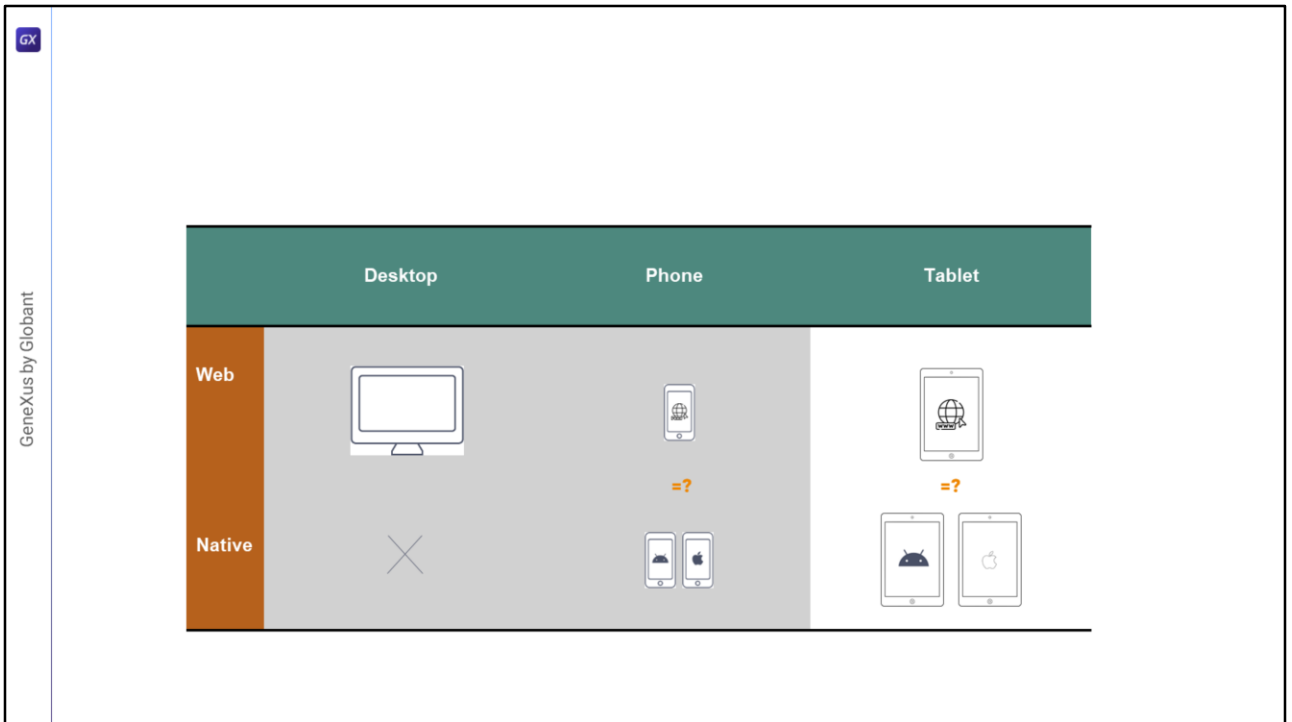


From Travel Agency Web to Native Mobile

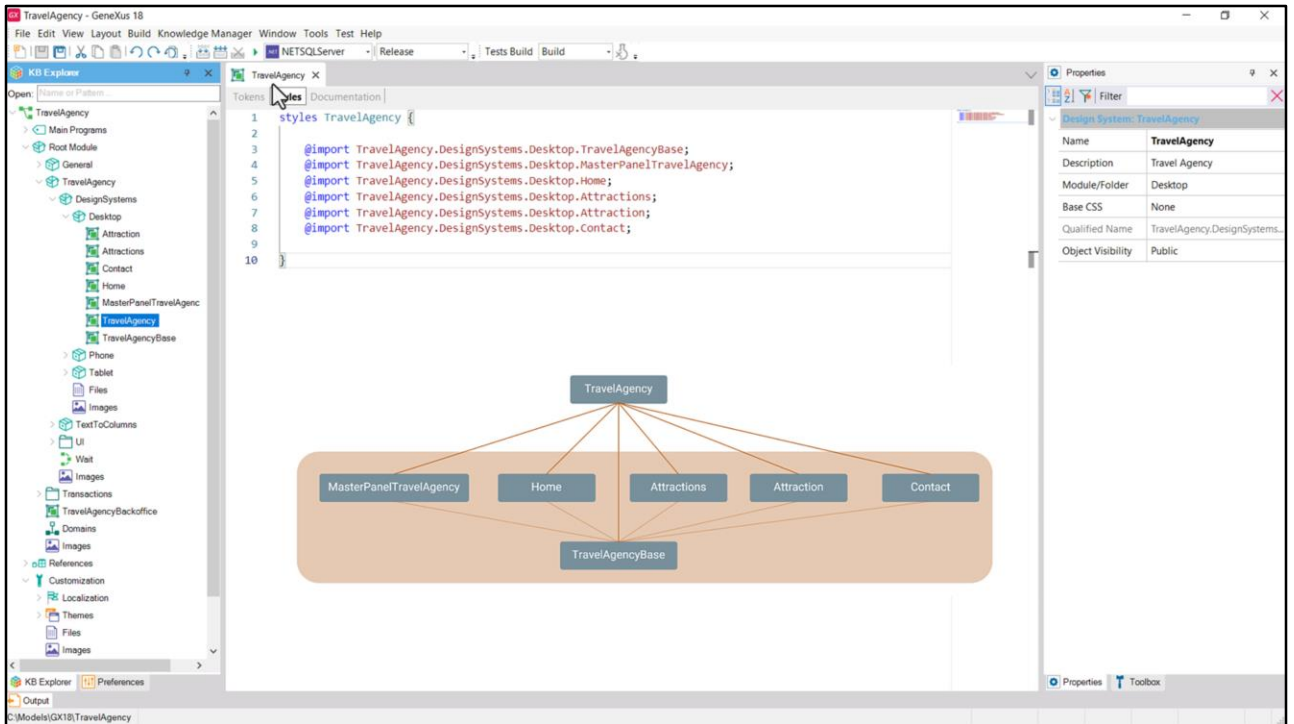


Cecilia Fernández

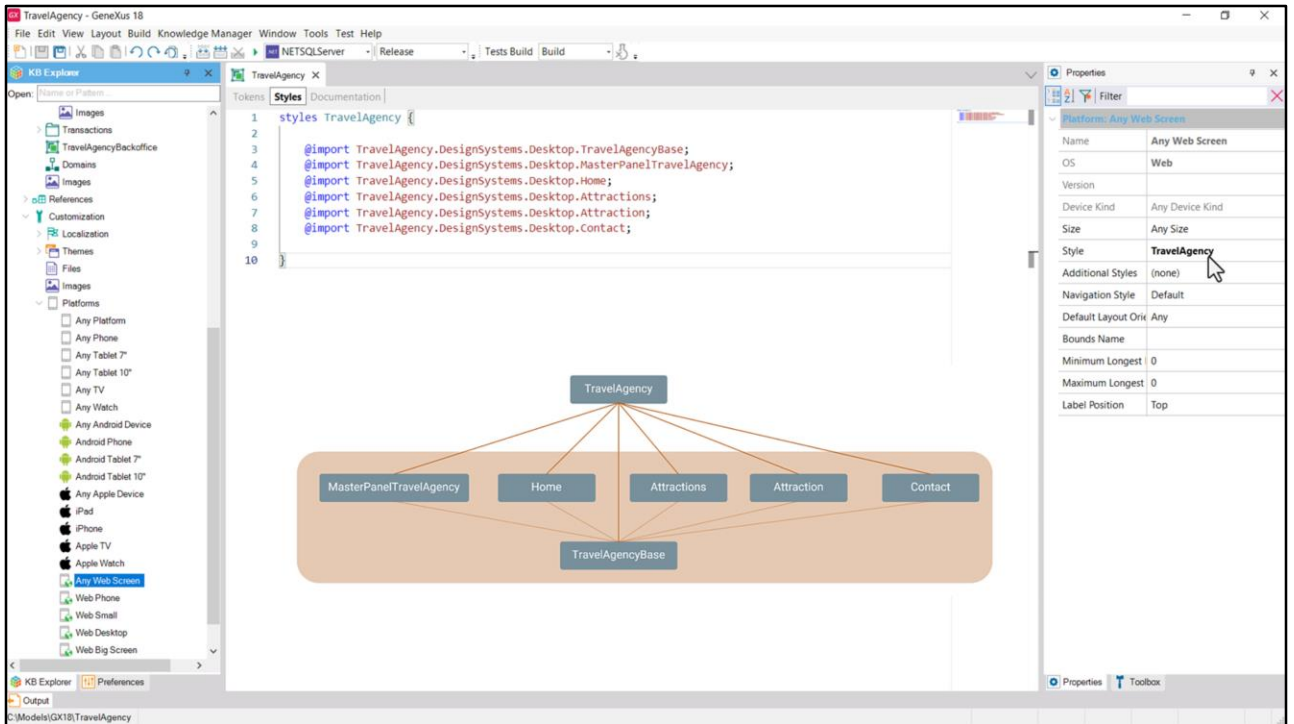


En el video anterior quedamos en la parte donde les quería empezar a mostrar algunas consecuencias del habernos concentrado en la solución Angular sin haber considerado las diferencias respecto a la aplicación nativa.

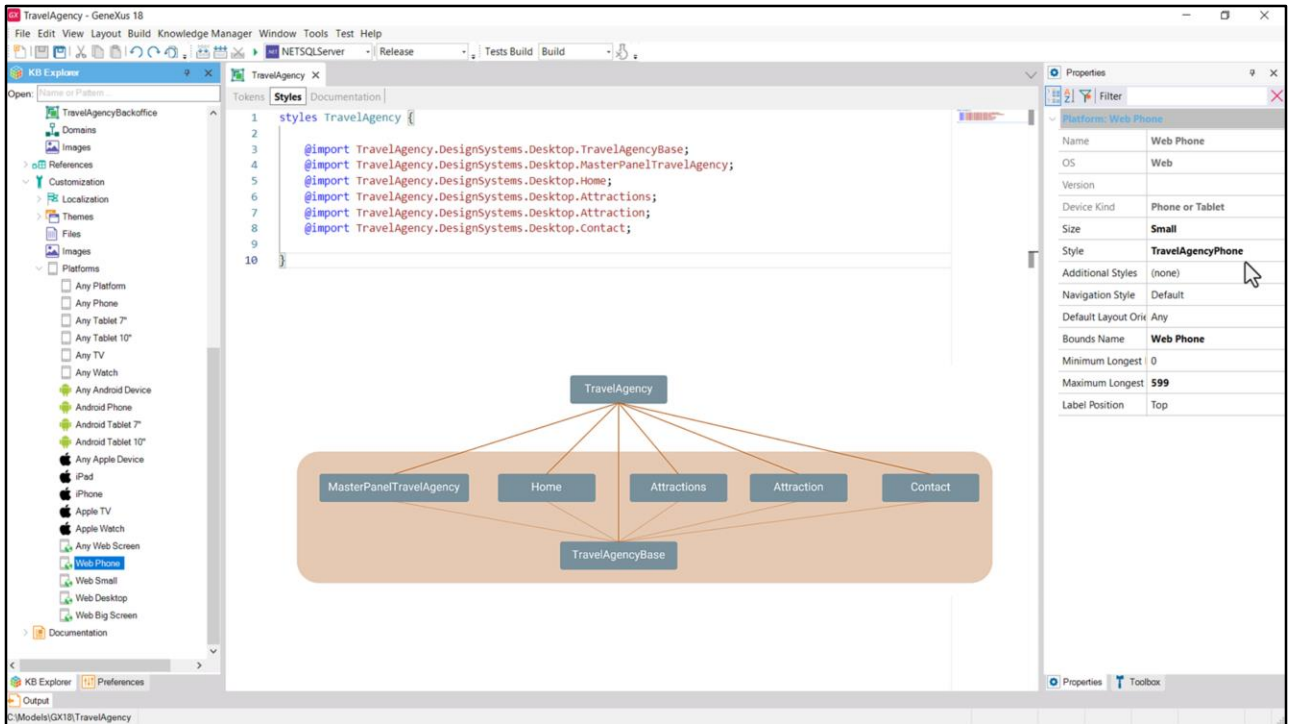
Design System Objects



Recordemos que habíamos definido un árbol de DSOs para implementar el estilo de las pantallas Desktop de la aplicación Angular.



Donde este era el DSO raíz, el DSO que indicábamos para la plataforma, tanto default (para toda plataforma), como default para todas las pantallas Angular. Que se iba a heredar por tanto para Web Desktop y Web Big Screen.



Para estas otras las habíamos personalizado con estos otros DSOs.

TravelAgency - Genexus 18

File Edit View Layout Build Knowledge Manager Window Tools Test Help

NETSQLServer Tests Build Build

KB Explorer TravelAgency X Properties

Open: Name or Pattern

- TravelAgencyBackoffice
 - Domains
 - Images
- References
- Customization
 - Localization
 - Themes
 - Files
 - Images
- 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 Small
 - Web Desktop
 - Web Big Screen
- Documentation

Token: Styles Documentation

```
1 styles TravelAgency {  
2  
3 @import TravelAgency.DesignSystems.Desktop.TravelAgencyBase;  
4 @import TravelAgency.DesignSystems.Desktop.MasterPanelTravelAgency;  
5 @import TravelAgency.DesignSystems.Desktop.Home;  
6 @import TravelAgency.DesignSystems.Desktop.Attractions;  
7 @import TravelAgency.DesignSystems.Desktop.Attraction;  
8 @import TravelAgency.DesignSystems.Desktop.Contact;  
9  
10 }
```

Properties

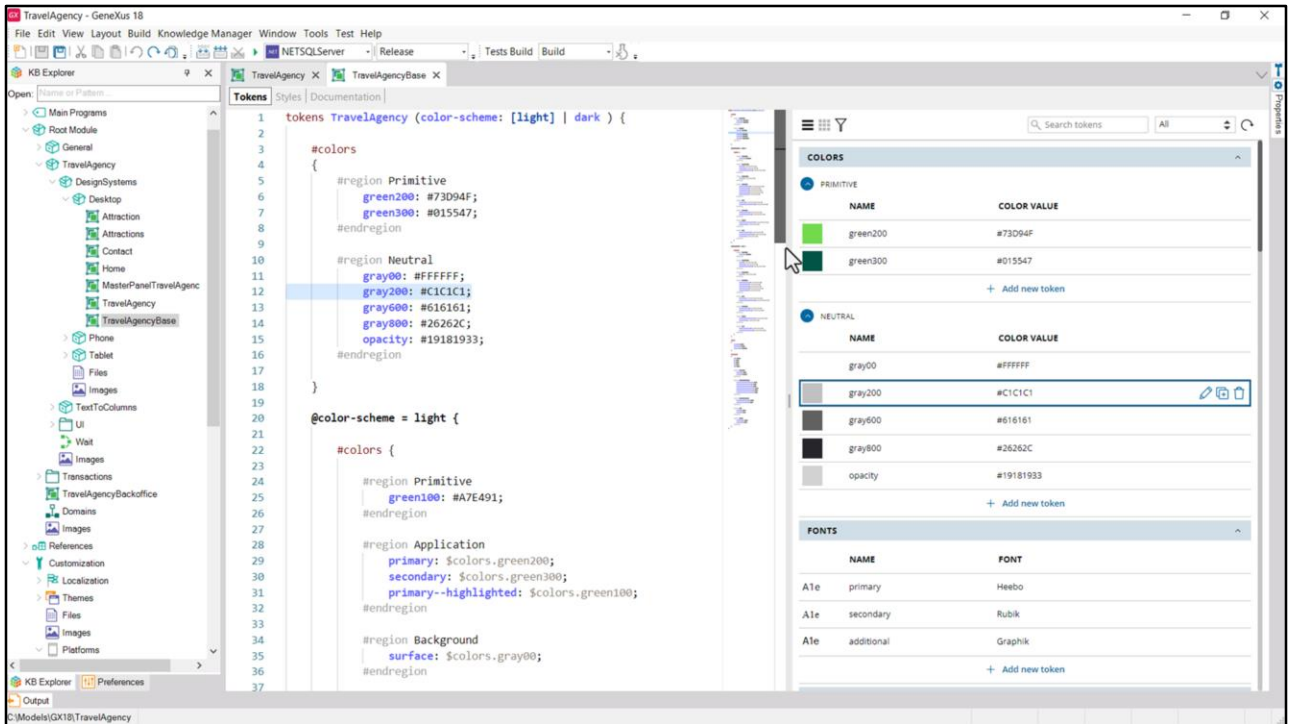
Platform: Web Small

Name	Web Small
OS	Web
Version	
Device Kind	Any Device Kind
Size	Medium
Style	TravelAgencyTablet
Additional Styles	(none)
Navigation Style	Default
Default Layout Ori	Any
Bounds Name	Web Small
Minimum Longest	0
Maximum Longest	719
Label Position	Top

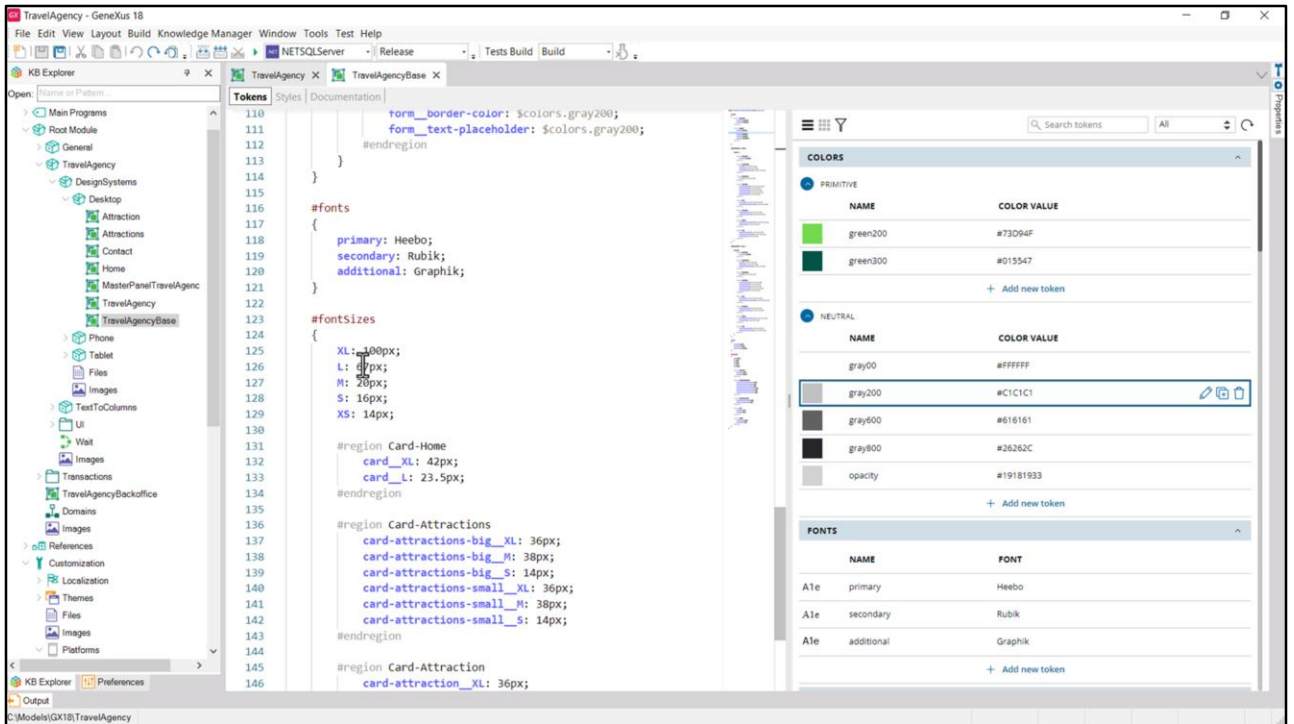
KB Explorer Preferences

Output

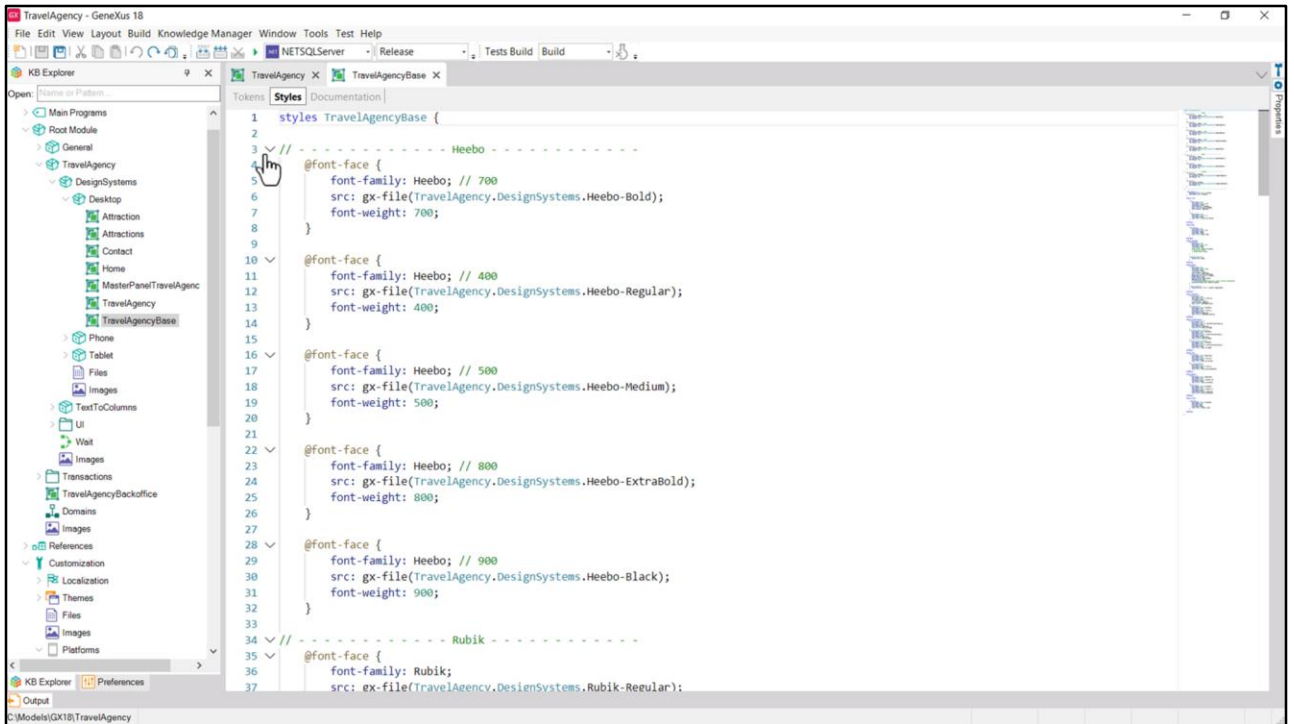
C:\Models\GX18\TravelAgency



Todos estos DSOs se construían sobre el DSO Base, que era en el que habíamos definido todos los tokens generales de la aplicación: tanto los de color (con sus variaciones light y dark)...



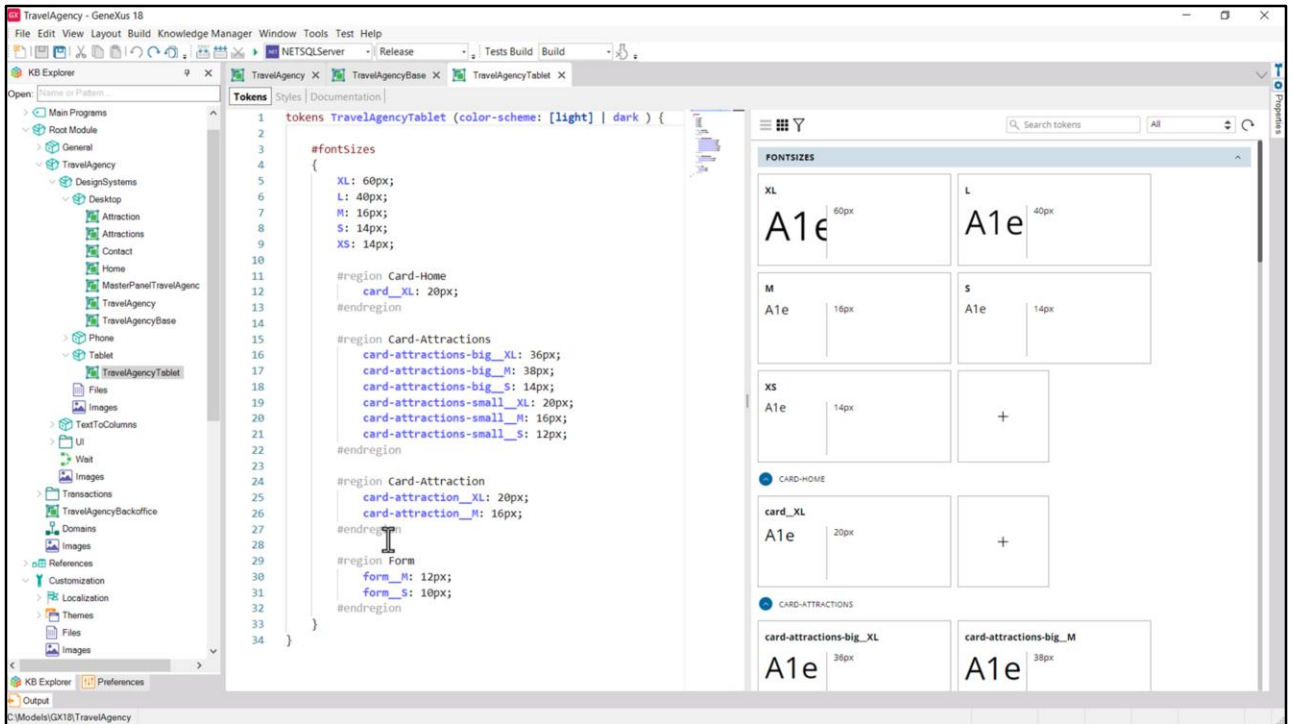
...como los de fuentes, como los de tamaño de fuentes.



Y también habíamos agregado allí las fuentes y las clases tipográficas que habíamos identificado en la etapa de Preparación.

	A	B	C	D	E	F	G	H	I	J	K	L
	Name	Region	Class Name	Font Token	Weight	Font	Style	Size D	Size Tablet	Size Phone	fontSize	Color token
2	H1	Title	h1	primary	900	Heebo	Black	100	60	40	XL	hero_title title_on-image
3	H2		h2	primary	700	Heebo	Bold	67	40	20	L	title_on-surface
4	Paragraph	Paragraph	paragraph	primary	400	Heebo	Regular	16	14	12	S	text_on-surface
5	Button	Button	button	primary	800	Heebo	ExtraBol	14	14	12	XS	text_on-primary
6	Menu Label	Menu	menu_label	primary	500	Heebo	Medium	20	16	14	M	menu_item
7	Copyright	Footer	copyright	secondary	400	Rubik	Regular	20	-	-		footer_text
8	Card Home / H1	Card-Home	card-home_h1	primary	800	Heebo	ExtraBol	42	20	15	card_XL	card-home_title
9	Card Home / H2		card-home_h2	secondary	500	Rubik	Medium	23.5	-	-	card_L	card-home_subtitle
10	Banner / H1	Banner	banner_h1	additional	600	Graphik	Sembok	36	-	-	banner_XL	banner_title text_on-primary
11	Banner / H2		banner_h2	secondary	500	Rubik	Medium	20	-	-	banner_L	banner_text text_on-primary
12	Card Attraction / H1	Card-Attraction	card-attractions_h1	primary	800	Heebo	ExtraBol	36	36	20	card-attractions-Big_XL	card-attraction_title title_on-image
13			card-attractions-small_h1					36	20	12	card-attractions-Small_XL	
14			card-attraction_h1					36	23	24	card-attraction_XL	
15	Card Attraction / Location		card-attractions_location	secondary	400	Rubik	Regular	14	14	12	card-attractions-Big_S	card-attraction_text text_on-image
16			card-attractions-small_location					14	12	10	card-attractions-Small_S	
17	Card Attraction / Rating		card-attractions_rating	secondary	500	Rubik	Medium	38	38	16	card-attractions-Big_M	card-attraction_text text_on-image
18			card-attractions-small_rating					38	16	12	card-attractions-Small_M	
19			card-attraction_rating					38	21	-	card-attraction_M	
20	Form / Regular Text	Form	form_text	additional	400	Graphik	Regular	20	12	12	form_M	form_text text_on-surface
21	Form / Place Holder		form_text-placeholder	primary	400	Heebo	Regular	16	10	10	form_S	form_text-placeholder

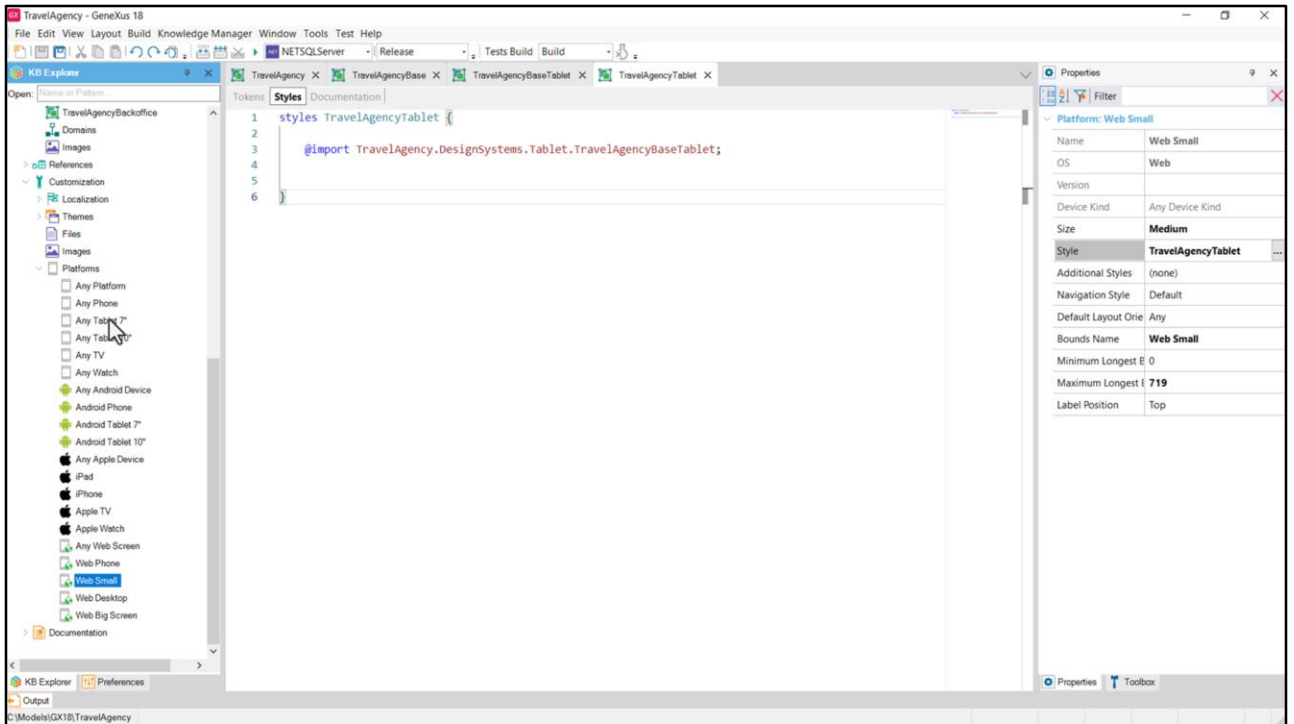
Si recuerdan, en aquella etapa ya habíamos analizado las variaciones que los tokens de tamaños de fuentes iban a sufrir, de acuerdo al tamaño de pantalla, y también habíamos identificado algunas variaciones para las clases tipográficas.



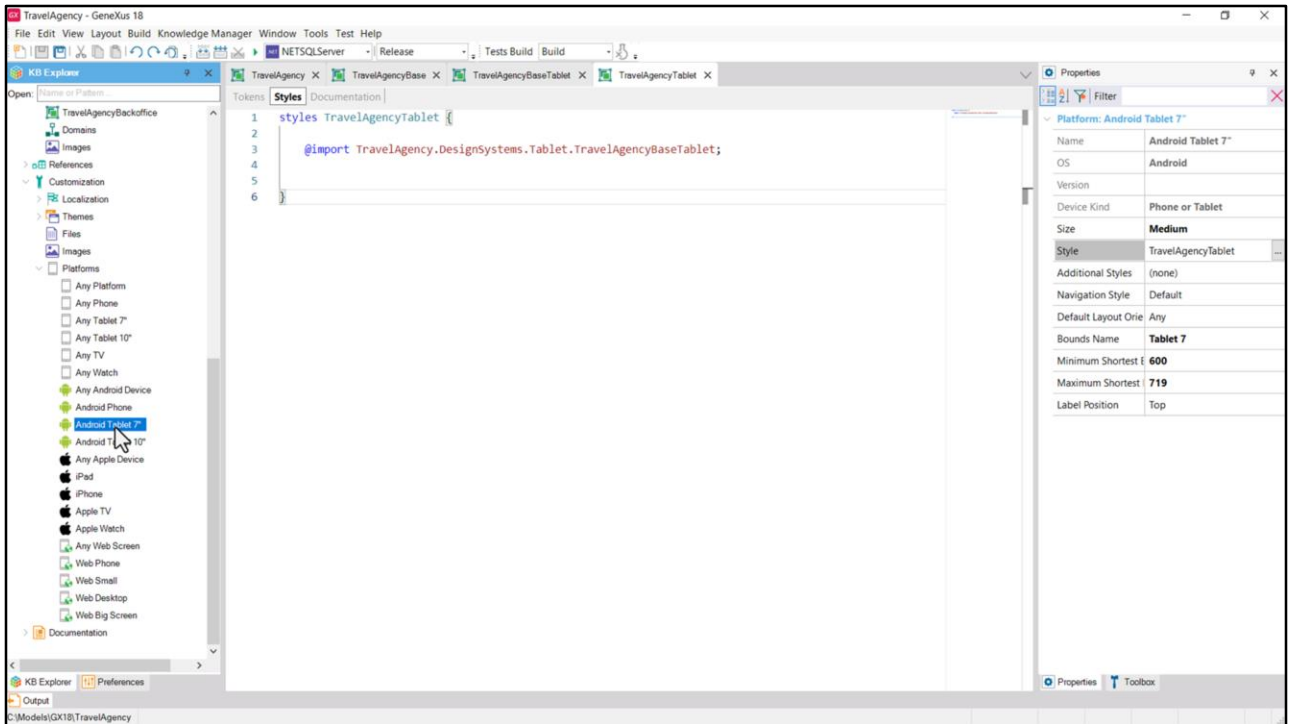
Y todo eso lo habíamos dejado expresado en sendos DSOs paralelos.

Así, para Tablet habíamos creado este DSO TravelAgencyTablet, donde habíamos especializado esas variaciones que habíamos identificado. Deberíamos, en verdad, llamarle análogamente a su paralelo de Desktop: TravelAgencyBaseTablet.

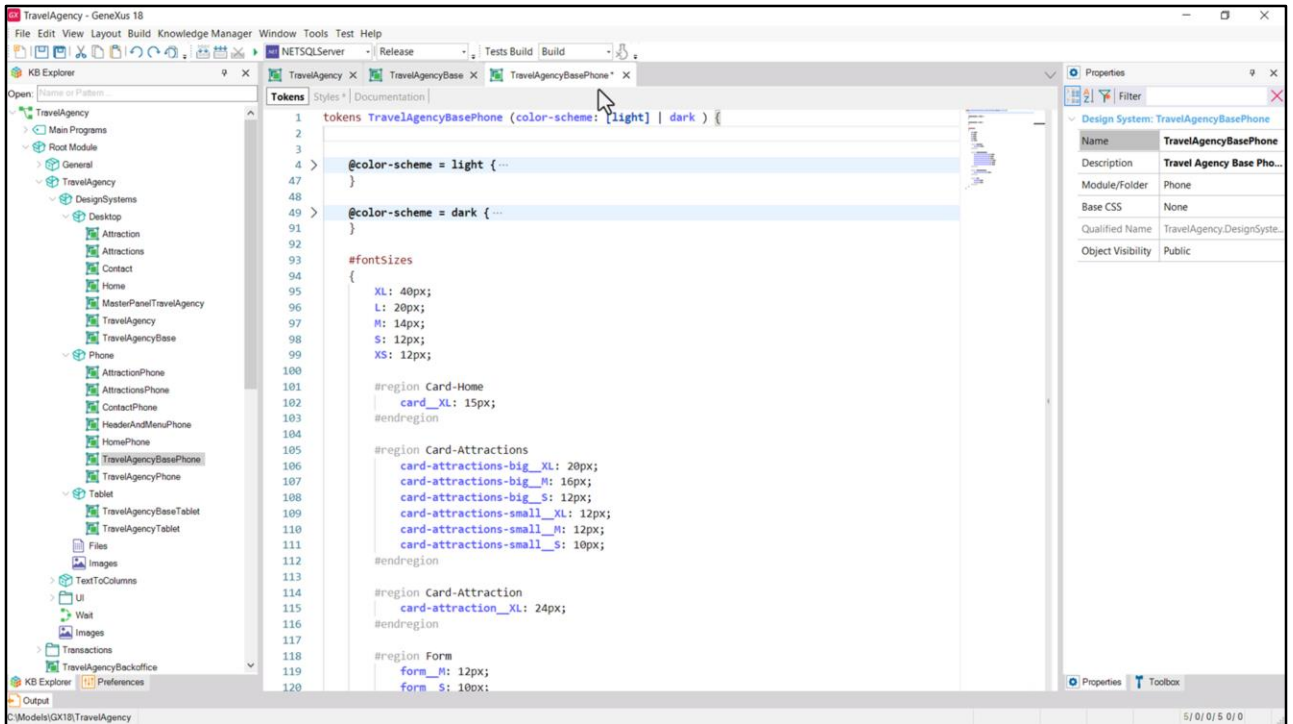
Y este debería importar al TravelAgencyBase, porque lo que hace es especializarlo. Importándolo, lo que hace es cambiarle los valores a los tokens de fontSizes...Y además agrega clases que no se necesitaban para Desktop pero sí para este otro tamaño de pantalla.



Por supuesto, necesitaremos tener el DSO raíz, TravelAgencyTablet, que es el que indicamos aquí para plataforma Web de ese tamaño.

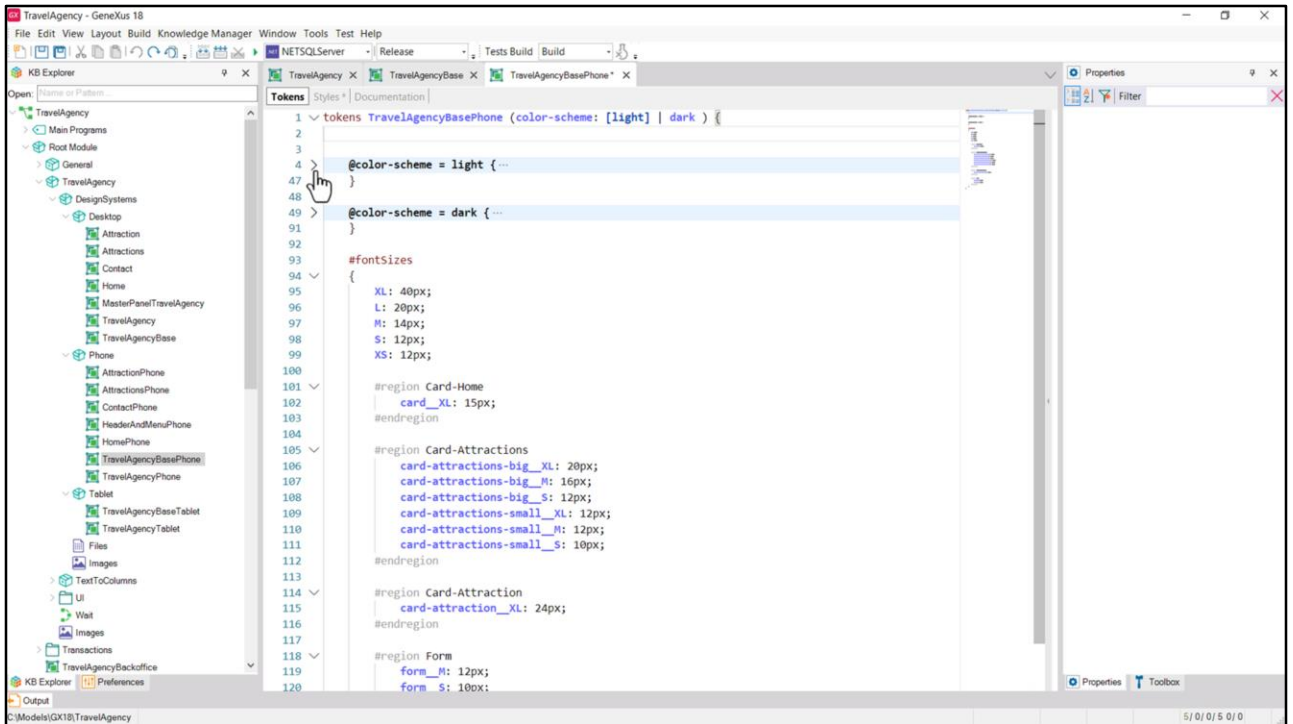


Y si queremos que el mismo DSO funcione también para la aplicación nativa en este tamaño, lo podemos colocar aquí también.

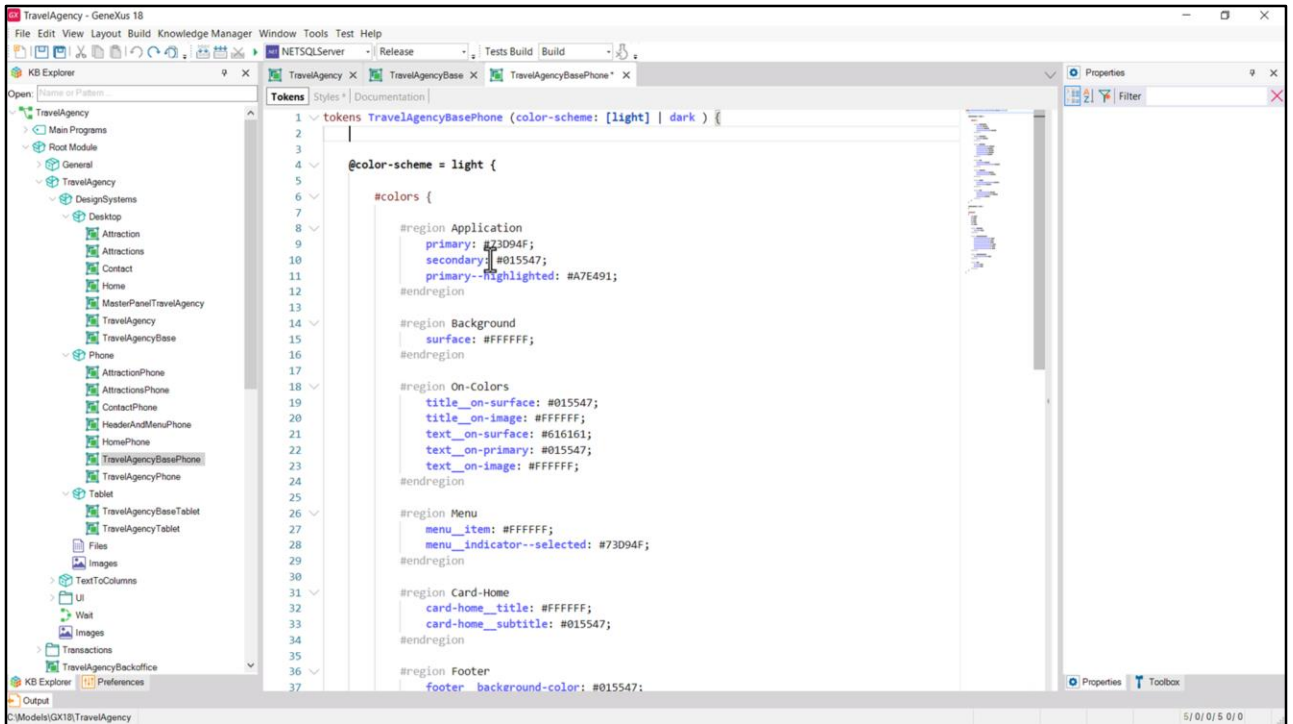


Y lo mismo valdrá para tamaño Phone. Aquí ya creé la estructura paralela.

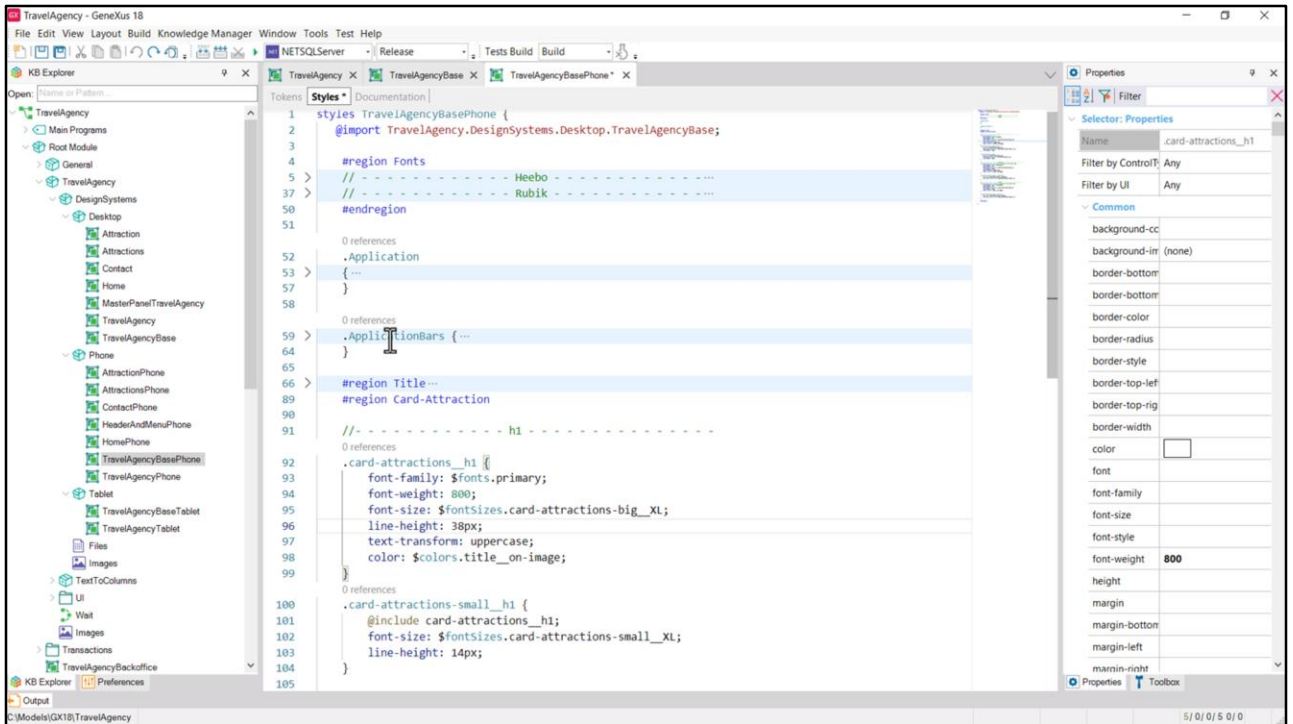
Este era el DSO que habíamos creado en la etapa de preparación (lo renombré), al que tuve que modificarle algunas cosas que necesito que vean para mostrarles justamente el punto.



Dado que importa al análogo de Desktop, aquí solamente deberían estar los tokens que cambian y no los que conservan los mismos valores, como los de color. ¿Por qué entonces aparecen todos estos de color, en lugar de sólo aparecer los de tamaños de fuentes?



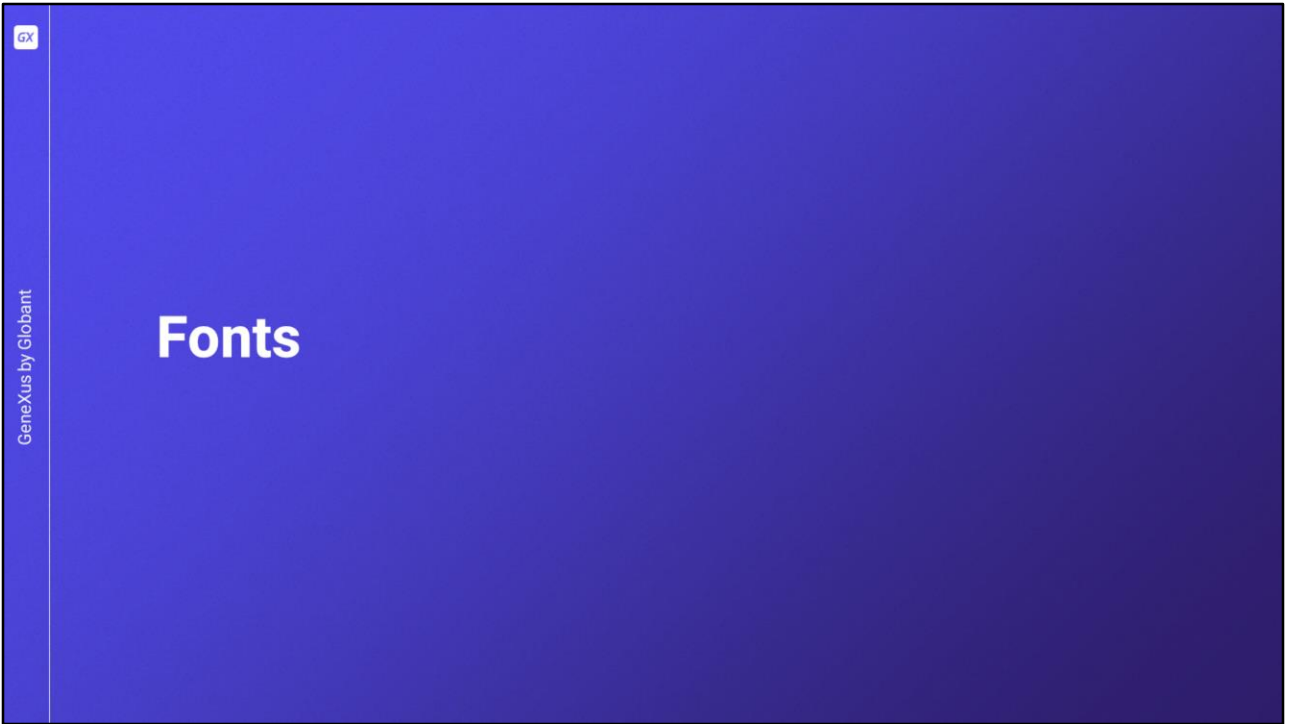
A esto no le den importancia. Tuve que copiar provisoriamente todos los tokens del de Desktop cuyo valor utilizaba otro token, porque no está funcionando esa indirectión para la aplicación nativa. Tuve que asignarles sus valores absolutos. Esto será arreglado, por eso les digo que no se preocupen. Toda esta sección no estará cuando se arregle ese bug.



De lo que sí debemos preocuparnos es de cómo definimos las fuentes.

Antes: todas estas clases son las que habíamos identificado en la etapa de preparación.

Las clases Application y ApplicationBars tendrán una semántica especial para las aplicaciones nativas, como veremos luego. Ahora no las mostraré.



Atendamos ahora a las fuentes...

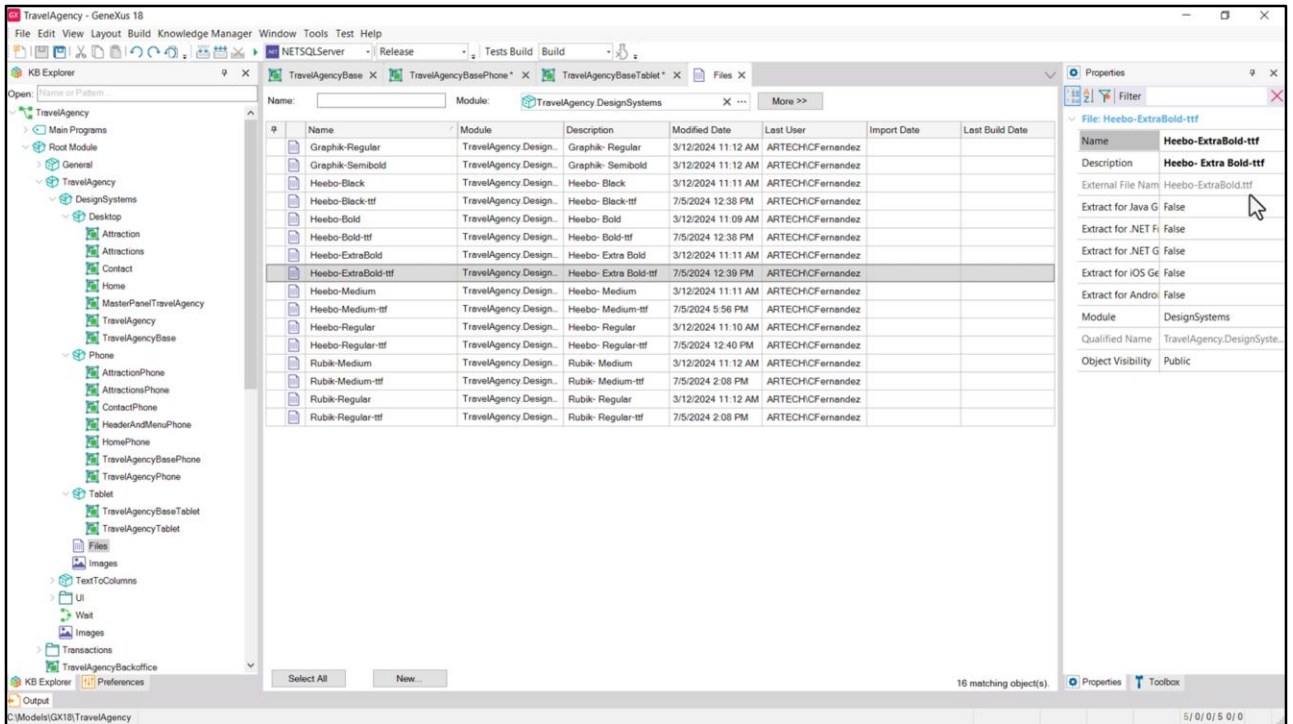
The screenshot shows the GeneXus 18 interface for the 'TravelAgency' project. The main window displays a table of font objects. The table has columns for Name, Module, Description, Modified Date, Last User, Import Date, and Last Build Date. The objects listed include various Heebo and Rubik fonts in different weights and formats (e.g., Regular, Bold, Medium, Extra Bold, ttf, woff2).

#	Name	Module	Description	Modified Date	Last User	Import Date	Last Build Date
	Graphik-Regular	TravelAgency.Design...	Graphik- Regular	3/12/2024 11:12 AM	ARTECHCFernandez		
	Graphik-Semibold	TravelAgency.Design...	Graphik- Semibold	3/12/2024 11:12 AM	ARTECHCFernandez		
	Heebo-Black	TravelAgency.Design...	Heebo- Black	3/12/2024 11:11 AM	ARTECHCFernandez		
	Heebo-Black-ttf	TravelAgency.Design...	Heebo- Black-ttf	7/5/2024 12:38 PM	ARTECHCFernandez		
	Heebo-Bold	TravelAgency.Design...	Heebo- Bold	3/12/2024 11:09 AM	ARTECHCFernandez		
	Heebo-Bold-ttf	TravelAgency.Design...	Heebo- Bold-ttf	7/5/2024 12:38 PM	ARTECHCFernandez		
	Heebo-ExtraBold	TravelAgency.Design...	Heebo- Extra Bold	3/12/2024 11:11 AM	ARTECHCFernandez		
	Heebo-ExtraBold-ttf	TravelAgency.Design...	Heebo- Extra Bold-ttf	7/5/2024 12:39 PM	ARTECHCFernandez		
	Heebo-Medium	TravelAgency.Design...	Heebo- Medium	3/12/2024 11:11 AM	ARTECHCFernandez		
	Heebo-Medium-ttf	TravelAgency.Design...	Heebo- Medium-ttf	7/5/2024 5:56 PM	ARTECHCFernandez		
	Heebo-Regular	TravelAgency.Design...	Heebo- Regular	3/12/2024 11:10 AM	ARTECHCFernandez		
	Heebo-Regular-ttf	TravelAgency.Design...	Heebo- Regular-ttf	7/5/2024 12:40 PM	ARTECHCFernandez		
	Rubik-Medium	TravelAgency.Design...	Rubik- Medium	3/12/2024 11:12 AM	ARTECHCFernandez		
	Rubik-Medium-ttf	TravelAgency.Design...	Rubik- Medium-ttf	7/5/2024 2:08 PM	ARTECHCFernandez		
	Rubik-Regular	TravelAgency.Design...	Rubik- Regular	3/12/2024 11:12 AM	ARTECHCFernandez		
	Rubik-Regular-ttf	TravelAgency.Design...	Rubik- Regular-ttf	7/5/2024 2:08 PM	ARTECHCFernandez		

The Properties panel on the right shows details for the selected 'Heebo-Black' object:

- Name: Heebo-Black
- Description: Heebo- Black
- External File Name: Heebo-Black.woff2
- Extract for Java G: False
- Extract for .NET F: False
- Extract for .NET G: False
- Extract for iOS Ge: False
- Extract for Android: False
- Module: DesignSystems
- Qualified Name: TravelAgency.DesignSystems
- Object Visibility: Public

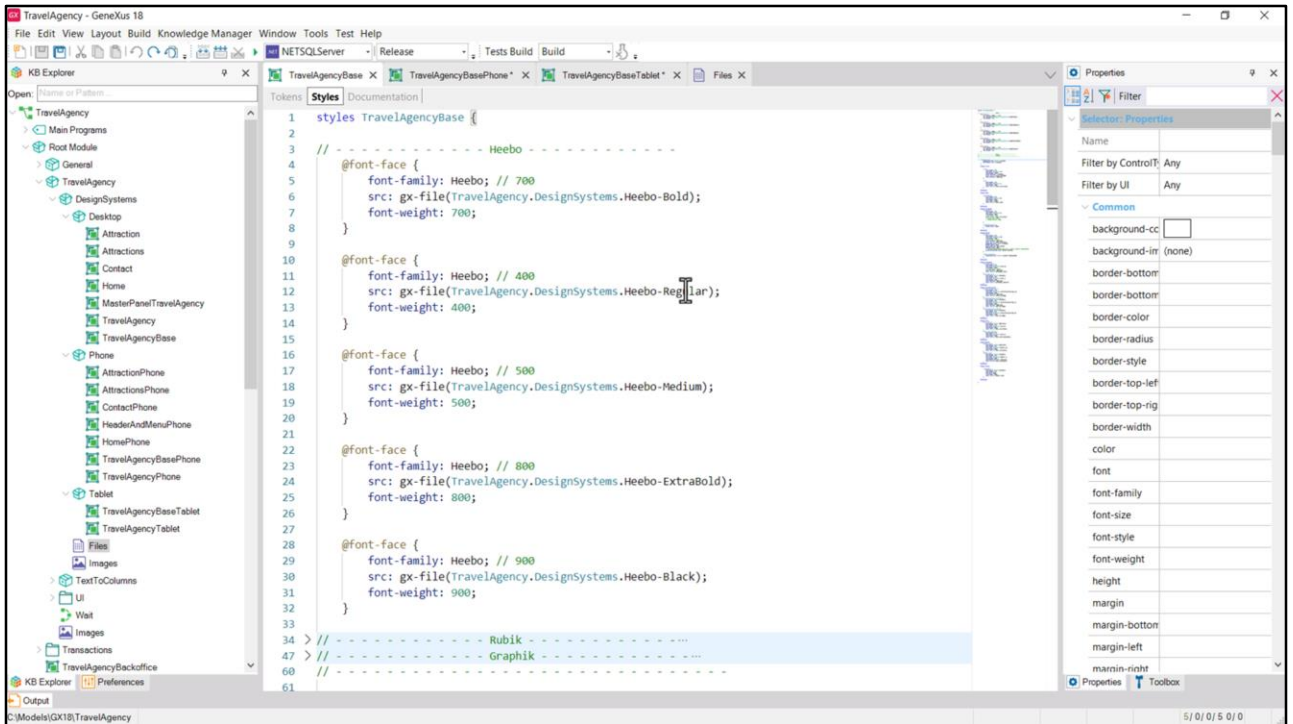
Habíamos insertado como archivos en la KB las fuentes no default que íbamos a utilizar, por ejemplo todas las Heebo de distintos pesos, ¿se acuerdan? Pero habíamos integrado exclusivamente las de formato woff2, que como habíamos indicado, eran las mejores para Angular. Sin embargo, también como dijimos en su momento, no se pueden utilizar en ambientes nativos.



Es por ello que agregué las mismas fuentes pero ahora en formato ttf.

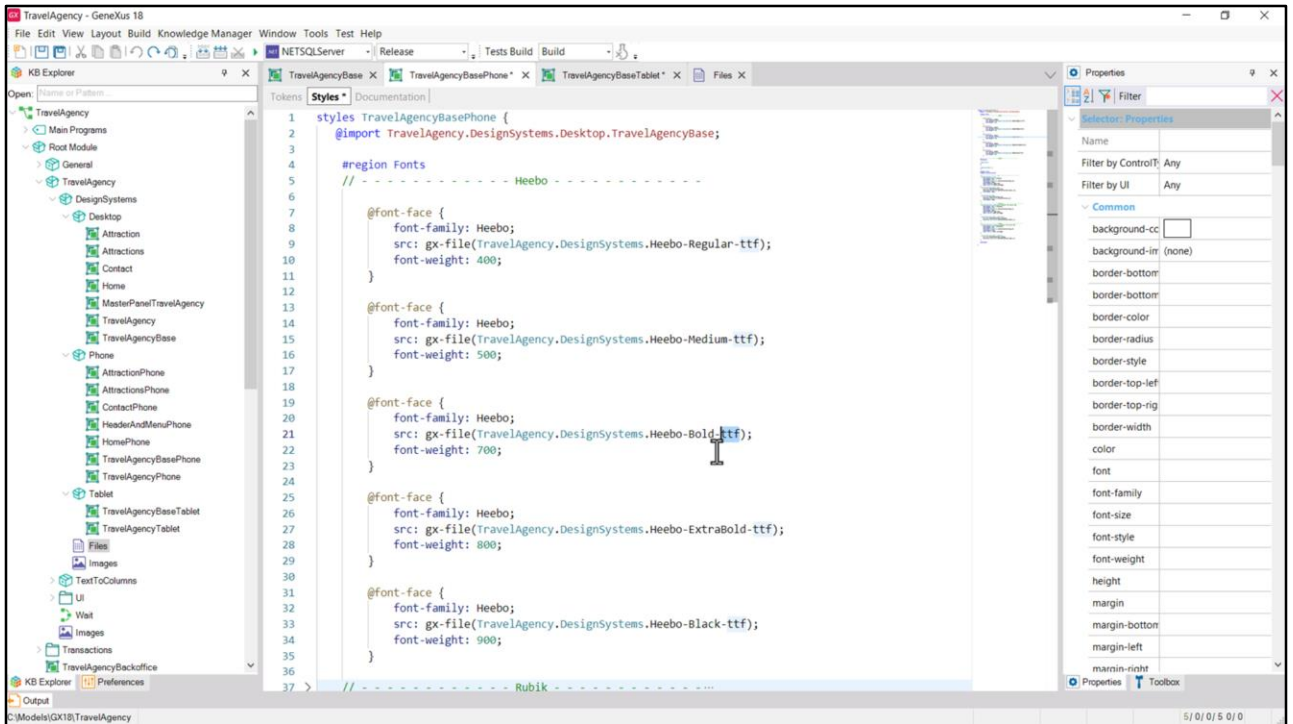
Si hubiéramos utilizado para Angular las ttf en lugar de las woff2, no tendríamos este problema, aunque las fuentes ttf no son las mejores para Angular.

¿Qué tendríamos que hacer entonces en nuestro caso donde sí tenemos los 2 tipos de fuentes?



En el DSO Base para Desktop teníamos las reglas font-face que declaraban las fuentes. Claramente tendremos que especializarlas para nuestros DSOs para aplicaciones nativas, para que ahora tomen el archivo ttf y no el woff2.

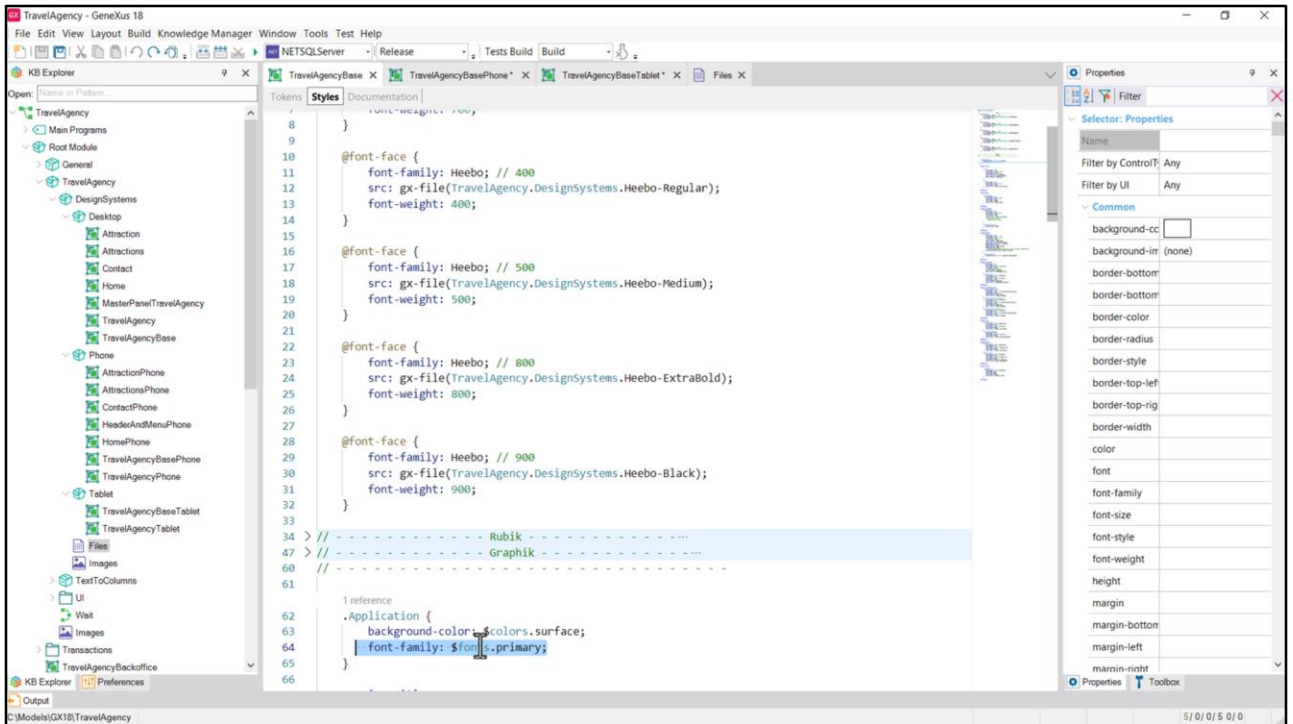
Es decir, tendríamos que copiar tanto para TravelAgencyBasePhone como para TravelAgencyBaseTablet todas estas reglas font-face y cambiar simplemente el archivo por el ttf.



Pero entonces, ¿si queremos que para la aplicación Angular en Phone utilice las woff2 y para la nativa en phone las ttf?
Allí no tendremos más remedio que especializar.

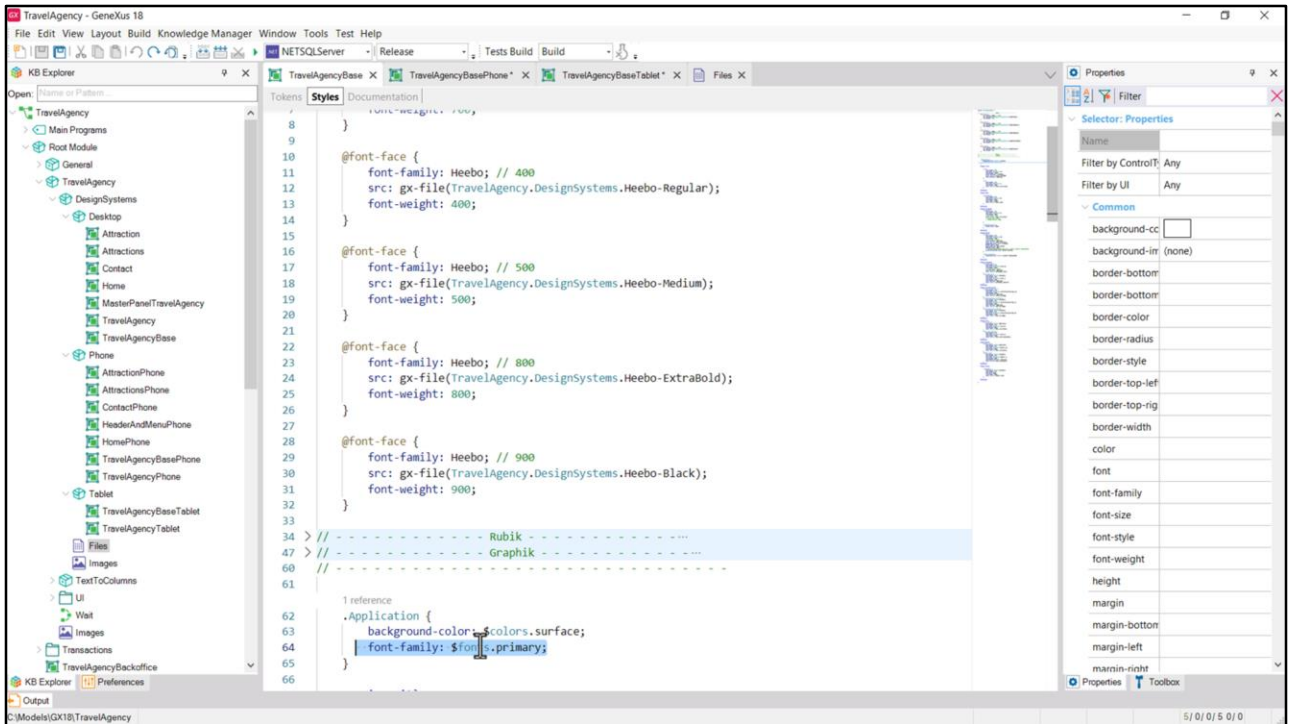
Es decir, tener un árbol de DSOs para plataforma Web Phone y otro para las nativas.

Entonces especificaríamos aquí el default para Phone que queremos que aplique sólo a las nativas y aquí su excepción, el de las Web, que ese sí utilizará el formato woff2.



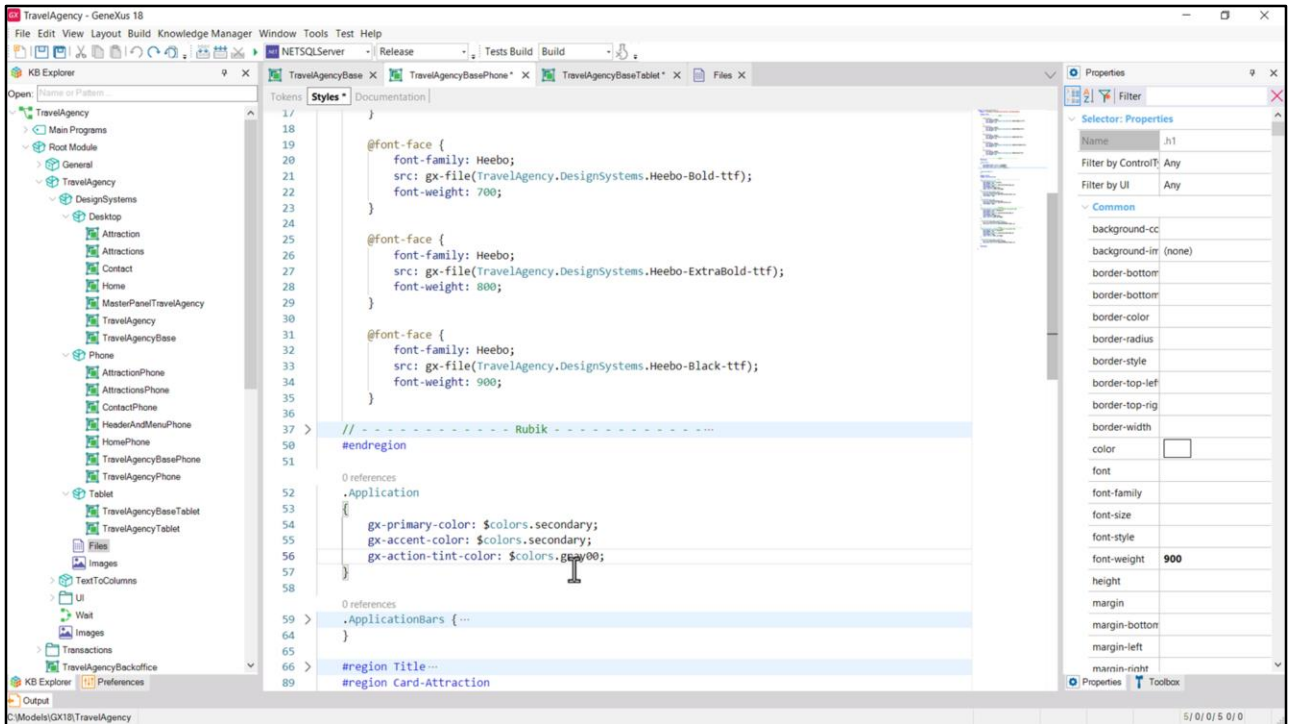
Pero tenemos un problema más importante que ese (supongamos que usemos las fuentes ttf para Angular también). El problema es que habíamos construido la solución en Angular aprovechándonos de que podíamos especificar propiedades que valieran como default para todas las clases especificándolas bajo la clase Application, ¿recuerdan? Y así fue como decidimos a todas las Heebo, independientemente de su peso, llamarlas igual, Heebo, y discriminarlas luego por la propiedad font-weight.

Así, habíamos podido definir un token primary para la familia Heebo, y en la clase Application indicar que esta será la familia default.



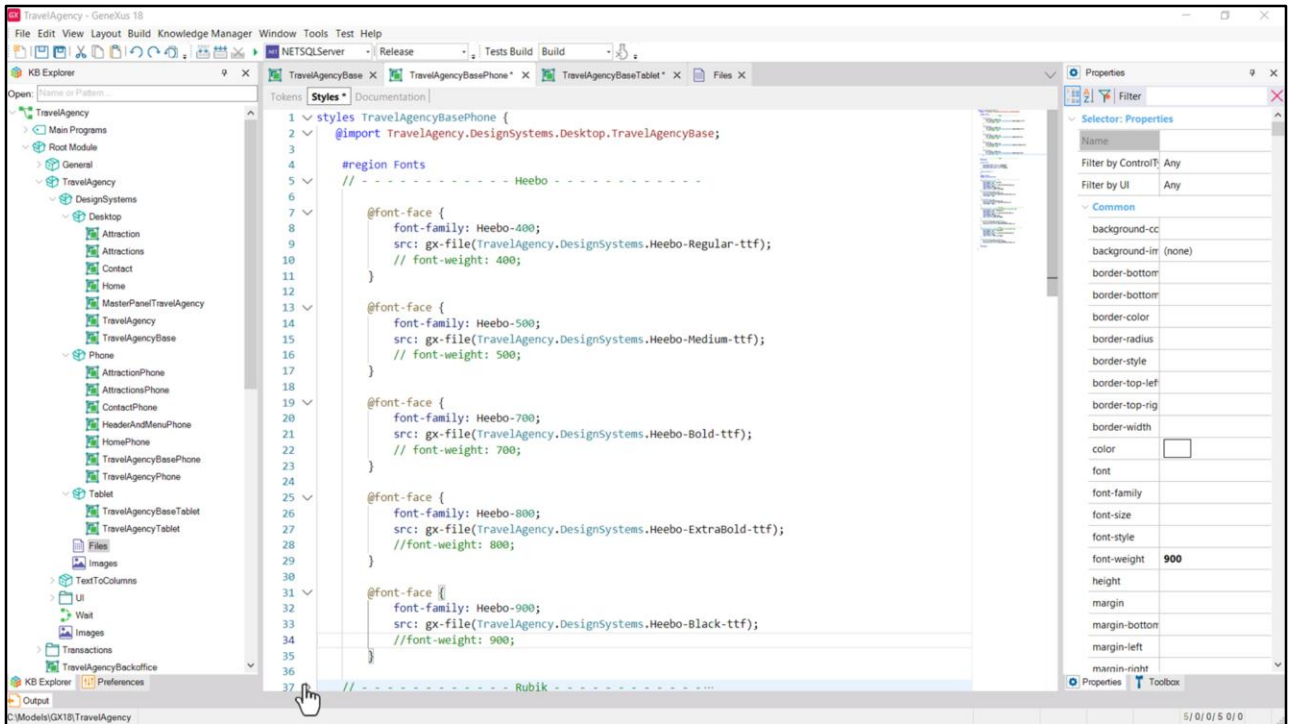
De esta manera luego en las clases que utilizaran esa familia, no teníamos que indicar la familia, sino solamente su peso. Aquí hay dos presupuestos que no funcionan en el mundo nativo con GeneXus (al menos no por el momento):

Uno: que las propiedades que se especifiquen en la clase Application valdrán por default para todas las clases. Esto es así para el mundo Angular porque esa clase se le aplica al tag body de todas las páginas HTML..



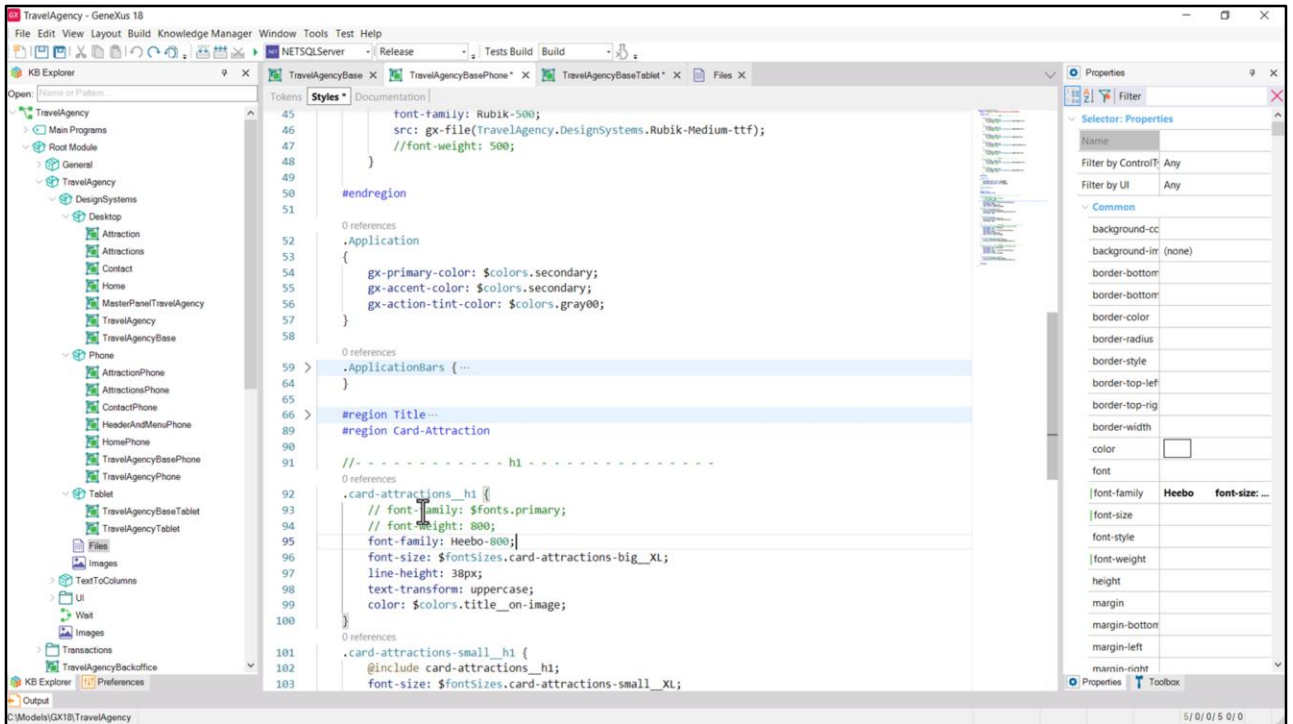
Pero en el mundo nativo esa clase sólo comanda algunos comportamientos generales de la aplicación, como los colores de la Application Bar, de las acciones, y demás. Por lo que si yo coloco aquí la propiedad font-family no valdrá por defecto para todos los controles de todas las pantallas.

Y segundo y más importante aún: en el caso Angular, dado que los nombres de familia de fuentes se repiten, lo que identifica a la fuente es el par nombre y peso. Esto no es así (al menos por el momento) para las aplicaciones nativas. Lo que identifica a la fuente es el nombre de la familia.

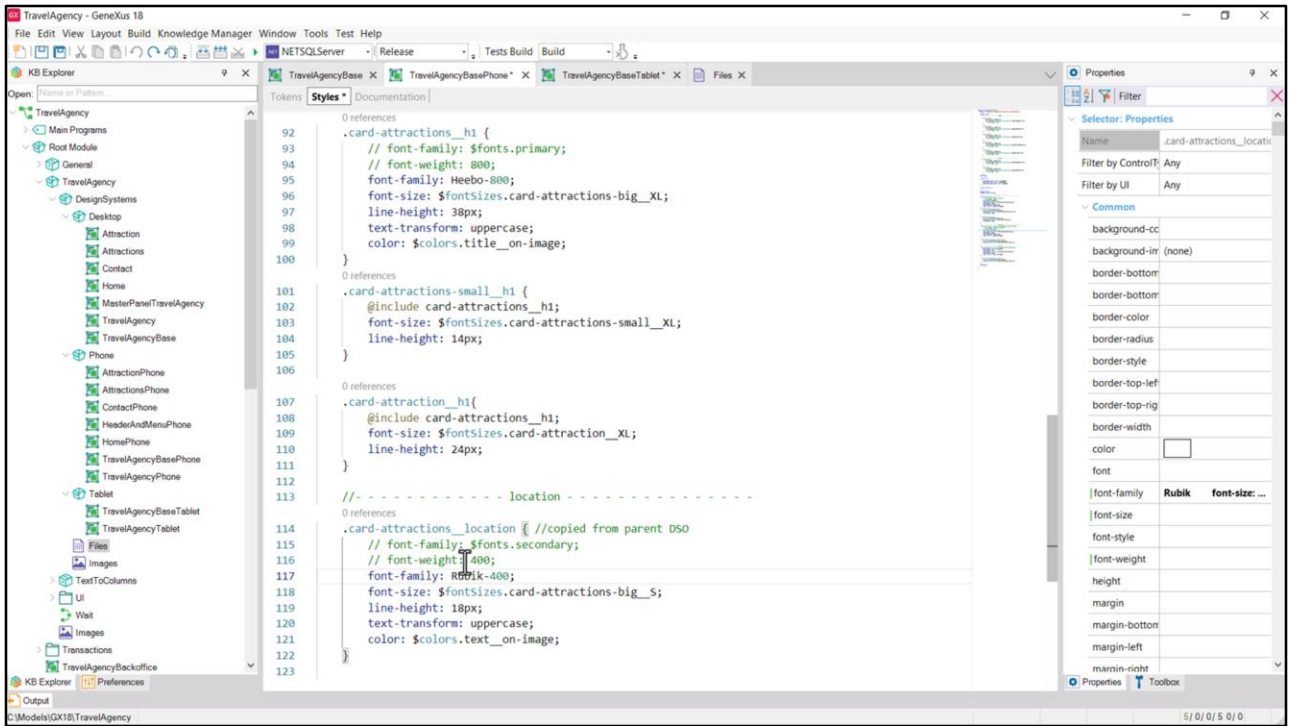


Así que tenemos que darles nombres distintos a las Heebo.

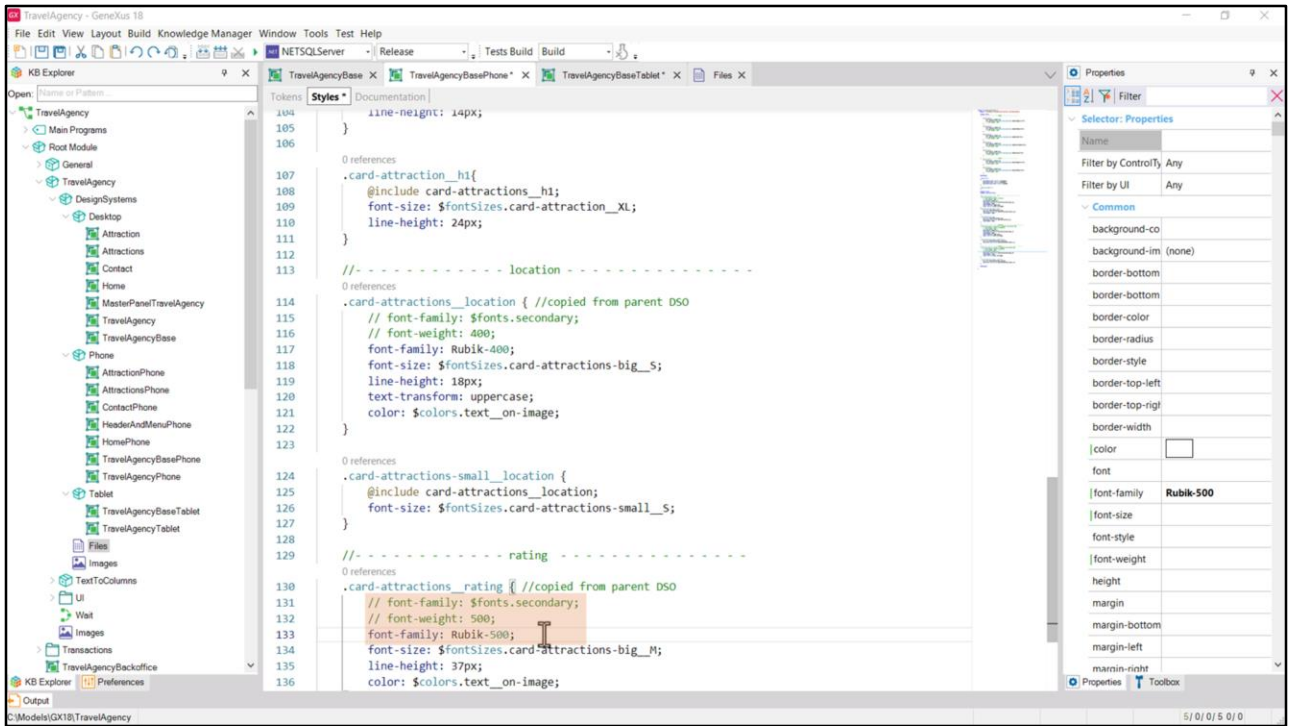
Y lo mismo a las otras familias.



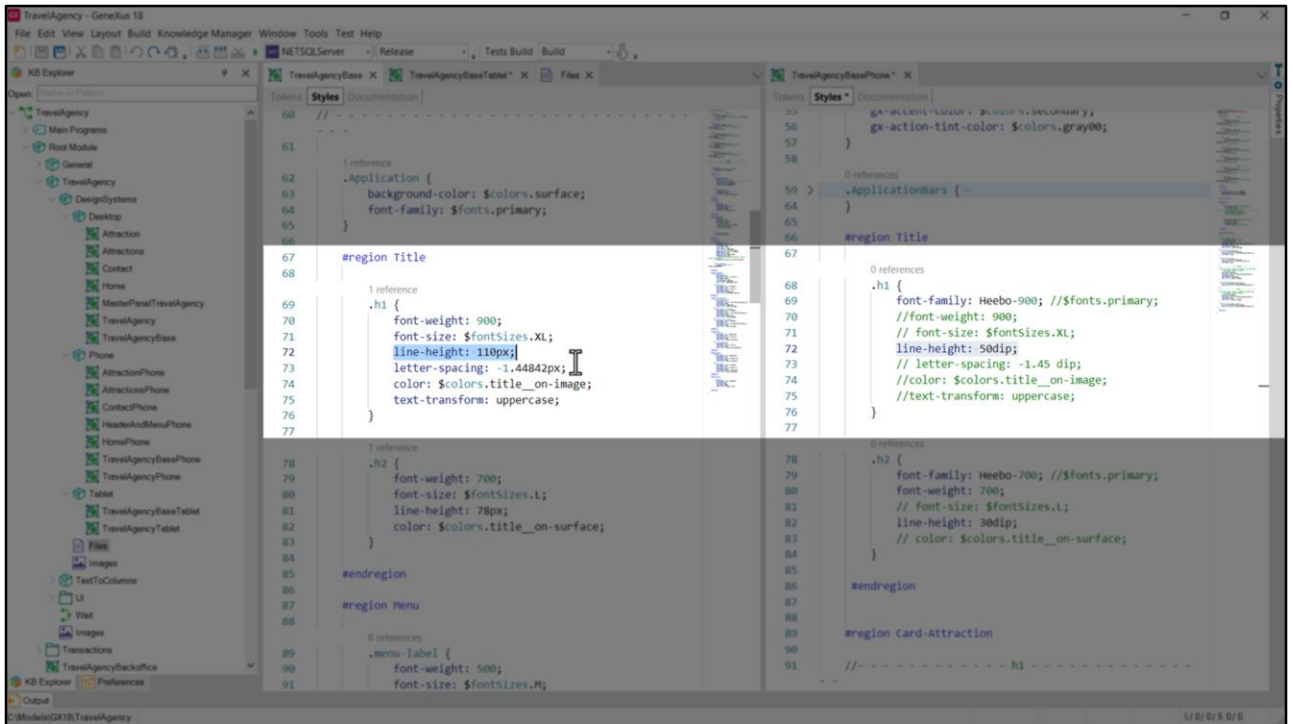
Pero esto conlleva que tengamos que redefinir todas las clases tipográficas. Por ejemplo estas...



Esta font secondary es la Rubik... así que tenemos que hacer este cambio...



Y aquí este otro...



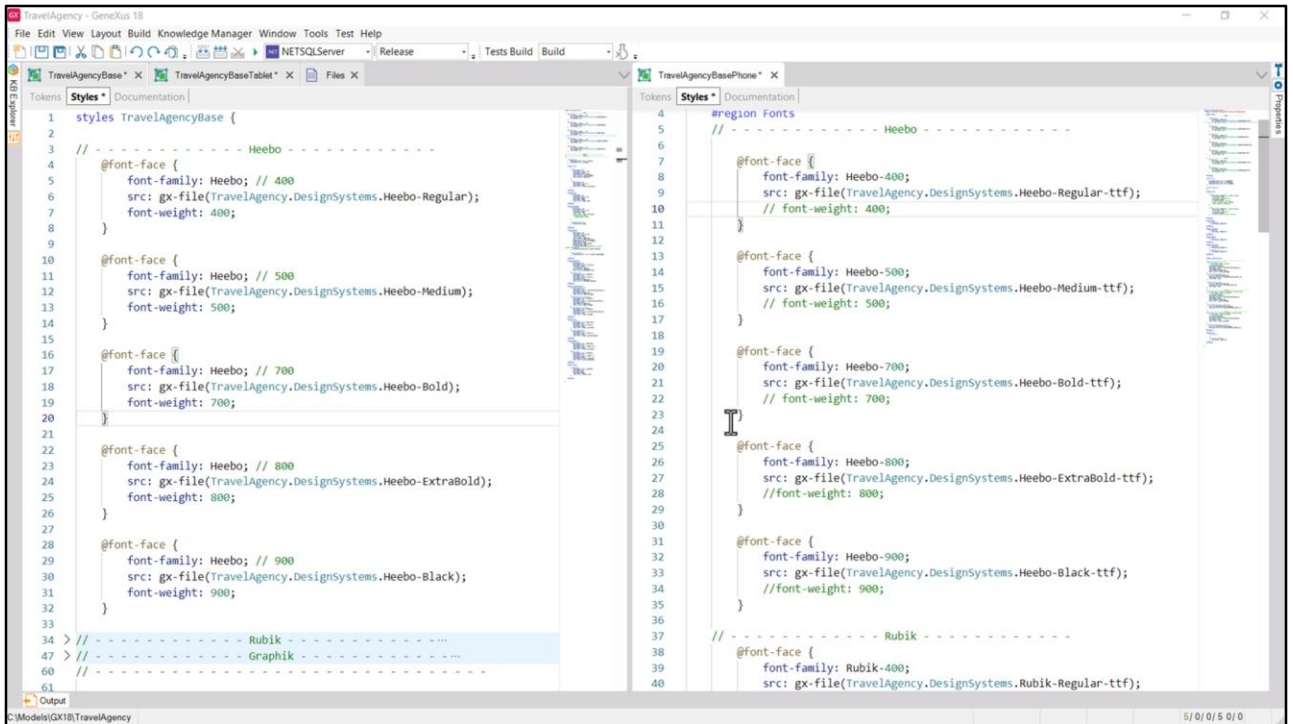
Pero no sólo eso, sino que también, las clases que no teníamos por qué modificar porque provenían del DSO TravelAgencyBase también tendremos que modificarlas.

Aquí vemos, y ya aprovecho a mostrarles, cómo tenemos la clase h1... a la que también podríamos quitarle el font-weight...

Vean dos cosas: primero, que si en este DSO que importa a este otro quiero a una clase cambiarle algunas propiedades y/o agregarle otras, no tengo que copiar todas las que no cambian. Esas valdrán.

Por eso dejé comentadas todas estas, porque son iguales, no tengo por qué volver a indicarlás aquí.

Pero sí las que varíen o las que se agreguen. En este caso el line-height cambia entre tamaño Desktop y tamaño Phone. Pero si no cambiara, el punto que quería mostrarles es que el haber desconocido esta diferencia entre Angular y las aplicaciones nativas en lo que hace a las definiciones de las fuentes, nos lleva a tener que especificar todas las clases tipográficas en el DSO hijo, así no cambien nada, solo para poder indicar la familia de fuentes correctamente. Y esto es costoso.



Si lo hubiéramos sabido de antemano no hubiéramos ido por esta solución que implementamos en Angular. Si no queremos tocar el DSO para Angular Desktop tendremos que seleccionar todas estas clases, al menos para cambiarles la indicación de la familia de fuentes.

Así nos quedarían los cambios: tenemos que declarar las reglas font-face para identificar a la familia de fuentes de manera única por el nombre...

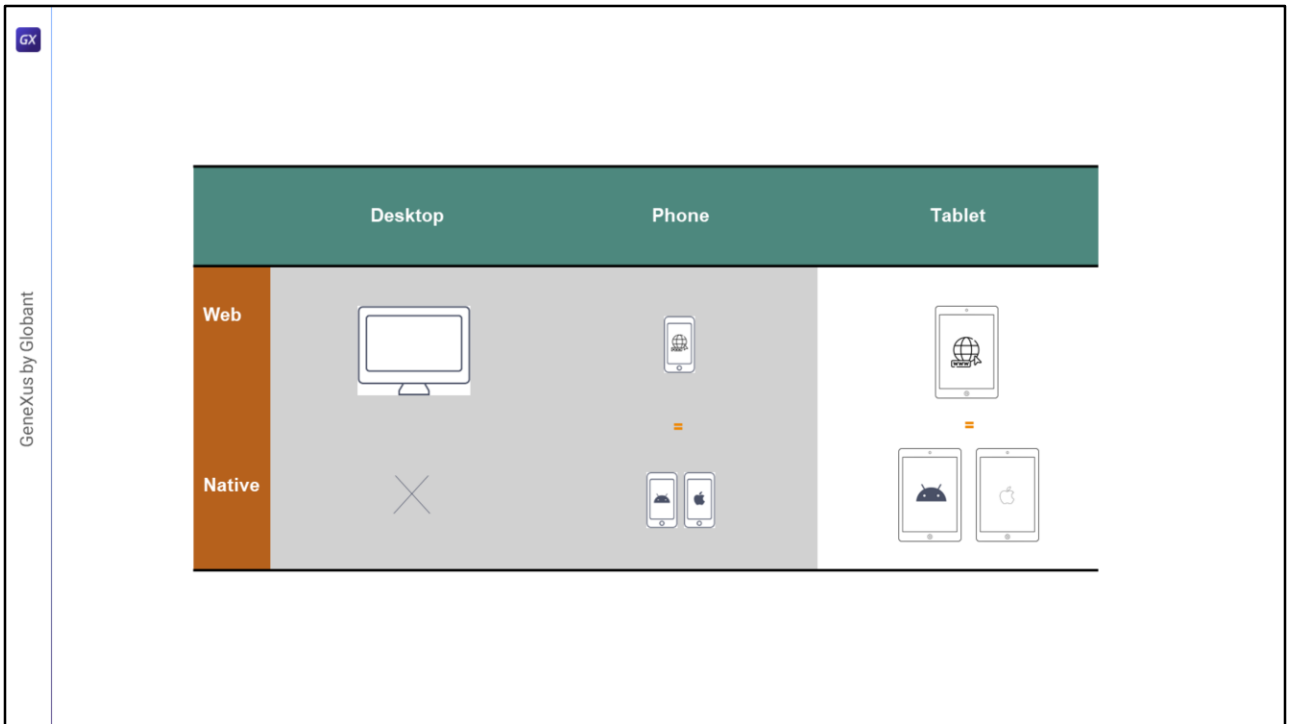
Y luego tenemos que agregar todas las clases tipográficas, aunque sea solamente para indicarle la familia de fuentes por su nombre.


```
TravelAgency - GeneXus 18
File Edit View Layout Build Knowledge Manager Window Tools Test Help
NETSQLServer Release Tests Build Build
TravelAgencyBase * x TravelAgencyBaseTablet * x Files x TravelAgencyBasePhone * x
Tokens Styles * Documentation Tokens Styles * Documentation
79 font-weight: 700;
80 font-size: $fontSizes.L;
81 line-height: 78px;
82 color: $colors.title_on-surface;
83 }
84 #endregion
85 #region Menu
86 0 references
87 .menu-label {
88 font-weight: 500;
89 font-size: $fontSizes.M;
90 line-height: 29px;
91 text-align: center;
92 color: $colors.menu_item;
93 }
94 #endregion
95 #region Paragraph
96 0 references
97 .paragraph {
98 font-weight: 400;
99 font-size: $fontSizes.S;
100 line-height: 29px;
101 //text-align: left;
102 color: $colors.text_on-surface;
103 // margin-top: 110px;
104 // margin-bottom: 110px;
105 }
106 #endregion
107 0 references
108 .paragraph-spacing {
109 margin-block: 110px;
110 }
111 }
74 //color: $colors.title_on-image;
75 //text-transform: uppercase;
76 }
77 }
78 0 references
79 .h2 {
80 font-family: Heebo-700; //$fonts.primary;
81 //font-weight: 700;
82 // font-size: $fontSizes.L;
83 line-height: 30dip;
84 // color: $colors.title_on-surface;
85 }
86 #endregion
87 #region Menu
88 0 references
89 .menu-label {
90 font-family: Heebo-500;
91 }
92 #endregion
93 #region Paragraph
94 0 references
95 .paragraph {
96 font-family: Heebo-400;
97 }
98 #endregion
99 #region Button
100 0 references
101 .button {
102 font-family: Heebo-800;
103 }
104 #endregion
105 #region Card-Home
106 0 references
107 .card-home_h1 {
108 font-family: Heebo-800;
109 }
110 }
Output
C:\Models\GX18\TravelAgency 5/0/0/5/0/0
```

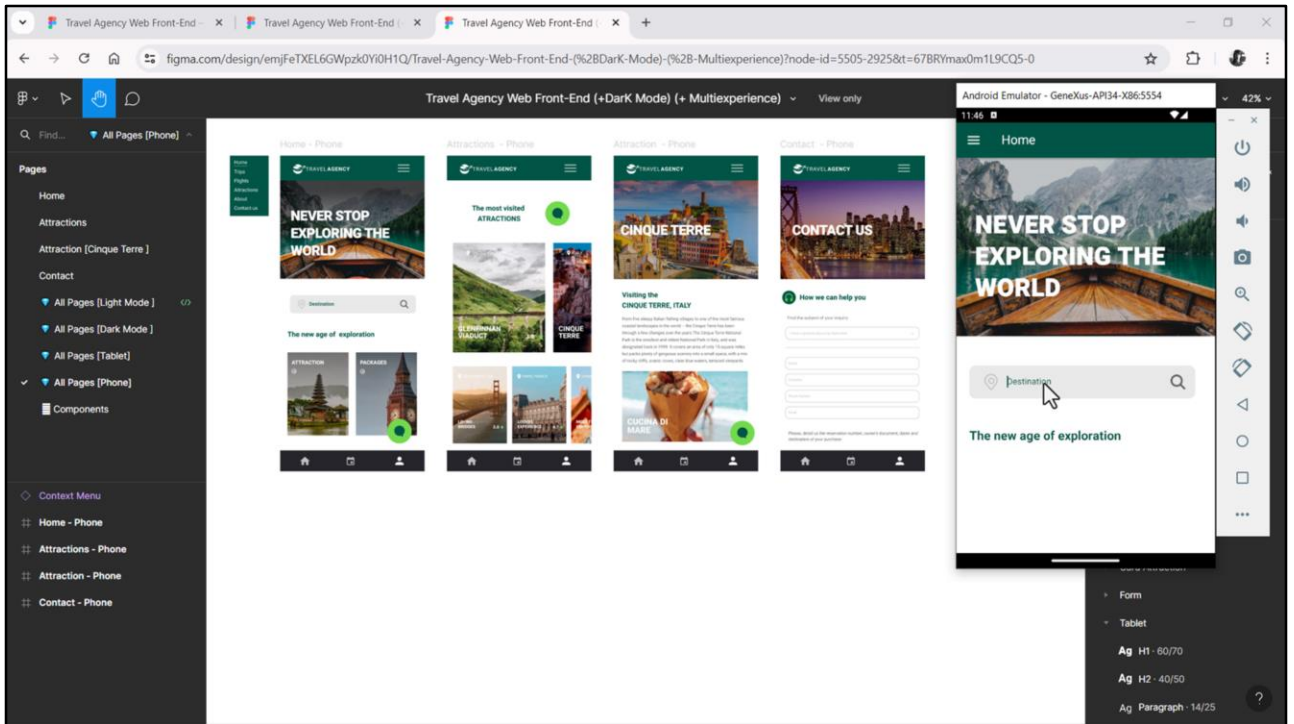
A estas les cambié algo más, a estas otras no las analicé para ver si debía cambiar algo más: solamente les cambié la familia de fuentes. Pero vean que tuve que hacerlo una por una con todas ellas.

MUX: Native vs Angular

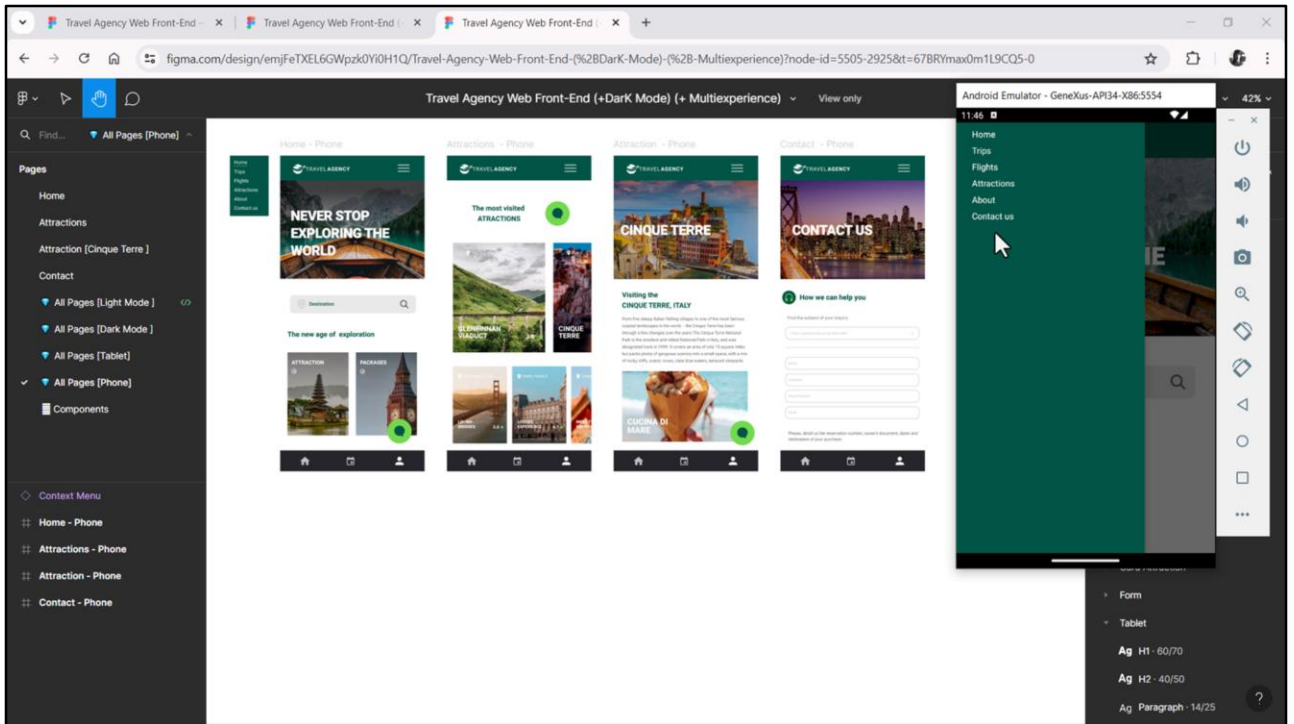
Esto seguramente les haya resultado engorroso. Era un ejemplo nada más, de los obstáculos que podrán surgir a la hora de implementar la multiexperiencia y que se pueden reducir al mínimo si uno conoce las diferencias desde el principio del desarrollo.



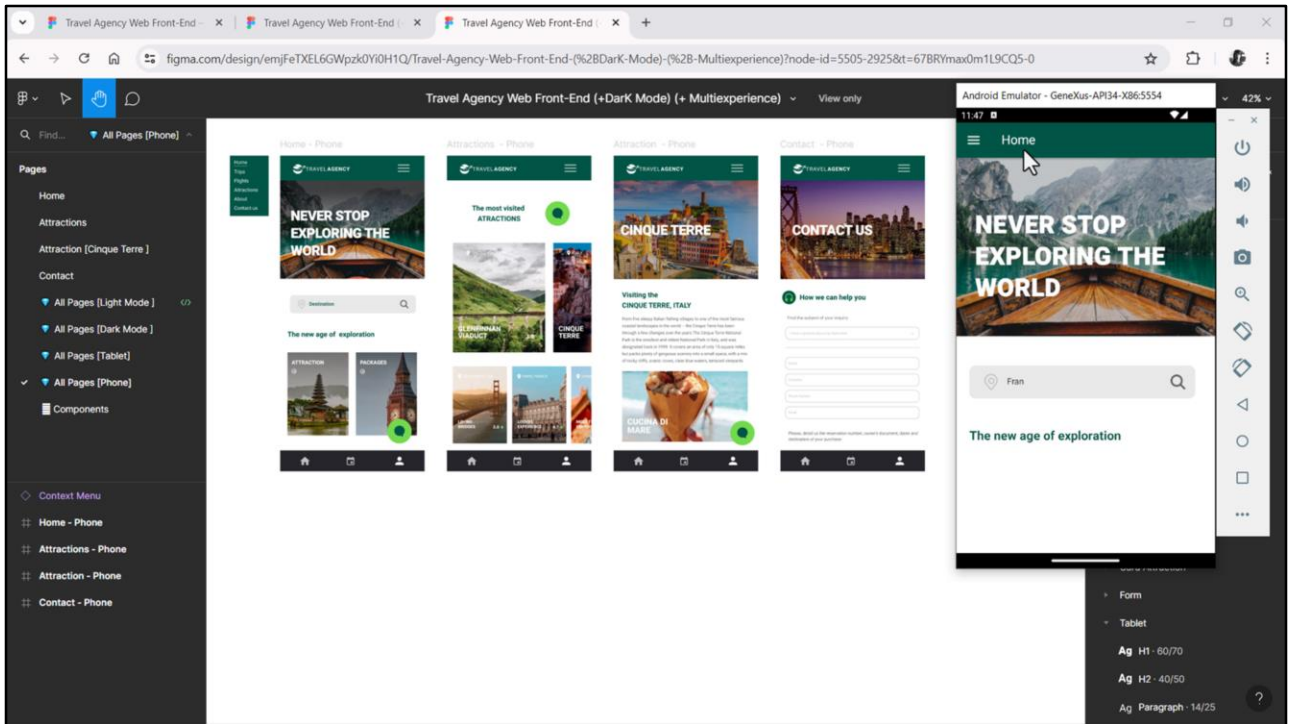
Es que las aplicaciones nativas no se han construido sobre el paradigma CSS y eso tiene sus efectos, si bien el horizonte de expectativas es acercar ambos mundos en GeneXus lo más que se pueda.



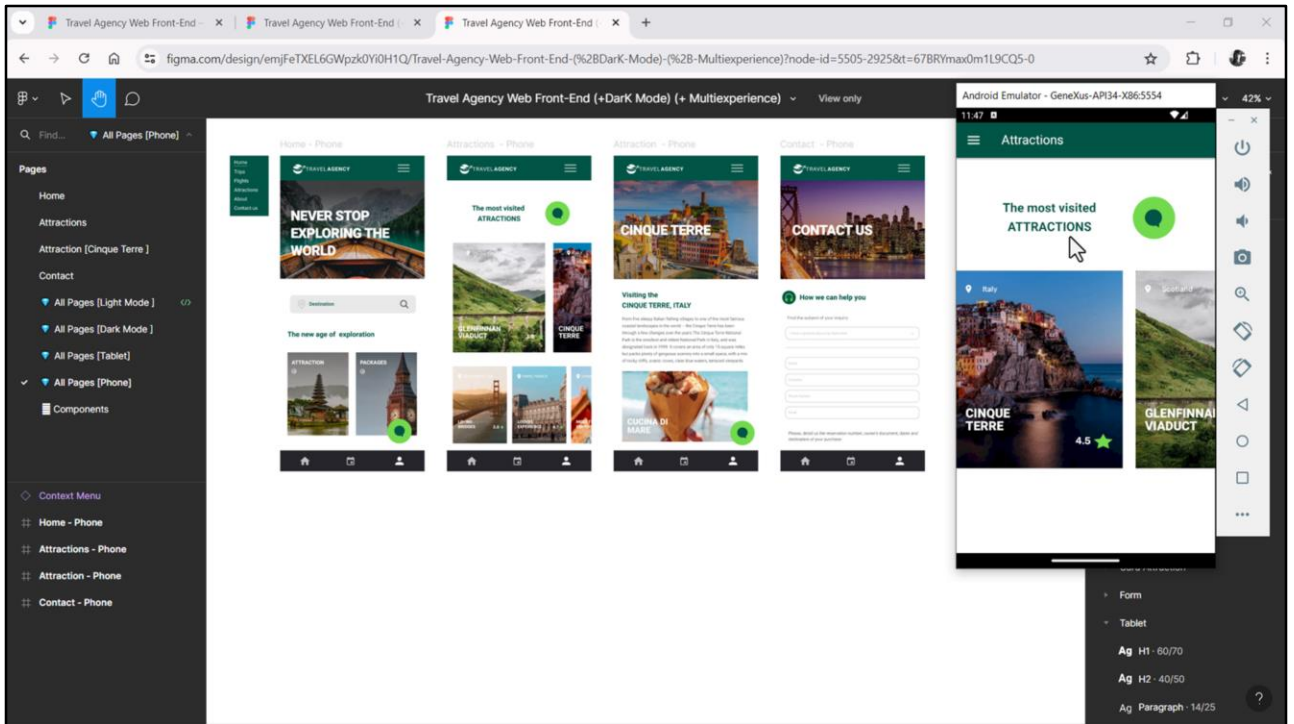
Para entusiasmarlos avancé rápidamente en mi KB para que vayan viendo en este emulador cómo me van quedando las pantallas de la aplicación Android para Phone.



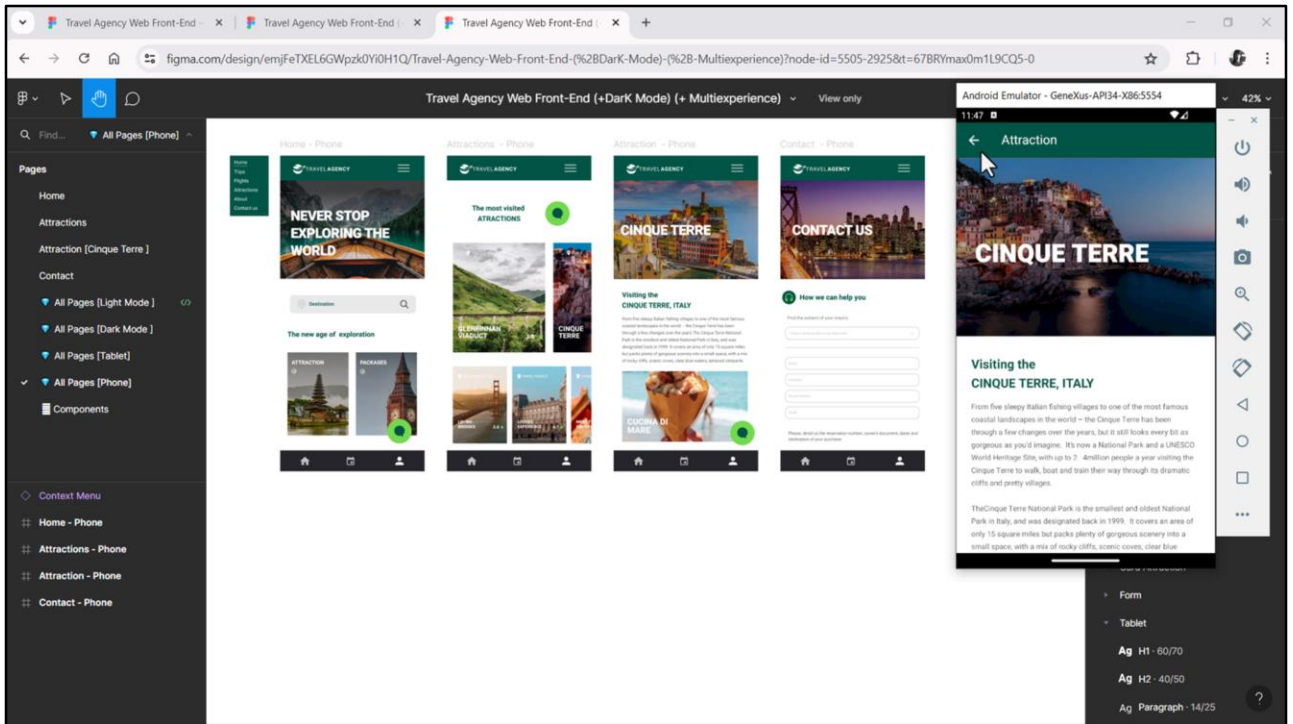
Por ejemplo, vemos el menú hamburguesa (a la izquierda, como es el estándar en su sistema operativo, y no a la derecha, como nos propone el diseño de Chechu)...



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.

Bueno, para que no se nos haga tedioso voy a interrumpir acá y vamos a continuar en el próximo video.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com