

Color system of Design System



Cecilia Fernández

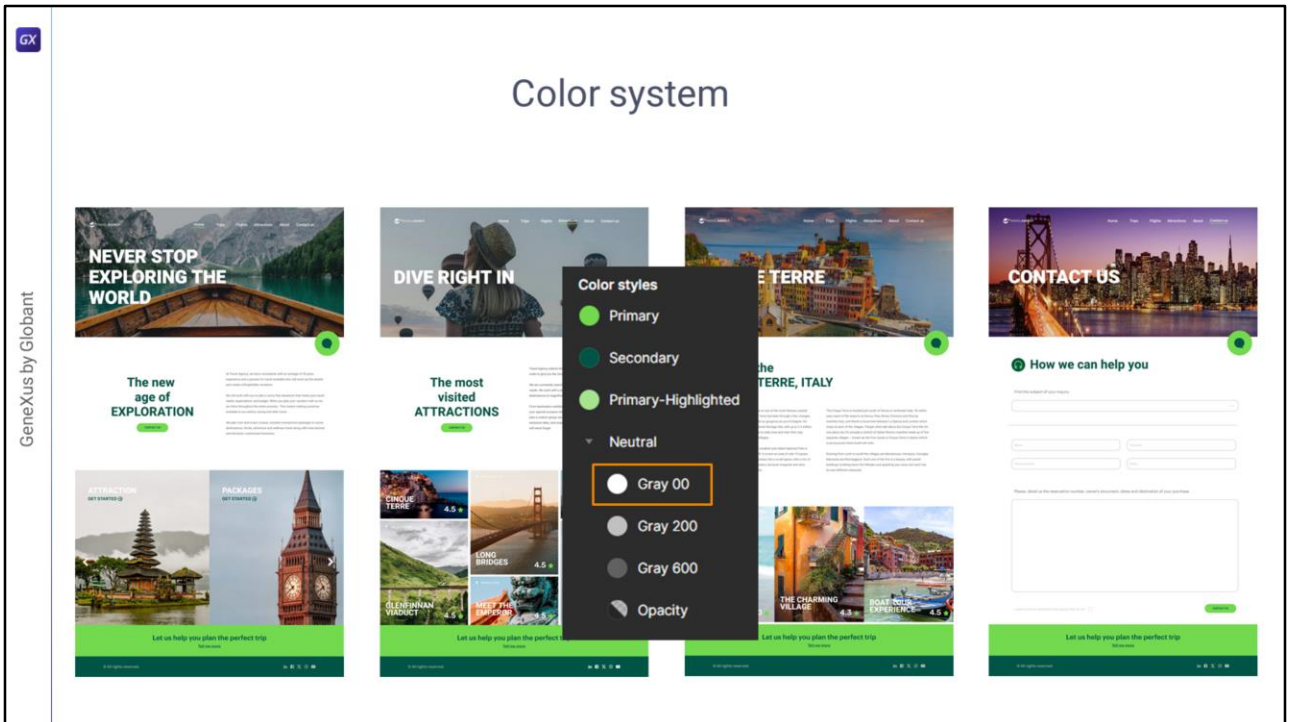
Color system

The screenshot displays the GeneXus Tokens editor interface. The main editor shows the following code for the 'TravelAgency' tokens:

```
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-hand panel, titled 'COLORS', provides a visual overview of the defined color tokens. It is organized into two sections: 'APPLICATION' and 'NEUTRAL'. Each section contains a table with columns for 'NAME' and 'COLOR VALUE'. The 'APPLICATION' section includes 'primary' (#73D94F), 'secondary' (#015547), and 'primary--highlighted' (#A7E491). The 'NEUTRAL' section includes 'gray00' (#FFFFFF), 'gray200' (#C1C1C1), 'gray600' (#616161), and 'opacity' (#19181933). Each row in the table is accompanied by a small color swatch. Below each table is a '+ Add new token' button. At the bottom of the panel, there is a 'FONTS' section header.

En el video anterior habíamos completado los tokens de color de nuestro DSO a partir de lo que habíamos recabado del proyecto en Figma, ¿se acuerdan?



Sin embargo, el sistema de colores que queda así expresado es muy básico: a nivel semántico solamente expresa el color **primario** de la aplicación y el **secundario**, pero no expresa cosas tales como qué color aplicará para el fondo, qué color aplicará para los textos que son títulos sobre los fondos, para los que son textos comunes sobre los fondos, para los textos sobre imágenes, y demás.

El sistema de colores podría representar de un modo mucho más semántico la función de los colores en la aplicación. Por ejemplo, este blanco, Gray00...

Color system


 Gray 00


The new
age of
EXPLORATION



Let us help you plan the perfect trip



DIVE RIGHT IN

The most
visited
ATTRACTIONS



Let us help you plan the perfect trip



CINQUE TERRE

Visiting the
CINQUE TERRE, ITALY



Let us help you plan the perfect trip



CONTACT US

How we can help you

First name of your name

Last name

Phone

Email

Please describe the services you are looking for, your destination, dates and duration of your booking

Let us help you plan the perfect trip

...podría utilizarse tanto como color de fondo, como...

Color system



Gray 00



The new
age of
EXPLORATION



The most
visited
ATTRACTIONS



Visiting the
CINQUE TERRE, ITALY

surface

title__on-image



Let us help you plan the perfect trip



Let us help you plan the perfect trip



Let us help you plan the perfect trip

...como color del **título sobre una imagen**, tanto de la de Hero como de la de las Cards...

Color system


 Gray 00


The new
age of
EXPLORATION



The most
visited
ATTRACTIONS



Visiting the
CINQUE TERRE, ITALY

surface



Let us help you plan the perfect trip



Let us help you plan the perfect trip



Let us help you plan the perfect trip

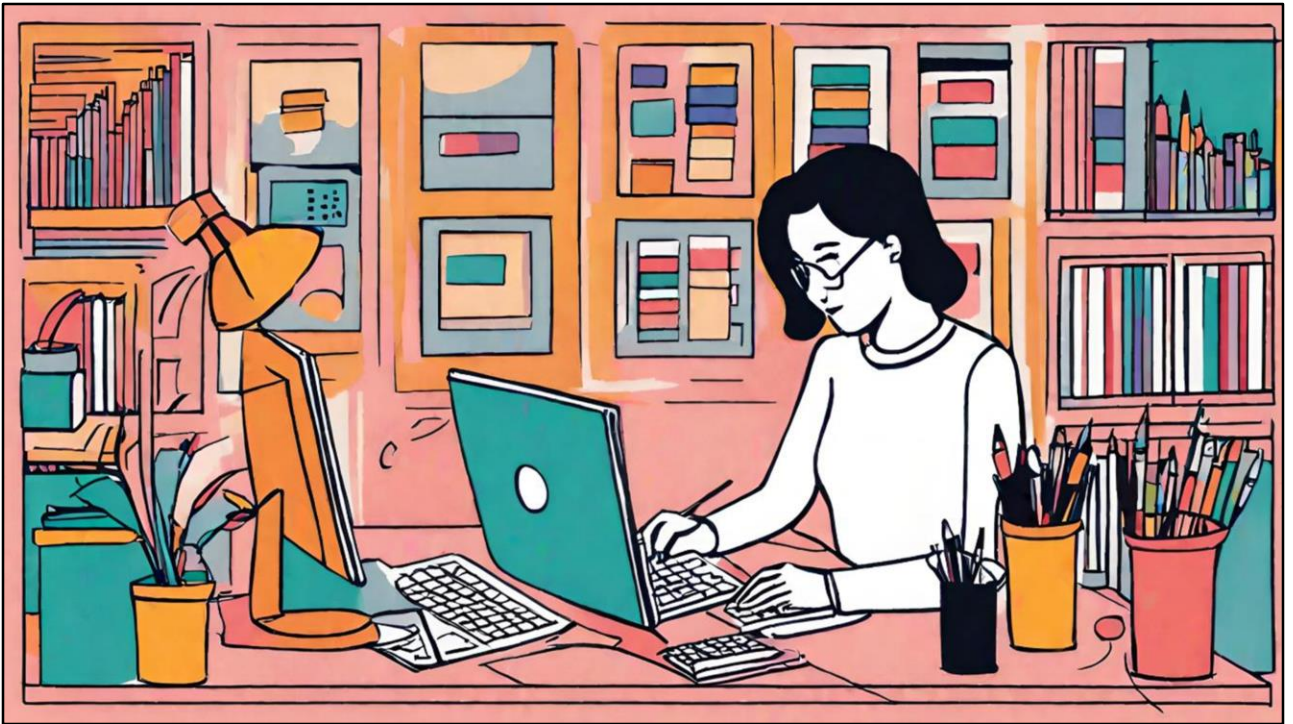
title_on-image

text_on-image

...así como para otros textos sobre imágenes, como los de estas cards de Attractions. Es decir, en al menos tres funciones distintas.

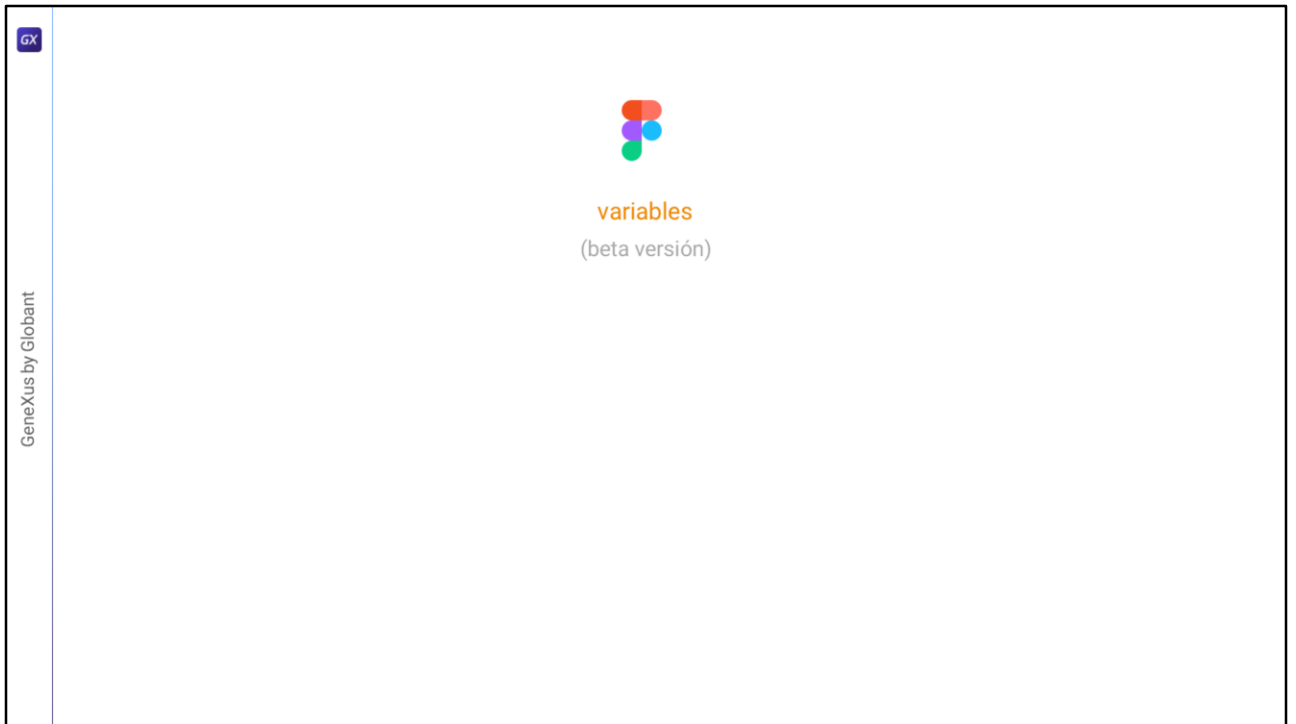
Si a estas funciones: color de **superficie**, color de **títulos** sobre imágenes, color de **textos** sobre imágenes, les damos un nombre, estamos construyendo un sistema de colores más verdaderamente sistémico, y por supuesto, más semántico.

Dirá mucho más y mejor del diseño de la aplicación. Será luego muy sencillo cambiar, por ejemplo, el color de fondo de las pantallas. O el color de los títulos sobre imágenes, o de los textos sobre imágenes.

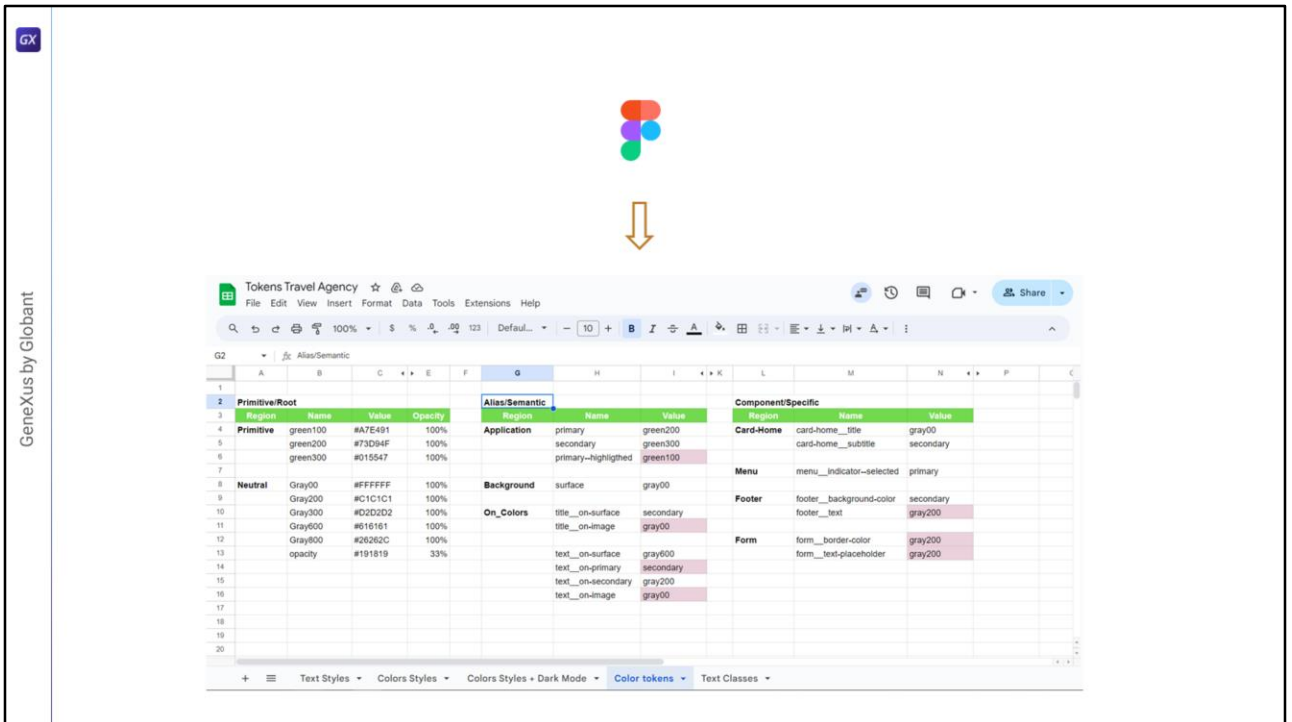


La tarea de construir un buen **sistema** de colores, es decir, de identificar y abstraer sus funciones, no es tan fácil, pero es una tarea muy importante, que si se posterga degradará al sistema en lo sucesivo, por lo que conviene ya de entrada destinar toda la energía posible en resolverla, cuanto antes, y entonces luego todo irá sobre ruedas. A lo sumo habrá que hacer pequeños ajustes.

En general esta tarea queda en manos del diseñador, que es quién ya trabaja más o menos conscientemente con estas abstracciones.



Si su herramienta de diseño no le permite ese nivel de expresividad (es el caso de Figma, que tiene en beta aún esa posibilidad)...



...podrá enviarle al frontend developer el modelado del sistema en una planilla, por ejemplo.

Eso no quita que nosotros, frontenders, aunque no seamos quienes construyamos el modelo, tengamos que entenderlo. Por supuesto que si no contamos con un diseñador, o el diseñador no es tan competente en la sistematización, tendremos que hacer ese trabajo nosotros.

Así que vamos a analizar este modelo que me armó Chechu (y que yo fui completando) para el sistema de colores de nuestra aplicación.

Lo primero que podemos ver es que está pensado en 3 niveles de abstracción. El más básico, este, corresponde a la paleta de colores, donde lo que hacemos es darle un nombre, simplemente, a cada color de la paleta y no mucho más.

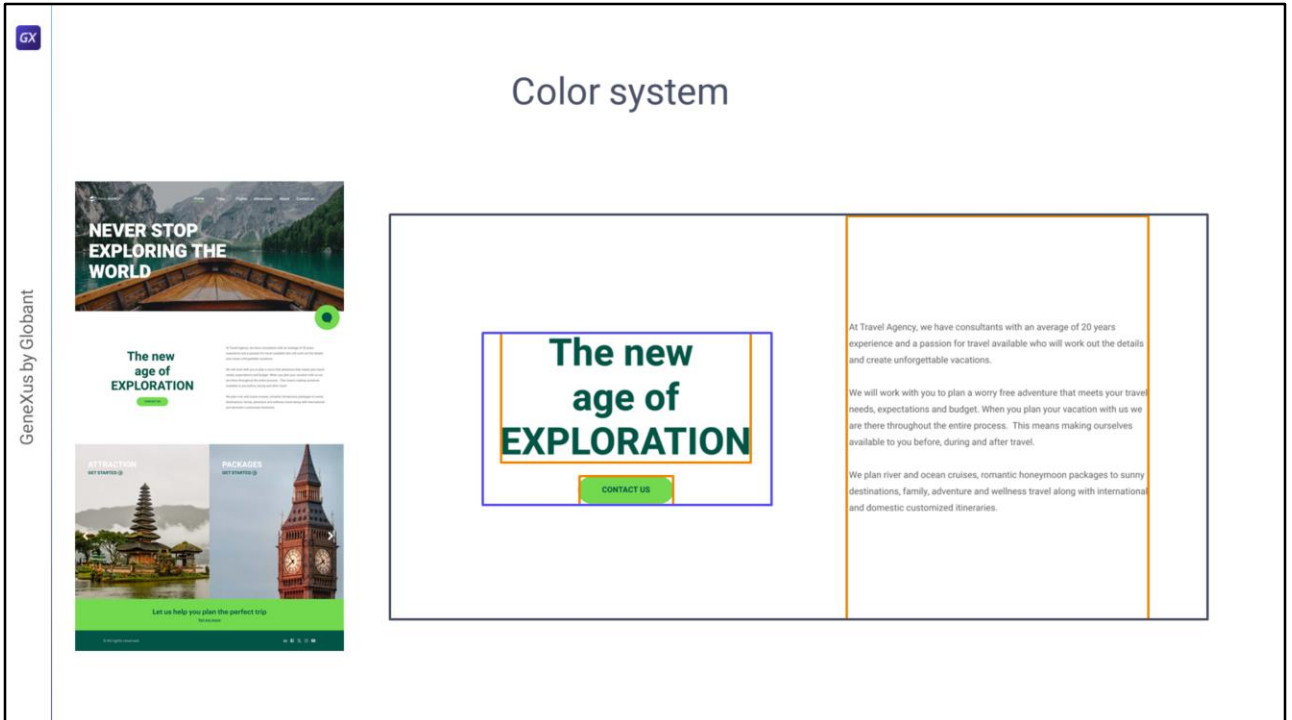
El segundo nivel sí es fundamental, se construye sobre ese primero, y es fundamental porque ya corresponde a la semántica global de los colores en la aplicación. Va a modelar, entonces, el sistema de colores de la manera más general posible.

Con los estilos de color que Chechu había creado en Figma hasta el momento, se acuerdan, tenemos una combinación muy básica de estos dos niveles.

Luego podrá haber o no un tercer nivel, que ya es para especificar los casos especiales que no pueden ser modelados según la generalidad que da el segundo nivel. Es decir, el segundo nivel se piensa como transversal a toda la aplicación, mientras que el tercero ya es mucho más específico de un componente o de una parte del diseño, en especial.

Vamos a analizar un poco todo esto para entenderlo bien. Aquí vemos tokens para el color primario, para el secundario, para el primario highlighteado (estos ya los habíamos incorporado antes) y estamos agregando un token para el color de fondo de las pantallas: este; y luego vemos que se están agregando tokens para representar los colores de los elementos que se ubicarán **sobre** colores de fondo.

Para entenderlo vamos a verlo con ejemplos.



Es que todo elemento de un layout tiene un color de fondo, el background-color, que eventualmente puede ser transparente.

Por ejemplo, estos dos textos tienen color de fondo transparente, pero este botón no.

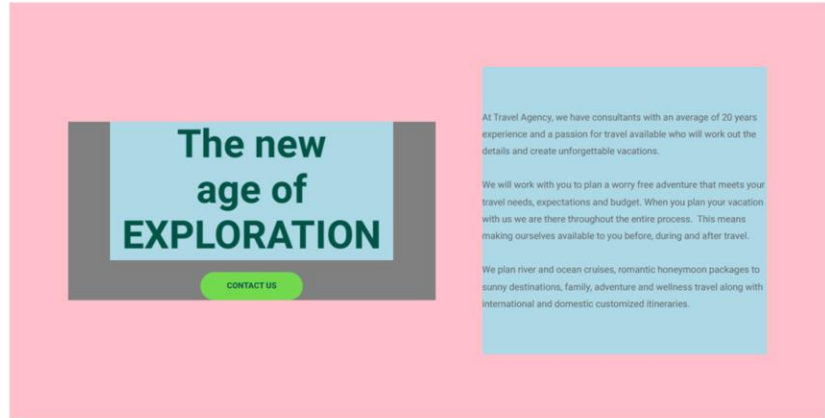
Pero estos dos controles están a su vez dentro de otro, tabla o flex, no importa, que también puede tener color de fondo. O, ser transparente.

Y este y este, a su vez, también están dentro de una tabla o flex que puede tener o no color de fondo.

Color system

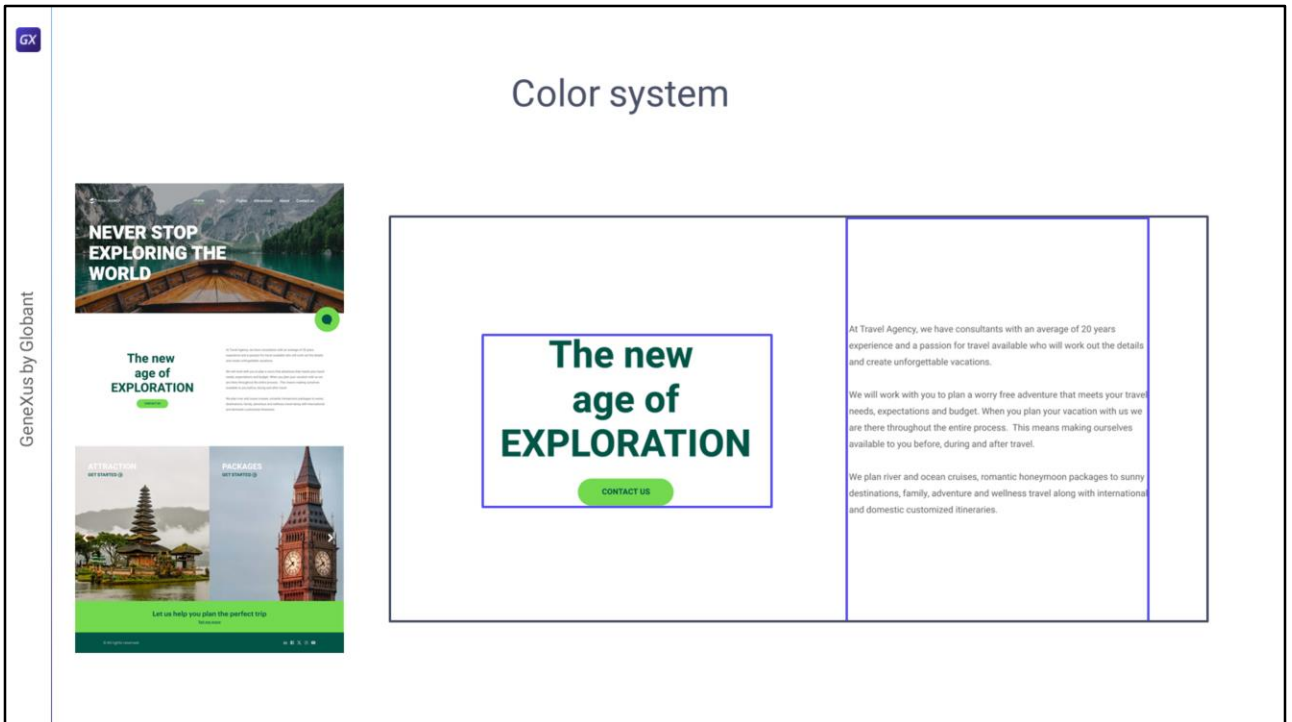


The new
age of
EXPLORATION



Por ejemplo, miren qué pasa si les proporciono background-colors diferentes a varios de esos elementos.

En definitiva, hay en todo layout una jerarquía de controles, unos dentro de otros, cada uno con color de fondo, que puede (o no) ser transparente.

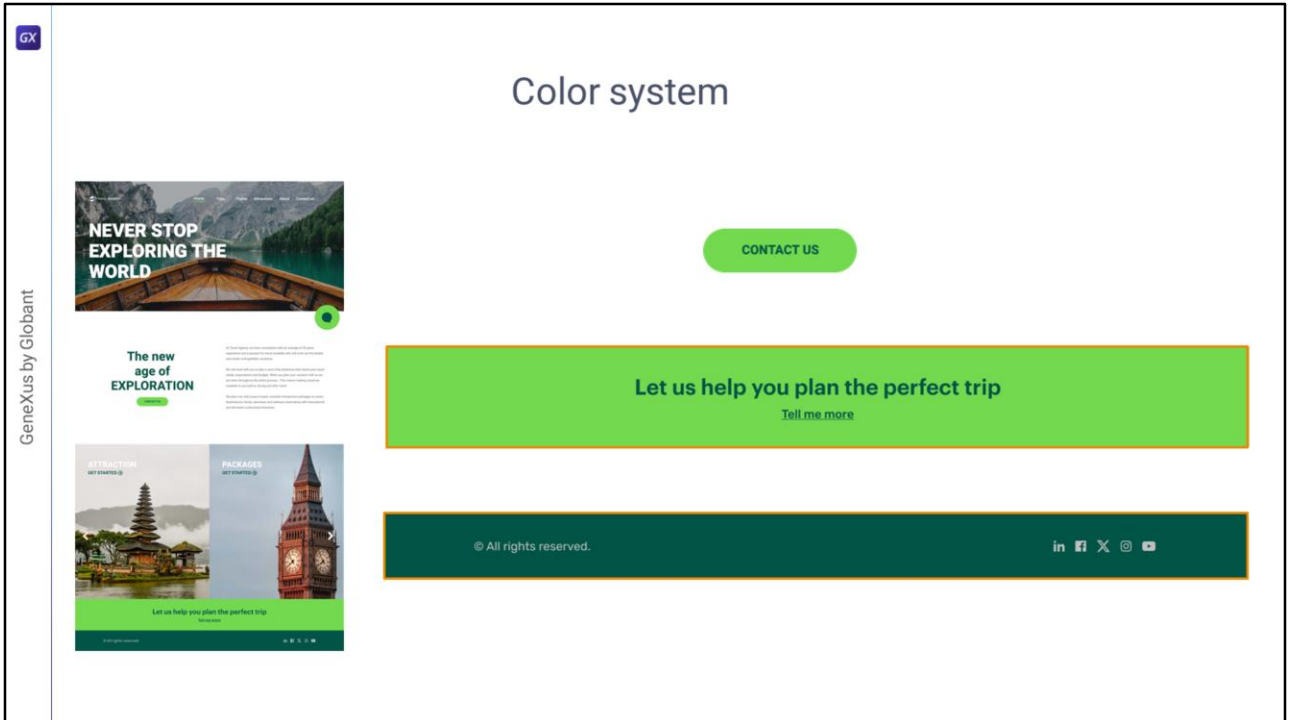


En el ejemplo de nuestro diseño, el único background-color que no es transparente (de hecho es blanco) es el del contenedor. Bueno, a excepción del botón, que tiene su background-color verde claro, obvio.

Bueno, tampoco es necesario lo que dije hace un instante: podría ser también transparente el background-color de esta tabla, si el que a su vez la contiene tiene el background-color blanco.

De hecho es nuestro caso, ¿se acuerdan que habíamos colocado el color de fondo a nivel de la clase Application en el DSO para que valiera universalmente para todos los layouts? Bueno, si no recuerdan no se preocupen que igual ese ahora no es el punto.

En lo que quiero que reparemos es en el contraste necesario de fondo y frente.



Así, esta por ejemplo puede ser una tabla o una celda de una tabla, y esta otra. Y esa tabla o celda tiene un background-color. En un caso el verde claro, en el otro el verde más oscuro.

Y los controles que están dentro tienen background-color transparente para que se ubiquen **sobre** esa superficie que es la del fondo.

Pero además tienen un color propio, en estos casos de textos son el **color del texto**, que es el que tiene que contrastar sobre esa **superficie de fondo**, y eso es todo lo que le interesa al diseñador: lograr un buen contraste, entonces, entre los elementos superpuestos, para que se puedan visualizar bien y no queden confusos.

El botón también puede pensarse así, como teniendo un background-color y el texto del botón **contrastado sobre** él.

Color system



NEVER STOP
EXPLORING THE
WORLD

The new
age of
EXPLORATION

CONTACT US



EXPERIENCES
LET US HELP YOU

PACKAGES
LET US HELP YOU

Let us help you plan the perfect trip

CONTACT US

The new age of EXPLORATION

CONTACT US

At Travel Agency, we have consultants with an average of 20 years experience and a passion for travel available who will work out the details and create unforgettable vacations.

We will work with you to plan a worry free adventure that meets your travel needs, expectations and budget. When you plan your vacation with us we are there throughout the entire process. This means making ourselves available to you before, during and after travel.

We plan river and ocean cruises, romantic honeymoon packages to sunny destinations, family, adventure and wellness travel along with international and domestic customized itineraries.

Let us help you plan the perfect trip

Tell me more

© All rights reserved.



CONTACT US

How we can help you

What is your name?

What is the destination, dates, services, dates and duration of your vacation?

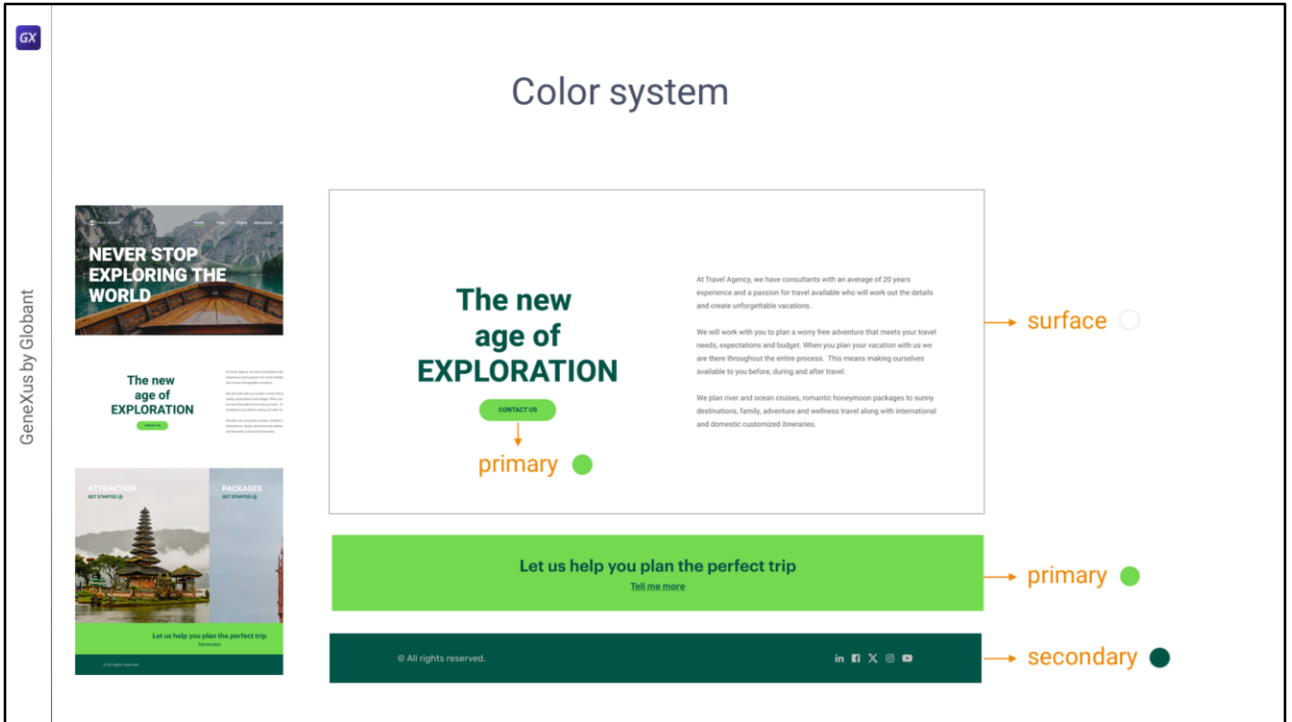
Let us help you plan the perfect trip

CONTACT US

© All rights reserved.



Así, si analizamos con esa óptica el sistema, podemos ver en funcionamiento 3 colores de fondo.



Este, que llamaremos **surface**, y corresponderá al blanco de la paleta de colores.

Este, al que llamamos **primary**, que es un verde de la paleta de colores (el verde clarito).

Y este otro, al que llamamos **secondary**, que corresponde al otro verde.

Para el botón también tenemos el color de fondo **primary**.

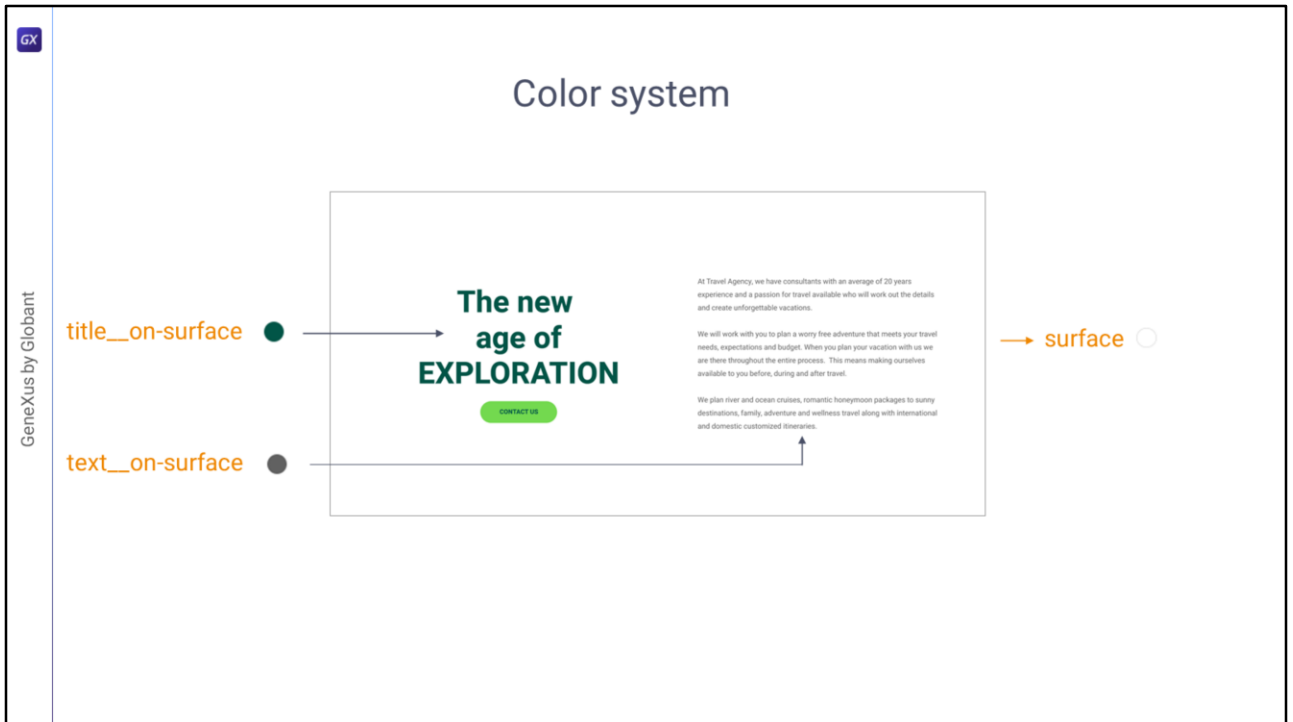


Bien, pero, decíamos, es fundamental para lograr un buen diseño el buen contraste de los colores que se ubiquen **sobre** esos **colores de fondo**.

Si observamos los dos elementos que tienen **fondo primary** (estos dos) vemos que los textos que se ubican **sobre** ese color **primary** corresponden a un mismo color.

Por ello podemos pensar en un token que especifique cuál es el color de los textos que se ubiquen sobre un fondo **primary**. Y a ese token es al que le llamamos **text_on-primary**. (Noten que aprovecho y ya estoy utilizando la nomenclatura BEM para nombrar ese token).

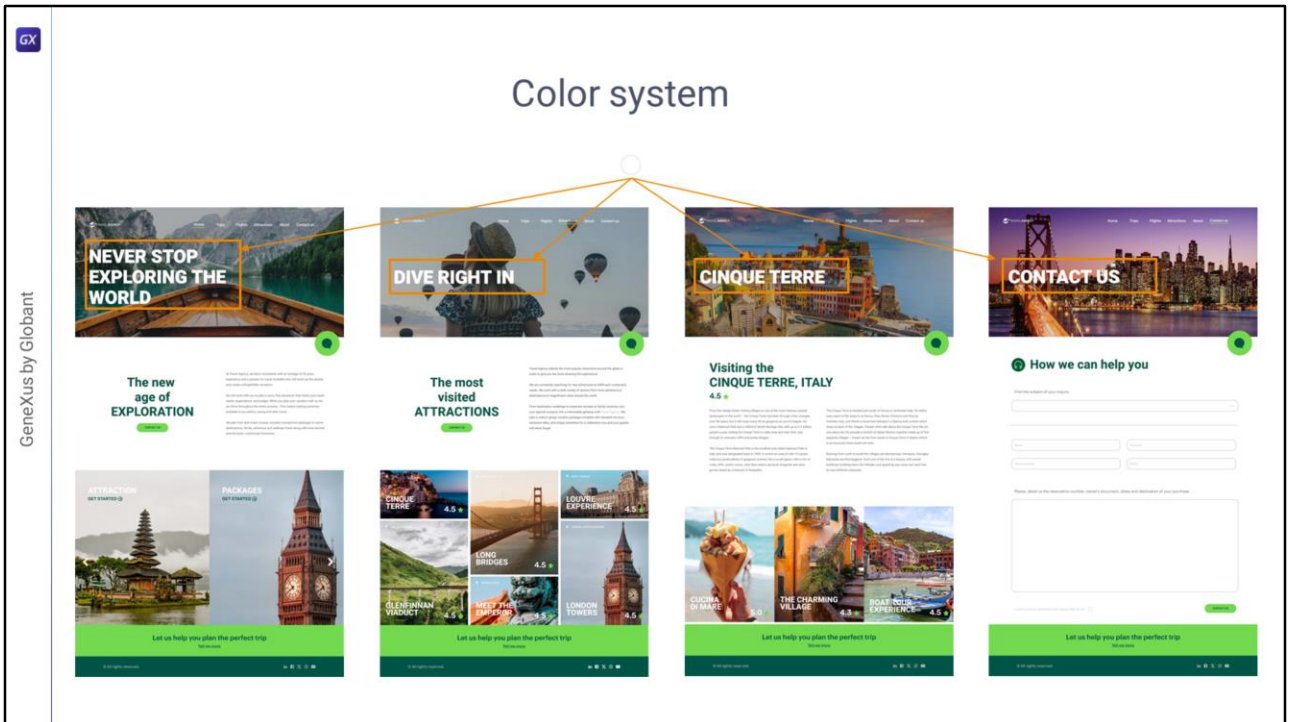
En este caso el valor que le daremos al token es el de otro token, el **secondary**, pero, más allá de eso, no importa qué valor le demos, lo que importa es que acabamos de aislar, de identificar, un token que va a cumplir esta función: de ser texto que se ubica **sobre** una superficie **primary**.



Y ya que estamos acá, también podemos pensar en qué colores tendrán los textos que se ubiquen **sobre** la superficie **surface**. Vemos dos tipos de elementos diferenciados:

El **color del título**, y el **color de un texto** común. Aquí no sé si se aprecia bien que son dos colores distintos. Pero aunque no lo fueran, es bueno diferenciarlos porque sus funciones son distintas, aunque para el diseño actual se concreticen, que no es el caso, porque son dos colores distintos, pero así se concretizaran con el mismo valor, igualmente sería bueno separarlo en dos elementos. Porque podrían cambiar más adelante y uno adquirir un valor y otro otro, porque corresponden a conceptos distintos.

Entonces podemos darles nombres que signifiquen exactamente los dos conceptos diferentes: title__on-surface, un **título** sobre la superficie y un **texto** sobre la superficie.



Por otro lado, podemos pensar que las **imágenes** también operan como **fondos**, cuando se les superponen elementos encima. Tanto las imágenes Hero de los encabezados de las páginas, como las de todos estos carruseles tienen textos superpuestos.

¿De qué tipos? Tenemos dos casos: el de los **títulos** sobre las Heros...

Color system



The new age of EXPLORATION



The most visited ATTRACTIONS



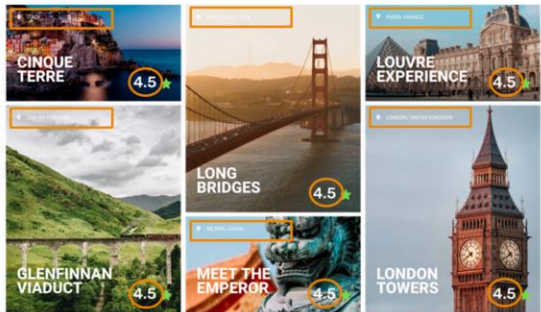
Visiting the CINQUE TERRE, ITALY



title_on-image

...y sobre las cards, por un lado. Se trata del mismo color en ambos casos, y seguramente la diseñadora lo haya pensado como concepto **"título sobre imagen"** (así que podemos llamarle **así** a ese concepto de color y no separarlo en dos, en principio al menos)

Color system



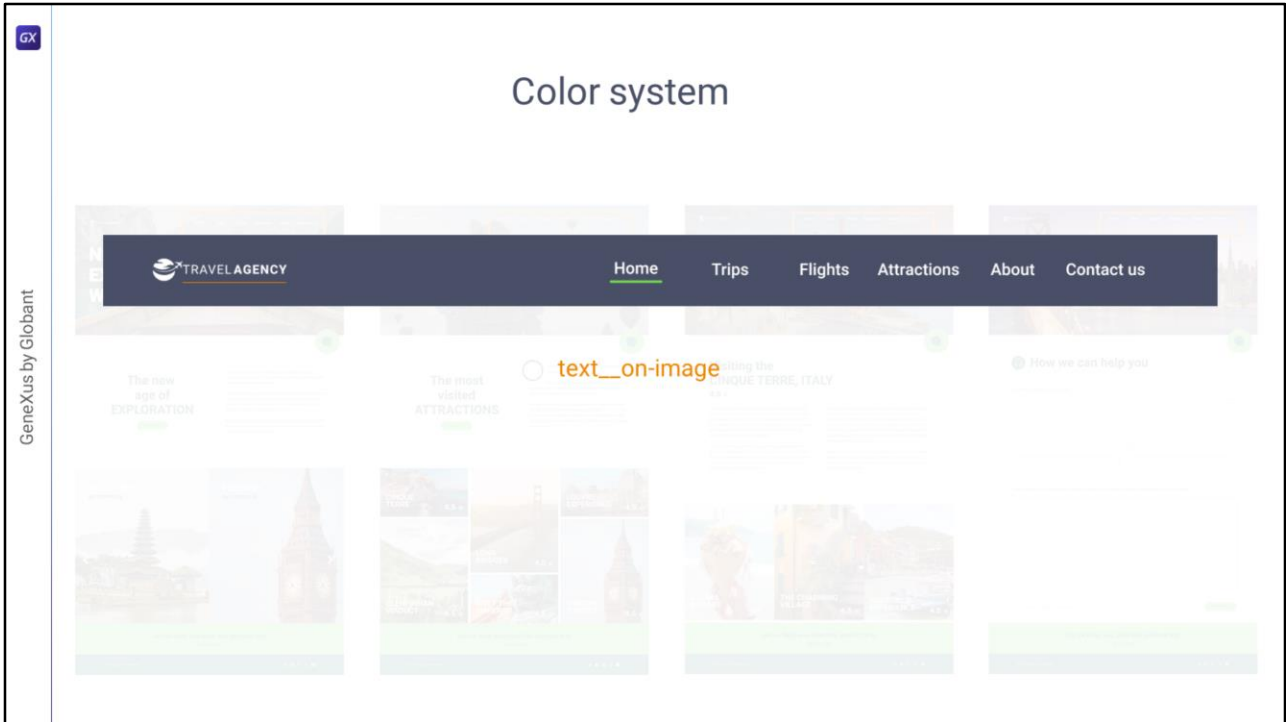
title_on-image

Using the
Color System



text_on-image

Y, por otro lado, el segundo caso que podemos individualizar es el concepto “**texto** sobre imagen”, para todos estos otros casos: los del rating sobre estas cards y estas otras, y también para la ubicación de la atracción turística.



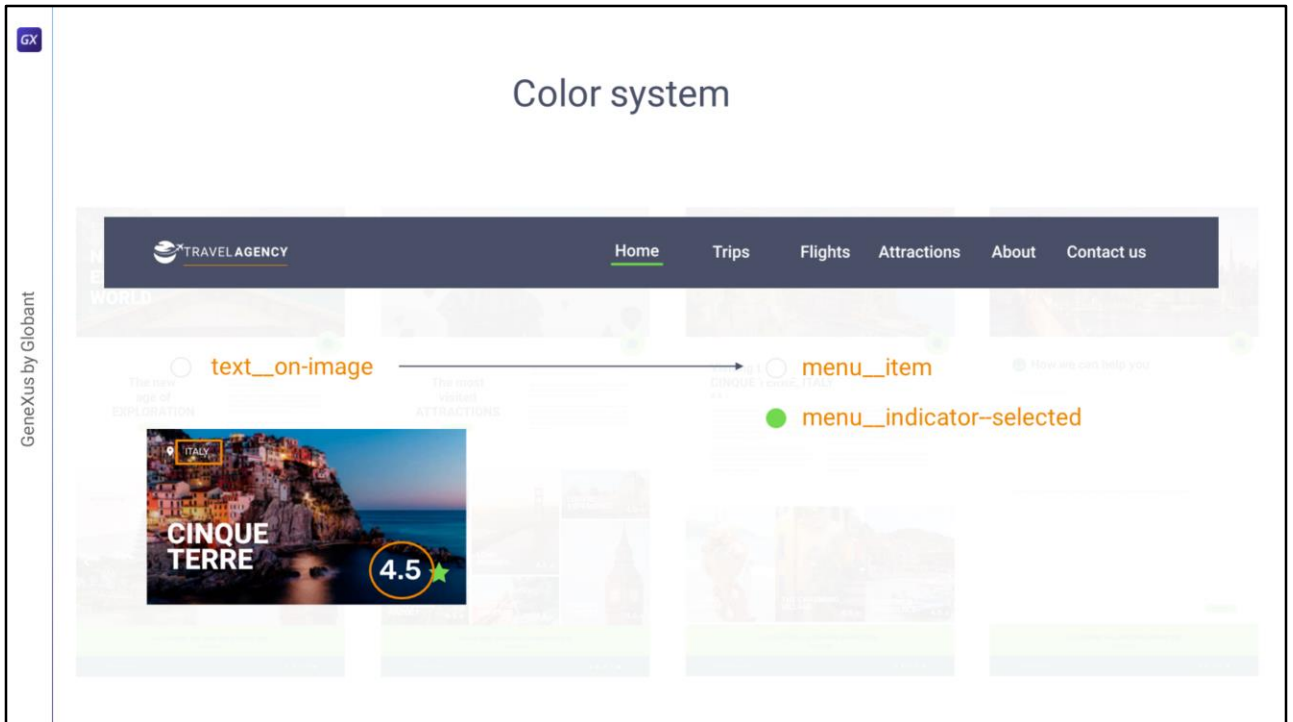
¿Y para los ítems del menú y estas dos palabras del logo? Podríamos utilizar, en principio, ese mismo concepto.

Color system

Primitive/Root				Alias/Semantic		
Region	Name	Value	Opacity	Region	Name	Value
Primitive	green100	#A7E491	100%	Application	primary	green200
	green200	#73D94F	100%		secondary	green300
	green300	#015547	100%		primary--highlighted	green100
Neutral	Gray00	#FFFFFF	100%	Background	surface	gray00
	Gray200	#C1C1C1	100%		On_Colors	title_on-surface
	Gray300	#D2D2D2	100%	title_on-image		gray00
	Gray600	#616161	100%	text_on-surface		gray600
	Gray800	#26262C	100%	text_on-primary		secondary
	opacity	#191819	33%	text_on-secondary	gray200	
				text_on-image	gray00	

Con todo esto que hemos analizado ya tenemos el nivel semántico mínimo.

¿Necesitaremos agregar un tercer nivel, específico? Y de ser así, ¿qué tan específico? ¿o será que con esto ya modelamos todo el sistema?

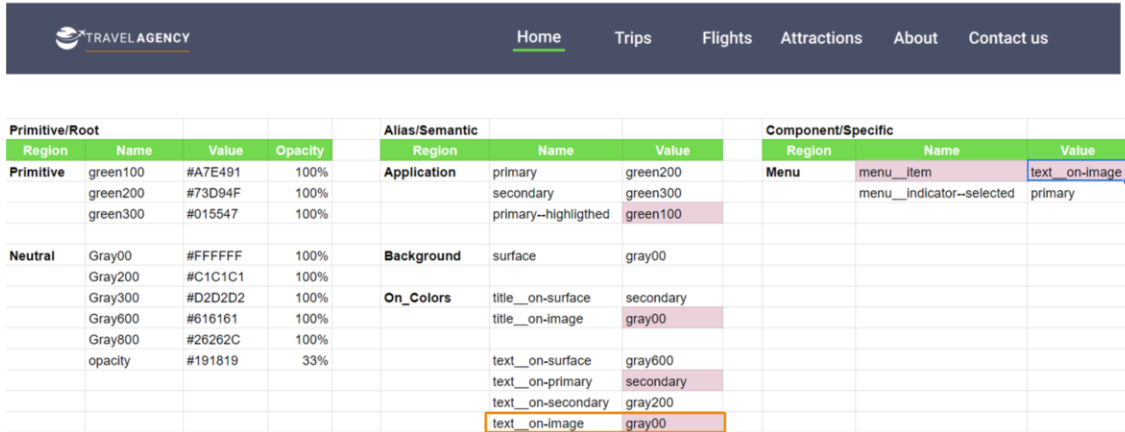


Bueno, por ejemplo, como dijimos, en principio podríamos utilizar para los ítems del menú el mismo color que para cualquier otro texto común que se coloque sobre una imagen, como estos otros. Es decir, el `text_on-image`.

Pero también podríamos pensar que el color de un ítem del menú tiene una especificidad tal que puede ameritar asignarle un concepto de color propio. Pensemos que en este diseño estamos implementando el menú siempre sobre la imagen Hero, pero podría ser que cambiáramos más adelante esa decisión. O incluso que se le quiera dar otro color al menú, por lo cual necesitaremos tenerlo independizado del resto de los usos de "textos sobre imagen".

Este será un concepto específico del componente menú. Incluso debería agregar el color de la barra que aparecerá indicando la opción seleccionada, como vemos acá, que vemos que asume este color verde, el primario.

Color system

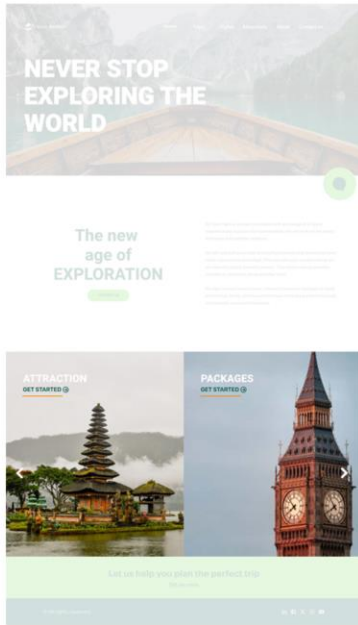


Primitive/Root				Alias/Semantic			Component/Specific		
Region	Name	Value	Opacity	Region	Name	Value	Region	Name	Value
Primitive	green100	#A7E491	100%	Application	primary	green200	Menu	menu_item	text_on-image
	green200	#73D94F	100%		secondary	green300		menu_indicator--selected	primary
	green300	#015547	100%		primary--highlighted	green100			
Neutral	Gray00	#FFFFFF	100%	Background	surface	gray00			
	Gray200	#C1C1C1	100%						
	Gray300	#D2D2D2	100%	On_Colors	title_on-surface	secondary			
	Gray600	#616161	100%		title_on-image	gray00			
	Gray800	#26262C	100%		text_on-surface	gray600			
	opacity	#191819	33%		text_on-primary	secondary			
					text_on-secondary	gray200			
			text_on-image	gray00					

Entonces agregaríamos el tercer nivel, porque, si bien estos colores solo aplicarán a un componente por lo que no sería tan indispensable aislarlos como tokens, hacerlo mejora la mantenibilidad del sistema y su consistencia.

Dejé marcado el primero porque en realidad, observemos que le asigné como valor el mismo que el del token `text_on-image`, lo que está indicando claramente que en principio no sería necesario tenerlo independizado.

Es el tipo de decisiones que hay que ir tomando, pensando en un equilibrio. Para un sistema pequeño como el nuestro, no se ve ninguna desventaja en principio en esta especialización, pero conforme el sistema crece, la proliferación de casos particulares claramente complejiza y degrada. Así que, sugerencia, ¡equilibrio!



Color system

Alias/Semantic			Component/Specific		
Region	Name	Value	Region	Name	Value
Application	primary	green200	Menu	menu_item	text_on-image
	secondary	green300		menu_indicator--selected	primary
	primary--highlighted	green100			
Background	surface	gray00			
On_Colors	title__on-surface	secondary			
	title__on-image	gray00			
	text__on-surface	gray600			
	text__on-primary	secondary			
	text__on-secondary	gray200			
	text__on-image	gray00			

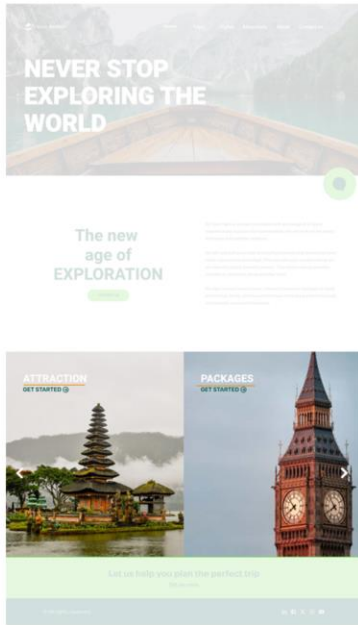
● text-alt__on-image

● card-home__subtitle

Por ejemplo, no hemos modelado el color de este texto. Es un texto sobre una imagen, pero no el blanco que teníamos para los otros casos de texto sobre imagen.

Por lo que podríamos definir un segundo token, alternativo, para texto sobre imagen, en el nivel 2, semántico. O podríamos definirlo específicamente como a nivel del componente donde lo encontraremos, al que nombraré card-home. Como en ese componente hace de subtítulo, le llamaré así.

Tomé esta decisión porque el token general no tiene demasiado sentido dado que no parece ser general en absoluto, sino ser completamente específico a este caso.



Color system

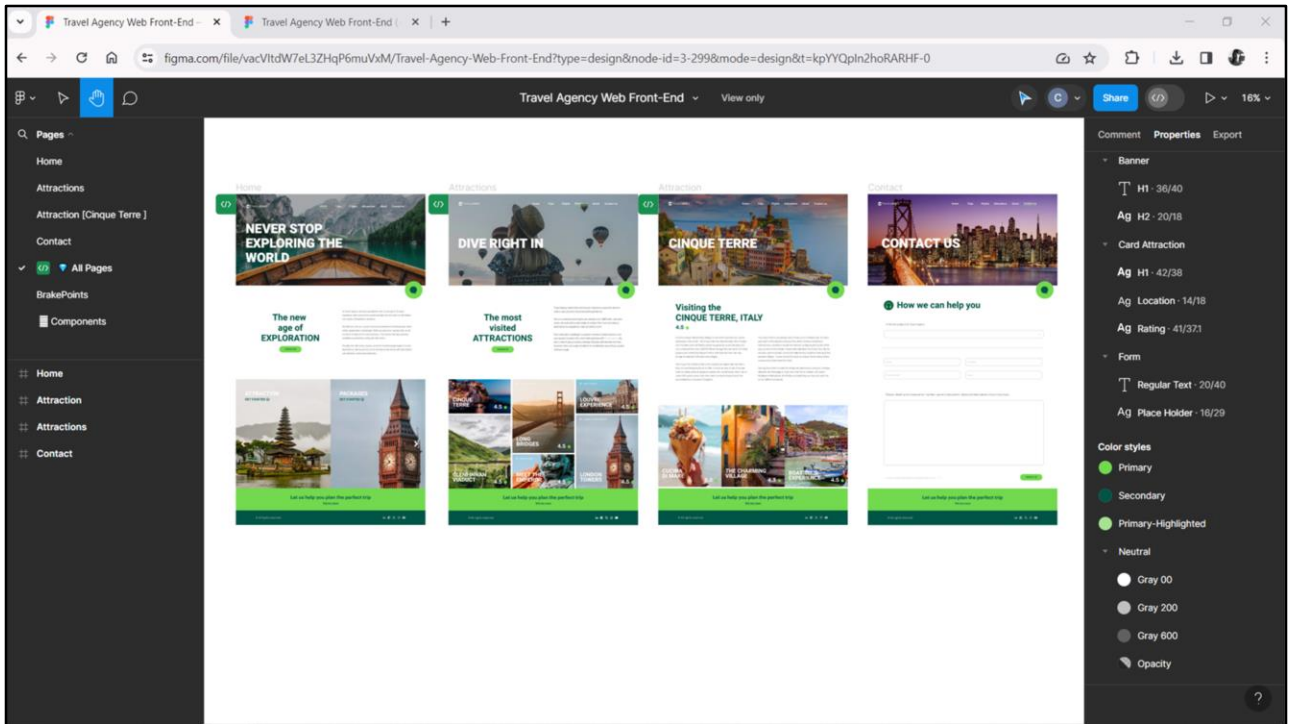
Alias/Semantic			Component/Specific		
Region	Name	Value	Region	Name	Value
Application	primary	green200	Menu	menu__item	text__on-image
	secondary	green300		menu__indicator--selected	primary
	primary--highlighed	green100	Card-Home	card-home__title	gray00
Background	surface	gray00		card-home__subtitle	secondary
	On_Colors	title__on-surface	secondary		
title__on-image		gray00			
text__on-surface		gray600			
text__on-primary		secondary			
text__on-secondary		gray200			
	text__on-image	gray00			

● card-home__subtitle

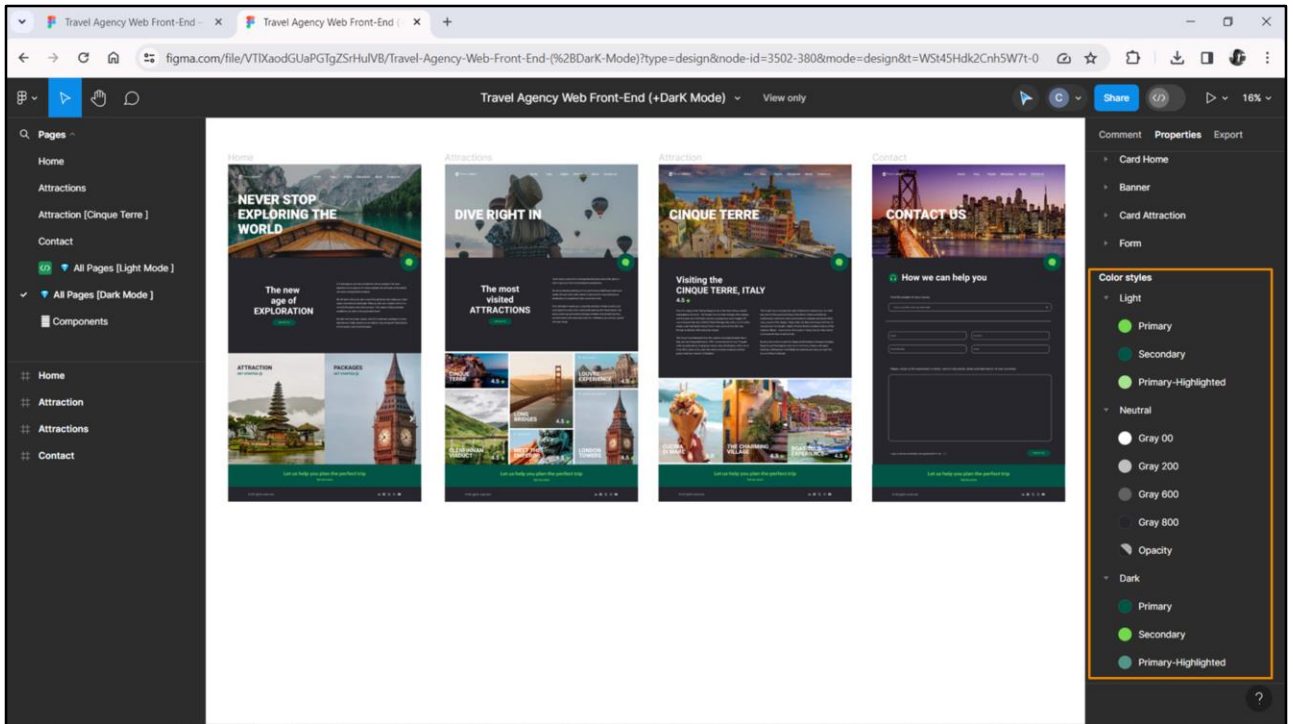
Entonces allí lo agregué, con el valor del token secondary.

Podrán preguntarse, con muy buen tino, por otra parte, por qué para este elemento especifiqué un token, card-home__title, si ya habíamos visto que me servía el general, title__on-image.

Podría parecer que esto se debió a que ya que tuve que definir uno específico para el subtítulo, entonces por consistencia sería buena idea definir uno para el título. Sin embargo... fue por algo mucho más fuerte que eso... y que tiene que ver con algo que hasta el momento no habíamos contemplado.



Que es esto... Este es el modo Light...



Este es el modo Dark.

Chechu me armó otro archivo, distinto de aquel con el que veníamos trabajando, donde agregó este modo.

Si hacemos clic aquí abajo, y nos posicionamos sobre las propiedades, vemos sobre los estilos de color que teníamos que aparecen ahora los del modo Dark.

Vemos que si bien este es el Primary Light, el Primary Dark es el que en este otro es el secundario. Y para el que es en modo Dark secundario va a ser el que en modo Light es el primario.

Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	primary
	green300	#015547		100%		primary--highlighted	green100	gray100			
Neutral	Gray00	#FFFFFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00
	Gray200	#C1C1C1		100%						card-home_subtitle	secondary
	Gray300	#D2D2D2		100%	On_Colors	title_on-surface	secondary	gray00			
	Gray600	#616161		100%		title_on-image	gray00	gray00			
	Gray800	#26262C		100%		text_on-surface	gray600	gray00			
opacity	#191819		33%	text_on-primary	secondary	secondary					
					text_on-secondary	gray200	primary?				
					text_on-image	gray00	gray00				

Entonces en la planilla, para cada token voy a tener que especificar otra columna para el valor que asume el token para el modo Dark.

La que teníamos, que decía Value (ahora lo cambié por Light) correspondía al modo Light (que era en el que veníamos trabajando), y tenemos que hacer estas asociaciones, para cada token especificar cuál va a ser su valor en el modo Dark también.

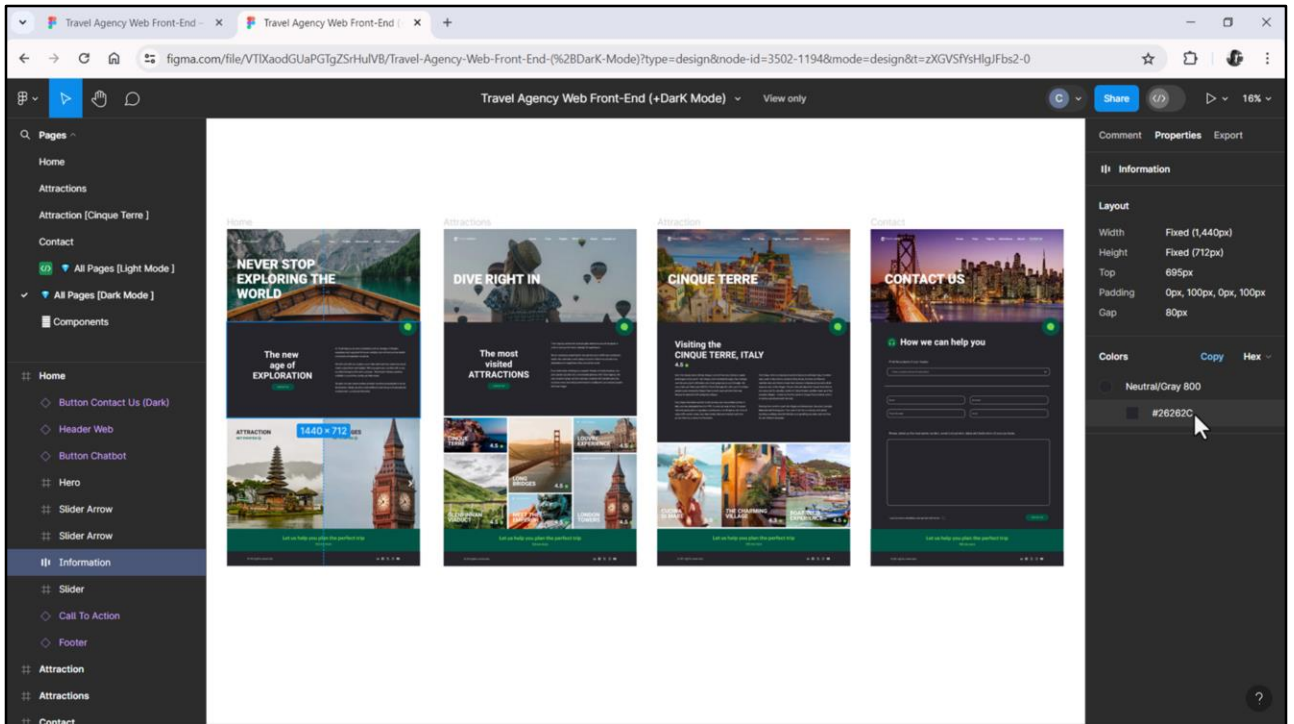
Light Value	Value Dark	Opacity	Alias/Semantic	Region	Name	Light	Dark	Component/Specific	
#A7E491	#54958A	100%	Application	primary		green200	green300	Menu	
#73D94F		100%		secondary		green300	green200		menu_item
#015547		100%		primary--highlighted		green100	green100		menu_indicator--selected
#FFFFFF		100%	Background	surface		gray00	gray800	Card-Home	
#C1C1C1		100%							card-home_title
#D2D2D2		100%	On_Colors	title_on-surface		secondary	gray00	card-home_subtitle	
#616161		100%		title_on-image		gray00	gray00	secondary	
#26262C		100%		text_on-surface		gray600	gray00	primary	
#191819		33%		text_on-primary		secondary	secondary	secondary	
				text_on-secondary		gray200	primary?	primary?	
				text_on-image		gray00	gray00	primary	

¿Entonces, por qué les decía esto? Porque si venimos acá... tendríamos que inspeccionar ahora todo esto para ver cómo varía el color de acuerdo al token. Esto que dejé marcado es cuando es el mismo valor en ambos modos.

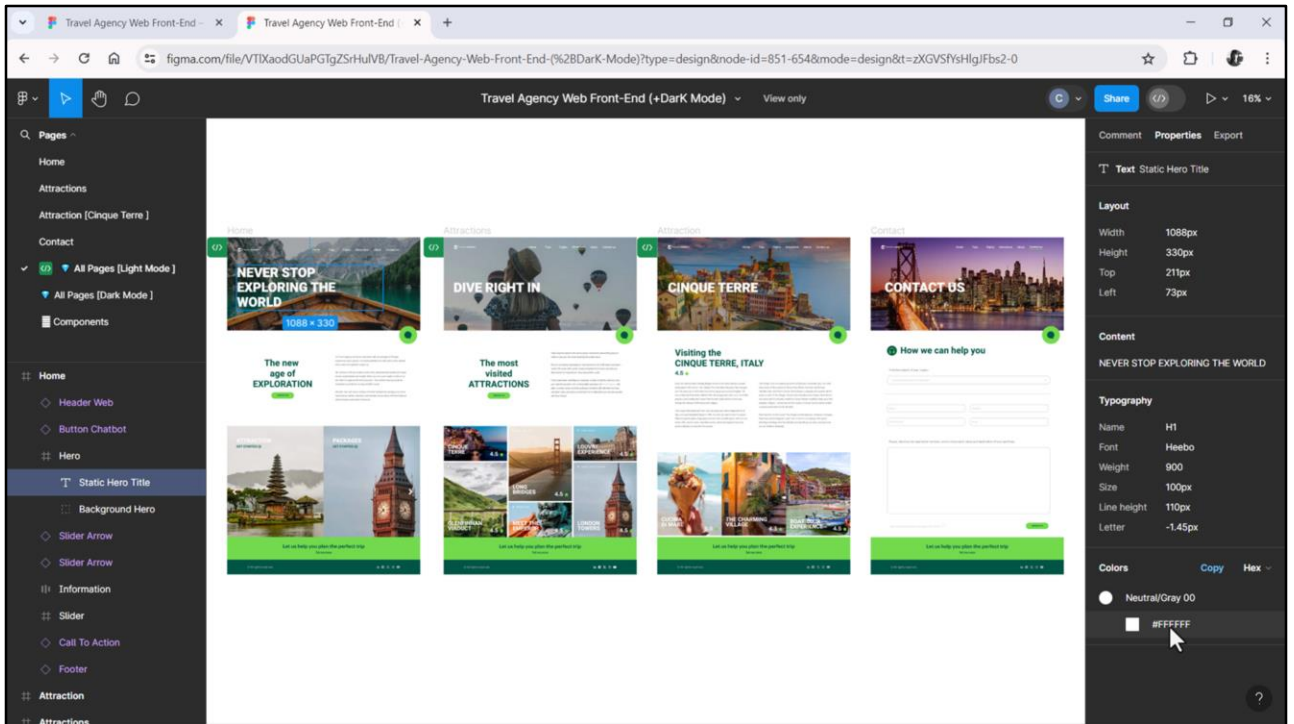
Entonces lo que no está marcado es que varía entre un modo y el otro.

Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	primary
	green300	#015547		100%		primary--highlighted	green100	green100			
Neutral	Gray00	#FFFFFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00
	Gray200	#C1C1C1		100%		On_Colors	title_on-surface	secondary		gray00	card-home_subtitle
	Gray300	#D2D2D2		100%	title_on-image		gray00	gray00			
	Gray600	#616161		100%	text_on-surface		gray600	gray00			
	Gray800	#26262C		100%	text_on-primary	secondary	secondary				
opacity	#191819		33%	text_on-secondary	gray200	primary?					
					text_on-image	gray00	gray00				

Por ejemplo, vamos a empezar por la **surface**, la superficie: mientras que en el modo Light es este blanco, el gray00, que lo tenemos aquí, para el modo Dark va a ser este gray800.

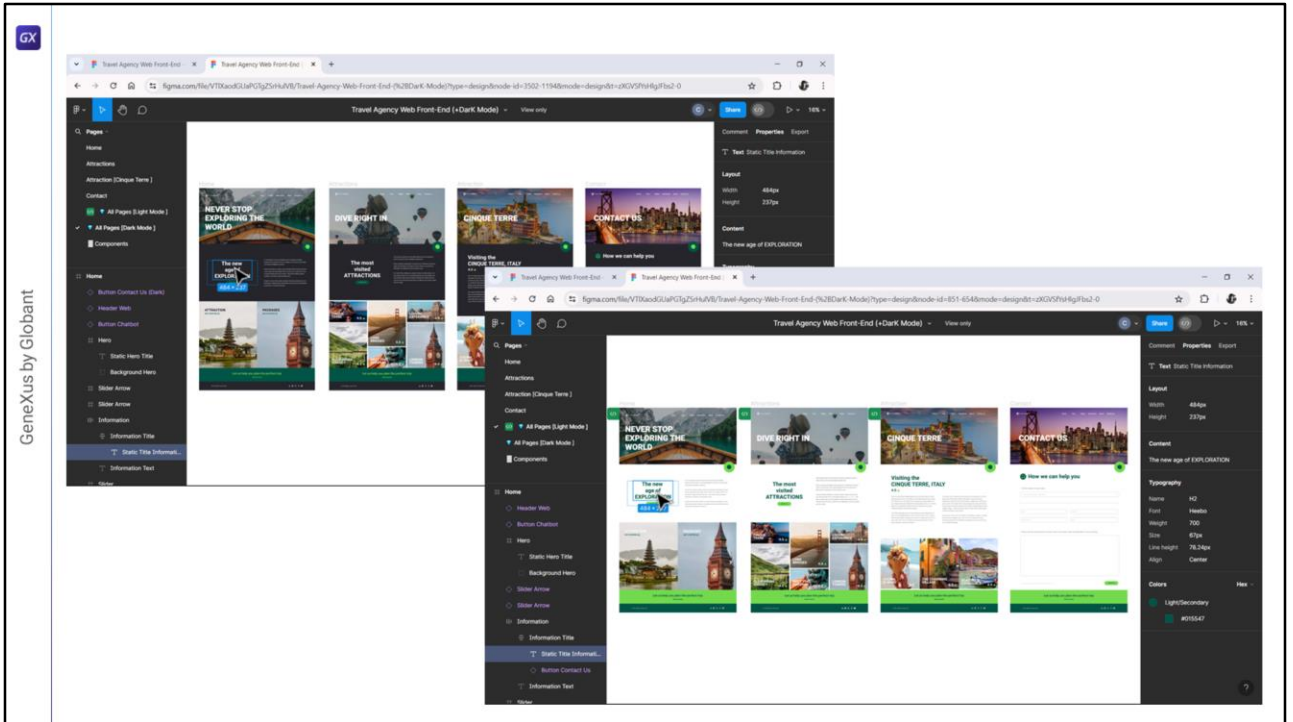


Y lo podemos ver claramente posicionándonos allí... y vemos: gray800... mientras que para el modo Light ya sabíamos que era el gray00.



Como les decía, no todo cambia.

Por ejemplo, el color del título sobre imagen que correspondía a nuestro token `title__on-image` es un `gray00` para la versión Light ...y para la Dark también. No se modifica.



En cambio, veamos qué pasa con el título sobre surface, `title__on-surface`. Acá en el modo Dark podemos ver que es un blanco, es el `gray00`, mientras que en el modo Light era el `secondary`.

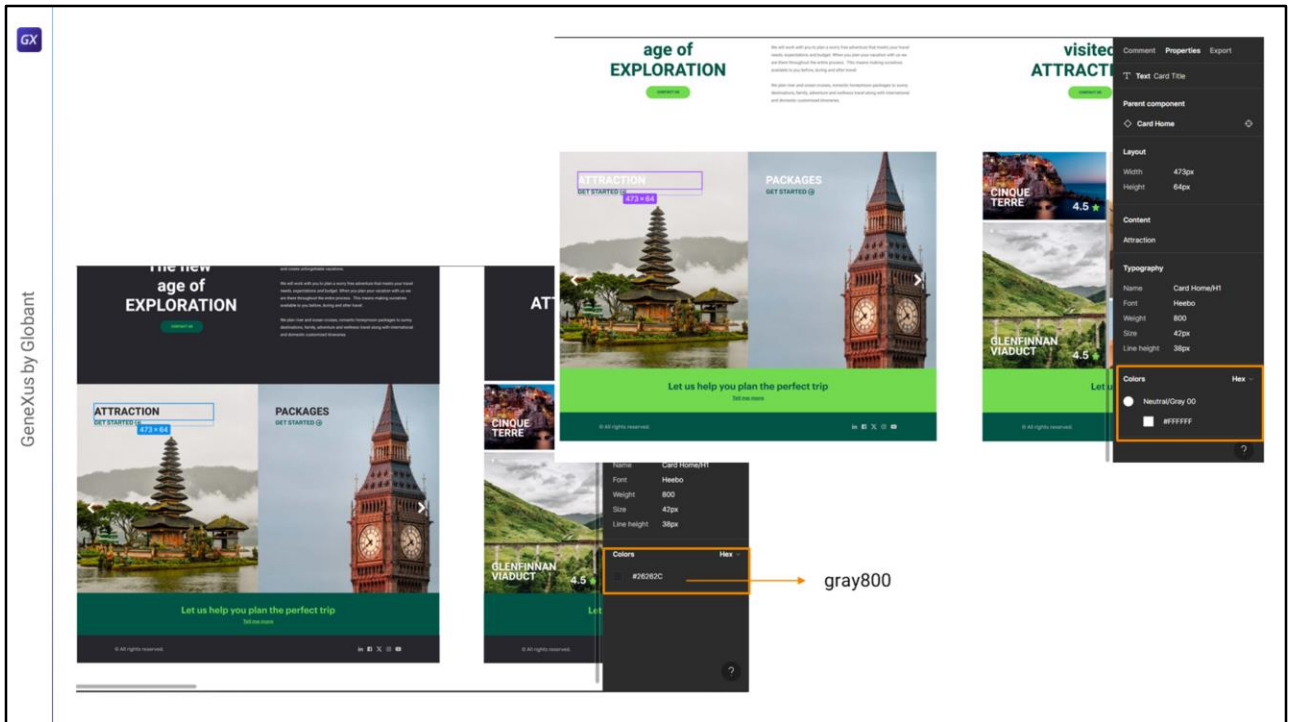
Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	primary
	green300	#015547		100%		primary--highlighted	green100	green100			
Neutral	Gray00	#FFFFFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00
	Gray200	#C1C1C1		100%		title__on-surface	secondary	gray00		card-home_subtitle	secondary
	Gray300	#D2D2D2		100%	On_Colors	title__on-image	gray00	gray00			
	Gray600	#616161		100%		text__on-surface	gray600	gray00			
	Gray800	#26262C		100%		text__on-primary	secondary	secondary			
opacity	#191819		33%	text__on-secondary	gray200	primary?					
					text__on-image	gray00	gray00				

Entonces esas cosas las vemos acá: surface, title__on-surface, secondary, gray00.

Mientras que title__on-image, el primero que habíamos visto, mantiene el mismo color blanco en ambos casos.

Value		Alias/Semantic				Component/Specific			
Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light	Dark
34958A	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	Card-Home	card-home_title	gray00	gray800
		Background	surface	gray00	gray800		card-home_subtitle	secondary	primary
	100%	On_Colors	title_on-surface	secondary	gray00				
	100%		title_on-image	gray00	gray00				
	100%		text_on-surface	gray600	gray00				
	33%		text_on-primary	secondary	secondary				
			text_on-secondary	gray200	primary?				
			text_on-image	gray00	gray00				

Ahora, vamos a lo que les preguntaba, que fue lo que hizo surgir esta introducción del modo Dark: ¿por qué había aislado también un token para el card-home__title? En lugar de utilizar, por ejemplo, el title__on-image.



Porque mientras este title__on-image, e incluso este title__on-image no varía (vemos que sigue siendo el mismo blanco)
¿qué pasa con este?

Aquí vemos que es una suerte de color gris oscuro... bueno, observemos que acá Chechu no está utilizando el estilo de color. Estas son inconsistencias que, por supuesto, nos pueden pasar en todo proyecto, ¿no? Pero claramente este color es el mismo que el de la superficie... así que es el gray800.

Bien, pero decíamos, entonces, acá es gray800, mientras que en la modalidad Light es el blanco este.

Entonces vemos que este color, el color de este título, tiene un comportamiento diferente que el color de los títulos sobre las imágenes que se usan para la generalidad de la aplicación. Entonces es un caso particular. Y es un caso particular que sólo aplica a las cards del Home, no a otra cosa, y es por eso que tomo esa decisión...

Tokens Travel Agency - Google

docs.google.com/spreadsheets/d/1oMvMncna8ZASN5_iTG6pcap3yiArNcfvMSgV0068e_/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

M7 | fx card-home_title

	D	E	F	G	H	I	J	K	L	M	N	O	P
1													
2													
3		Value Dark	Opacity		Alias/Semantic				Component/Specific				
4		34958A	100%		Region	Name	Light	Dark	Region	Name	Light	Dark	
5			100%		Application	primary	green200	green300	Menu	menu_item	text_on-image	text-on-image	
6			100%			secondary	green300	green200		menu_indicator--selected	primary	secondary	
7						primary--highlighted	green100	green100	Card-Home	card-home_title	gray00	gray800	
8			100%		Background	surface	gray00	gray800		card-home_subtl	secondary	primary	
9			100%										
10			100%		On_Colors	title__on-surface	secondary	gray00					
11			100%			title__on-image	gray00	gray00					
12			100%										
13			33%			text__on-surface	gray600	gray00					
14						text__on-primary	secondary	secondary					
15						text__on-secondary	gray200	primary?					
16						text__on-image	gray00	gray00					
17													
18													
19													
20													
21													
22													

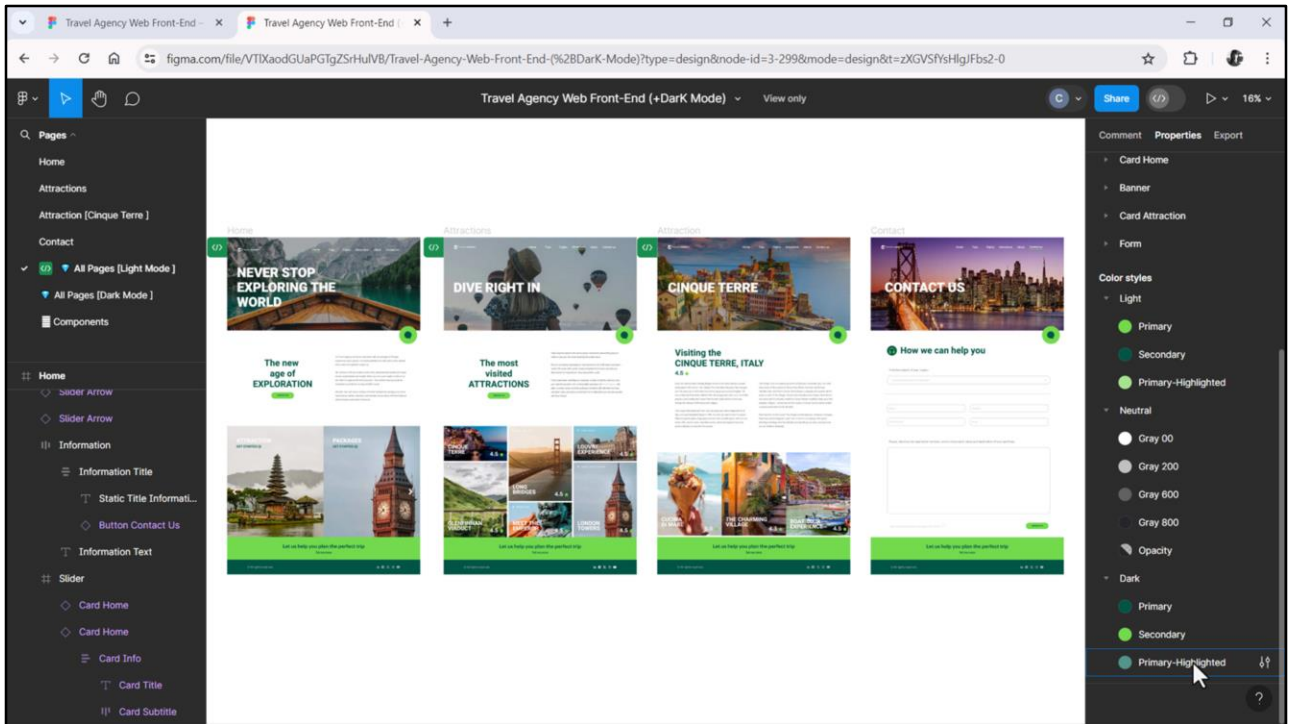
+ Text Styles Text Styles + Multiexperience Colors Styles Colors Styles + Dark Mode Color tokens Text Classes

...de definirlo como un token específico del componente.

Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	primary
	green300	#015547		100%		primary--highlighted	green100	green100			
Neutral	Gray00	#FFFFFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00
	Gray200	#C1C1C1		100%		On_Colors	title_on-surface	secondary		gray00	card-home_subtitle
	Gray300	#D2D2D2		100%	title_on-image		gray00	gray00			
	Gray600	#616161		100%	text_on-surface		gray600	gray00			
	Gray800	#26262C		100%	text_on-primary	secondary	secondary				
opacity	#191819		33%	text_on-secondary	gray200	primary?					
					text_on-image	gray00	gray00				

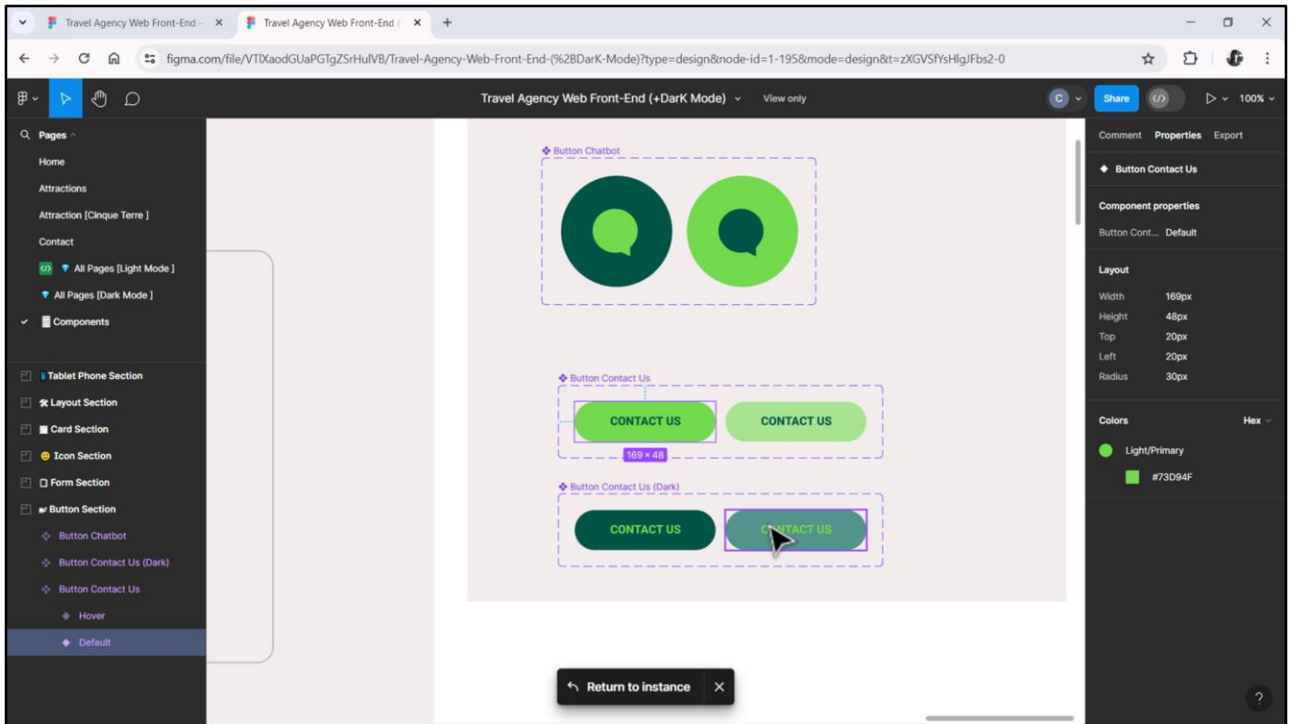
Veamos también esto que les decía antes, que el token primary va a ser un tipo de verde, el verde clarito, el verde 200, que es el verde clarito, para el modo Light; y el verde oscuro para el Dark. Y viceversa, se invierten, para el token secondary.

Aquí observemos que el primary--highlighted que está utilizando el mismo token primitivo green100, le llamé green100... estos no varían por modo Light y Dark, van a tener todos el mismo valor para los dos modos, salvo este primero que sí lo hago variar por modo Light y Dark. ¿Y este a qué color corresponde, qué color es este?



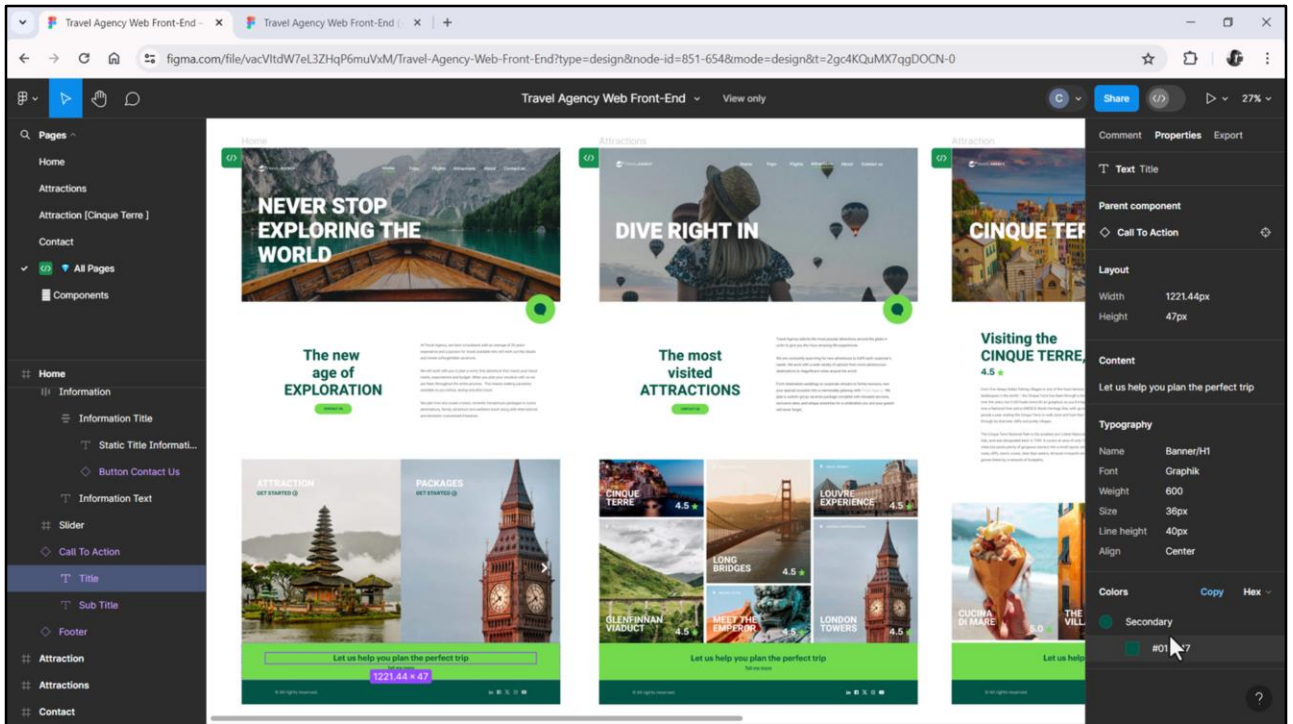
El primary-highlighted. Vean que tengo este primary-highlighted para el modo Light, y para el modo Dark es este otro, que es más claro que el primary, en ambos casos.

Que era el que usábamos, justamente, para que el botón...



...vamos a ir al botón para mostrarles... para hacer las variantes del botón cuando se hace hover sobre él. En el otro archivo solamente teníamos esto, y en este tenemos las dos versiones: la del Light y la del Dark.

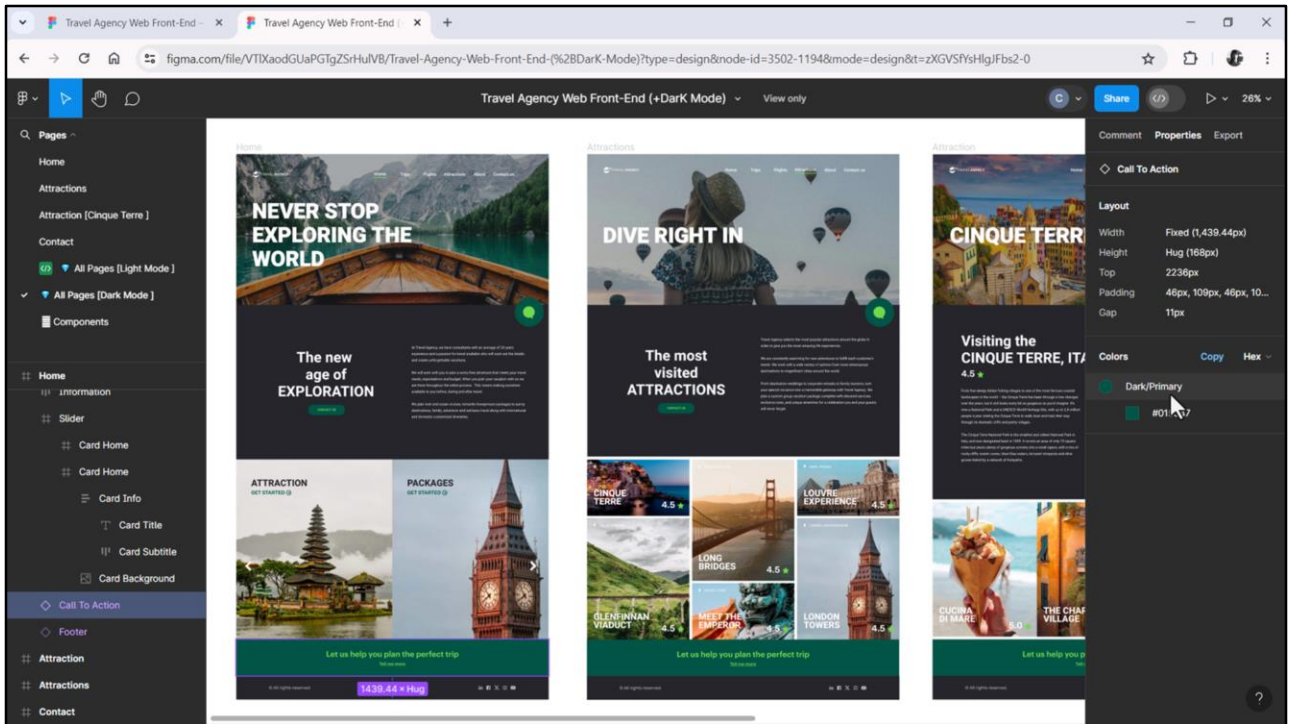
Así como para el chatbot (que también se los había contado en el video en el que hablamos de las imágenes, pero allí todavía no se los había presentado porque no tenía todavía este archivo, no les quería mostrar todavía este archivo que me había pasado la diseñadora).



Para que nos sea más fácil la comparación voy a tener los dos archivos a la vez. Este es el anterior, que tiene el modo Light únicamente, y este es el que tiene los dos modos, y me voy a quedar posicionada en el modo Dark, para poder comparar fácil.

Bueno, ¿qué es lo que tendríamos que ver? Que con los tokens que tenemos nos alcance, para ver si no tenemos que crear otros tokens, tal como nos pasó con el token que habíamos asociado a la card-home para el título.

Entonces, por ejemplo, quiero ver qué pasa con el primary como fondo. Teníamos que para el modo Light habíamos definido que con el primary, con el verde este, luego definíamos un token text_on-primary, que servía tanto para el caso del botón como para el banner. Porque el color de lo que se superponía a ese color primary, es decir, text_on-primary, era siempre el mismo. Que en este caso, en el modo Light, sabíamos que el primary era este verde, el on-primary correspondía al secondary.

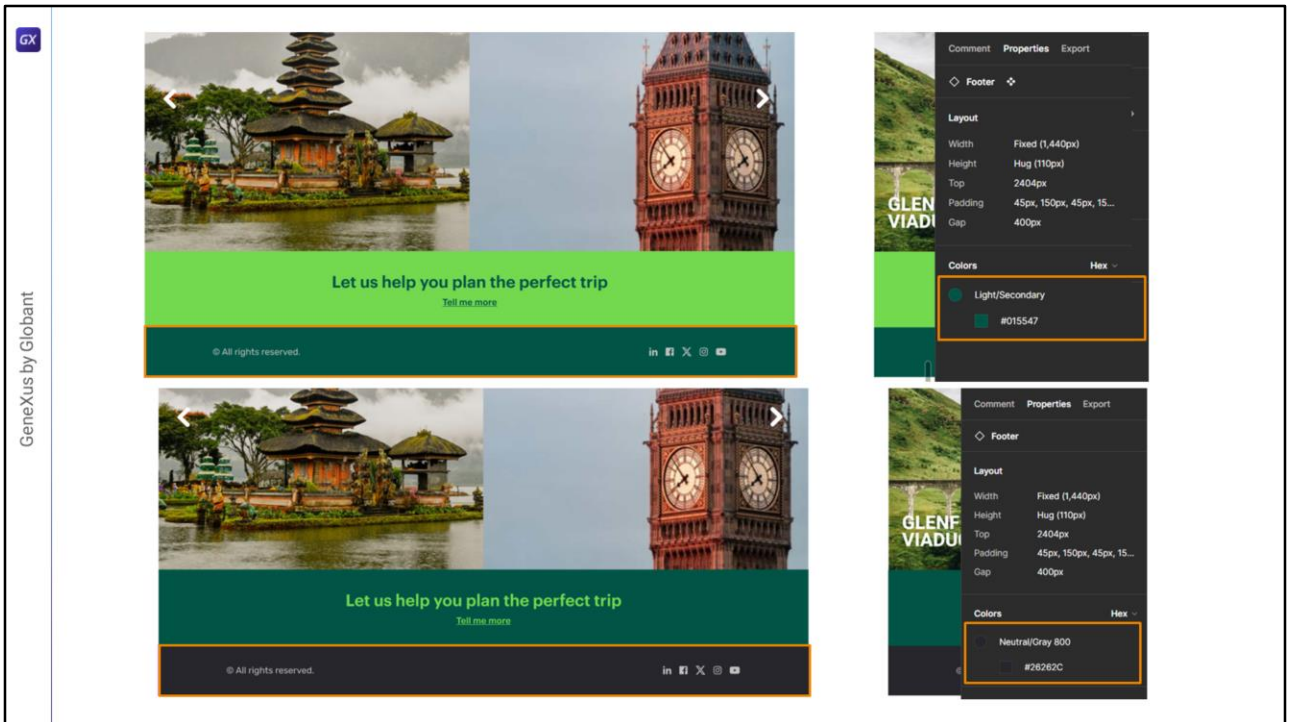


¿Qué pasa con el modo Dark? Bueno, vemos que se invierten: lo que en el otro caso era primary aquí es secondary y viceversa. Y eso nos funciona, ¿ven? Porque entonces ¿qué es lo que tenemos que colocar acá?

Sabemos que en este caso el primary es este, y el secondary es en el que en el otro caso era el primary.

Alias/Semantic					Component/Specific			
Region	Name	Light	Dark	Region	Name	Light	Dark	
Application	primary	green200	green300	Menu	menu_item	text_on-image	text-on-image	
	secondary	green300	green200		menu_indicator--selected	primary	secondary	
	primary--highlighted	green100	green100	Card-Home	card-home_title	gray00	gray800	
Background	surface	gray00	gray800		card-home_subtitle	secondary	primary	
On_Colors	title_on-surface	secondary	gray00					
	title_on-image	gray00	gray00					
	text_on-surface	gray600	gray00					
	text_on-primary	secondary	secondary					
	text_on-secondary	gray200	primary?					
	text_on-image	gray00	gray00					

Parece un trabalenguas pero aquí es donde se entiende por qué le pusimos el mismo valor, cuando en los hechos no es el mismo. Lo que pasa es que este secondary está remitiendo a este token secondary que en realidad varía entre un modo y el otro.



Ahora bien, veamos que hay una excepción a esta conversión de primary en secondary y de secondary en primary entre los modos Light y Dark.

Si miramos acá, vemos por ejemplo en el chatbot que lo que es primario y secundario sí se convierte en su inversión. Lo mismo para el botón, habíamos visto, lo mismo para el banner.

Ahora, ¿qué pasa con el footer? Aquí no se está cumpliendo esa inversión. Porque el background-color del footer que es este verde del secondary no se está convirtiendo en el verde primary del modo Light, es decir en este verde claro... si se siguiera la misma regla tendría que ser este color. Pero está siendo este otro.

Entonces podemos ver claramente que se trata de una excepción a esta regla (que de cumplirse debería indicar que el secondary de aquí debería ser el secondary de aquí...) Que el secondary de aquí ¿cuál es? Este. Y no está sucediendo.

¿Qué color está asumiendo en cambio? El gray800.

Tokens Travel Agency - Google

docs.google.com/spreadsheets/d/1oMvInca8ZASn5_iTG6pcap3yiArNcfvMSgVO068e_/edit#gid=1337893737

GeneXus DL Portal Issues

Tokens Travel Agency

File Edit View Insert Format Data Tools Extensions Help

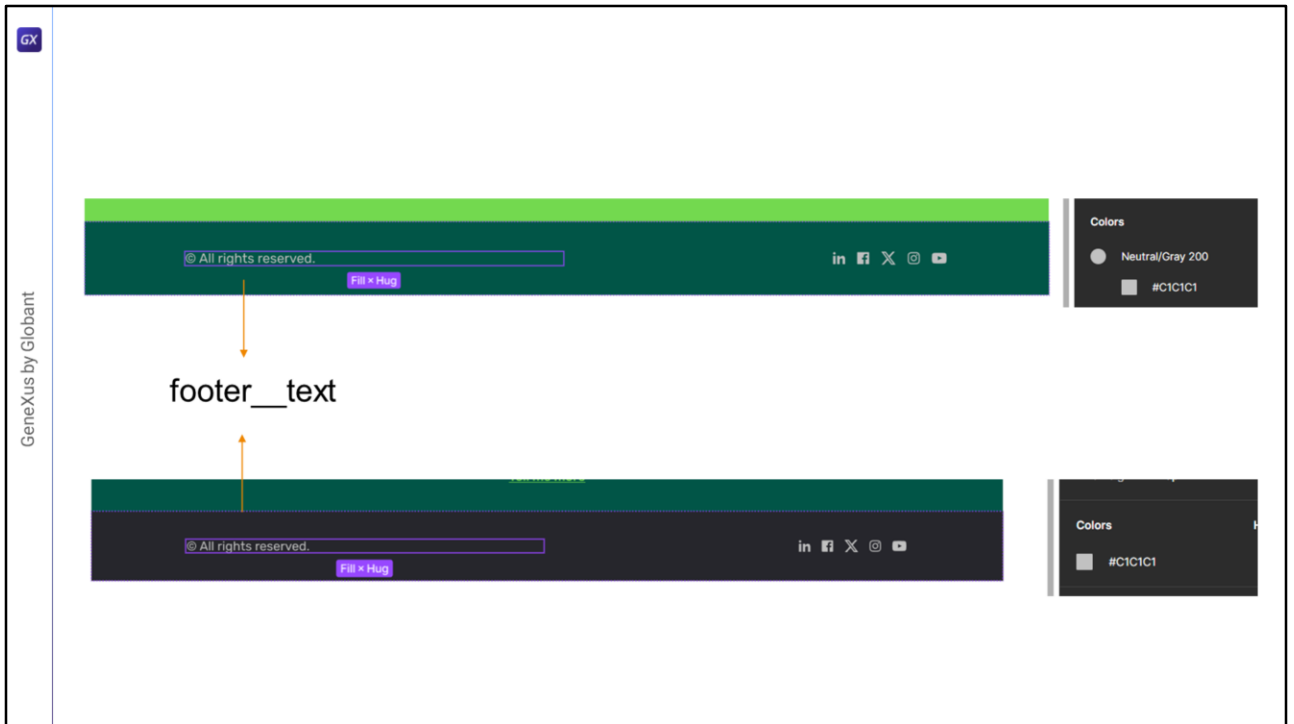
100% 123 Default... 10 B I A

O10 gray800

	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1														
2														
3		if Value	Value Dark	Opacity	Alias/Semantic				Component/Specific					
4		#491	#54958A	100%	Region	Name	Light	Dark	Region	Name	Light	Dark		
5		94F		100%	Application	primary	green200	green300	Menu	menu_item	text_on-image	text-on-image		
6		547		100%		secondary	green300	green200		menu_indicator--selected	primary	secondary		
7						primary--highlighted	green100	green100						
8		FFF		100%	Background	surface	gray00	gray800	Card-Home	card-home_title	gray00	gray800		
9		1C1		100%						card-home_subtitle	secondary	primary		
10		2D2		100%	On_Colors	title__on-surface	secondary	gray00		Footer	footer_background-color	secondary	gray800	
11		161		100%		title__on-image	gray00	gray00						
12		62C		100%										
13		819		33%		text__on-surface	gray600	gray00						
14						text__on-primary	secondary	secondary					maybe better would be: surface	
15						text__on-secondary	gray200	primary?						
16						text__on-image	gray00	gray00						
17														
18														
19														
20														
21														
22														

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

Entonces creo un token especial, footer__background-color, justamente para esa excepción. Va a asumir el color secondary del modo Light. Y para el modo Dark va a asumir este gray800.



¿Y qué pasa con el texto sobre ese fondo?

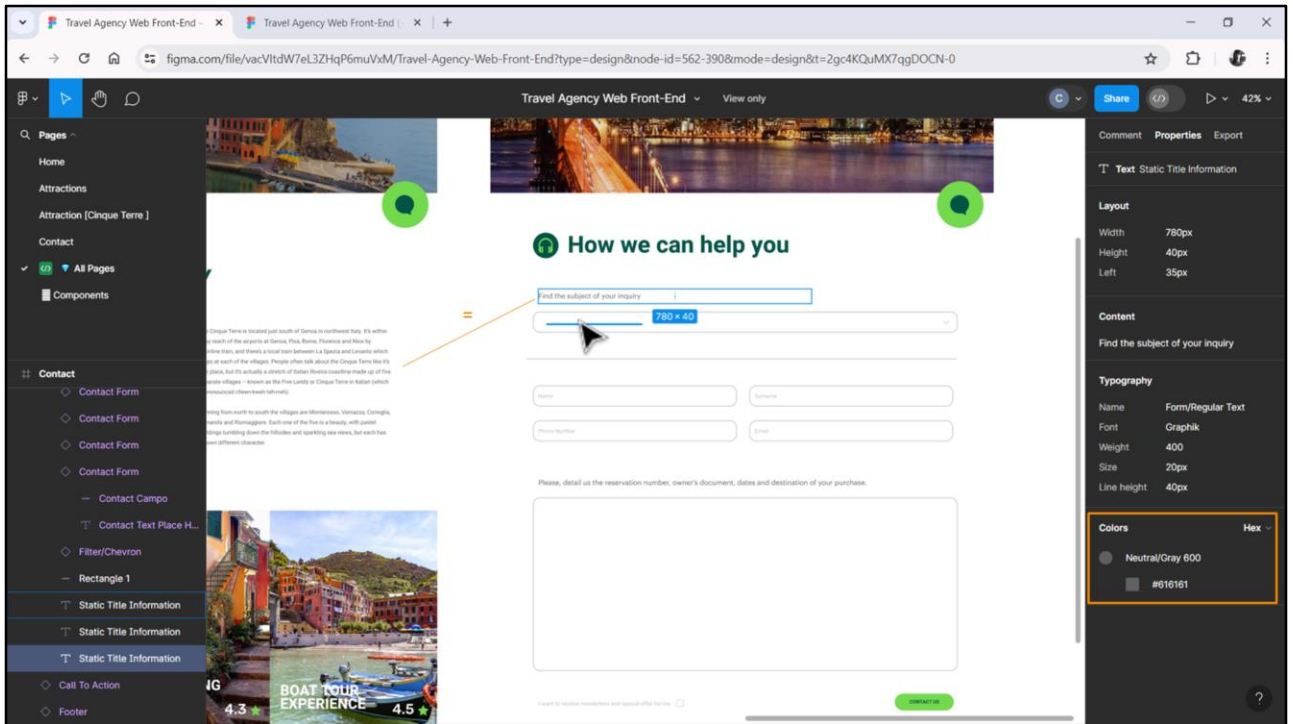
Vemos que se trata de este color... si vamos al modo Light es el mismo, es el gray200. Y por tanto lo agrego como token a este, footer__text.

Alias/Semantic			Component/Specific					
Region	Name	Light	Dark	Region	Name	Light	Dark	
Application	primary	green200	green300	Menu	menu_item	text_on-image	text-on-image	
	secondary	green300	green200		menu_indicator--selected	primary	secondary	
	primary--highlighted	green100	green100					
Background	surface	gray00	gray800	Card-Home	card-home_title	gray00	gray800	
					card-home_subtitle	secondary	primary	
On_Colors	title_on-surface	secondary	gray00	Footer	footer_background-color	secondary	gray800	
	title_on-image	gray00	gray00		footer_text	gray200	gray200	
	text_on-surface	gray600	gray00					
	text_on-primary	secondary	secondary					
	text_on-secondary	gray200	primary?					
	text_on-image	gray00	gray00					

Porque no va a corresponder a text__on-secondary, porque no va a ser, por ejemplo, para el modo Dark secondary el fondo, va a ser este otro, va a ser este color de fondo. Por tanto necesito agregar un token especial.

De hecho text__on-secondary me quedó así... esto es para pensarlo porque no lo estamos usando en verdad. Uno a priori con esta inversión de colores que decíamos, que el primario de uno se convierte en el secundario del otro modo y viceversa, pensaría que en realidad un text__on-secondary debería ser primary en ambos casos, si siguiera esa regla.

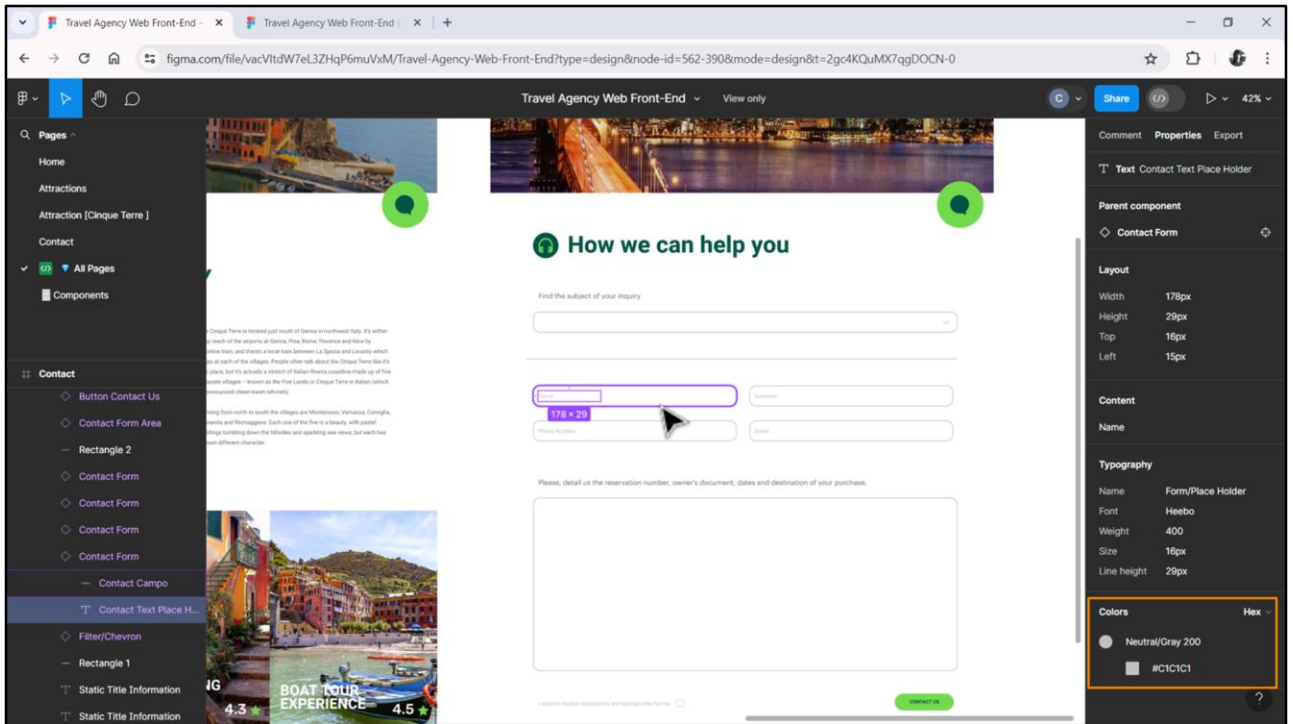
Voy a dejar los dos valores con signo de pregunta por lo que les decía... el único caso por el momento de secondary como fondo es este, el del footer. No tenemos otro caso.



Para terminar de expresar todo nuestro sistema de colores nos está faltando modelar solamente una parte. Vamos a observarla acá: tiene que ver con el panel de Contacto, que es el único panel que tiene campos de entrada. Todos los demás paneles tienen solamente campos de salida.

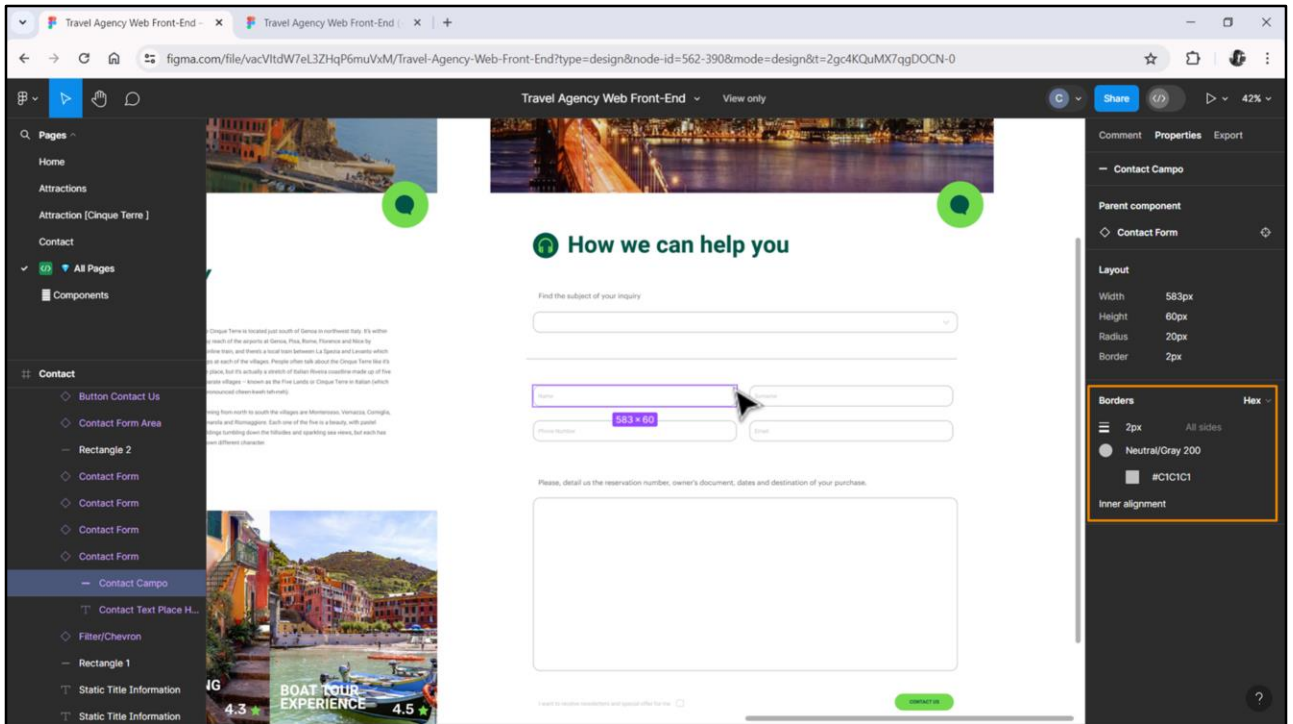
Acá el usuario va a tener que ingresar información y presionar el botón.

Si observamos (lo voy a ampliar un poco más) vemos que este texto que está describiendo lo que el usuario tiene que ingresar en el campo siguiente, asume como color este, gray600, que, en este caso es el mismo que asume cualquier texto sobre la superficie de fondo. Así que le colocaríamos como color el del mismo token que a este. Ahí no hay nada que especializar.



Sin embargo, observemos qué pasa con el texto que aparece acá adentro, que es un texto de sugerencia, es decir, no es la descripción sino que corresponde a indicaciones para el usuario, que una vez que empiece a digitar desaparecen, a diferencia de este que es una descripción que nunca desaparece.

Bueno, si observamos este es un color más claro, no es el gray600, es este gray200. Corresponde a un concepto diferente, por eso esa diferencia de color.

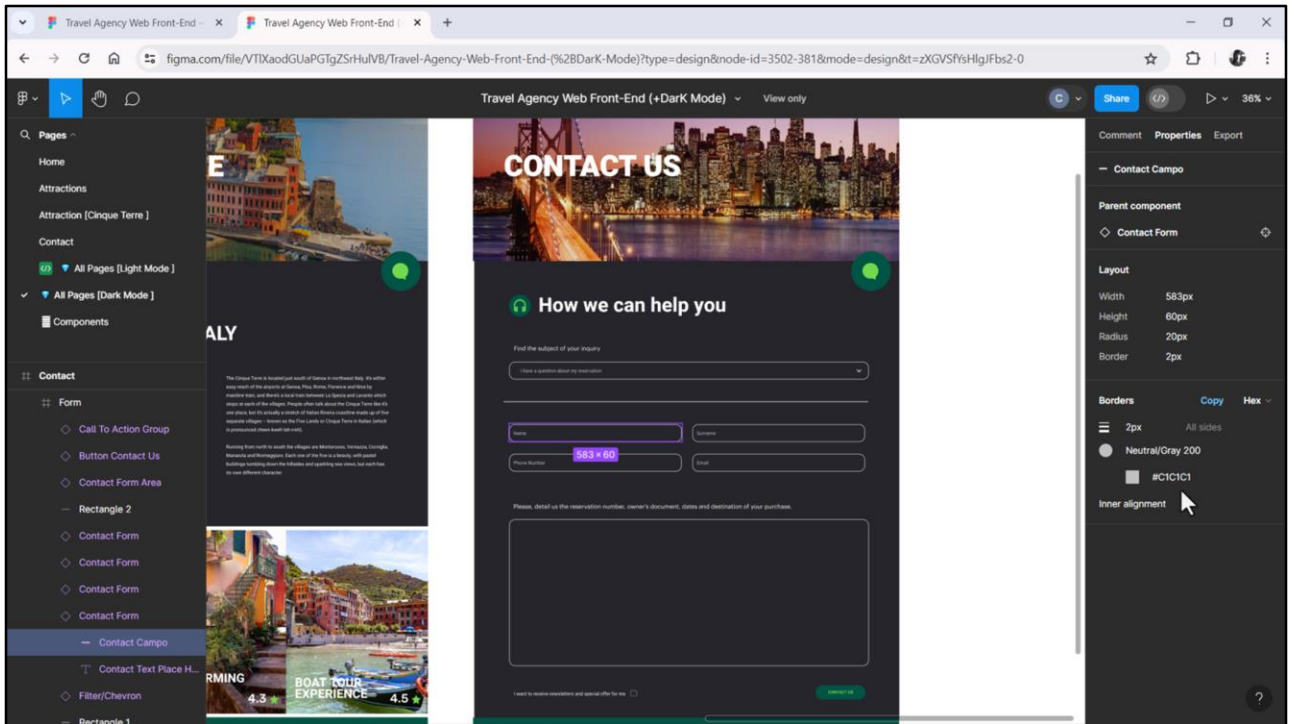


Y además aquí nos aparece otro tipo de color, que es el de los bordes de estos campos. Los otros campos no tenían bordes. Este sí. Y es un gray200 también.

Entonces nos están faltando modelar dos tokens: uno para esos bordes y otros para el texto en estos casos.

Alias/Semantic	Region	Name	Light	Dark	Component/Specific	Region	Name	Light	Dark
Application	primary		green200	green300	Menu	menu_item	text_on-image	text-on-image	
	secondary		green300	green200		menu_indicator--selected	primary	secondary	
	primary--highlighted		green100	green100					
Background	surface		gray00	gray800	Card-Home	card-home_title	gray00	gray800	
						card-home_subtitle	secondary	primary	
On_Colors	title_on-surface		secondary	gray00	Footer	footer_background-color	secondary	gray800	
	title_on-image		gray00	gray00		footer_text	gray200	gray200	
	text_on-surface		gray600	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					

Y es por eso que construyo entonces dos tokens dentro de un componente al que le voy a llamar form, porque es específico, es bien específico. A uno le llamé border-color y al otro text-placeholder, es el texto que va dentro del placeholder.



Veán que va a asumir para el modo Dark, no lo mostré, el mismo valor gray200. Ese es para el borde, y para el texto interno también.

	C	D	E	F	G	H	I	J	K	L	M	N	O
5	D94F		100%			secondary	green300	green200		menu_indicator--selected	primary	secondary	
6	5547		100%			primary--highlighed	green100	green100		Card-Home	card-home_title	gray00	gray800
7										card-home_subtitle	secondary	primary	
8	FFFF		100%		Background	surface	gray00	gray800		Footer	footer_background-color	secondary	gray800
9	C1C1		100%							footer_text	gray200	gray200	
10	D2D2		100%		On_Colors	title_on-surface	secondary	gray00		Form	form_border-color	gray200	gray200
11	6161		100%			title_on-image	gray00	gray00		form_text-placeholder	gray200	gray200	
12	262C		100%			text_on-surface	gray600	gray00		form_text	text_on-surface	text_on-surface	
13	1819		33%			text_on-primary	secondary	secondary		Card-Attractions	card-attraction_title	title-on-image	title-on-image
14						text_on-secondary	gray200?	primary?		card-attraction_text	text-on-image	text-on-image	
15						text_on-image	gray00	gray00		Hero	hero_title	title-on-image	title-on-image
16										Banner	banner_background-color	primary	primary
17										banner_title	secondary	secondary	
18										banner_text	secondary	secondary	
19													
20													
21													
22													
23													
24													
25													
26													

Con esto tendríamos la expresión mínima necesaria del sistema de colores de la aplicación.

Como dije antes, para un sistema tan pequeño como el nuestro, podríamos crear también tokens para estos otros componentes, lo que nos deja mucho más fácil de entender el sistema.

Por ejemplo, tenemos bien individualizado el título que va sobre la imagen Hero, el color, por tanto si queremos cambiar ese color, que en principio estaba usando el mismo token `title__on-image`, podemos cambiarlo acá directamente y nos queda separado lo que es el token de `title__on-image` de lo que es el `hero__title`, es decir, podemos empezar a independizarlos. De modo tal que por ejemplo este color pueda ser distinto de este color de acá, de este color de acá, de este de acá.

Bueno, y lo mismo para los demás. Para el banner, por ejemplo, que no era necesario, para las cards de atracciones, que tampoco eran necesarios.

Dejé marcado acá todo lo que en realidad en principio no sería necesario.

Y acá lo que solamente en estas dos filas... las que se repite exactamente el mismo valor. Y ahora después vamos a ver por qué nos puede interesar esto.

Bueno, ahora nos queda por ver cómo llevar todo esto a GeneXus, pero como este video ya me quedó larguísimo voy a dejar por acá y seguimos en el siguiente.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com