

Color system in GeneXus



Cecilia Fernández

Color system

The screenshot displays a design system tool interface for 'TravelAgency'. The left pane shows a JSON configuration for tokens, and the right pane shows a visual representation of the color system.

```
tokens TravelAgency {  
  #colors  
  #region Application  
    primary: #73D94F;  
    secondary: #015547;  
    primary--highlighted: #A7E491;  
  #endregion  
  #region Neutral  
    gray00: #FFFFFF;  
    gray200: #C1C1C1;  
    gray600: #616161;  
    opacity: #19181933;  
  #endregion  
  #fonts  
  {  
    primary: Heebo;  
    secondary: Rubik;  
    additional: Graphik;  
  }  
}
```

The right pane, titled 'COLORS', shows 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 a similar 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.

No vídeo anterior havíamos partido destes tokens definidos em nosso Design System, que representavam de forma muito básica o sistema de cores de nossa aplicação, e havíamos chegado, no final do vídeo, com um sistema de cores muito mais complexo, por um lado, mas ao mesmo tempo funcionalmente muito mais preciso, muito mais semântico.

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

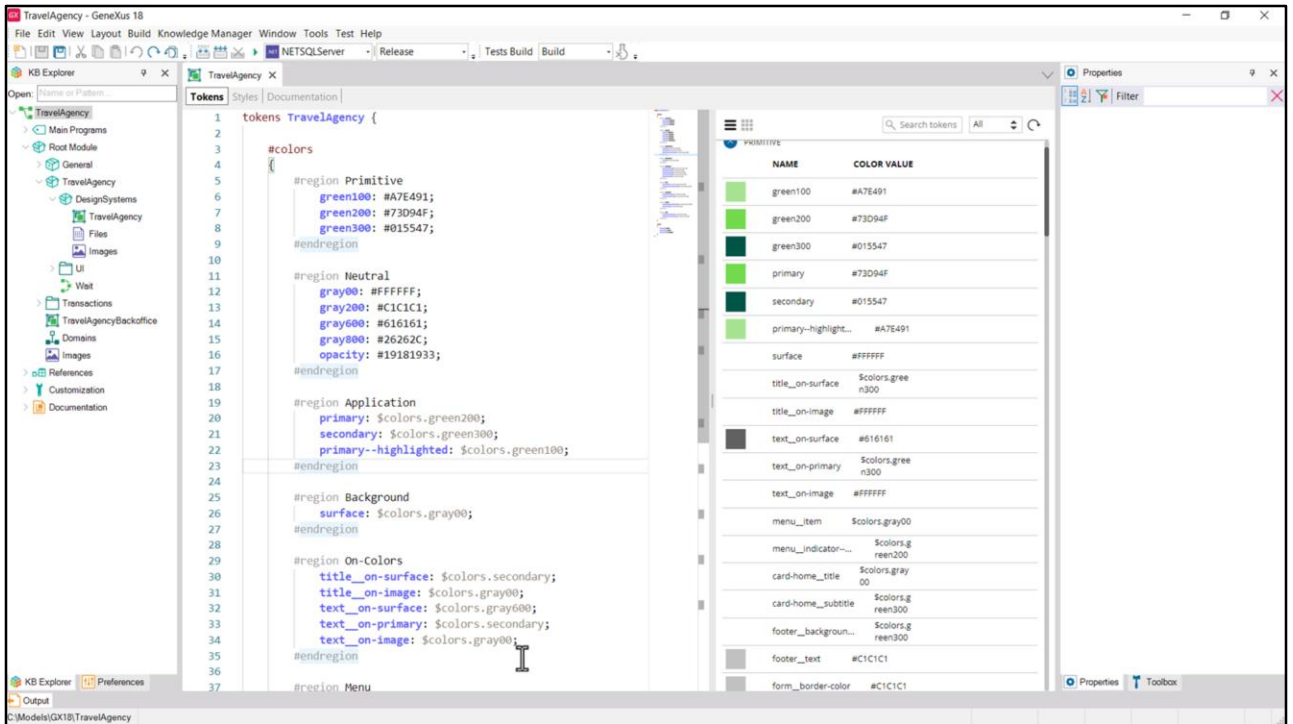
	C	D	E	F	G	H	I	J	K	L	M	N
1												
2												
3		Light Value	Value Dark	Opacity	Alias/Semantic				Component/Specific			
4		#A7E491	#54958A	100%	Region	Name	Light	Dark	Region	Name	Light	
5		#73D94F		100%	Application	primary	green200	green300	Menu	menu_item	text_on-image	text_o
6		#015547		100%		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?				
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 o que tínhamos nesta planilha.

	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									Hero	hero_title	title_on-image	title_on-image	
20													
21									Banner	banner_background-color	primary	primary	
22										banner_title	text_on-primary	text_on-primary	
23										banner_text	text_on-primary	text_on-primary	
24													
25													
26													

Havíamos dito que estes tokens, estes marcados assim, que davam mais especificidade ao sistema, ao mesmo tempo que davam mais complexidade, então nesta ocasião não irei utilizá-los. Eu os movo aqui para baixo para não os perder.

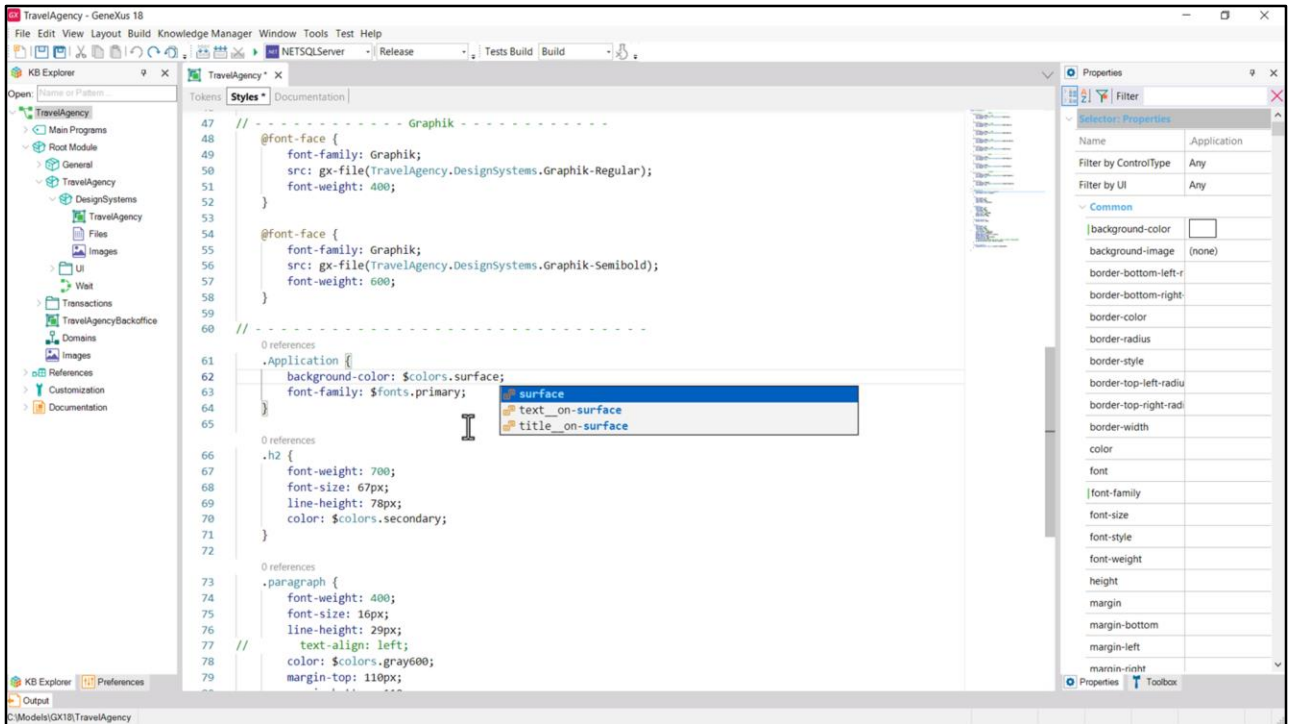


Bom, aqui tenho nosso DSO já com todos os tokens indispensáveis para modelar o sistema, e seus valores para o modo Light, unicamente.

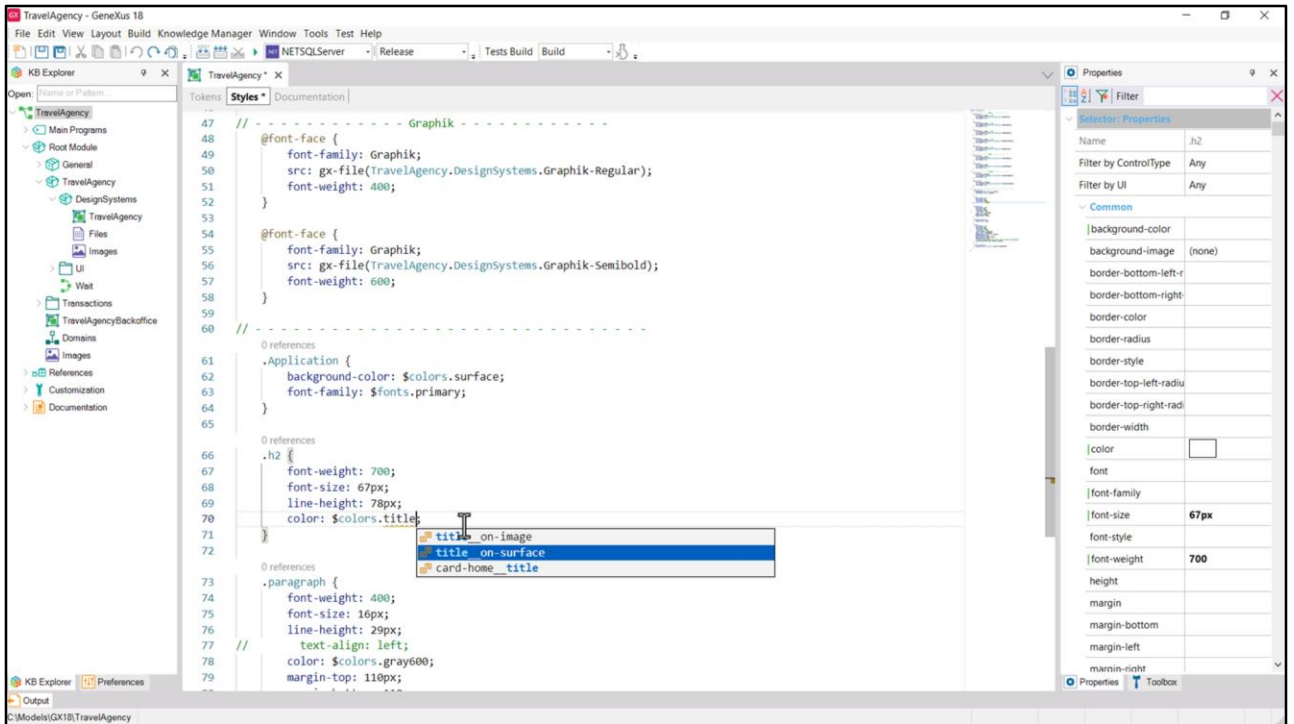
Vemos que aqueles que colocamos na região Primitive e Neutral possuem valores absolutos, em hexadecimal, enquanto todos os demais possuem como valores, referências a outros tokens.

Até aqui, coisas que mais ou menos já sabíamos. Resta ver como representamos o modo Dark.

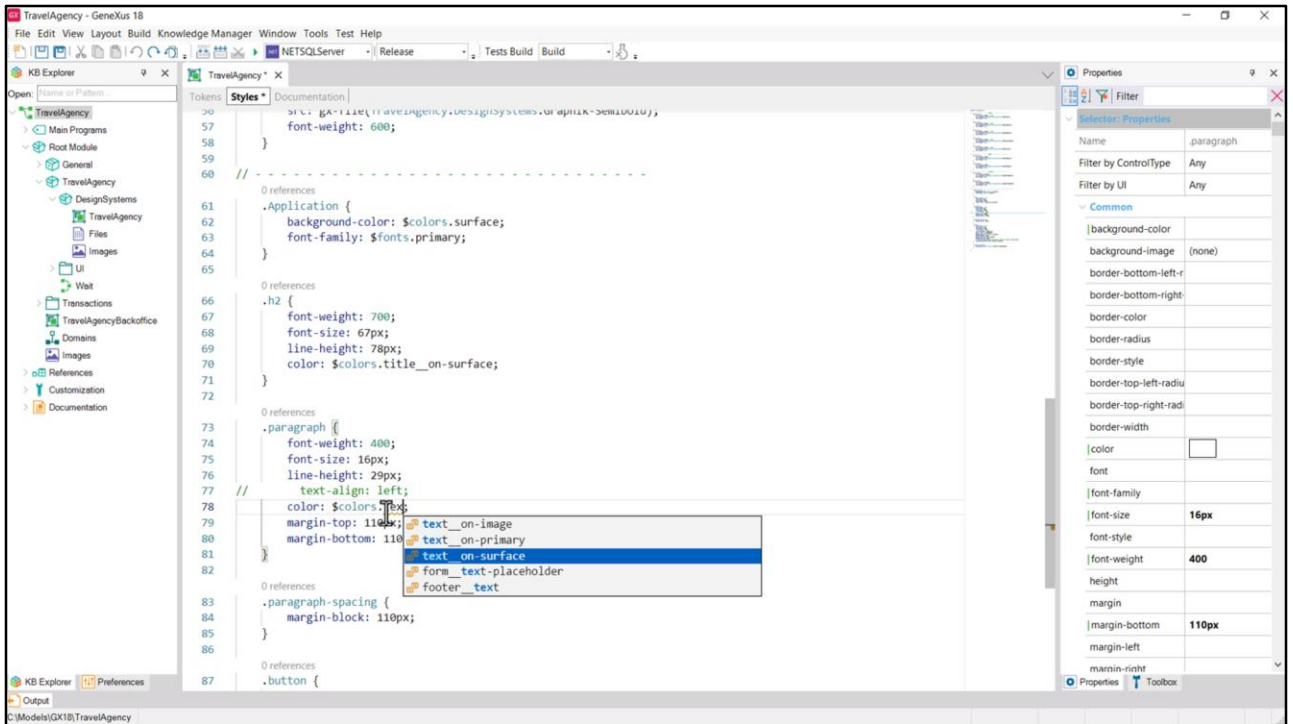
Mas antes... vamos modificar as cores que usamos nas classes, para que utilizem os novos tokens. Que cores tínhamos?



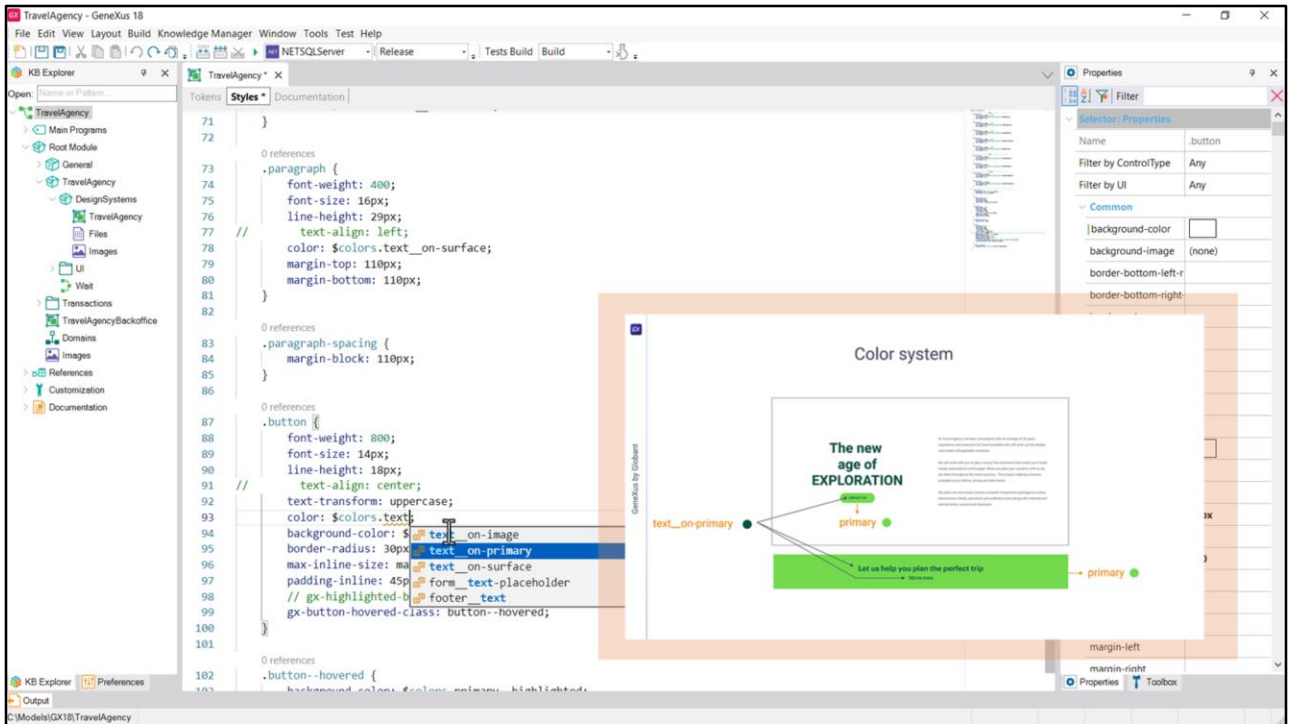
A background-color das telas, que tínhamos colocado dentro da classe Application, agora corresponderá ao token de cor surface...



A classe `h2` era a do título sobre a superfície `Surface`, tem como cor o token `title__on-surface`.



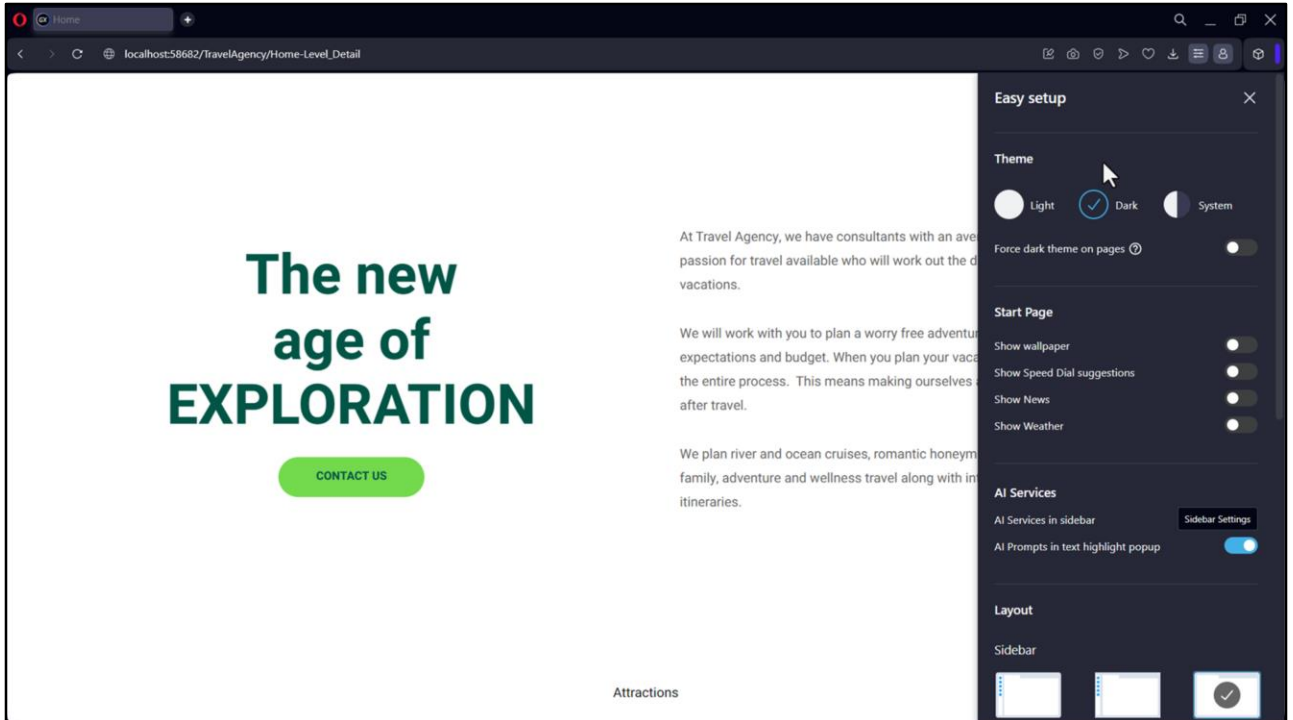
A classe paragraph correspondia ao texto comum sobre a superfície, então aqui substituímos pelo token de cor text__on-Surface.



E para o botão, temos a background-color correta, pois é a primary como cor de fundo.

E a cor deveria ser text__on-primary.

E com a ação de hover sobre o botão, a cor de fundo muda para a primary—highlighted, o que está correto.



Executamos para verificar que tudo fique tal qual esperamos.

Vou copiar a url para executar a aplicação no navegador Opera, que me permite alternar entre os modos Light e Dark com mais facilidade.

Vemos que se eu escolher o modo Dark, o que vemos é exatamente o mesmo que em modo Light.

The new age of EXPLORATI

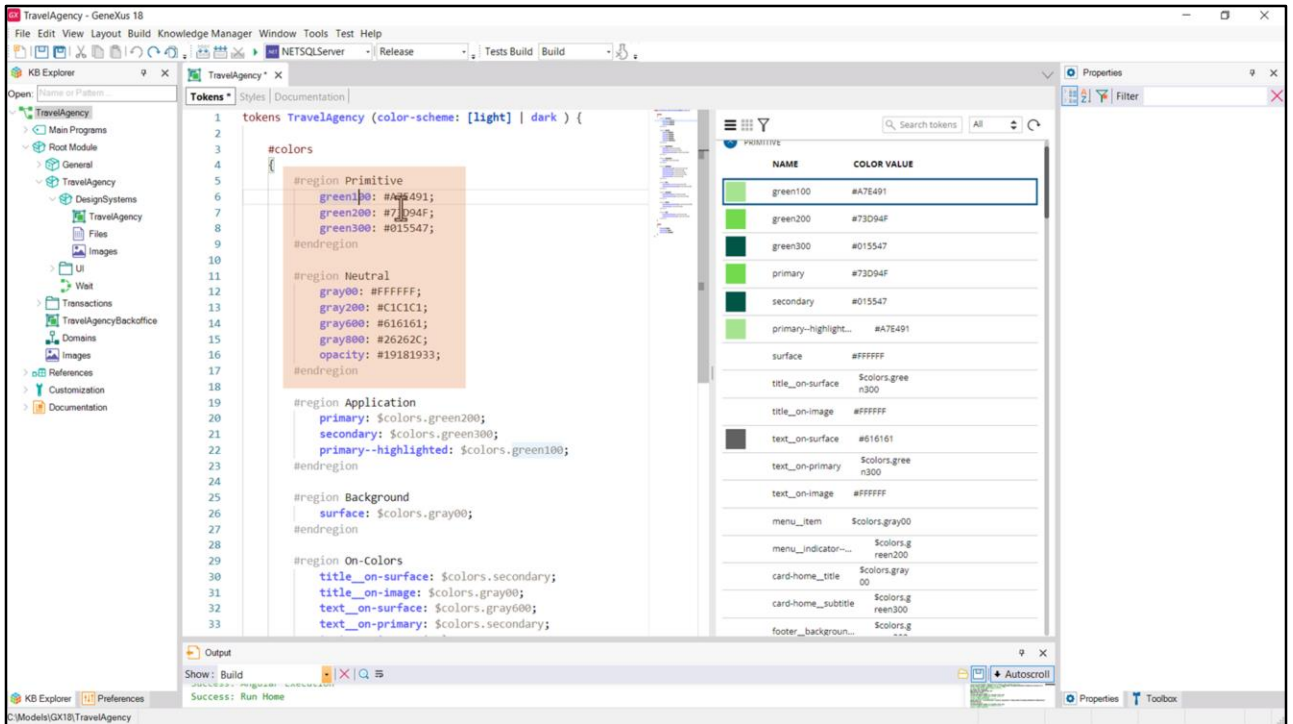
CONTACT US

Category	Alias/Semantic	Region	Light	Dark	Component/Specific	Region	Light	Dark
100%	Application	primary	green200	green300	Menu	menu_item	text_on-image	text_on-image
100%		secondary	green300	green200		menu_indicator--selected	primary	secondary
100%		primary--highlighted	green100	green100				
100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00 surface	gray800 surface
100%						card-home_subtitle	secondary	primary
100%	On_Colors	title_on-surface	secondary	gray00	Footer	footer_background-color	secondary	surface
100%		title_on-image	gray00	gray00		footer_text	gray200	gray200
100%								
33%		text_on-surface	gray00	gray00	Form	form_border-color	gray200	gray200
		text_on-primary	secondary	secondary		form_text-placeholder	gray200	gray200
		text_on-secondary	gray200?	primary?				
		text_on-image	gray00	gray00				

Attractions

Bem, não vamos atrasar mais a pergunta que estamos querendo responder desde o início: como faço para especificar as alterações nos tokens de cor para poder fazê-los variar por modo?

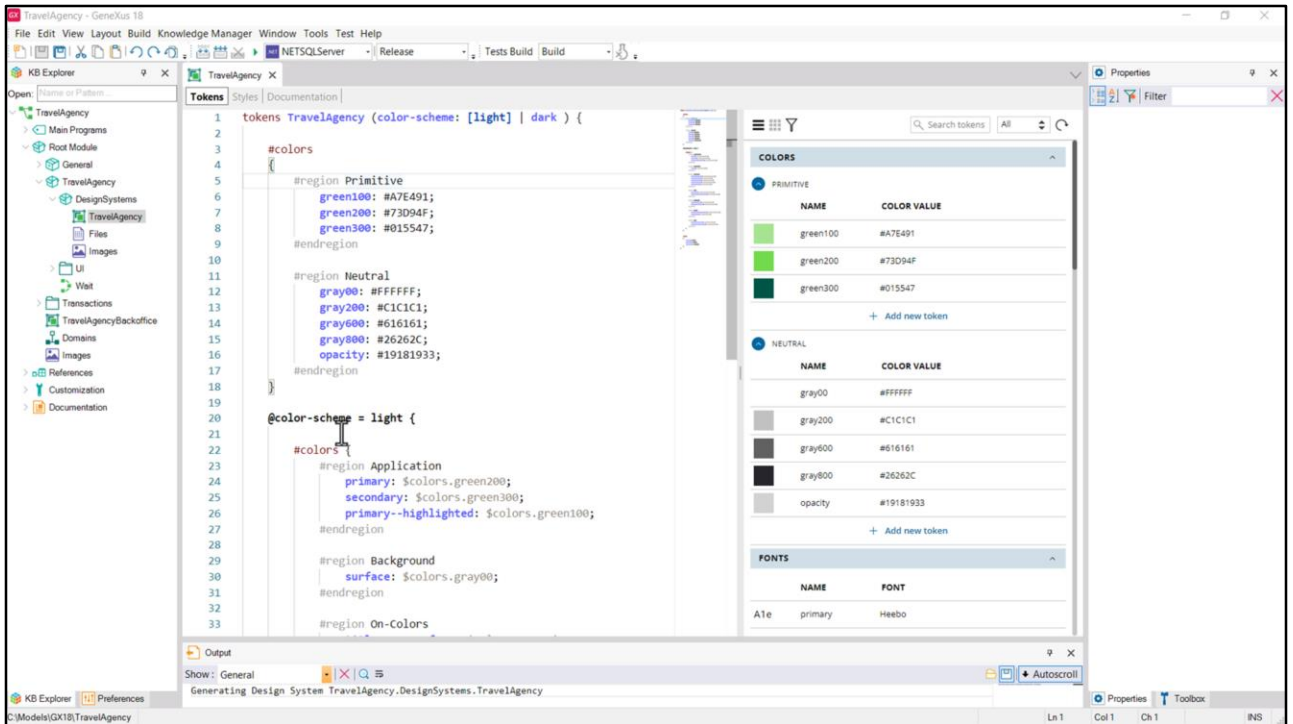
Ali aparecem o que chamamos de options, opções do conjunto de tokens.



Para este conjunto de tokens chamado TravelAgency, podemos especificar variações e fazemos isso entre parênteses. No nosso caso queremos fazer variar alguns dos tokens de cor de acordo com o esquema de cor, color-scheme se chamará nossa opção, que poderá assumir dois valores: light e dark. Podemos especificar com colchetes qual será a opção default.

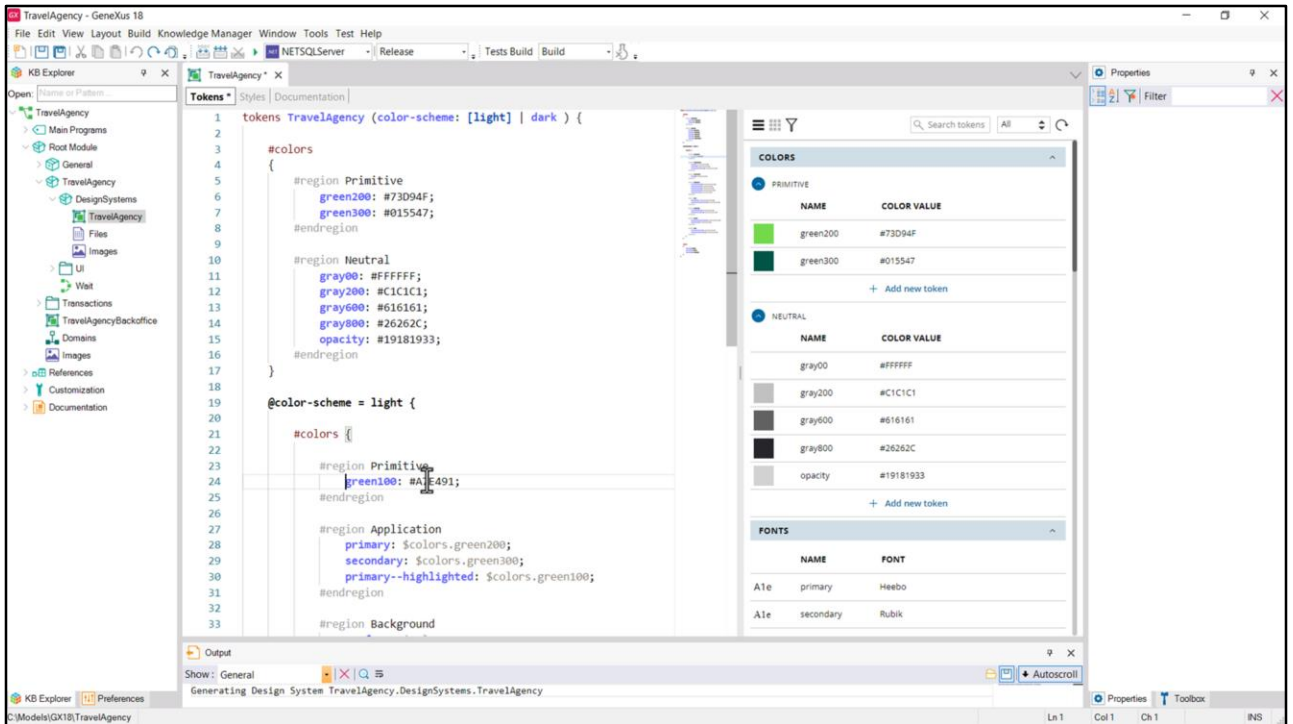
E então nos resta especificar quais valores adquirem os tokens, para um valor da opção e para o outro.

Por exemplo, os valores desses tokens, para qual esquema de cor valem? Exceto o primeiro, que variava, os demais não variam, então seriam para ambos. Então vamos deixá-los assim como está.



E podemos a partir dali especificar que a todos que temos indicados aqui, que são aqueles dos níveis dois e três que havíamos identificado em nossa planilha, lhes daremos valor para ambas as opções.

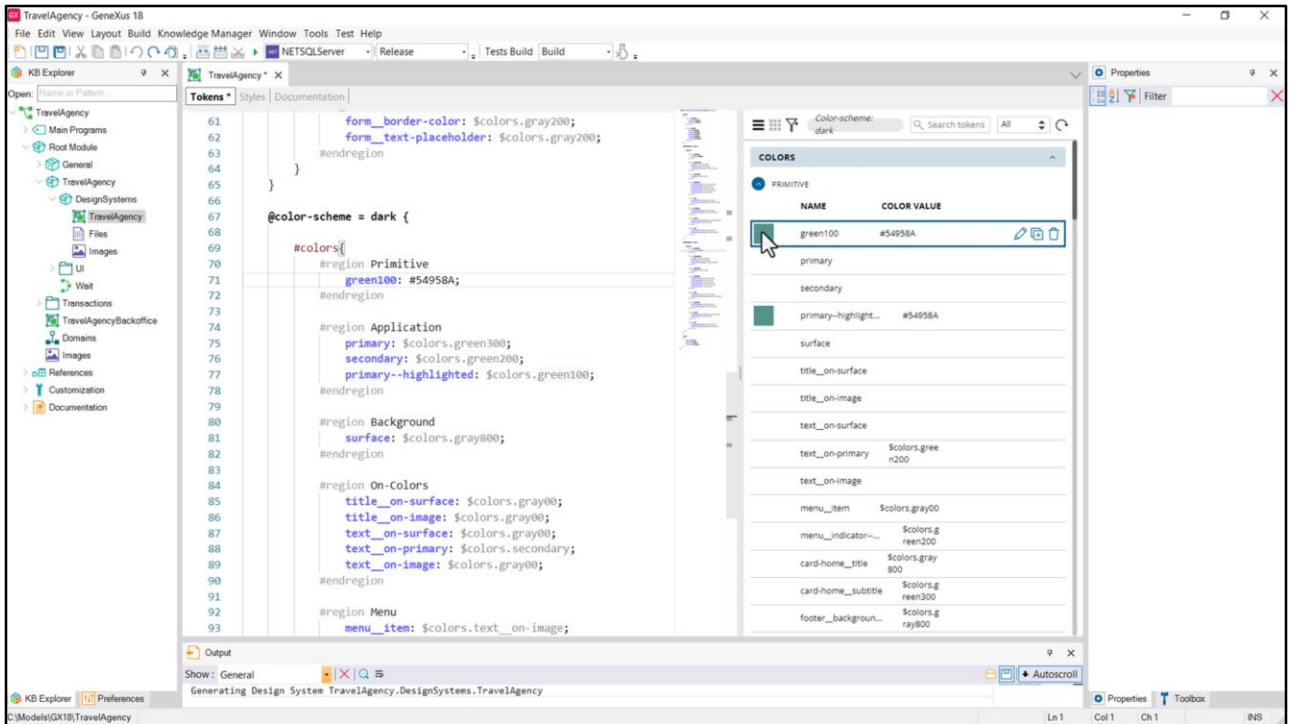
Então, a todos estes, colocamos como atribuições para o valor de color-scheme light... assim. Vejam que temos os tokens de cor universais, e aqui temos para a opção light, a definição deste conjunto de tokens de cor. Que serão válidos, então, somente quando essa for a opção escolhida.



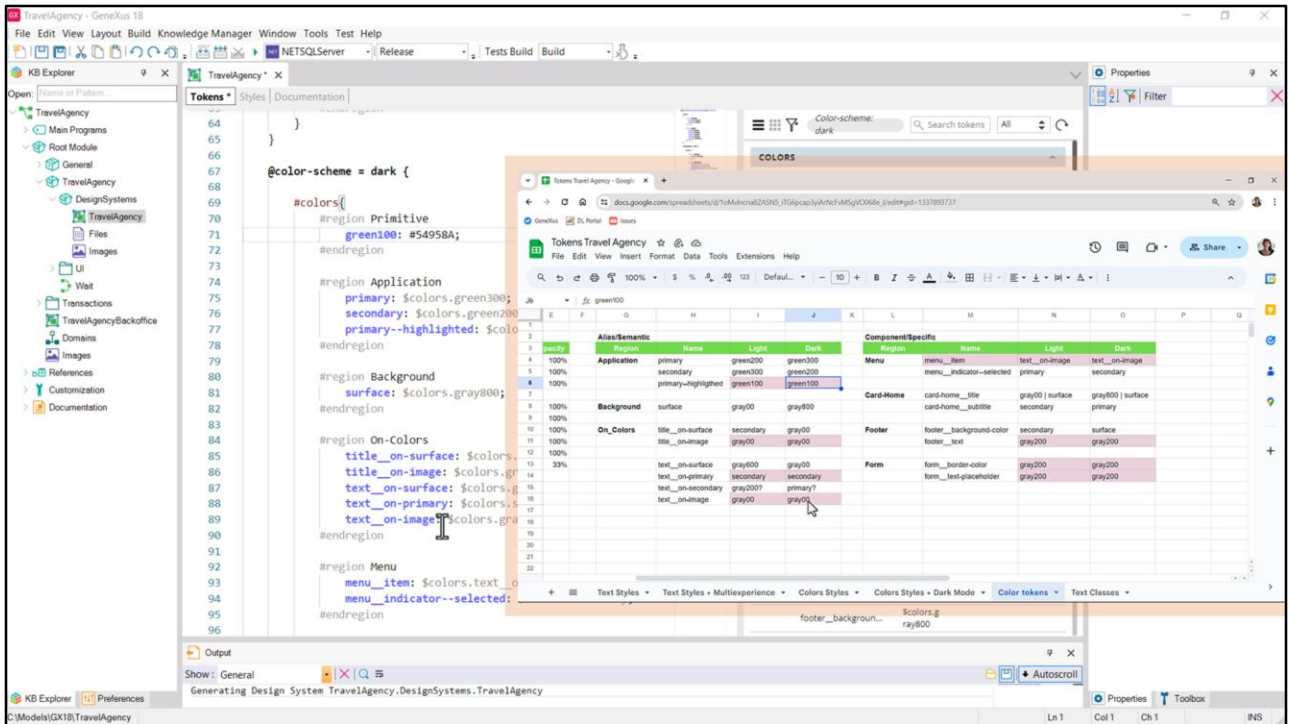
Vamos adicionar, aqui dentro, a região Primitivo. E ali dentro vamos copiar este token, pois este sabíamos que iria variar por modo. Portanto, este será o valor do token para o modo light.

Então, resumindo o que foi feito até agora: definimos a opção color-scheme, que assume um destes dois valores. O valor default é este, light. Aqui temos tokens de cor que não variam de acordo com o modo. Vão corresponder sempre a esses valores. E aqui estamos definindo os tokens de cor quando o color-scheme for light, certo? Estamos indicando, então, os valores de todos esses tokens.

O que nos resta? Bem, dizer quais serão os valores dos tokens quando o color-scheme for dark.

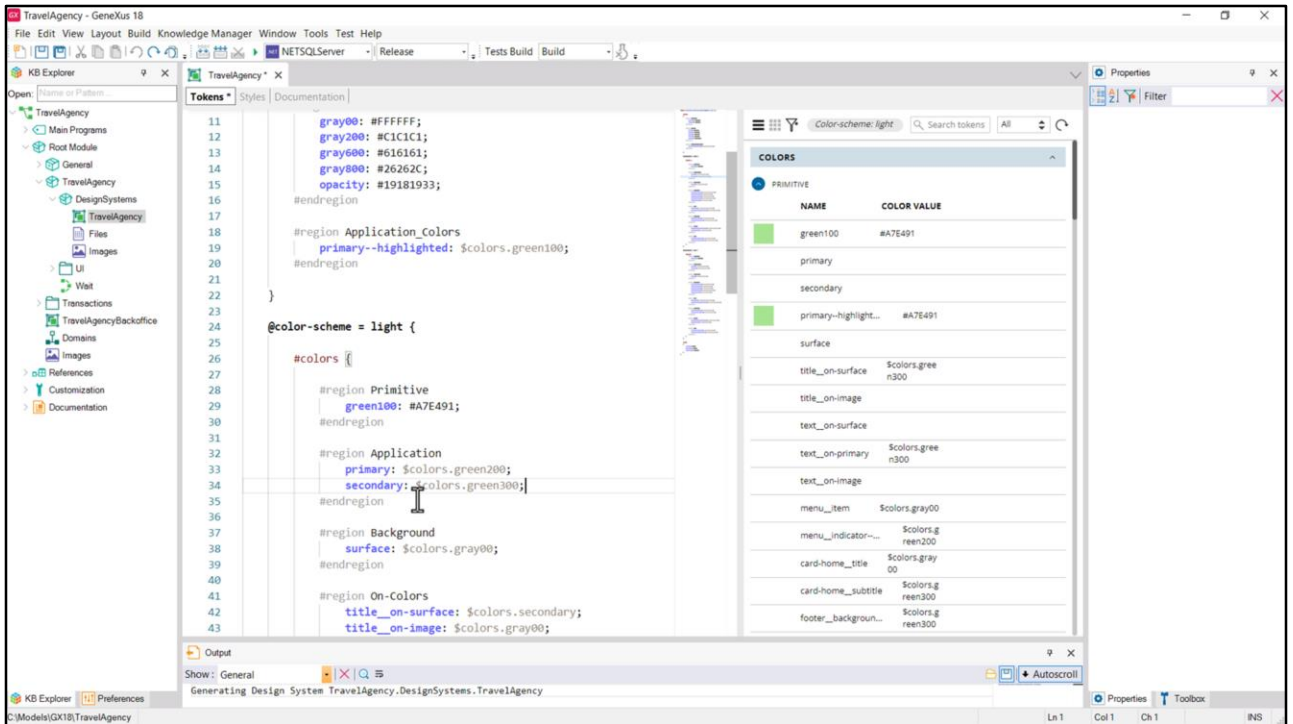


E isso, então, coloco aqui embaixo. Para o color-scheme dark... quanto vale o token green100, que vemos que é esse outro verde claro, o primary, o secondary e todos os demais...



...cujos valores, claramente tirei da planilha, das colunas Dark de cada tabela.

Bem, e aqui entendemos o sentido de ter deixado marcados os tokens para os quais o valor no modo light e dark eram o mesmo.

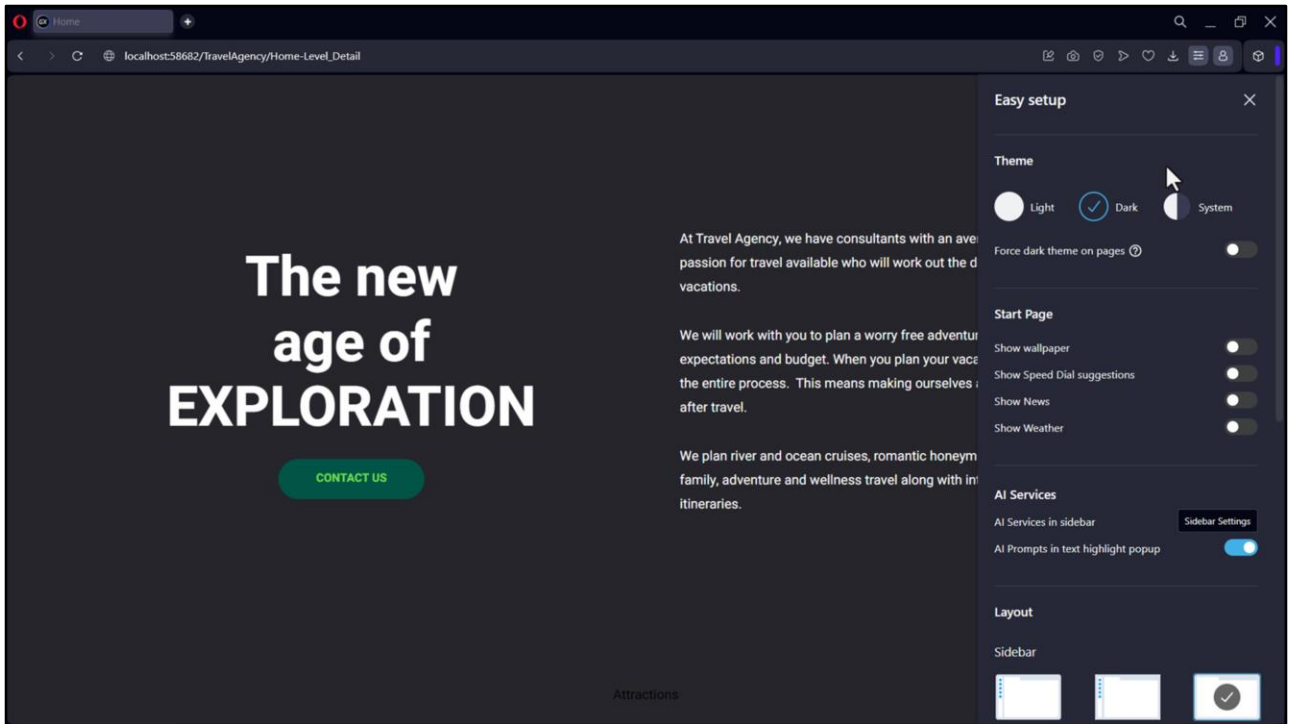


Porque estes eu poderia defini-los uma única vez, no início, onde tenho os tokens invariáveis, por color-scheme. Então, movo, por exemplo, o primary—highlighted para aqui, e o remove das duas seções, light e dark.

Fazer desse modo tem suas vantagens: não haverá inconsistências; e suas desvantagens: ficamos com os tokens mais espalhados.

Por exemplo, terei a primary e a secondary variáveis por color-scheme, mas por outro lado, se quero procurar a primary—highlighted não vou encontrar ali, filtrando por light ou dark.

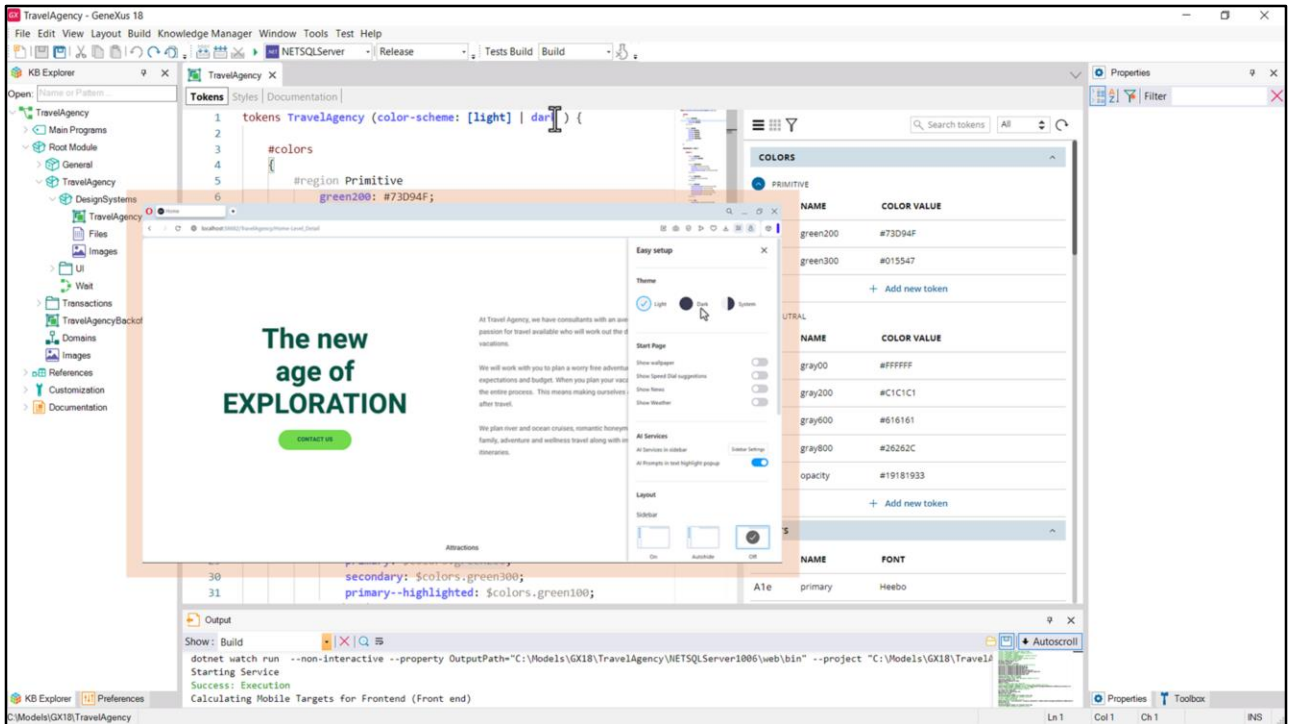
Outra vez, a decisão é um pouco a gosto e risco do frontend. Vou deixá-los por enquanto como os tinha.



Bem, e agora vamos testar.

Aqui temos o modo Light. E vemos, se mudarmos para o Dark, como foram modificadas as cores, tal como queríamos.

Se agora voltarmos ao Light...

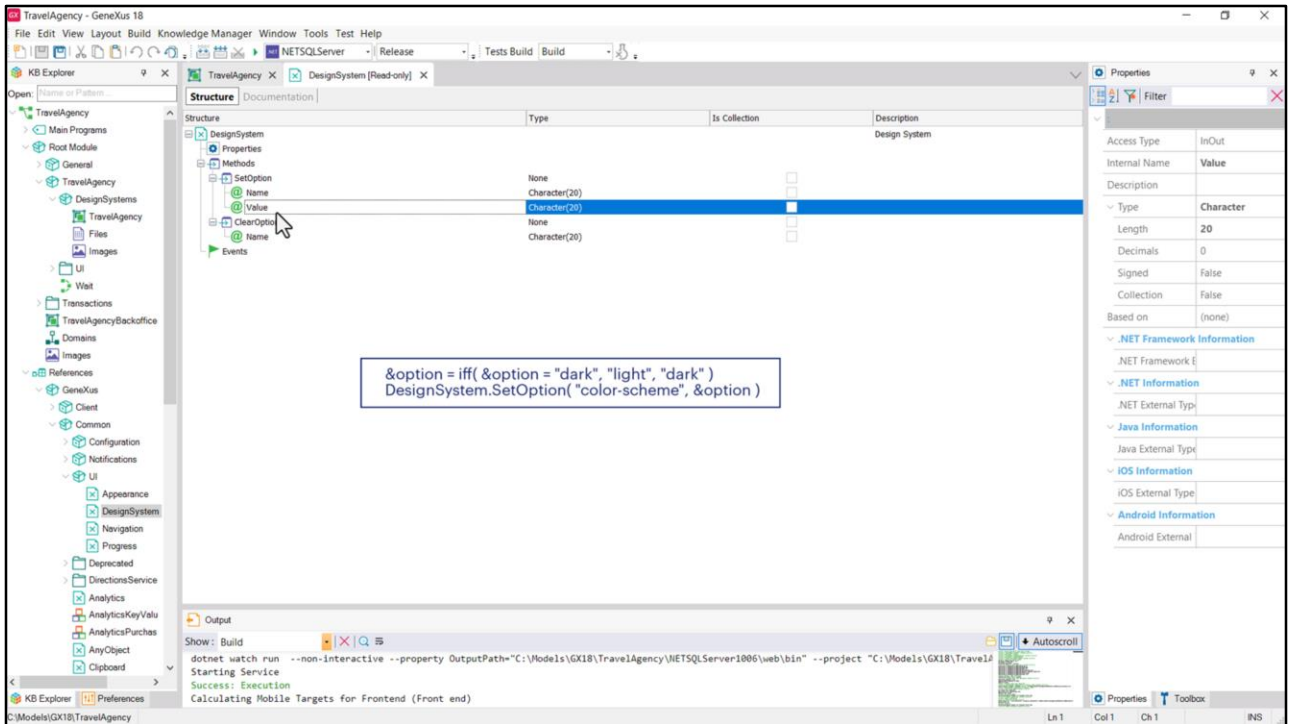


Bem, ao chegar a este ponto seria conveniente fazer algumas observações.

Começamos pelo mais simples: isto que pareceu mágica, na realidade corresponde a uma correta sistematização onde para as propriedades de cor das classes, tanto background-color como color, lhes atribuímos tokens de cor bem semantizados por função. E esses tokens de cor fazemos variar por esquema de cor: light ou dark.

Por outro lado, isto funcionou assim sem mais, sem que tivéssemos que especificar nada no nível programação porque o navegador já está entendendo que o que ali se chama theme light e dark se está correspondendo exatamente com os valores aos quais chamamos de light e dark, para a opção que chamamos exatamente de color-scheme em nosso DSO.

Isto é por ter colocado exatamente esse nome. Não é obrigatório, poderíamos ter chamado de qualquer maneira a opção, mas se assim o tivéssemos feito, o navegador não teria entendido que é a mesma coisa, e não teria mudado as cores, como fez.



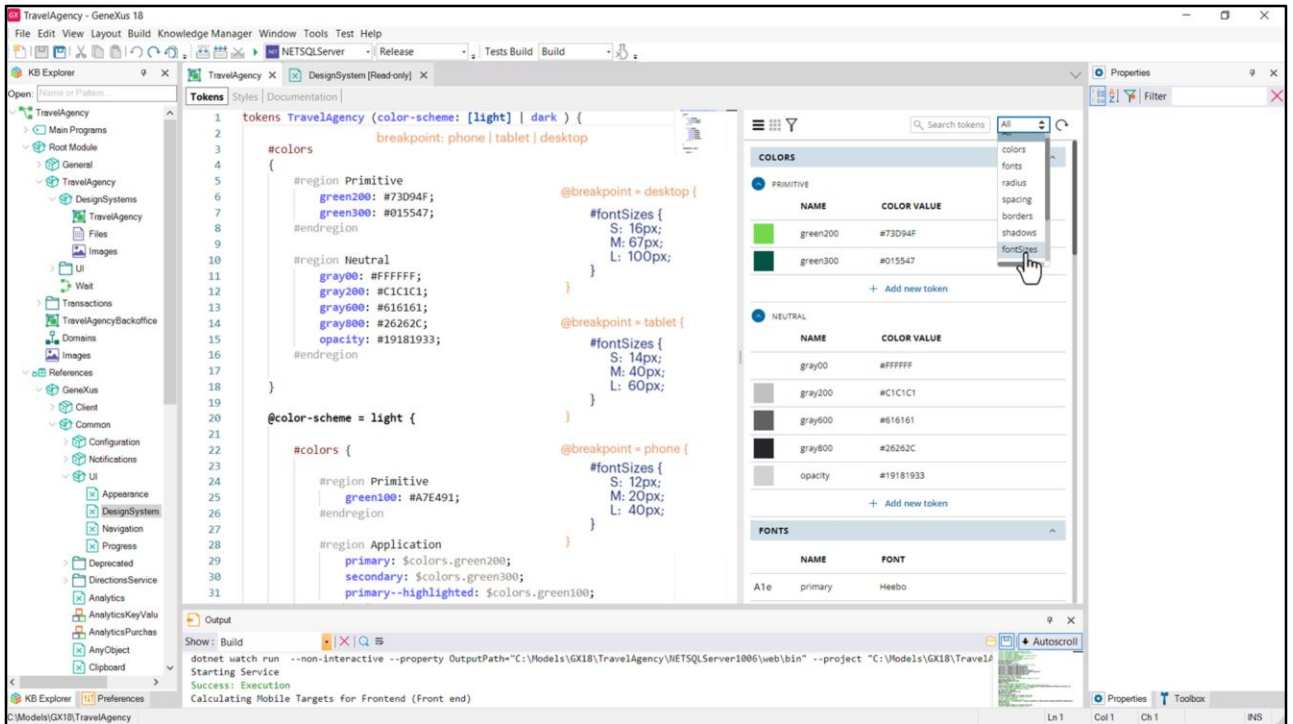
Temos, apesar disso, uma forma de especificar a nível programático qual dos valores da opção queremos que aplique em um determinado momento.

Nos será útil, por exemplo, se quisermos oferecer ao usuário a opção no nível da aplicação, e não do navegador, de escolher o esquema de cor.

Para isso se conta dentro do módulo GeneXus que vem instalado com toda a KB, com o objeto externo DesignSystem.

Ao abri-lo podemos ver que oferece um método SetOption que serve justamente para configurar para uma opção em particular (no nosso caso a única que temos é a color-scheme) um de seus valores possíveis.

Assim, por exemplo, poderíamos colocar este código no evento associado a um controle do layout que seja para modificar o color-scheme. Se na variável que inserimos, option, havia o valor "dark" se altera para "light" e vice-versa (isso é o que faz o comando iff), e logo se define a opção "color-scheme" do DSO associado ao objeto.



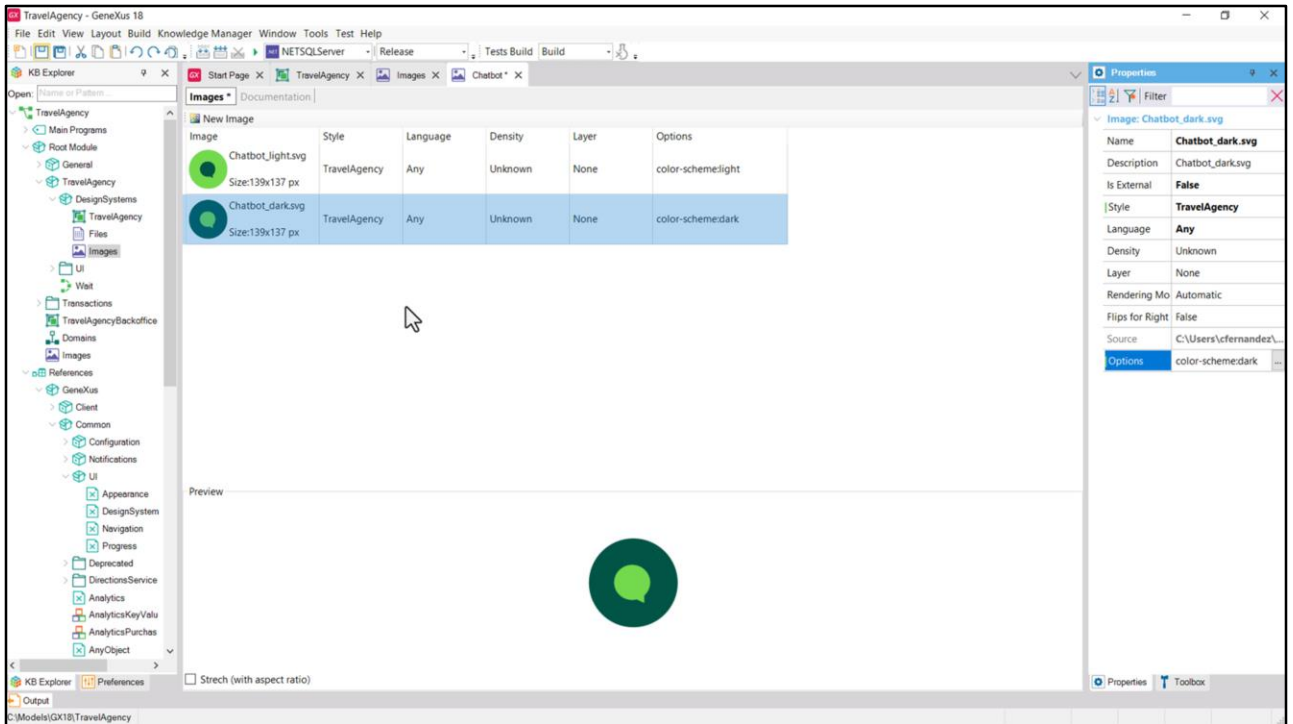
Claro, isso significa que podemos ter por DSO várias opções, não apenas uma, nem tem porquê ser unicamente para poder variar os tokens de cor. Por exemplo, poderíamos adicionar uma opção, além desta, separando-a por vírgula, ou no lugar desta, que seja para variar tokens por tamanho de tela.

Vamos pensar, por exemplo, se tivéssemos definidos tokens de espaçamento ou de tamanho de fontes. Por exemplo, imaginemos que tivéssemos tokens S, M, L para tamanhos de fontes. Para os diferentes tamanhos, Small, Medium e Large.

Poderíamos querer variar seus valores de acordo com o tamanho de tela, para o que podemos definir uma opção breakpoint que assume os valores phone, tablet e desktop, e a partir daí definir que para o breakpoint Desktop estes sejam os valores desses 3 tokens, mas para os outros breakpoints sejam estes outros.

Nos será conveniente utilizar tokens para os tamanhos de fontes em nosso sistema?

Analisaremos isso no próximo vídeo.

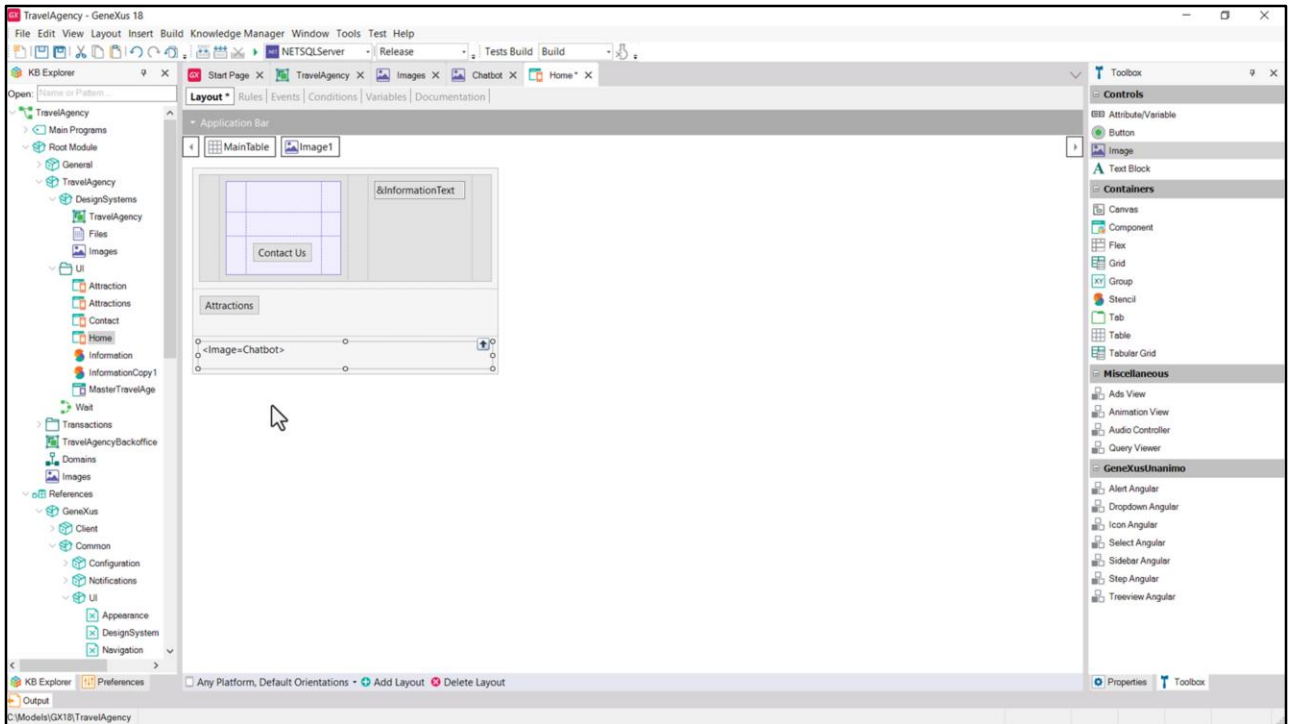


O que me resta dizer agora, é que a partir do aparecimento desta opção em nosso DSO, podemos entender essa dimensão que aparecia no nível das imagens.

Se eu agora associar a ambas as variações como Style nosso DSO... comecemos com esta primeira... vemos aparecer a propriedade Options. Que me oferecerá dar valor a cada uma das opções que esse DSO tenha especificadas. No nosso caso por enquanto só temos uma, a color-scheme.

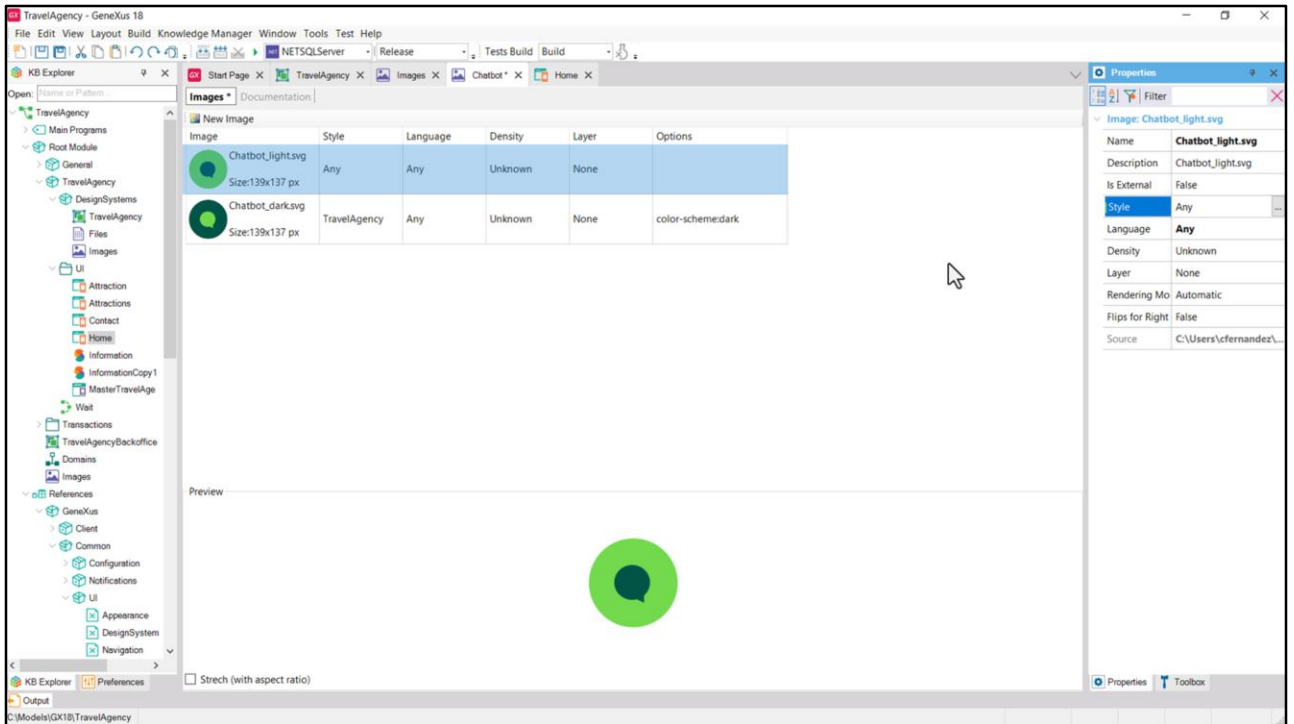
Então diremos que a imagem de Chatbot corresponderá a esta variação apenas se o color-scheme é Light.

E em vez disso a esta outra, se o DSO for TravelAgency também, mas com color-scheme Dark.

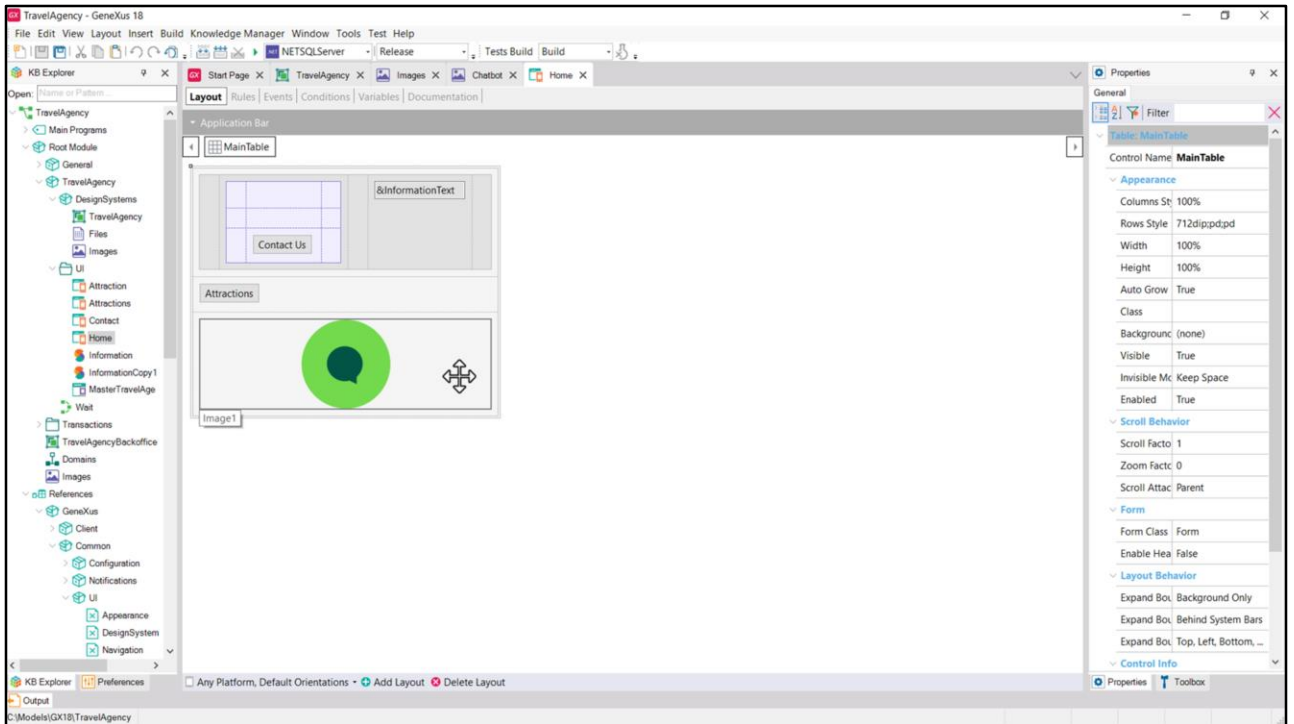


Ainda não implementamos o Master Panel, que será onde teremos o botão com esta imagem, mas poderíamos, só para testar isto em execução, inserir uma imagem solta aqui...

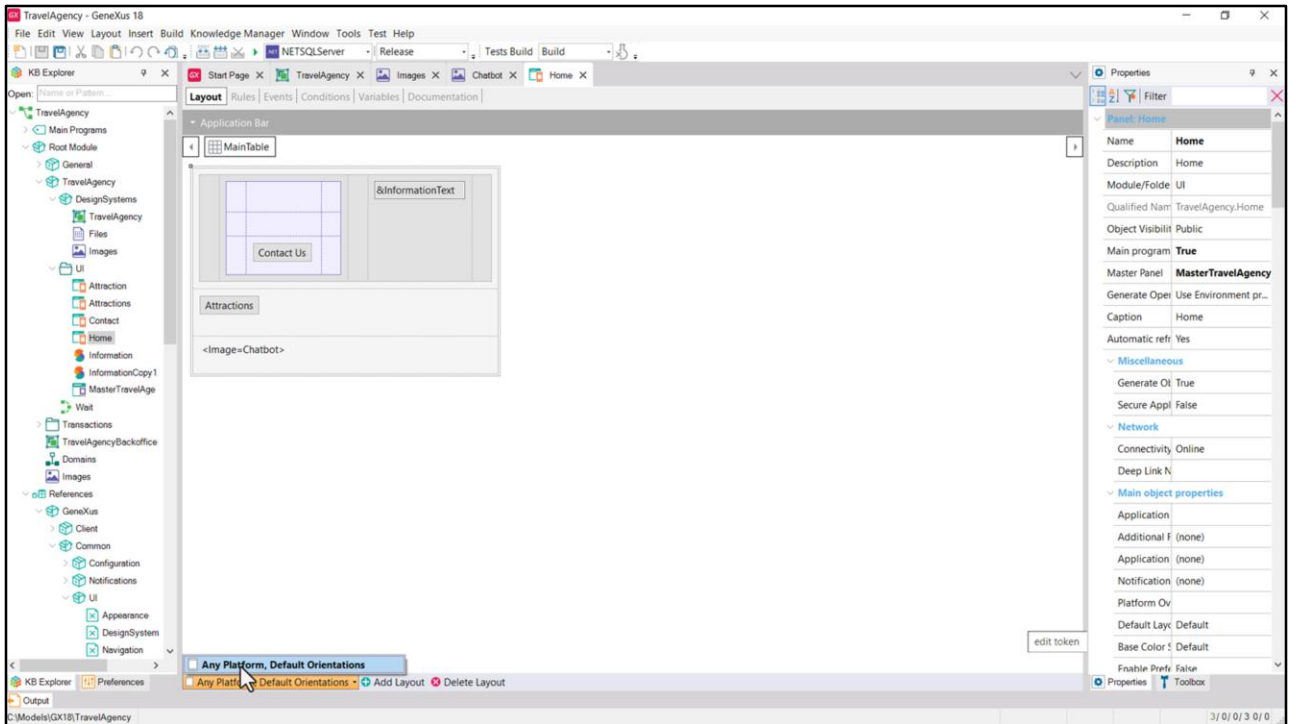
Não está nos mostrando a imagem que se chama Chatbot, não a vemos... diz que essa é a imagem que será renderizada, mas visualmente estamos às cegas. E gravar não muda nada.



Vamos ver o que acontece se eu retirar para uma das variações da imagem o Style... deixo o valor Any, para que valha para qualquer DSO.... Gravo...

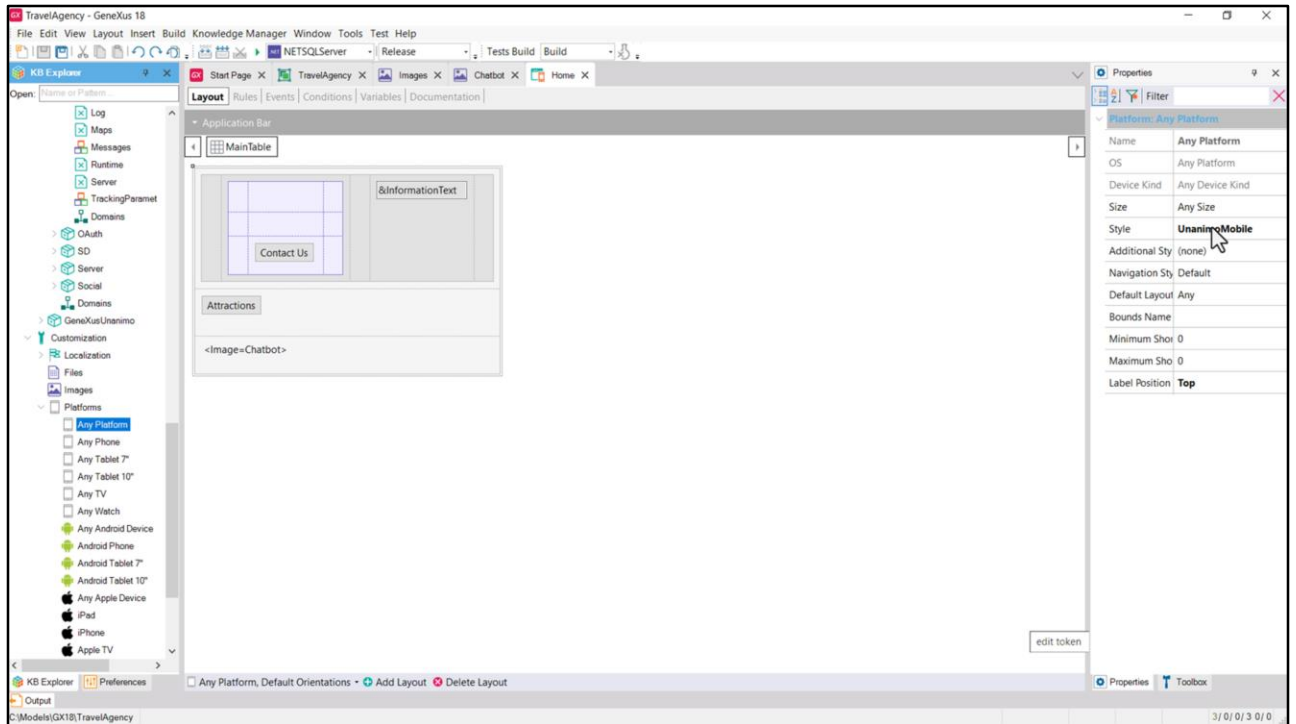


... e agora volto ao panel, o fecho e volto a abri-lo. E agora a vejo!

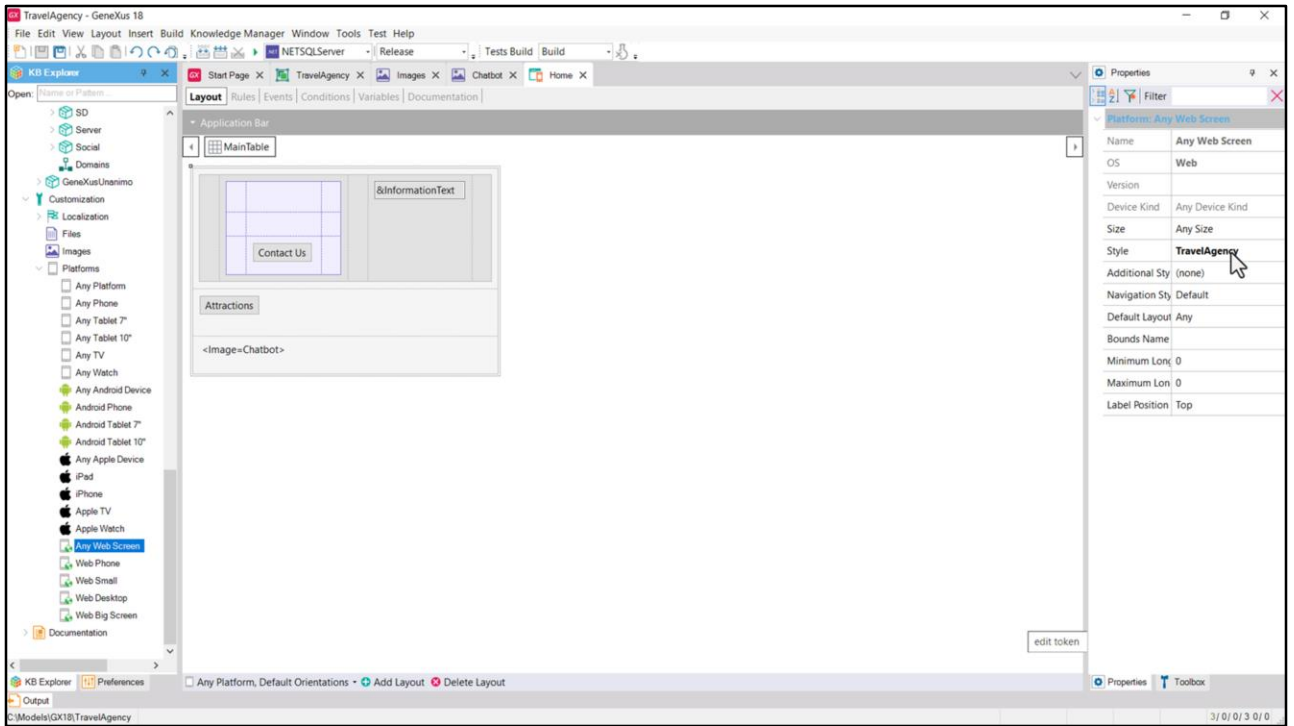


E se volto a restringi-la apenas ao DSO e em modo Light?... Fecho e abro outra vez, e outra vez!

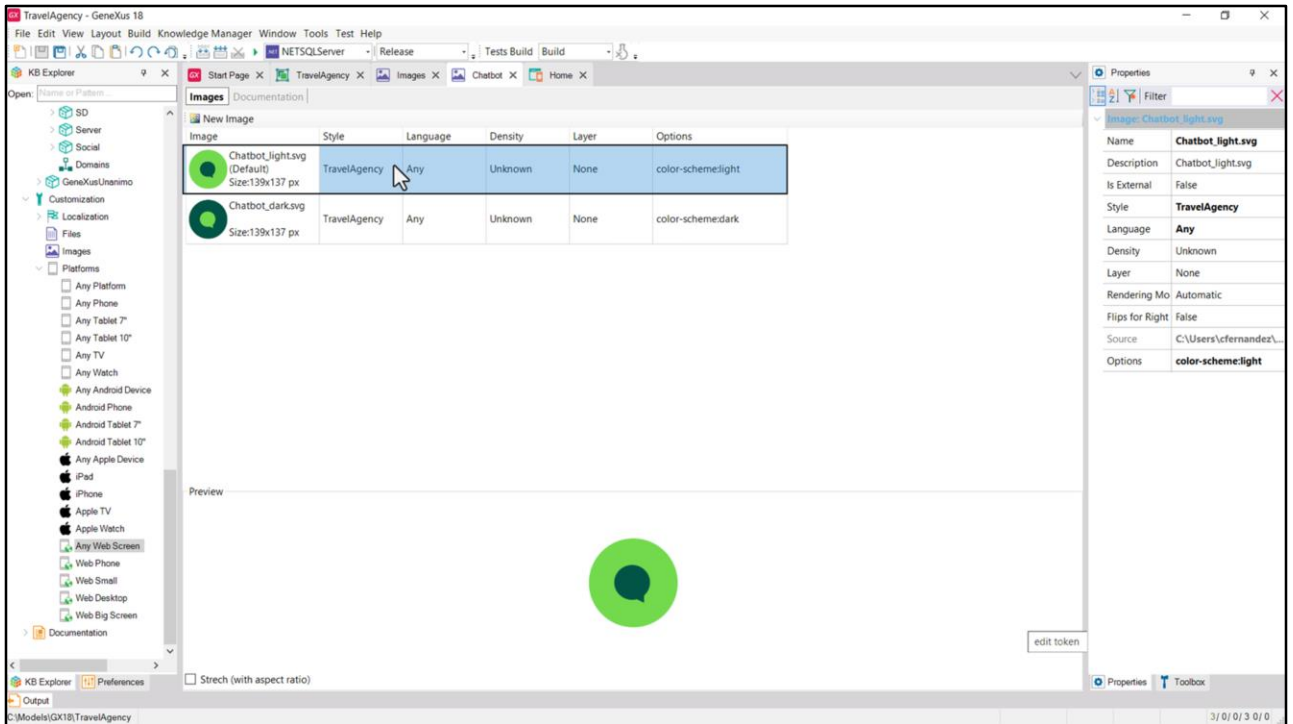
Bem, isto faz sentido porque este objeto Panel é válido para todas as plataformas, não apenas para a Angular que estamos desenvolvendo.



E se viermos para ver os DSOs associados à Any Platform, vemos que é Unanimomobile.

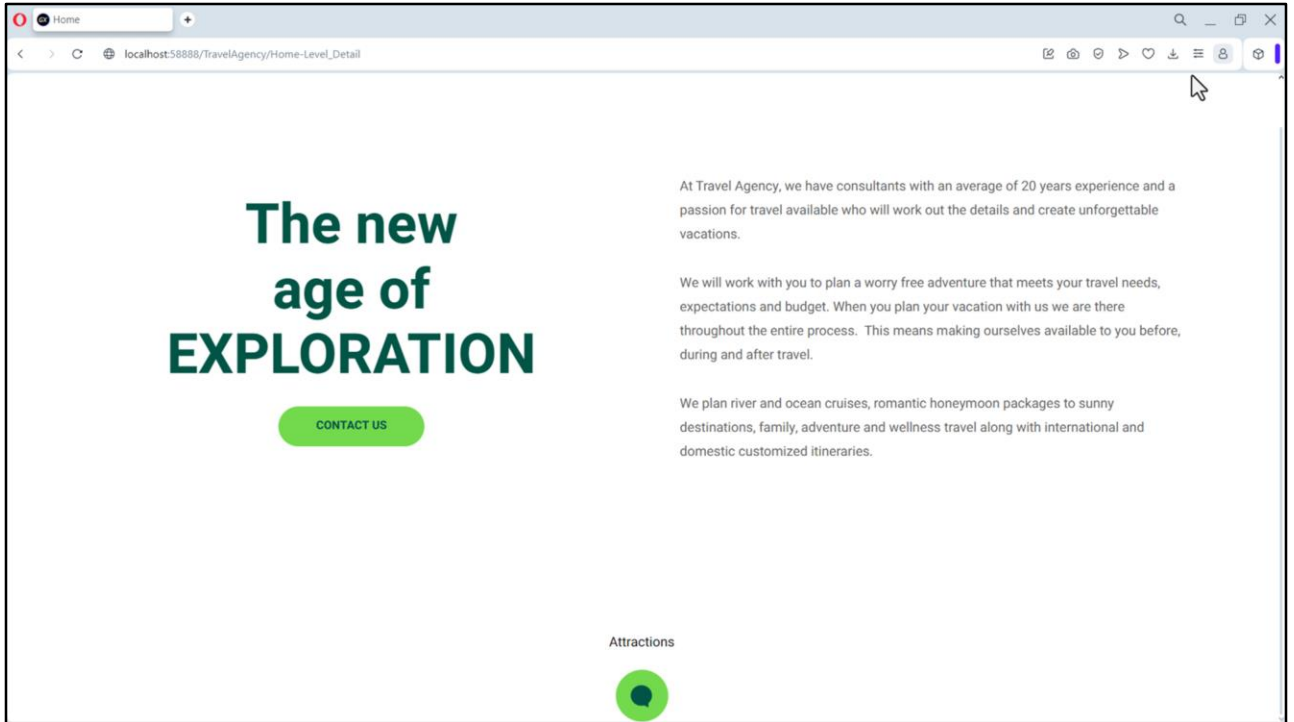


Que não é o TravelAgency que tínhamos aqui.

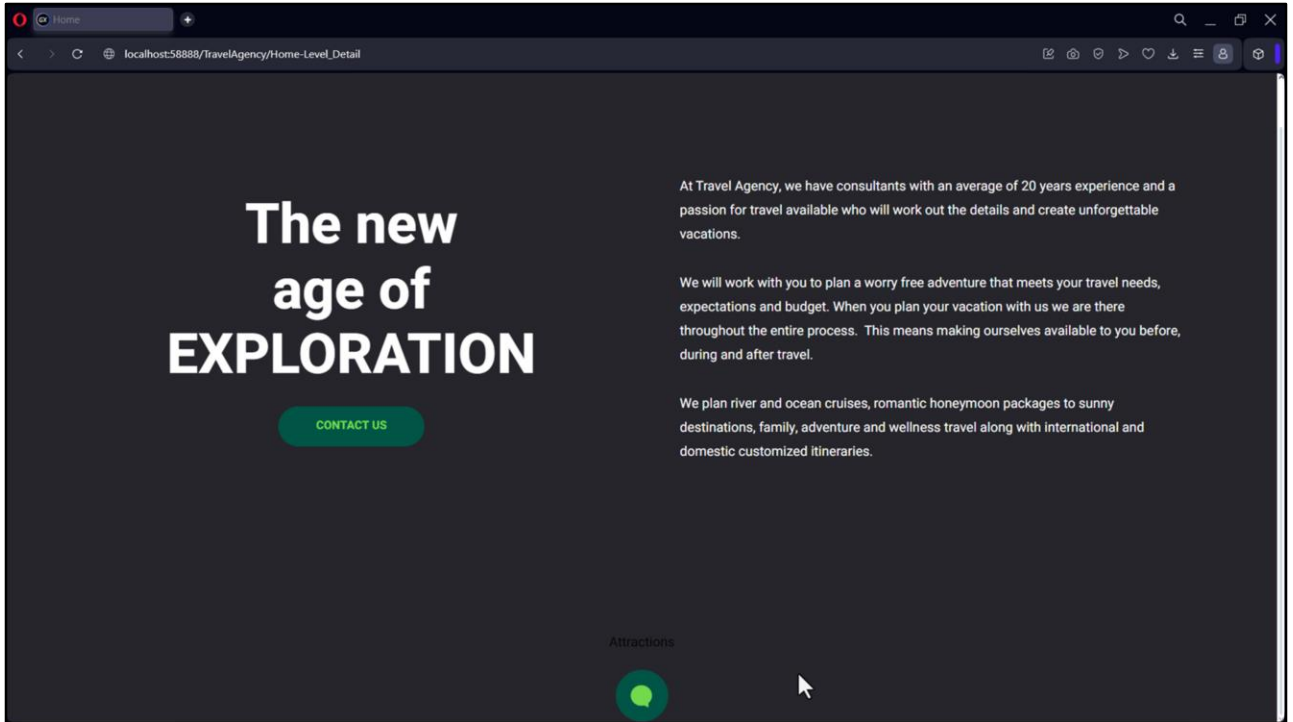


E o que estamos especificando é que pegue um desses dois valores apenas para este Design System Object. Não temos especificado o valor para outro Design System Object.

Porém, observemos que acontece o mesmo, ainda que isto apareça desta maneira, quando executamos para Angular...



Bem, aqui vemos a imagem correta para light...



... e vamos ver o que acontece em modo dark... muito bom!

Vemos que está realizando, como esperávamos, as variações.

Bem, continuamos no próximo vídeo.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com