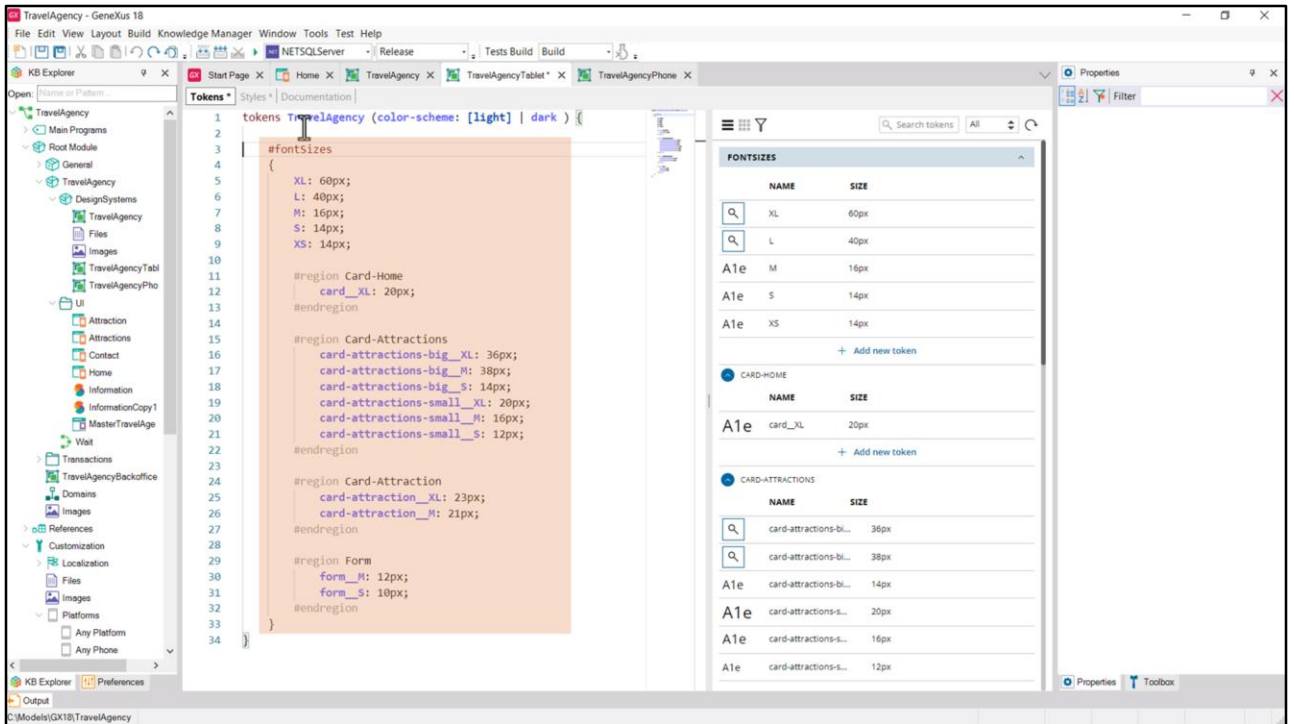


Es decir, haría esto...

Eliminar de aquí todos los tokens que ya están idénticos en TravelAgency...

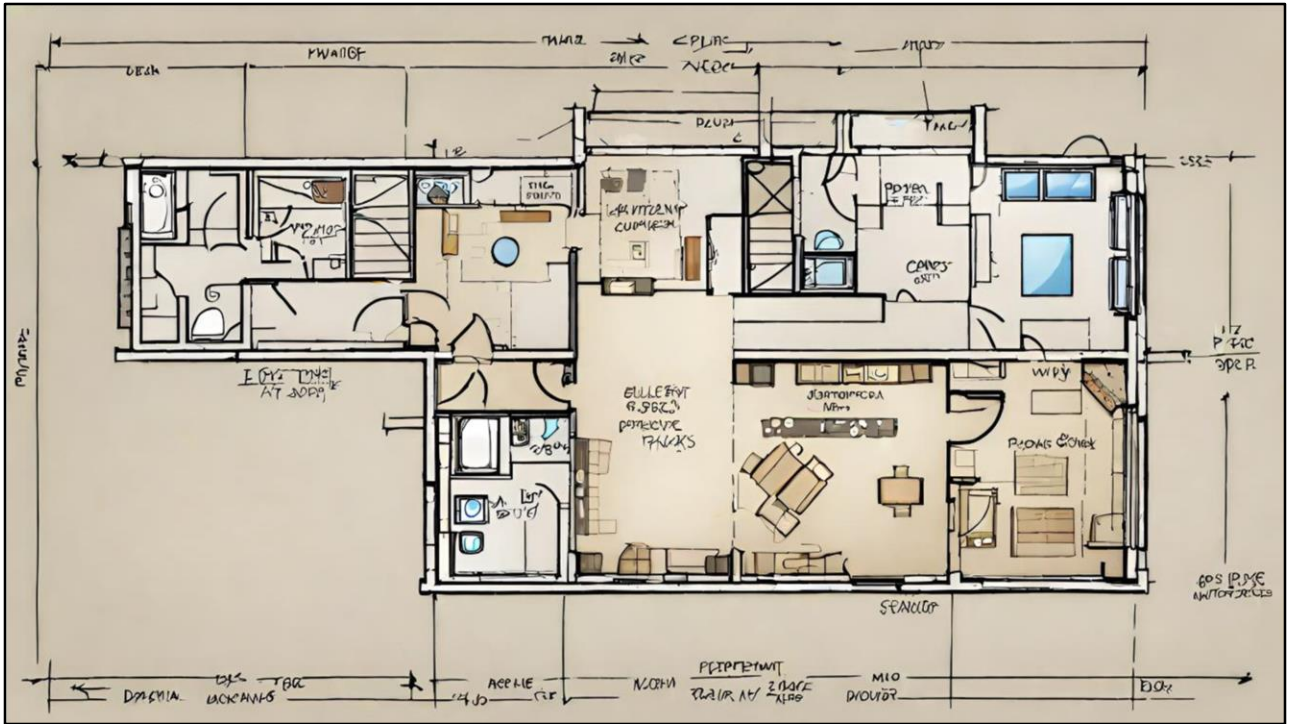


Eliminar todas las definiciones idénticas en la pestaña de Styles, pero importar todo lo del DSO que está en el módulo TravelAgency, submódulo DesignSystems, y se llama TravelAgency (tanto lo de la pestaña de Tokens como de Styles: todo esto estoy importando de TravelAgency).



Se importa todo, y se sobrescriben estos tokens.

Y lo mismo haríamos para el otro DSO.



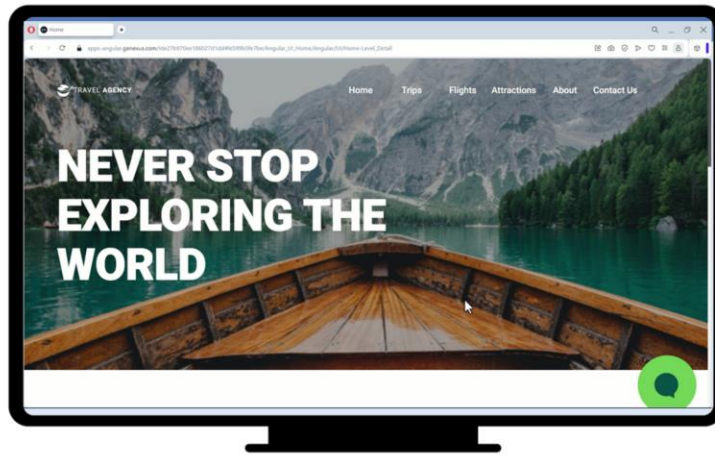
Ustedes a lo mejor se preguntan para qué me metí en este baile si dije que uno arranca por una de las variaciones...



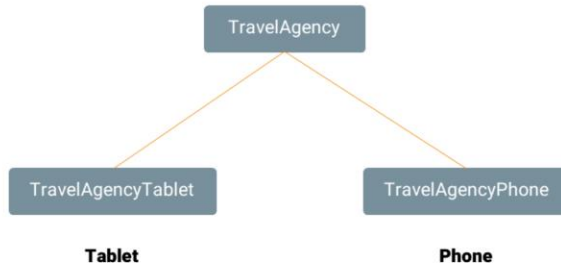
...y recién después desarrolla las otras. Si bien eso es cierto en general...



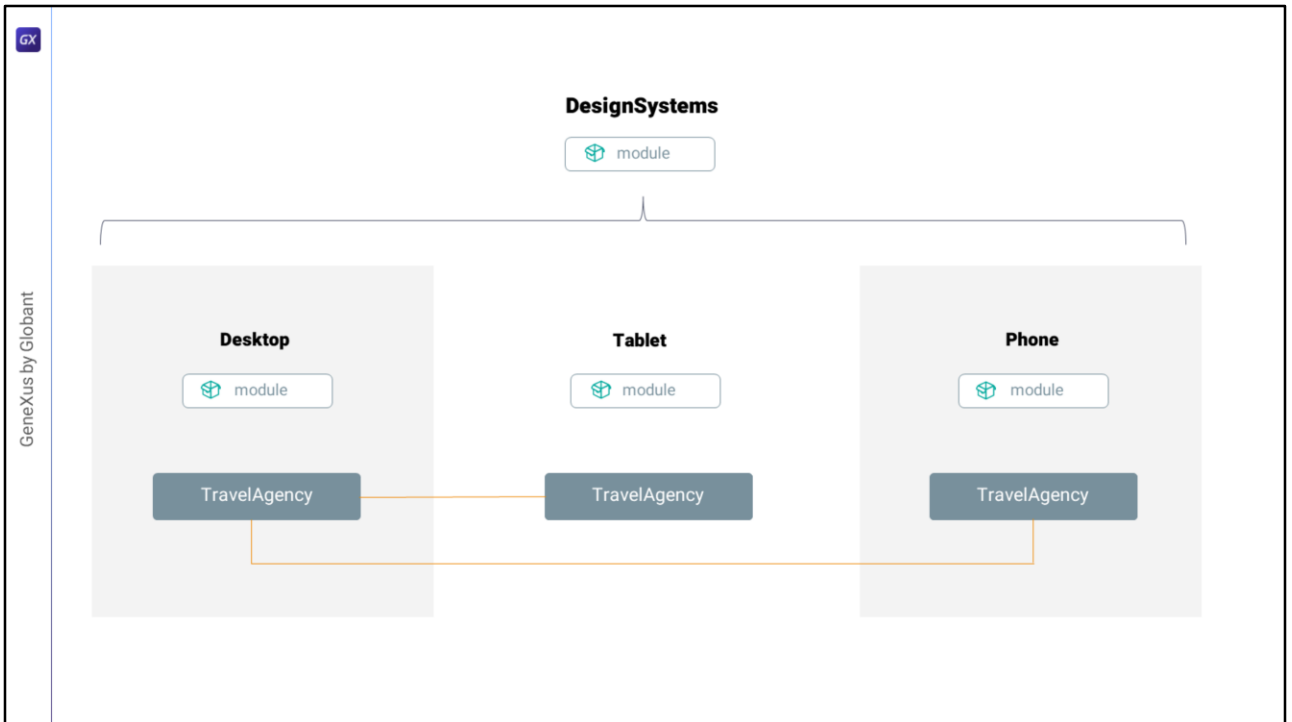
...previsualizar cómo vamos a implementar esas variaciones permite tomar decisiones tempranas que harán más mantenible al sistema luego.



En los próximos videos nos vamos a concentrar en el desarrollo web para Desktop, y recién vamos a volver sobre todo esto al final, cuando encaremos las otras versiones.



De hecho, creo que no será esta organización exactamente la que voy a terminar eligiendo...



...sino que voy a colocar el Design System para cada variación en un módulo distinto. De esa manera les podré llamar igual (los módulos permiten eso, justamente, que haya objetos entre ellos con el mismo nombre), y cada módulo, conceptualmente, implementará lo mismo, el design system de TravelAgency, pero cada uno para un universo diferente. Uno para desktop, otro para Phone, otro para Tablet. Por supuesto uno será el modelo de los otros, es decir, no serán completamente independientes.

Bueno, pero esto no lo vamos a pensar ahora. Lo dejo simplemente esbozado para analizarlo y resolver más adelante, cuando encaremos estas otras variaciones. Seguimos en el próximo video.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com