

Color system in GeneXus



Cecilia Fernández

Color system

The screenshot displays a design system tool interface for 'TravelAgency'. The left pane shows a code editor with the following configuration:

```
1 tokens TravelAgency {  
2  
3 #colors  
4 {  
5 #region Application  
6 primary: #73D94F;  
7 secondary: #015547;  
8 primary--highlighted: #A7E491;  
9  
10 #endregion  
11  
12 #region Neutral  
13 gray00: #FFFFFF;  
14 gray200: #C1C1C1;  
15 gray600: #616161;  
16 opacity: #19181933;  
17 #endregion  
18 }  
19 }  
20  
21 #fonts  
22 {  
23 primary: Heebo;  
24 secondary: Rubik;  
25 additional: Graphik;  
26 }  
27 }
```

The right pane shows a visual representation of the color system, organized into two sections: APPLICATION and NEUTRAL. Each section contains a table of color tokens with their names and values.

NAME	COLOR VALUE
primary	#73D94F
secondary	#015547
primary-highlighted	#A7E491

Below the APPLICATION table is a '+ Add new token' button. The NEUTRAL section contains the following table:

NAME	COLOR VALUE
gray00	#FFFFFF
gray200	#C1C1C1
gray600	#616161
opacity	#19181933

Below the NEUTRAL table is another '+ Add new token' button. At the bottom of the right pane, there is a 'FONTS' section header.

En el video anterior habíamos partido de estos tokens definidos en nuestro Design System, que representaban de un modo demasiado básico al sistema de colores de nuestra aplicación, y habíamos llegado, al final del video, con un sistema de colores mucho más complejo, por un lado, pero a la vez funcionalmente mucho más preciso, mucho más semántico.

Tokens Travel Agency - Google

docs.google.com/spreadsheets/d/1oMvMncna8ZASn5_iTG6pcap3yiArNcfvMSgVO068e_/edit#gid=1337893737

GeneXus DL Portal Issues

Tokens Travel Agency

File Edit View Insert Format Data Tools Extensions Help

100% 123 Default... 10 B I A

L15

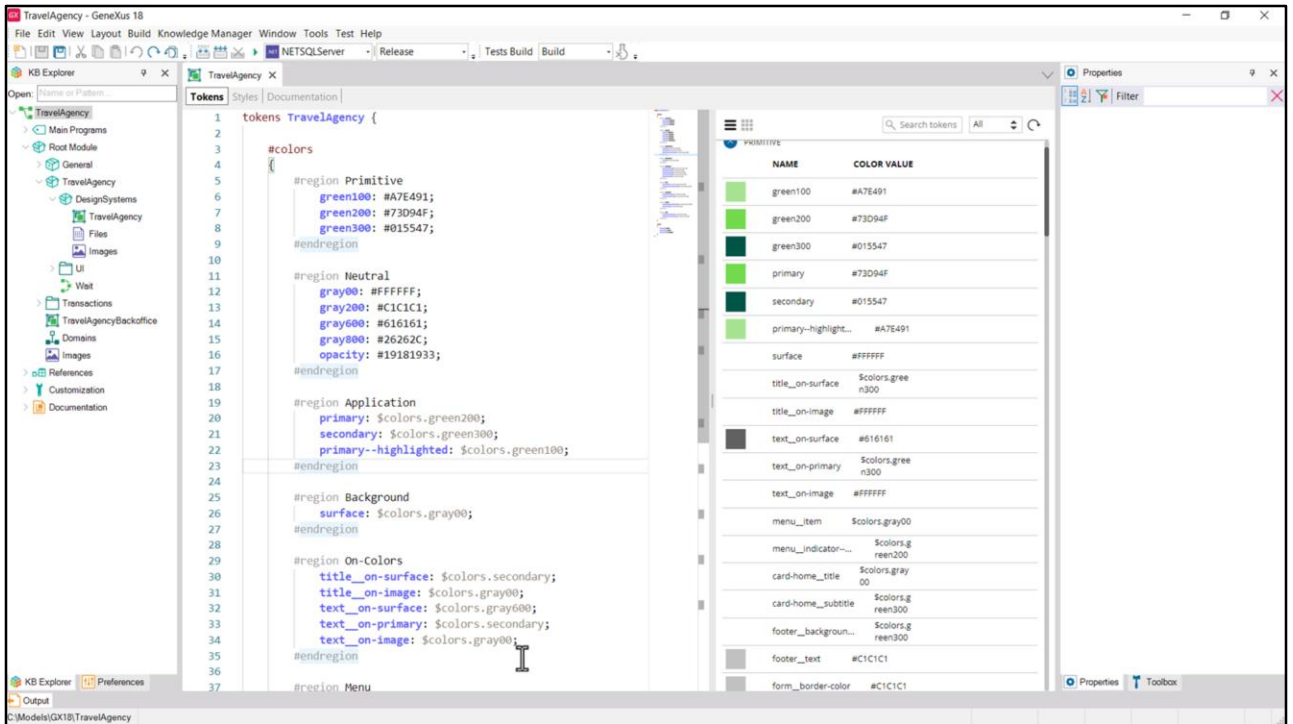
	C	D	E	F	G	H	I	J	K	L	M	N
1												
2												
3		Light Value	Value Dark	Opacity	Alias/Semantic				Component/Specific			
4		#A7E491	#54958A	100%	Region	Name	Light	Dark	Region	Name	Light	
5		#73D94F		100%	Application	primary	green200	green300	Menu	menu_item	text_on-image	text_o
6		#015547		100%		secondary	green300	green200		menu_indicator--selected	primary	second
7						primary--highlighted	green100	green100				
8		#FFFFFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00 surface	gray800
9		#C1C1C1		100%						card-home_subtitle	secondary	primary
10		#D2D2D2		100%	On_Colors	title_on-surface	secondary	gray00	Footer	footer_background-color	secondary	surface
11		#616161		100%		title_on-image	gray00	gray00		footer_text	gray200	gray200
12		#26262C		100%					Form	form_border-color	gray200	gray200
13		#191819		33%		text_on-surface	gray600	gray00		form_text	text_on-surface	text_o
14						text_on-primary	secondary	secondary		form_text-placeholder	gray200	gray200
15						text_on-secondary	gray200?	primary?		form_text	text_on-surface	text_o
16						text_on-image	gray00	gray00				
17									Card-Attractions	card-attraction_title	title_on-image	title_o
18										card-attraction_text	text_on-image	text_o
19									Hero	hero_title	title_on-image	title_o
20									Banner	banner_background-color	primary	primary
21												
22												

Text Styles Text Styles + Multiexperience Colors Styles Colors Styles + Dark Mode Color tokens Text Classes

Era el que teníamos expresado en esta planilla.

	D	E	F	G	H	I	J	K	L	M	N	O	P
5		100%			secondary	green300	green200			menu_indicator--selected	primary	secondary	
6		100%			primary--highlighted	green100	green100						
7									Card-Home	card-home_title	gray00 surface	gray800 surface	
8		100%	Background	surface	gray00	gray800			card-home_subtitle	secondary	primary		
9		100%											
10		100%	On_Colors	title_on-surface	secondary	gray00			Footer	footer_background-color	secondary	surface	
11		100%		title_on-image	gray00	gray00			footer_text	gray200	gray200		
12		100%											
13		33%		text_on-surface	gray600	gray00			Form	form_border-color	gray200	gray200	
14				text_on-primary	secondary	secondary			form_text-placeholder	gray200	gray200		
15				text_on-secondary	gray200?	primary?			form_text	text_on-surface	text_on-surface		
16				text_on-image	gray00	gray00							
17									Card-Attractions	card-attraction_title	title_on-image	title_on-image	
18									card-attraction_text	text_on-image	text_on-image		
19													
20									Hero	hero_title	title_on-image	title_on-image	
21													
22									Banner	banner_background-color	primary	primary	
23									banner_title	text_on-primary	text_on-primary		
24									banner_text	text_on-primary	text_on-primary		
25													
26													

Habíamos dicho que los tokens estos, estos así marcados, que le daban más especificidad al sistema, a la vez le daban más complejidad, por lo que en esta oportunidad no los utilizaré. Los nuevo para acá abajo para no perderlos.

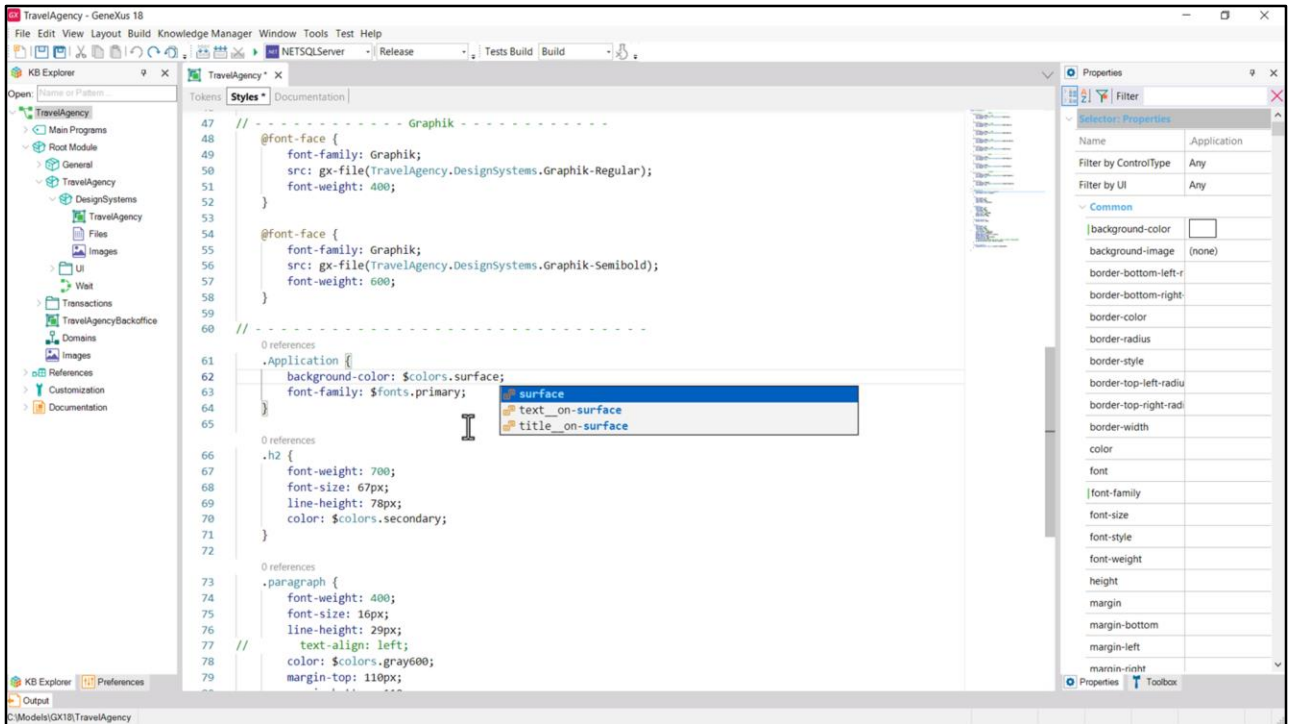


Bueno, aquí tengo nuestro DSO ya con todos los tokens indispensables para modelar el sistema, y sus valores para modo Light, únicamente.

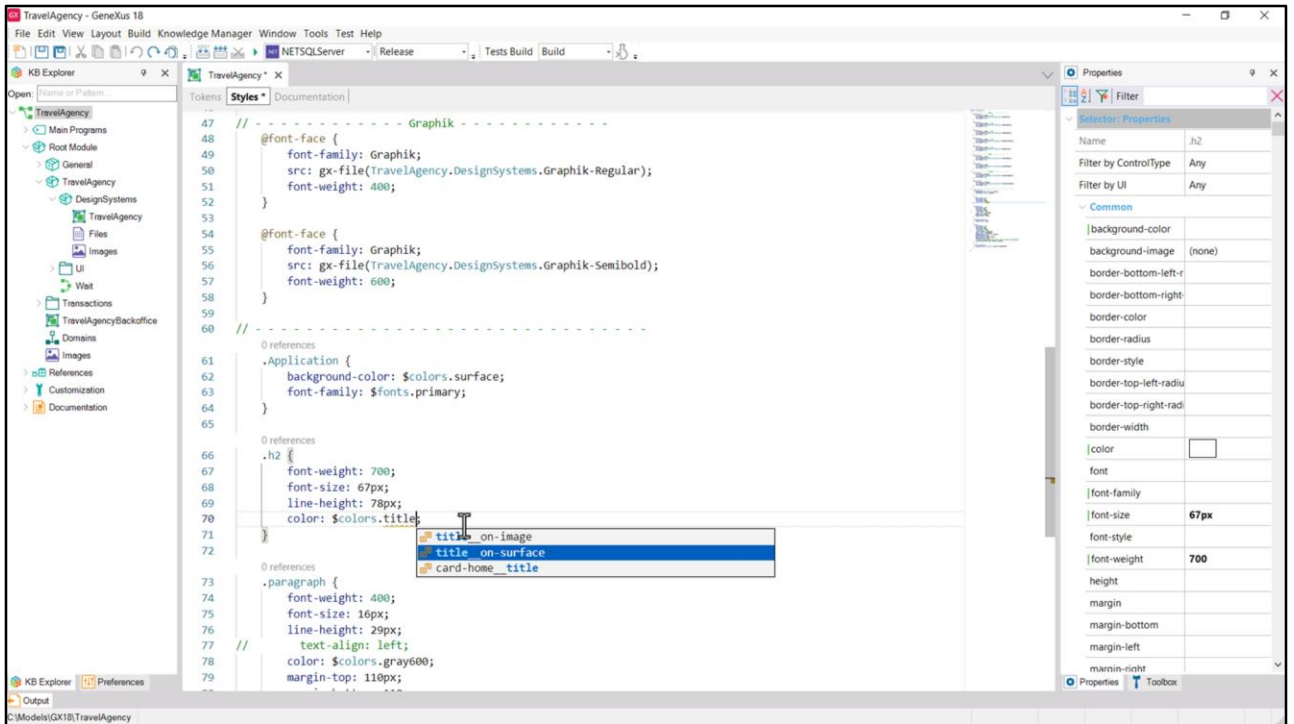
Vemos que los que colocamos en la región Primitive y Neutral tienen valores absolutos, en hexadecimales, mientras que todos los demás tienen como valores referencias a otros tokens.

Hasta acá cosas que más o menos ya sabíamos. Nos queda por ver cómo representamos el modo Dark.

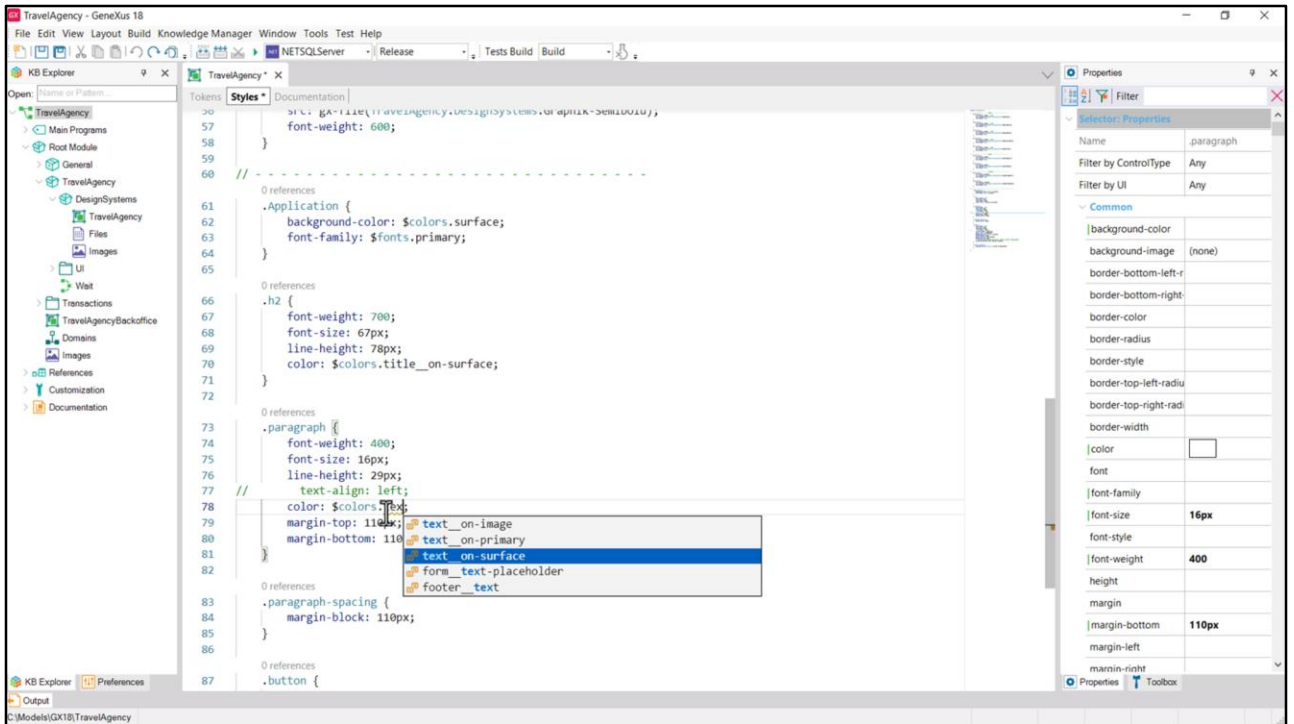
Pero antes... vamos a modificar los colores que usamos en las clases, para que utilicen los nuevos tokens. ¿Qué colores teníamos?



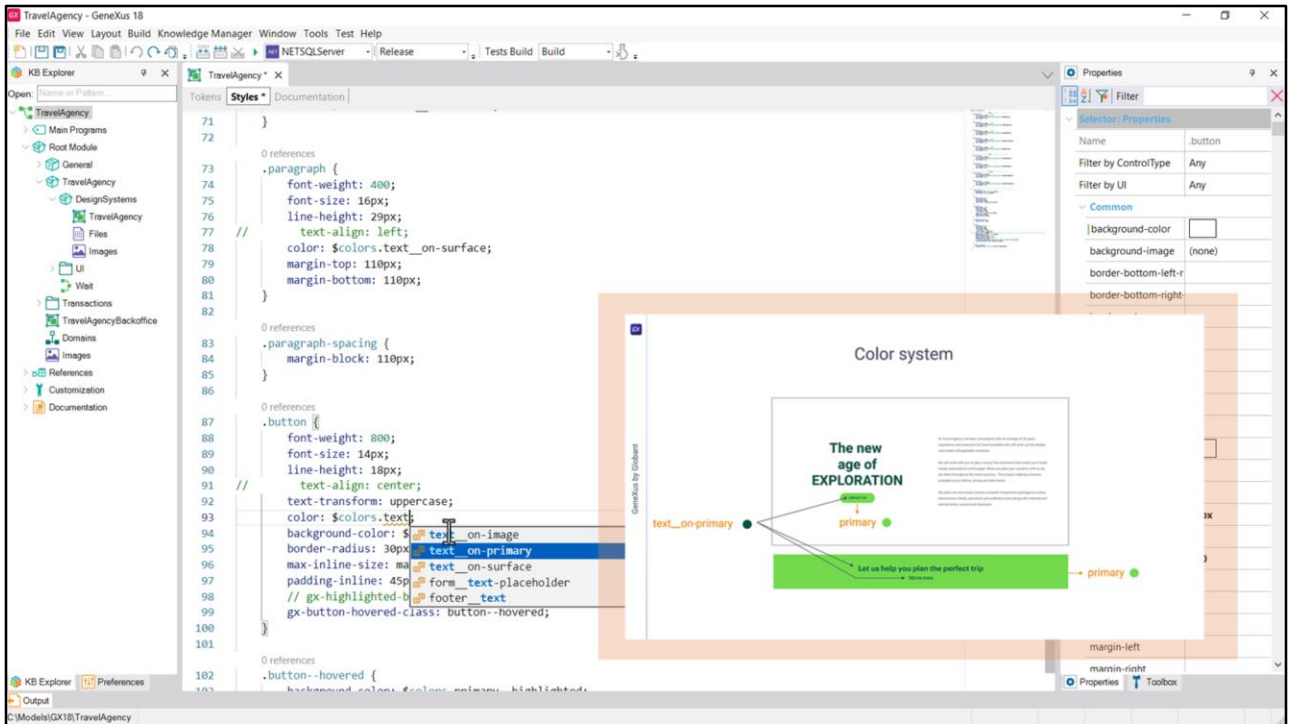
El background-color de las pantallas, que lo habíamos colocado dentro de la clase Application, ahora corresponderá al token de color surface...



La clase h2 era la del título sobre la superficie Surface, tiene como color el token title__on-surface.



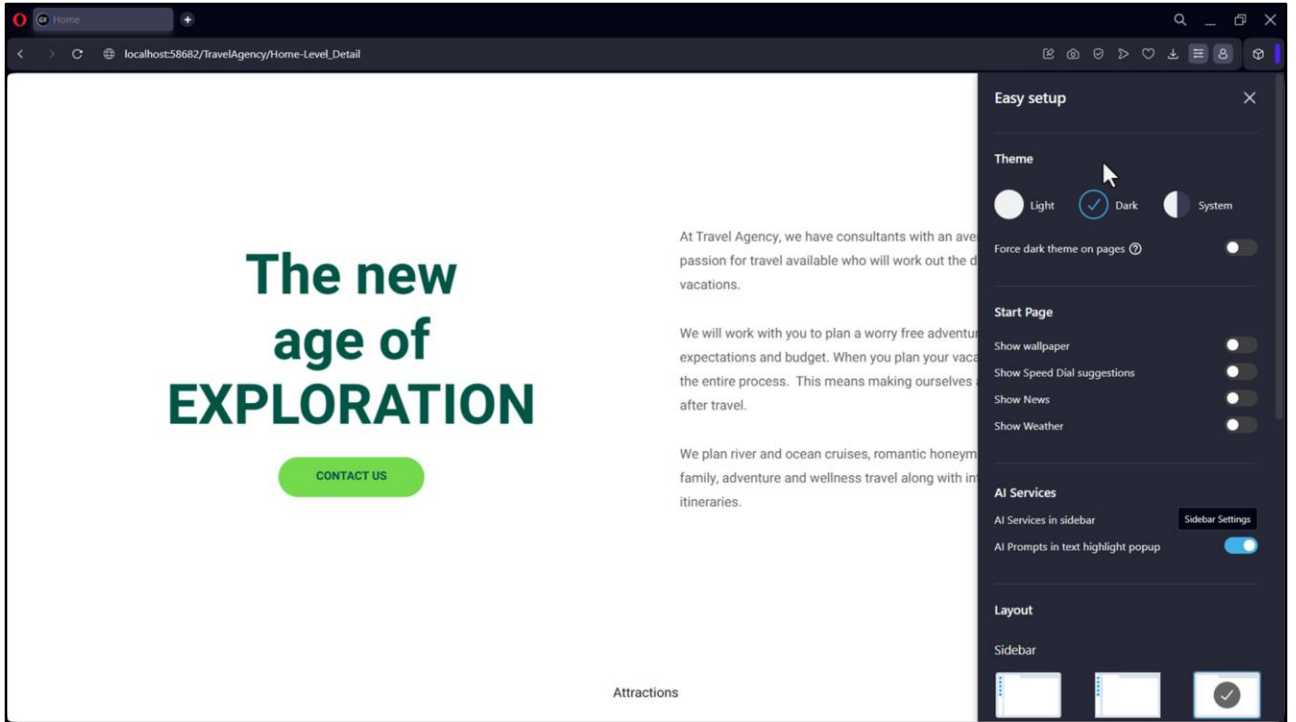
La clase paragraph correspondía al texto común sobre la superficie, así que aquí sustituimos por el token de color text_on-Surface.



Y para el botón, tenemos el background-color correcto, porque es el primary como color de fondo.

Y el color debería ser text__on-primary.

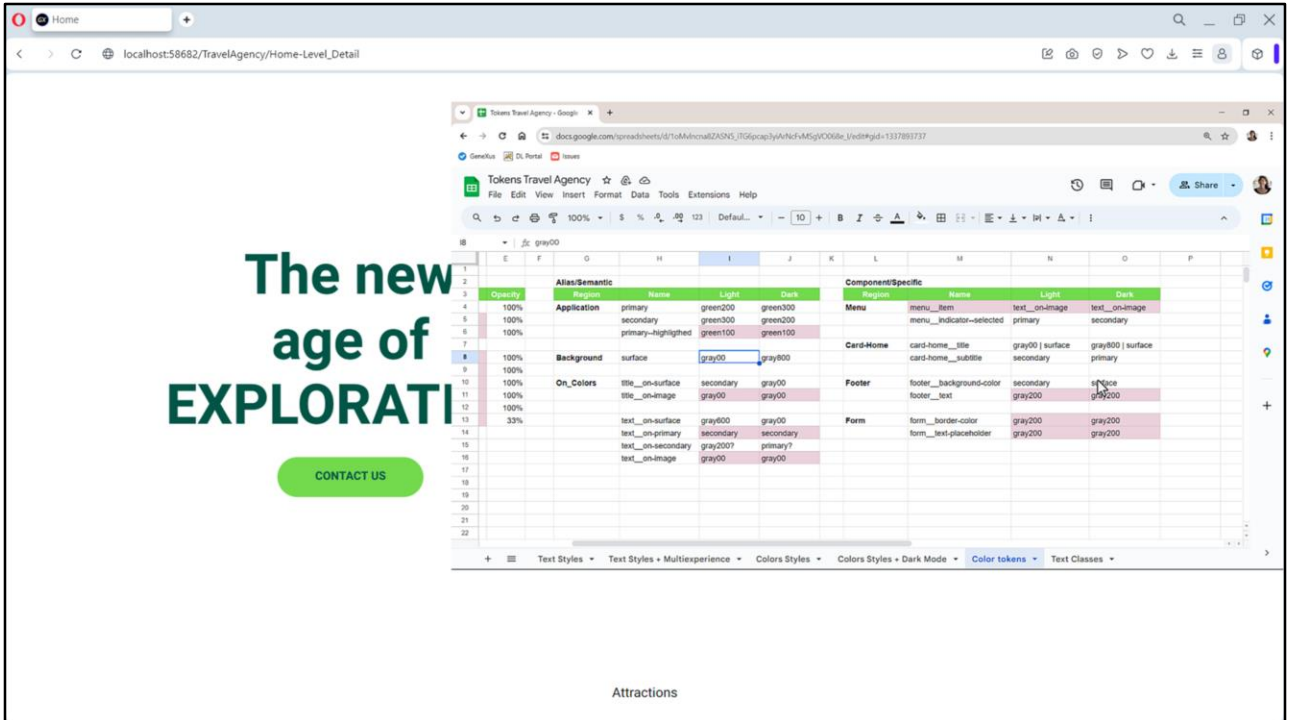
Y con la acción de hover sobre el botón, el color de fondo cambia por el primary—highlighted, lo que es correcto.



Ejecutemos para verificar que todo se vea tal cual lo esperamos.

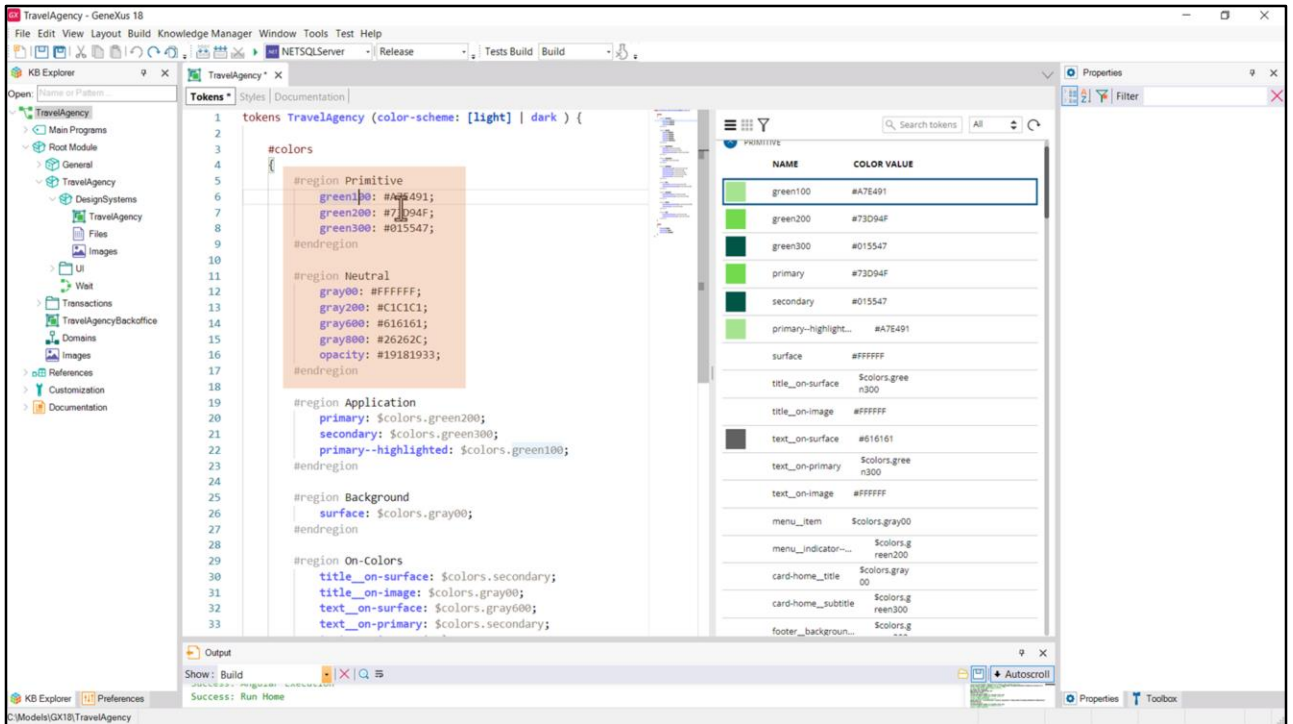
Voy a copiarme la url para ejecutar la aplicación en el navegador Opera que me permite cambiar entre modo Light y Dark más fácilmente.

Vemos que si elijo el modo Dark lo que vemos es exactamente igual que en modo Light.



Bueno, no demoremos más la pregunta que estamos deseando contestar desde el principio: ¿cómo hago para especificar los cambios en los tokens de color para poder hacerlos variar por modo?

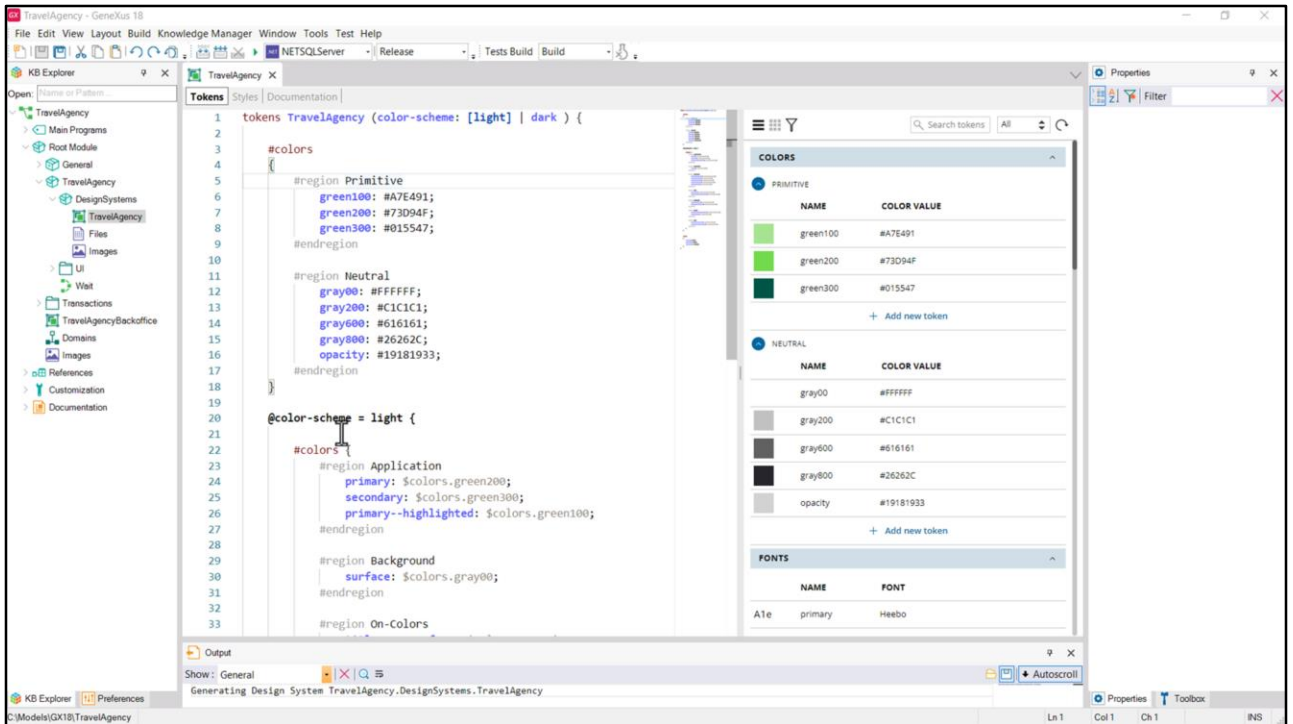
Allí aparecen lo que llamamos options, opciones del conjunto de tokens.



A este conjunto de tokens llamado TravelAgency le podemos especificar variaciones, y lo hacemos entre paréntesis. En nuestro caso queremos hacer variar algunos de los tokens de color según esquema de color, color-scheme se llamará nuestra opción, que podrá asumir dos valores: light y dark. Podemos especificar con paréntesis rectos cuál será la opción default.

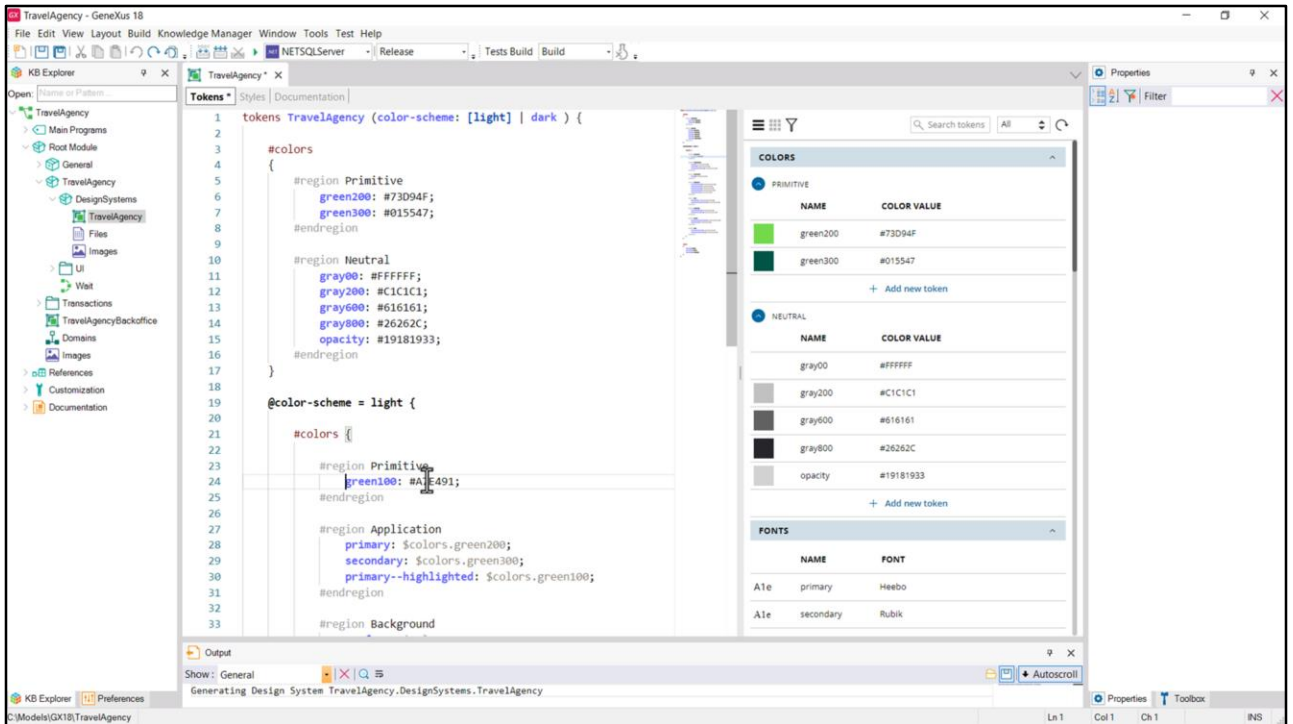
Y luego nos queda por especificar qué valores adquieren los tokens para un valor de la opción y para el otro.

Por ejemplo, los valores de estos tokens, ¿para qué esquema de color valen? Salvo el primero, que variaba, los demás no varían, así que serían para ambos. Así que los dejamos así tal cual.



Y podemos a partir de allí especificar que a todos los que tenemos indicados aquí, que son los de los niveles dos y tres que habíamos identificado en nuestra planilla, les daremos valor para ambas opciones.

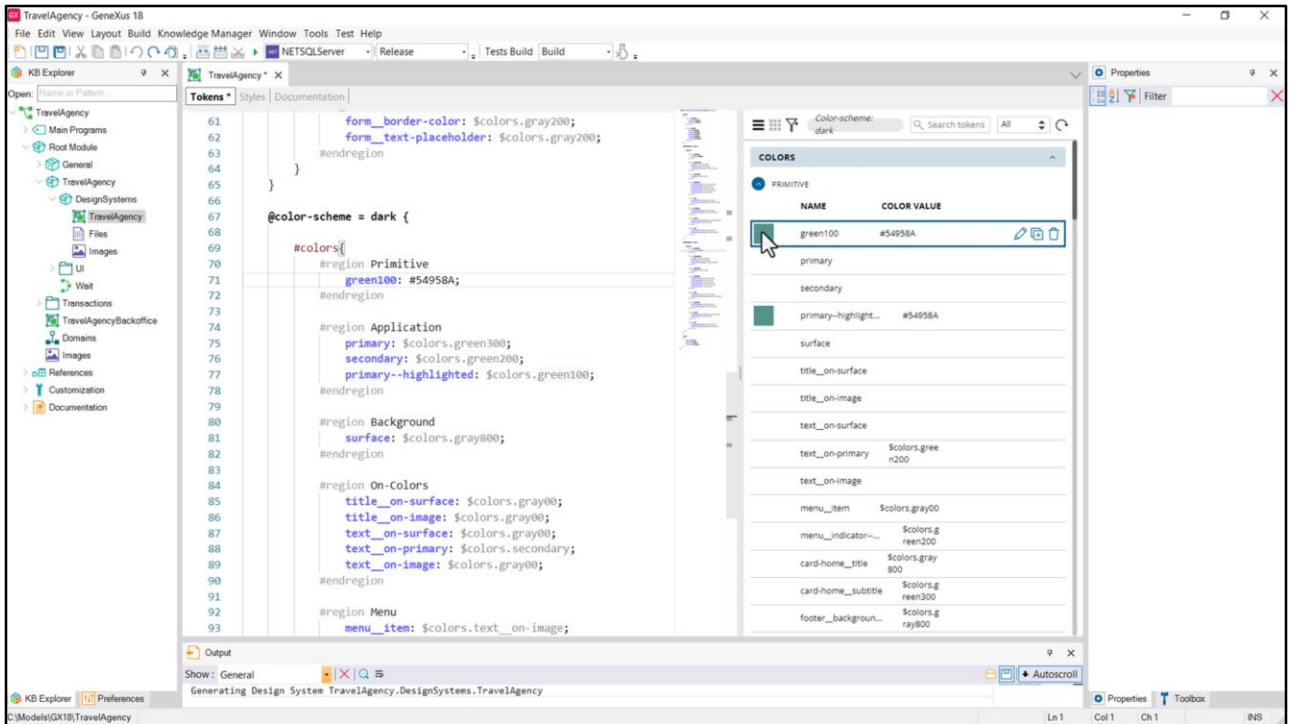
Entonces a todos estos los colocamos como asignaciones para el valor de color-scheme light... así. Vean que tenemos los tokens de color universales, y acá tenemos para la opción light, la definición de este conjunto de tokens de color. Que van a valer, entonces, solamente cuando esa sea la opción elegida.



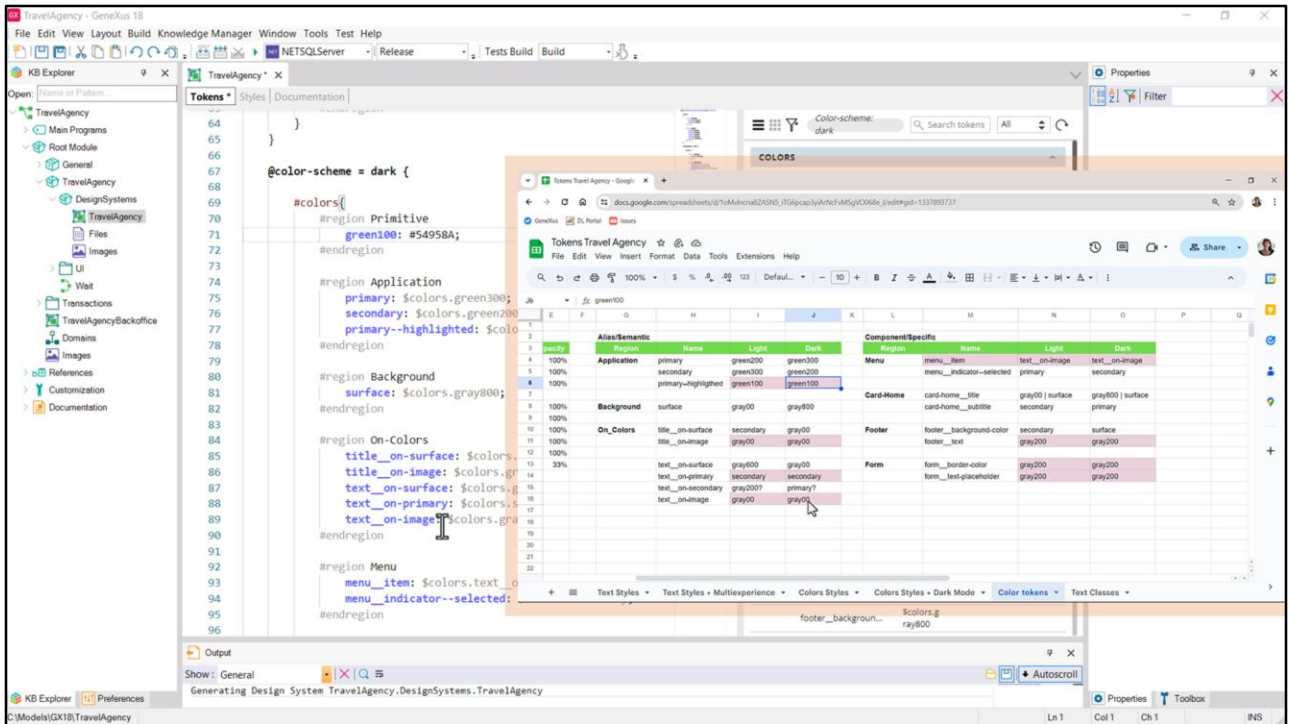
Vamos a agregar, aquí dentro, la región Primitive. Y allí dentro vamos a copiar este token, porque este sabíamos que sí iba a variar por modo. Así que este va a ser el valor del token para el modo light.

Entonces, resumiendo lo hecho hasta ahora: definimos la opción color-scheme, que asume uno de estos dos valores. El valor por default es este, light. Aquí tenemos tokens de color que no varían de acuerdo al modo. Van a corresponder siempre a estos valores. Y aquí estamos definiendo los tokens de color cuando el color-scheme sea light, ¿sí? Estamos indicando, entonces, los valores de todos esos tokens.

¿Qué nos queda? Bueno, decir cuáles van a ser los valores de los tokens cuando el color-scheme sea dark.

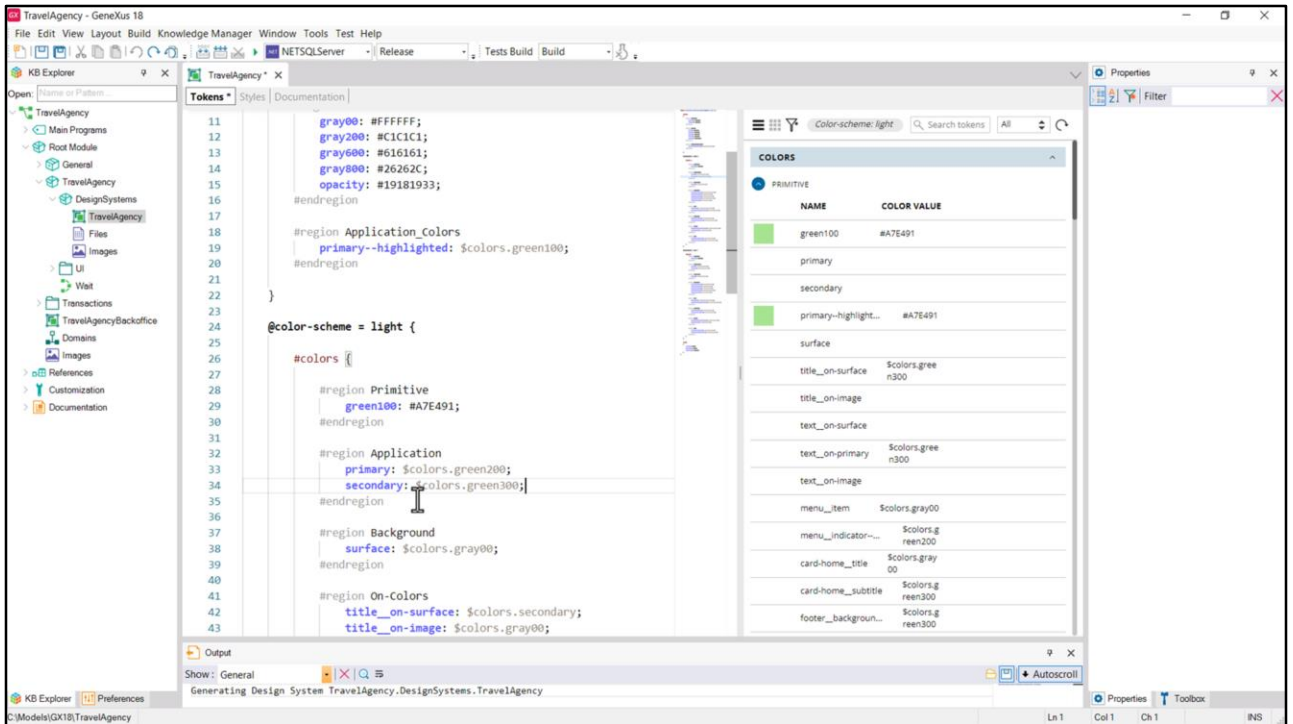


Y eso, entonces, lo coloco aquí debajo. Para el color-scheme dark... cuánto vale el token green100, que vemos que es este otro verde claro, el primary, el secondary, y todos los demás...



...cuyos valores, claramente, tomé de la planilla, de las columnas Dark de cada tablita.

Bueno, y acá se entiende el sentido de haber dejado marcados los tokens para los cuales el valor en el modo light y el dark eran el mismo.

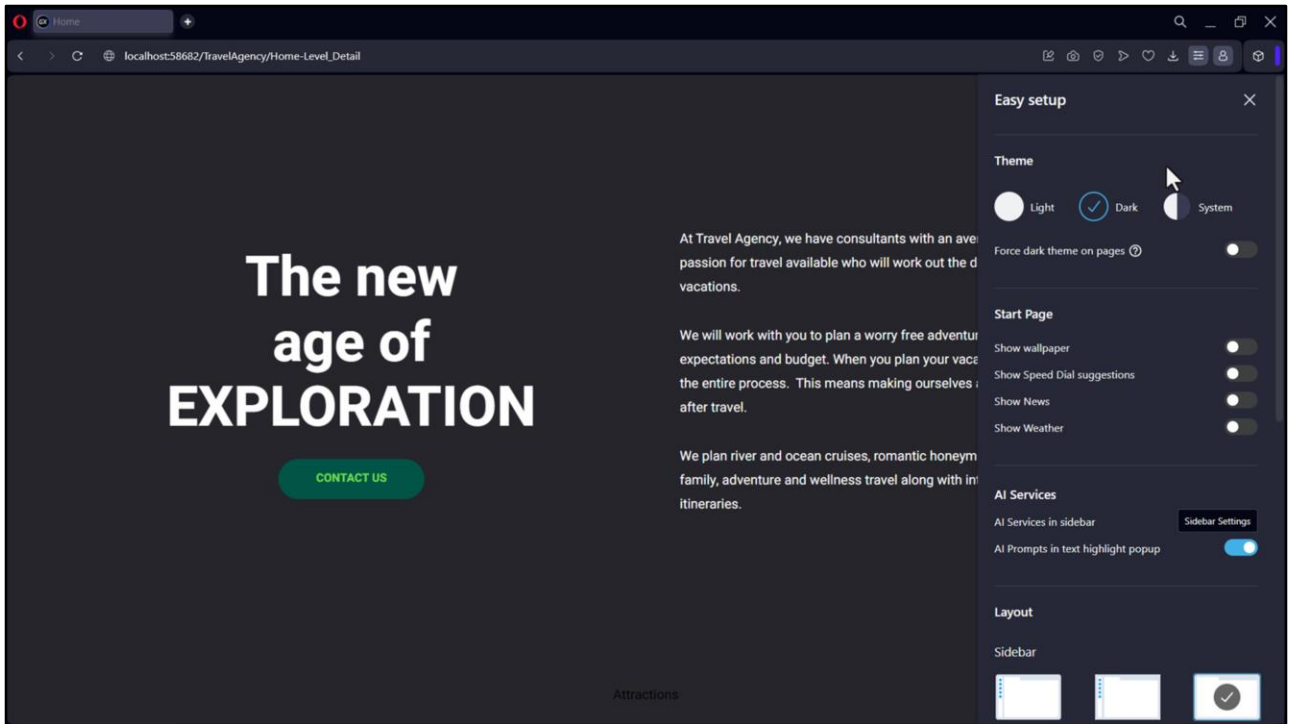


Porque a estos podría definirlos una única vez, al principio, donde tengo los tokens invariables, por color-scheme. Entonces, muevo, por ejemplo, el primary—highlighted para aquí, y lo elimino de las dos secciones, light y dark.

Hacerlo de ese modo tiene sus ventajas: no habrá inconsistencias; y sus desventajas: nos quedan los tokens más desperdigados.

Por ejemplo tendré el primary y el secondary variables por color-scheme, pero en cambio si quiero ir a buscar el primary—highlighted no lo encontraré allí, filtrando por light o dark.

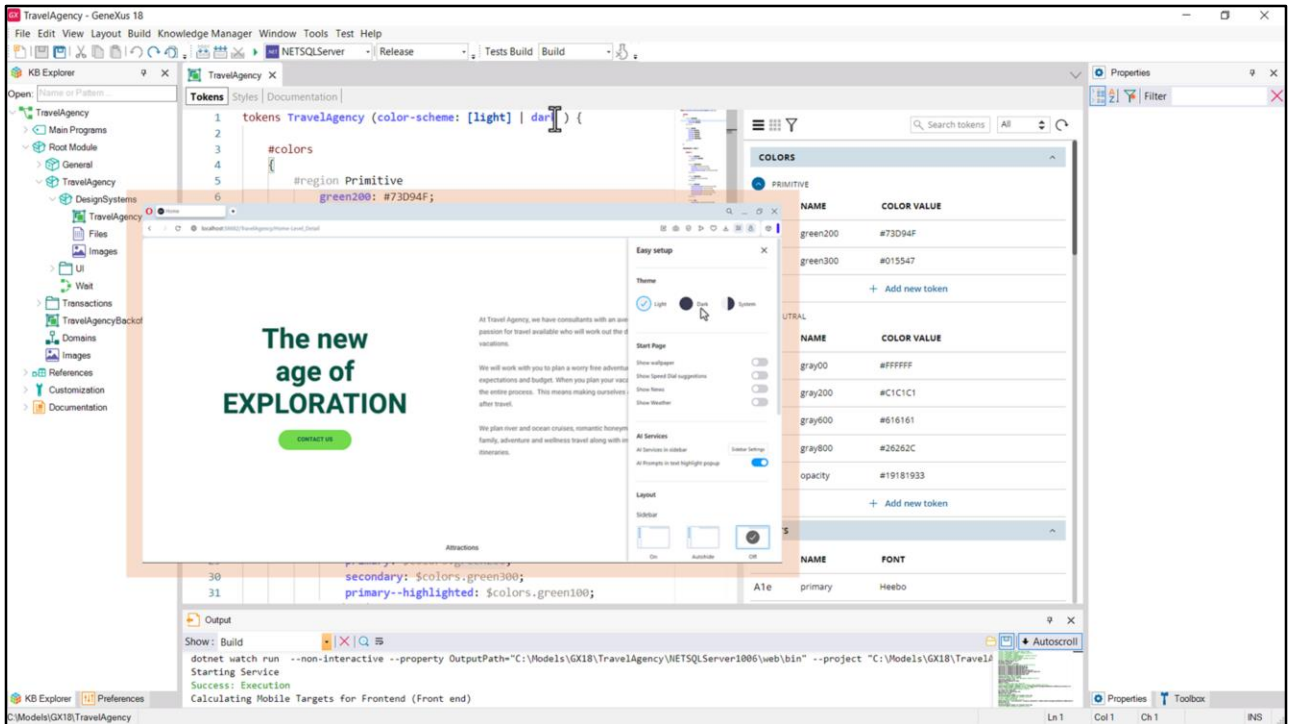
Otra vez, la decisión es un poco a gusto y riesgo del fontender. Yo los voy a dejar por ahora como los tenía.



Bueno, y ahora vamos a probar.

Aquí tenemos el modo Light. Y vemos, si nos cambiamos al Dark, cómo se modificaron los colores, tal como queríamos.

Si ahora volvemos al Light...

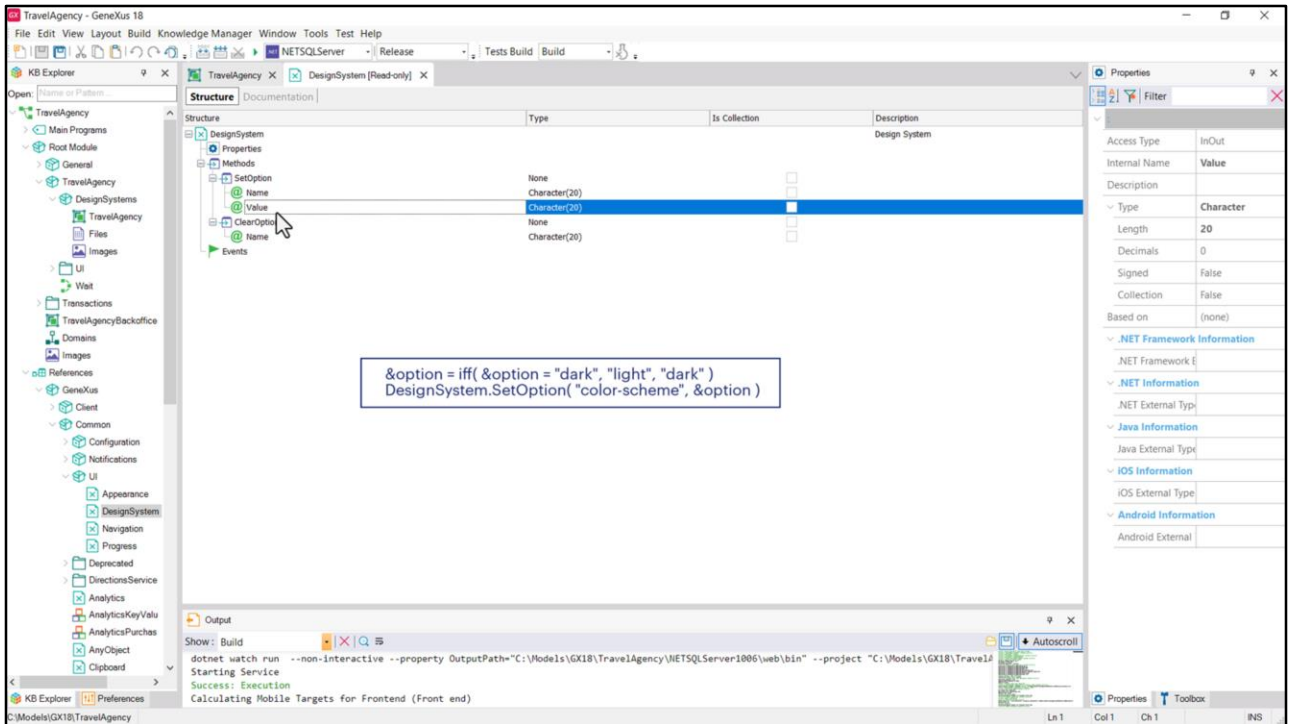


Bueno, llegados a este punto convendría realizar algunas observaciones.

Empecemos por lo más simple: esto que pareció magia, en realidad corresponde a una correcta sistematización donde a las propiedades de color de las clases, tanto background-color como color, les asignamos tokens de color bien semantizados por función. Y a esos tokens de color los hacemos variar por esquema de color: light o dark.

Por otro lado, esto funcionó así sin más, sin que hayamos tenido que especificar nada a nivel programación porque el navegador ya está entendiendo que lo que allí llama theme light y dark se está correspondiendo exactamente con los valores a los que nosotros llamamos light y dark, para la opción a la que llamamos exactamente color-scheme en nuestro DSO.

Esto es por haberle colocado exactamente ese nombre. No es obligatorio, podríamos haberle llamado de cualquier manera a la opción, pero si así lo hubiéramos hecho, el navegador no habría entendido que se trata de lo mismo, y no nos habría cambiado los colores, como lo hizo.



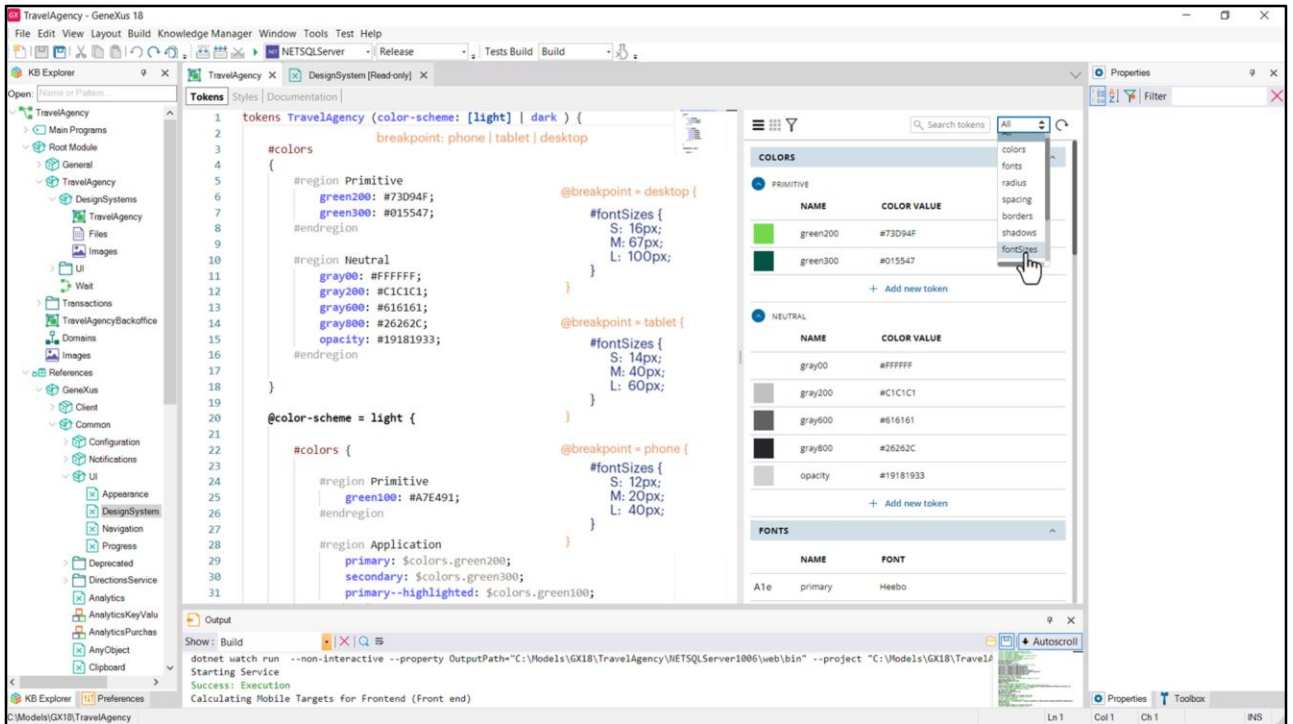
Tenemos, a pesar de eso, una forma de especificar a nivel programático cuál de los valores de la opción queremos que aplique en un momento dado.

Nos será útil, por ejemplo, si queremos brindar al usuario la opción a nivel de la aplicación, y no del navegador, de elegir el esquema de color.

Para ello se cuenta dentro del módulo GeneXus que viene instalado con toda KB, con el objeto externo DesignSystem.

Al abrirlo podemos ver que ofrece un método SetOption que es justamente para setear para una opción en particular (en nuestro caso la única que tenemos es la color-scheme) uno de sus valores posibles.

Así, por ejemplo, podríamos colocar este código en el evento asociado a un control del layout que sea para modificar el color-scheme. Si en la variable que insertamos, option, había valor "dark" se cambia a "light" y viceversa (eso es lo que hace el comando iff), y luego se setea la opción "color-scheme" del DSO asociado al objeto.



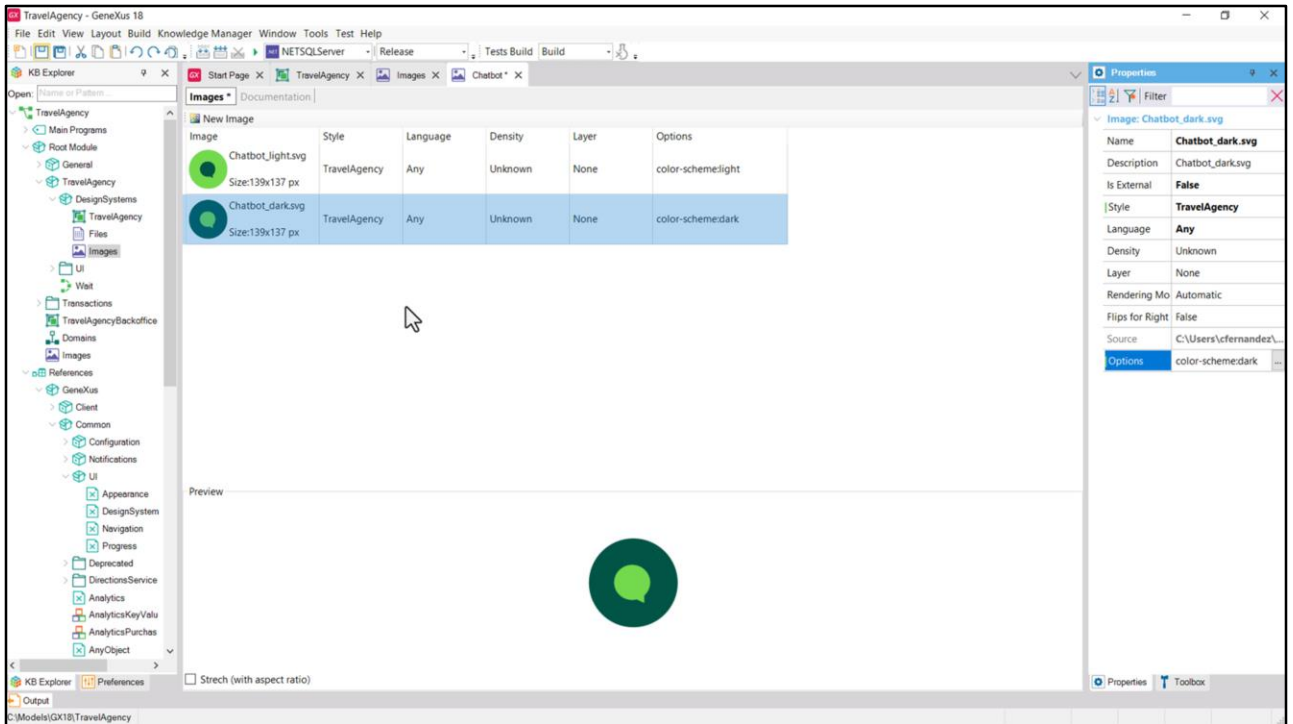
Por supuesto, esto significa que podemos tener por DSO varias opciones, no solamente una, ni tiene por qué ser únicamente para poder variar los tokens de color. Por ejemplo, podríamos agregar una opción, además de esta, separándola con coma, o en lugar de esta, que sea para variar tokens por tamaño de pantalla.

Pensemos, por ejemplo, si tuviéramos definidos tokens de espaciado o de tamaño de fuentes. Por ejemplo imaginemos que tuviéramos tokens S, M, L para tamaños de fuentes. Para los distintos tamaños, Small, Medium y Large.

Podríamos querer variar sus valores de acuerdo al tamaño de pantalla, para lo que podemos definir una opción breakpoint que asuma los valores phone, tablet y desktop, y a partir de allí definir que para el breakpoint Desktop estos sean los valores de esos 3 tokens, pero para los otros breakpoints sean estos otros.

¿Nos convendrá utilizar tokens para los tamaños de fuentes en nuestro sistema?

Analizaremos eso en el próximo video.

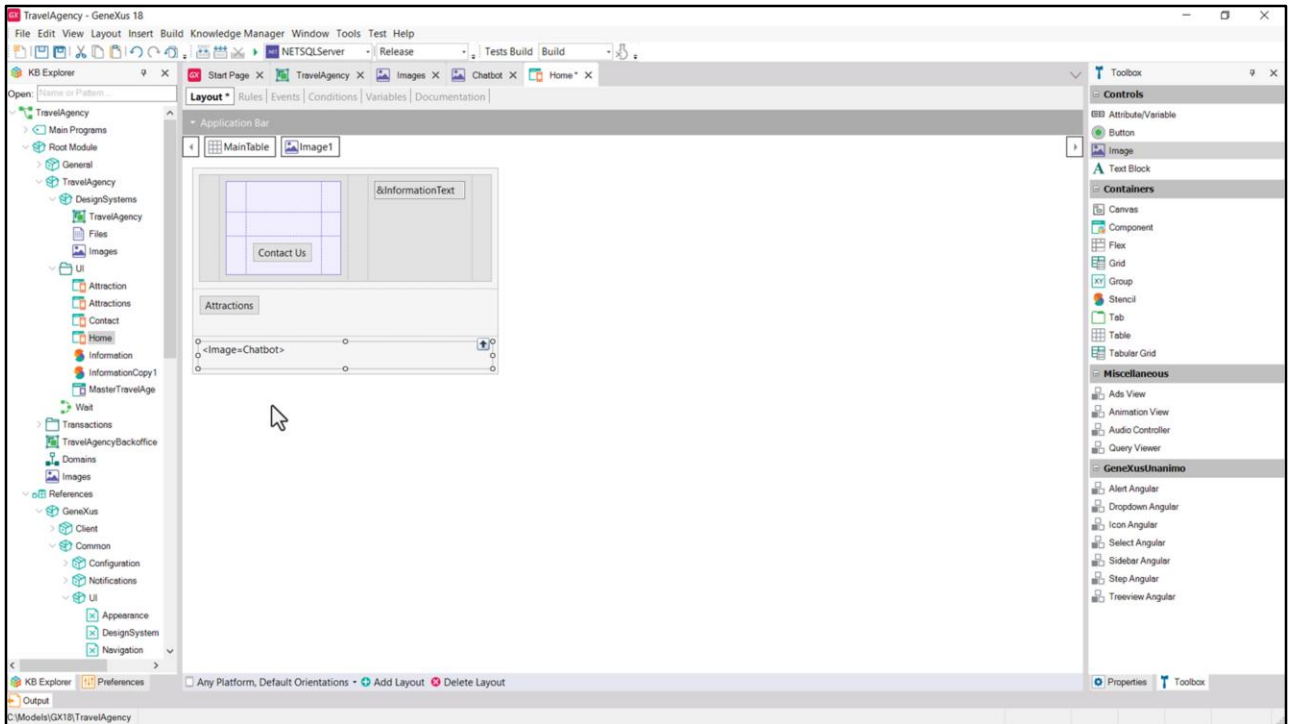


Lo que me queda ahora por decir, es que a partir de la aparición de esta opción en nuestro DSO, podemos entender esa dimensión que aparecía a nivel de las imágenes.

Si yo ahora le asocio a ambas variaciones como Style nuestro DSO... empezamos por esta primera...vemos aparecer la propiedad Options. Que me ofrecerá darle valor a cada una de las opciones que ese DSO tenga especificadas. En nuestro caso por ahora solamente tenemos una, la color-scheme.

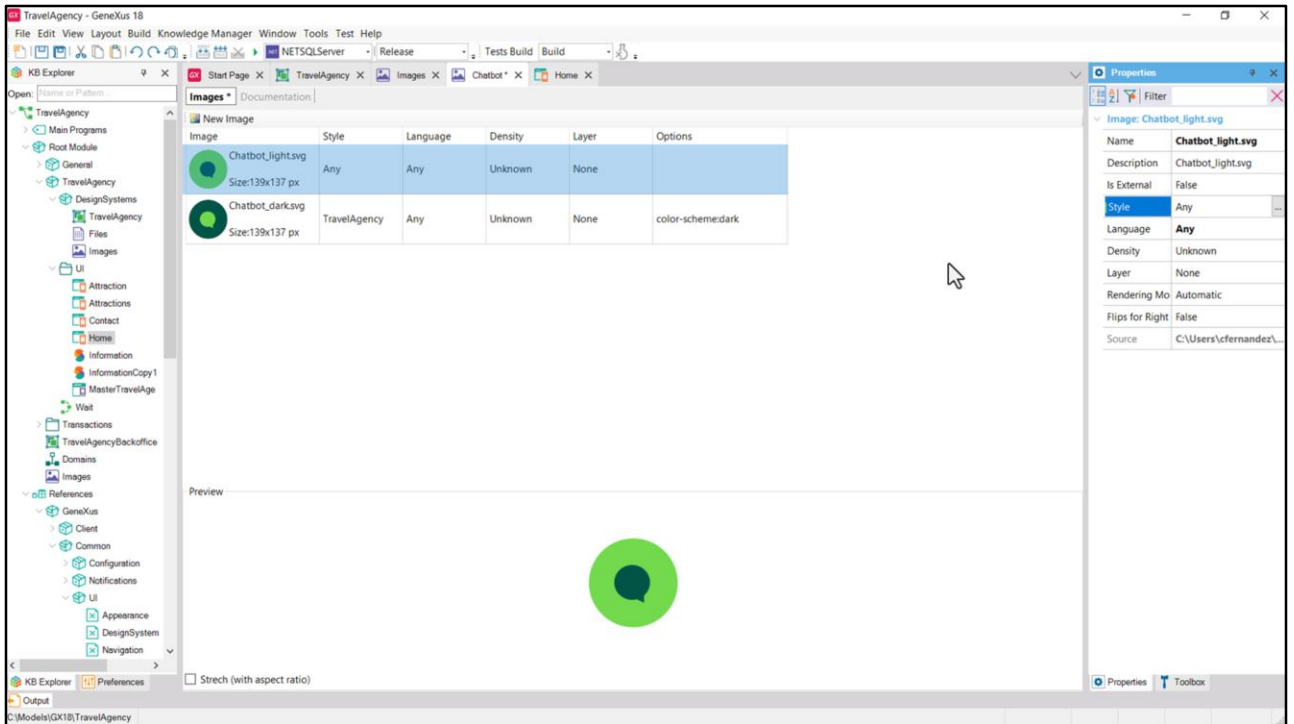
Entonces diremos que la imagen de Chatbot corresponderá a esta variación solo si el color-scheme es Light.

Y en cambio a esta otra, si el DSO es TravelAgency también, pero con color-scheme Dark.

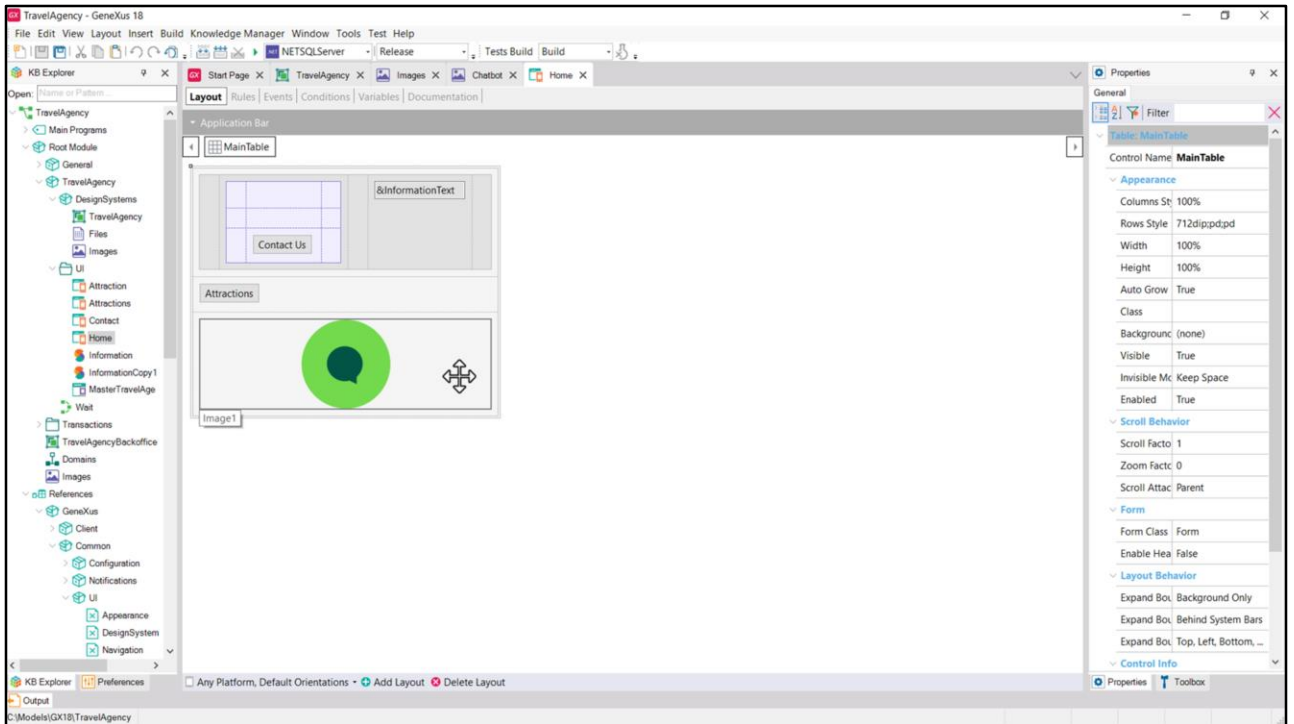


Aún no implementamos el Master Panel, que será donde tendremos el botón con esta imagen, pero podríamos, solo para probar esto en ejecución, insertar una imagen suelta aquí...

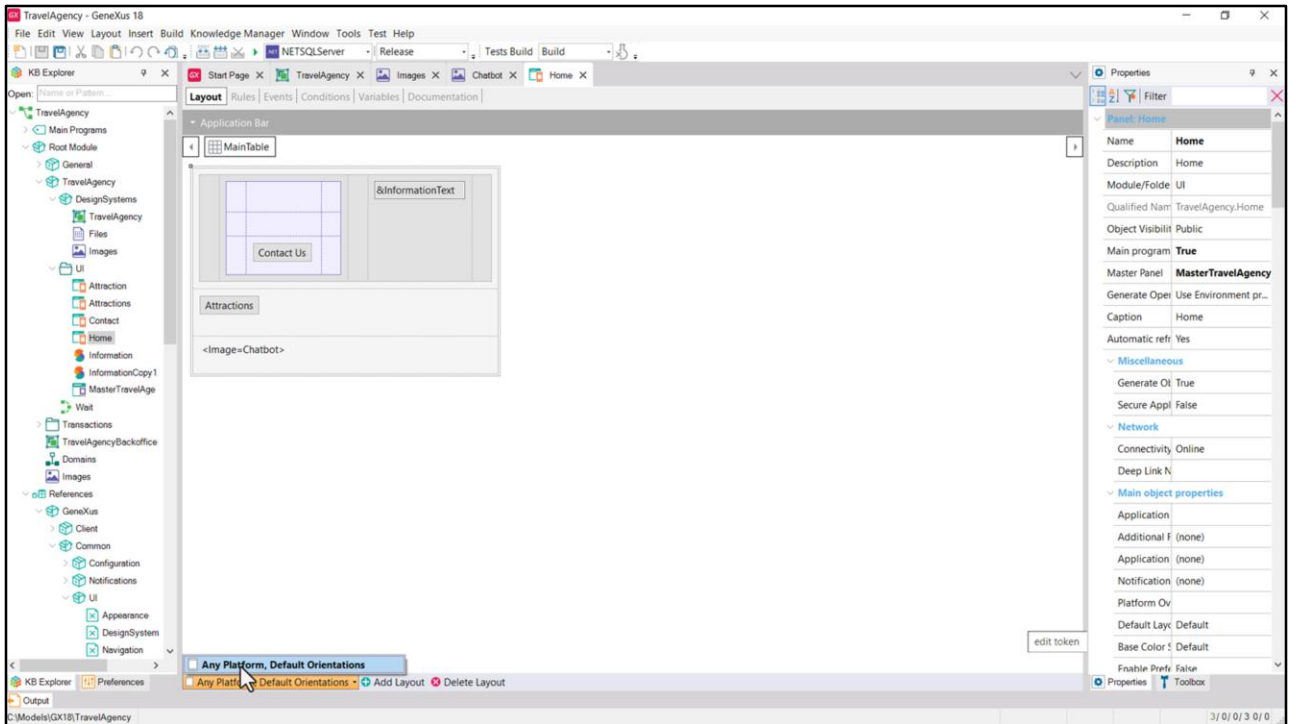
No nos está mostrando la imagen que se llama Chatbot, no la vemos... dice que esa es la imagen que se renderizará, pero visualmente estamos a ciegas. Y grabar no cambia nada.



A ver qué pasa si le quito a una de las variaciones de la imagen el Style... le dejo el valor Any, para que valga para cualquier DSO.... Grabo...

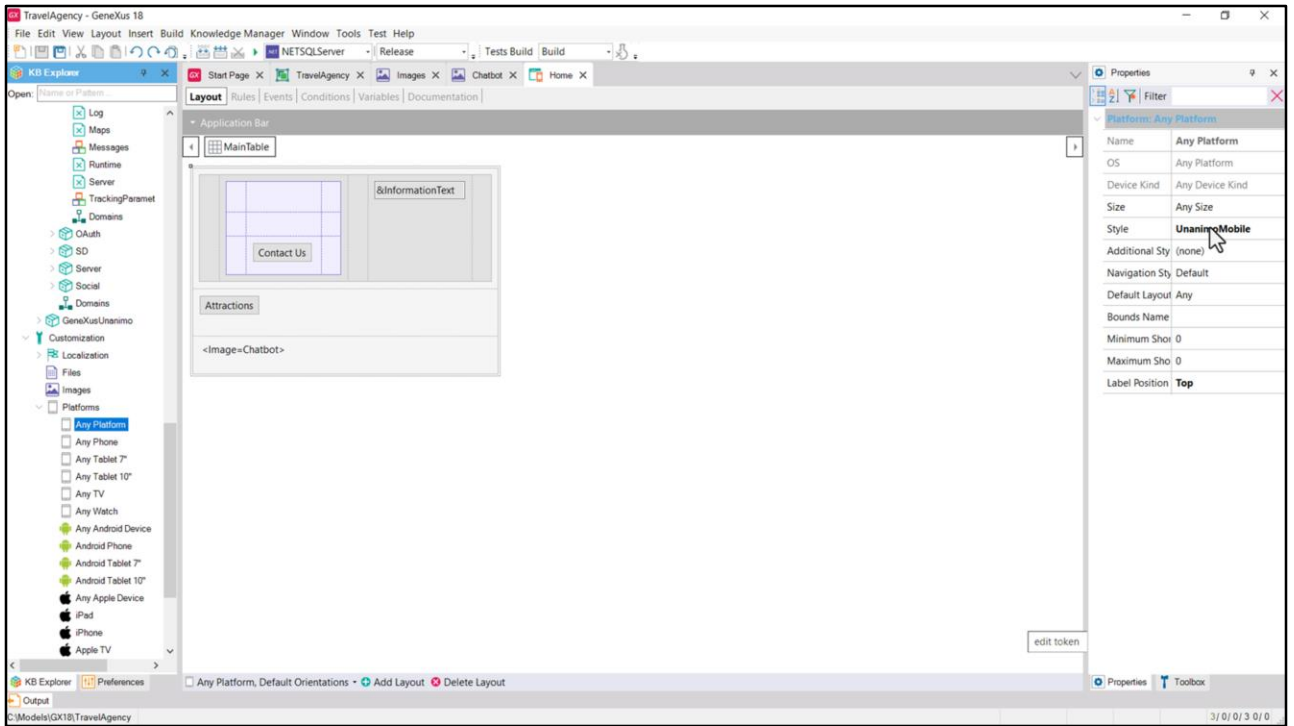


... y ahora vuelvo al panel, lo cierro y lo vuelvo a abrir. ¡Y ahora la veo!

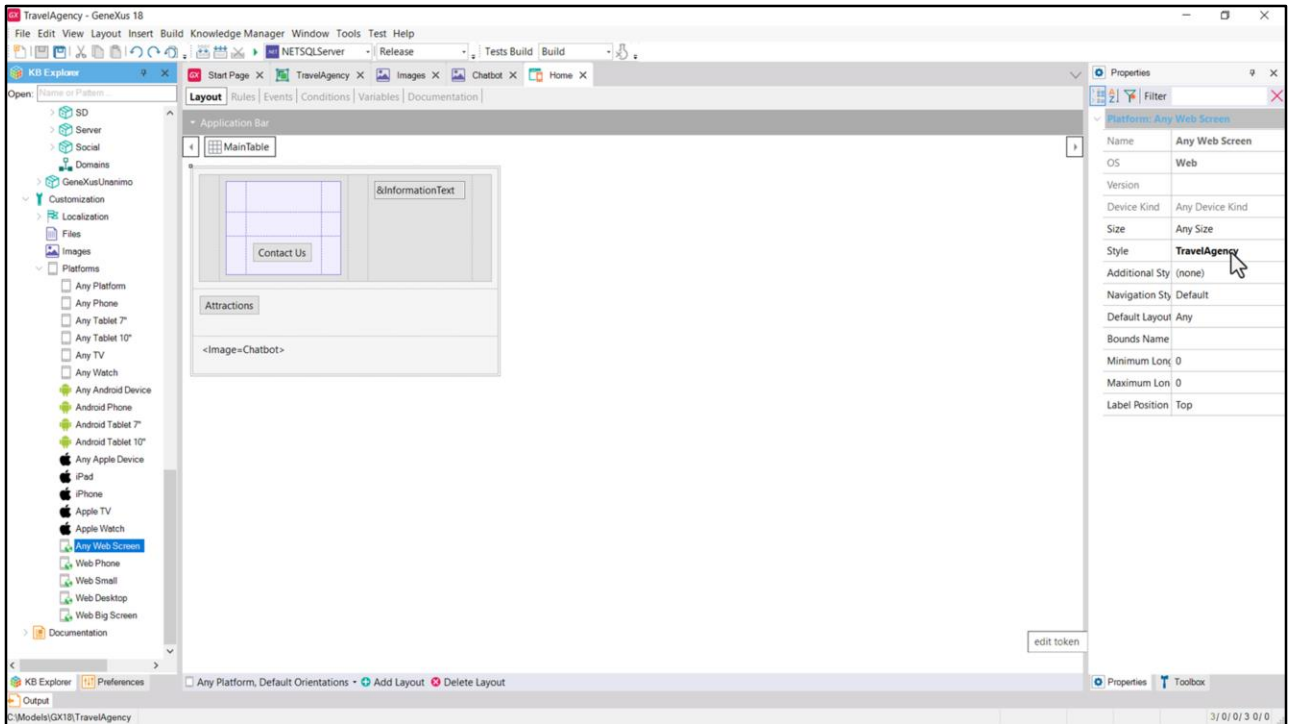


¿Y si vuelvo a restringirla solo al DSO y en modo Light?... Cierro y abro otra vez, ¡y otra vez!

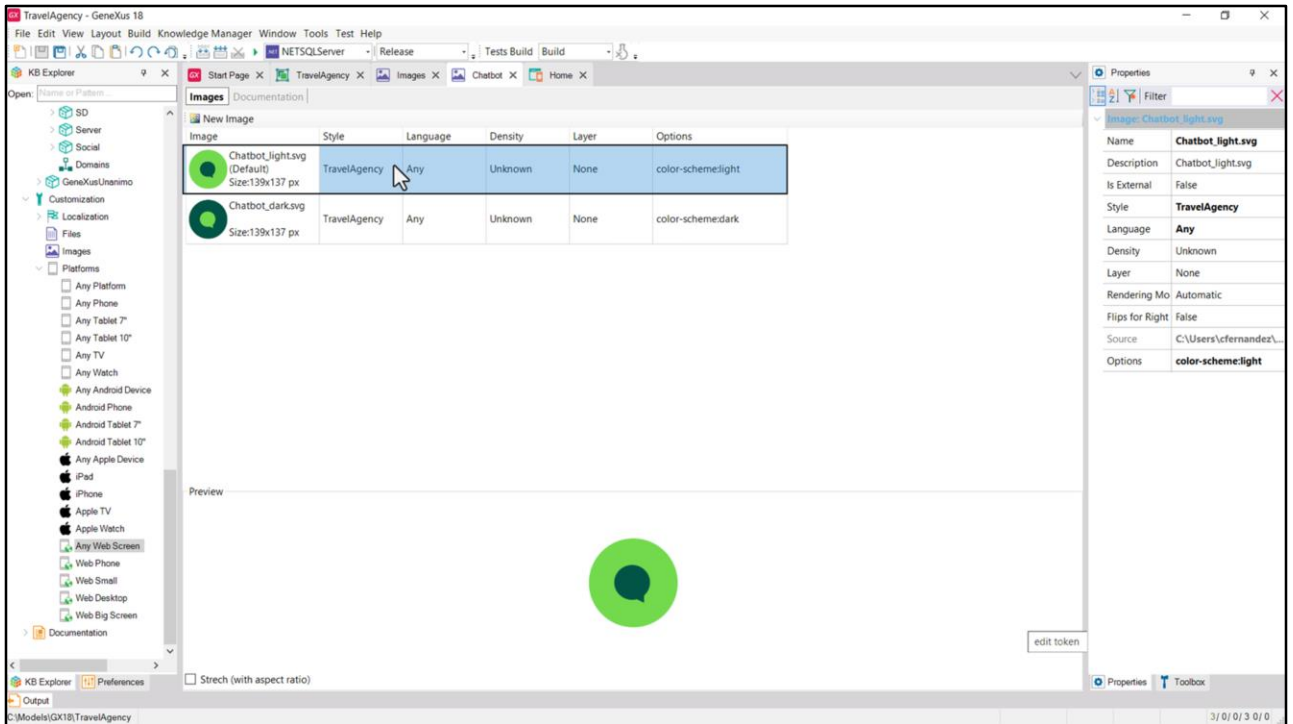
Bueno, esto tiene sentido porque este objeto Panel vale para toda plataforma, no solamente para la Angular que estamos desarrollando.



Y si venimos a ver los DSOs asociados al Any Platform, vemos que es Unanimomobile.

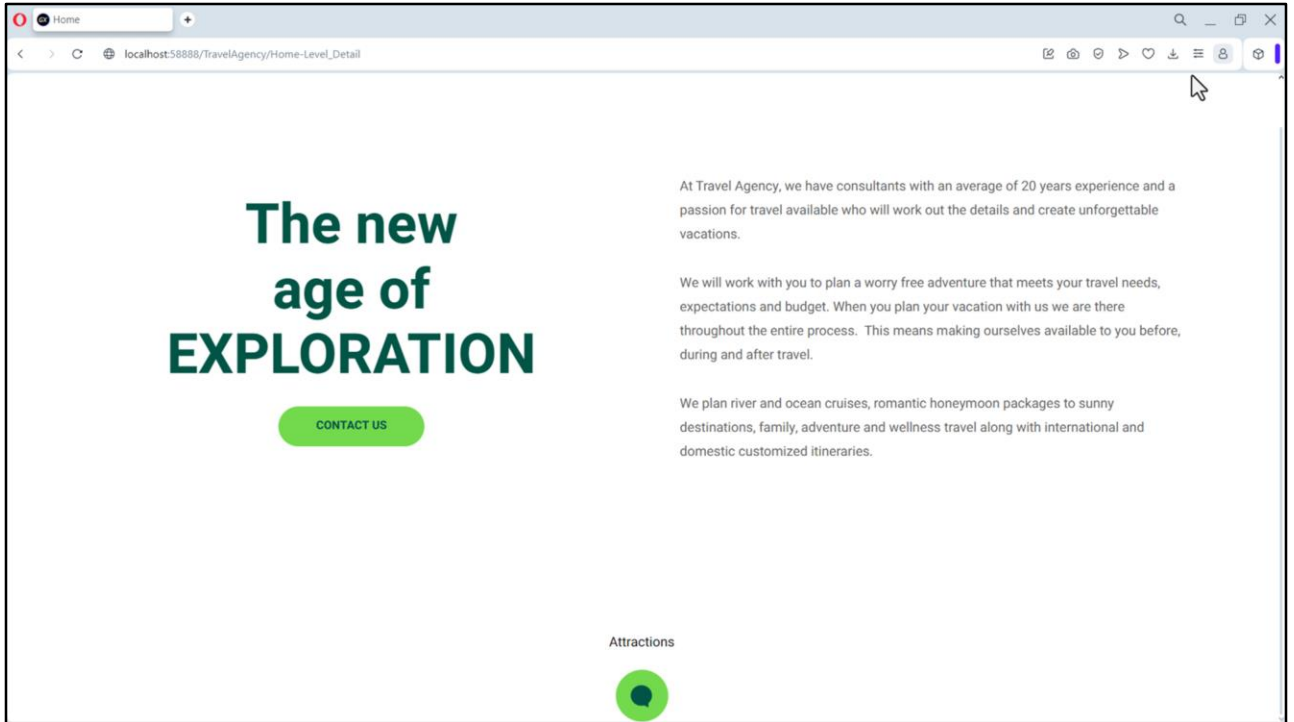


Que no es el TravelAgency que teníamos acá.

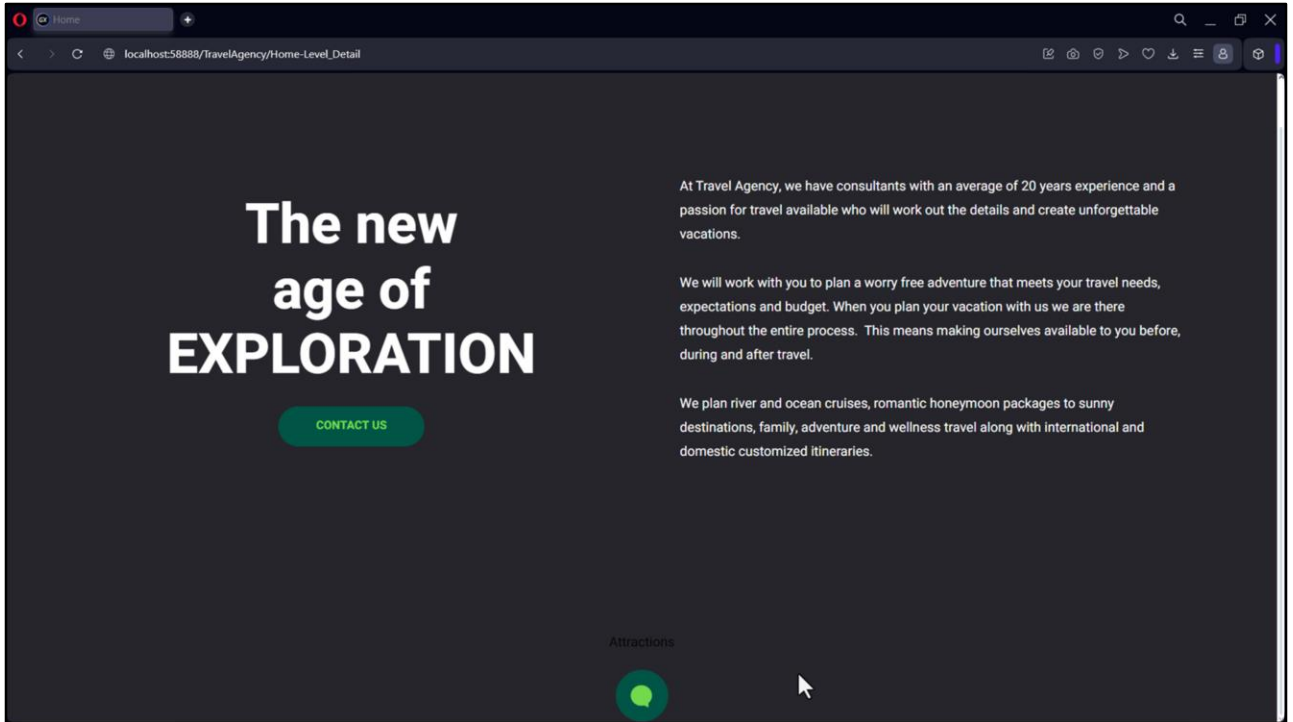


Y lo que estamos especificando es que tome uno de estos dos valores sólo para este Design System Object. No tenemos especificado el valor para otro Design System Object.

Sin embargo observemos qué pasa igual aunque esto aparezca de esta manera, cuando ejecutamos para Angular...



Bueno, acá vemos la imagen correcta para light...



... y a ver qué pasa en modo dark... ¡muy bien!

Vemos que está realizando, tal como esperábamos, las variaciones.

Bueno, seguimos en el próximo video.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com