

# Color system of Design System



Cecilia Fernández

## Color system

The screenshot displays a design tool interface with a code editor on the left and a visual token palette on the right. The code editor shows the following JSON structure for the color system:

```
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 visual token palette on the right is titled 'COLORS' and is divided into two sections: 'APPLICATION' and 'NEUTRAL'. Each section contains a table of color tokens with their names and values.

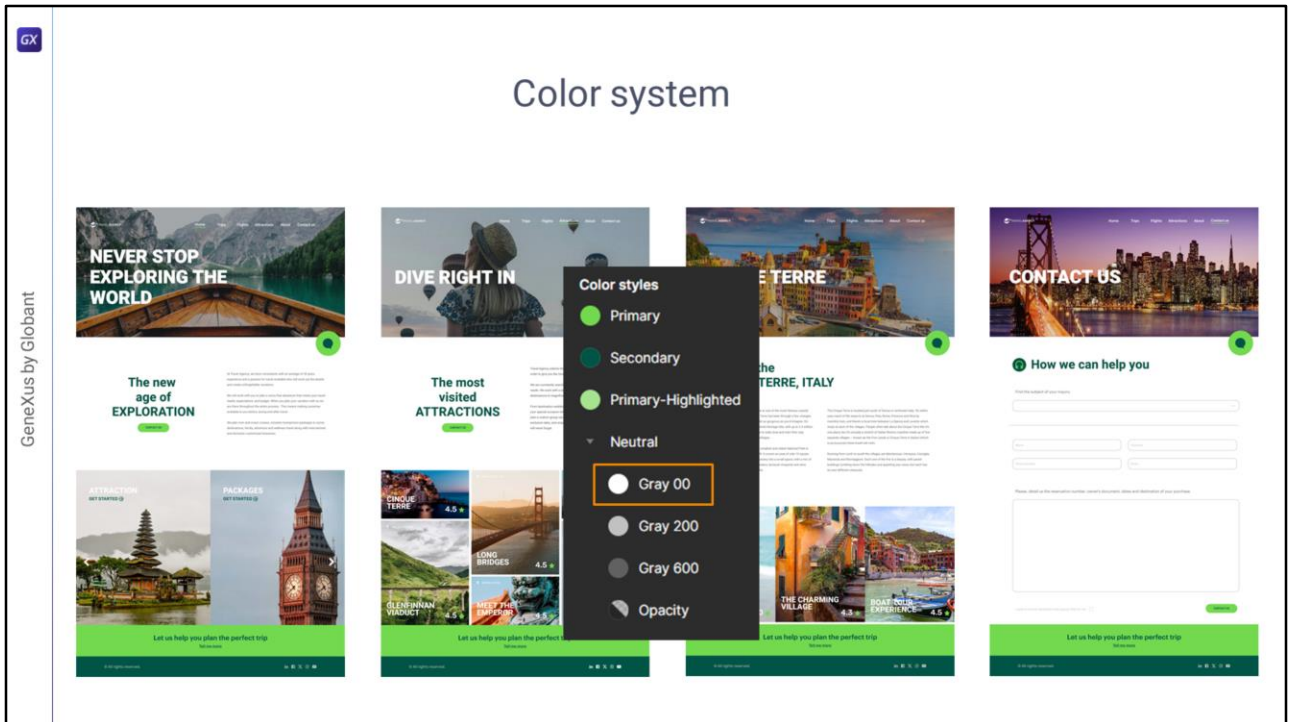
NAME	COLOR VALUE
primary	#73D94F
secondary	#015547
primary--highlighted	#A7E491
+ Add new token	

NAME	COLOR VALUE
gray00	#FFFFFF
gray200	#C1C1C1
gray600	#616161
opacity	#19181933
+ Add new token	

Below the 'COLORS' section, there is a 'FONTS' section which is currently empty.

No vídeo anterior, completamos os tokens de cor de nosso DSO a partir do que coletamos em Figma, se lembram?



No entanto, o sistema de cores expresso dessa maneira é muito básico: a nível semântico, expressa apenas a cor **primária** da aplicação e a **secundária**, mas não expressa coisas tais como que cor será aplicada para o fundo, que cor será aplicada para os textos que são títulos sobre os fundos, para aqueles que são textos comuns sobre os fundos, para os textos sobre imagens, e outras.

O sistema de cores poderia representar de uma maneira muito mais semântica a função das cores na aplicação. Por exemplo, este branco, Gray00...

## Color system


 Gray 00


### The new age of EXPLORATION

Discover the world's most beautiful destinations and experiences. From the majestic mountains of the Alps to the pristine beaches of the Maldives, we have it all. Let us help you plan the perfect trip.



Let us help you plan the perfect trip



### DIVE RIGHT IN

### The most visited ATTRACTIONS

Discover the world's most beautiful destinations and experiences. From the majestic mountains of the Alps to the pristine beaches of the Maldives, we have it all. Let us help you plan the perfect trip.



Let us help you plan the perfect trip



### CINQUE TERRE

### Visiting the CINQUE TERRE, ITALY

Discover the world's most beautiful destinations and experiences. From the majestic mountains of the Alps to the pristine beaches of the Maldives, we have it all. Let us help you plan the perfect trip.



Let us help you plan the perfect trip



### CONTACT US

#### How we can help you

First Name of your name

Last Name

Email

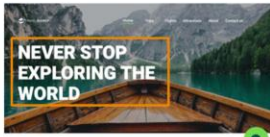
Phone

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

Let us help you plan the perfect trip

...poderia ser utilizado tanto como cor de fundo, quanto...

## Color system


 Gray 00


The new  
age of  
EXPLORATION



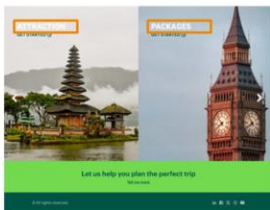
The most  
visited  
ATTRACTIONS



Visiting the  
CINQUE TERRE, ITALY

surface

title\_\_on-image



Let us help you plan the perfect trip

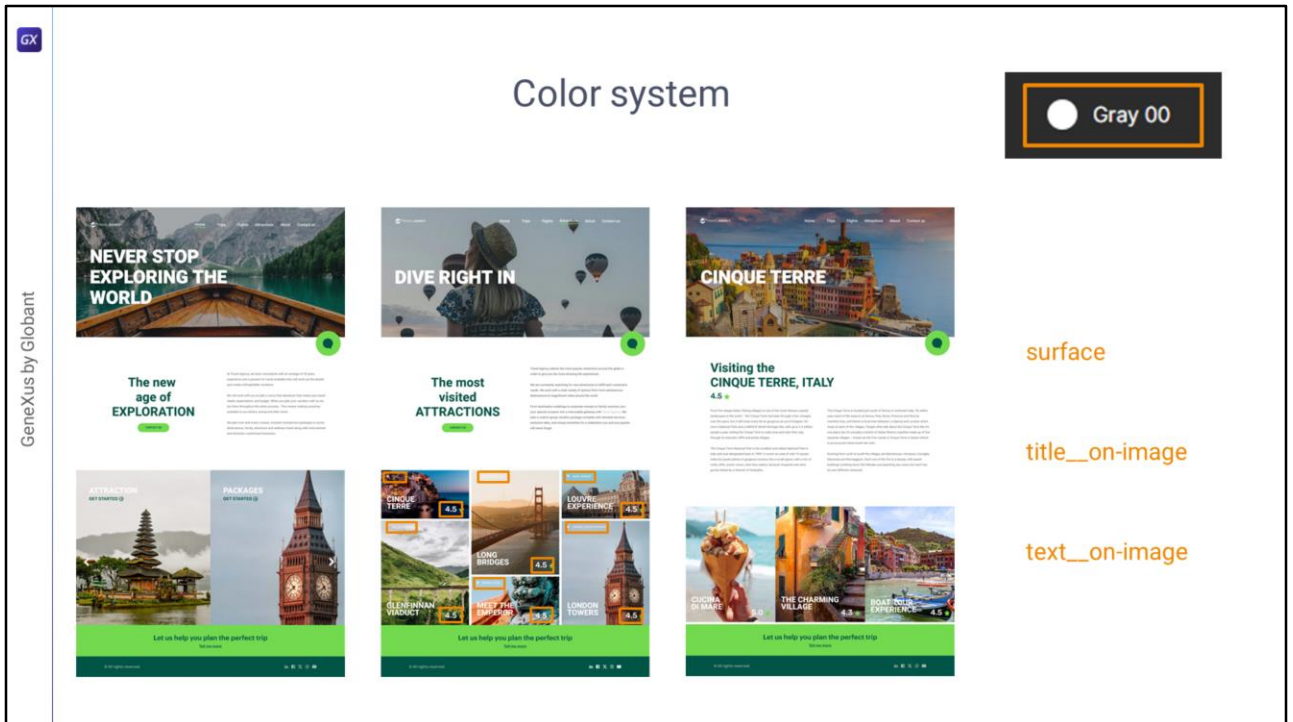


Let us help you plan the perfect trip



Let us help you plan the perfect trip

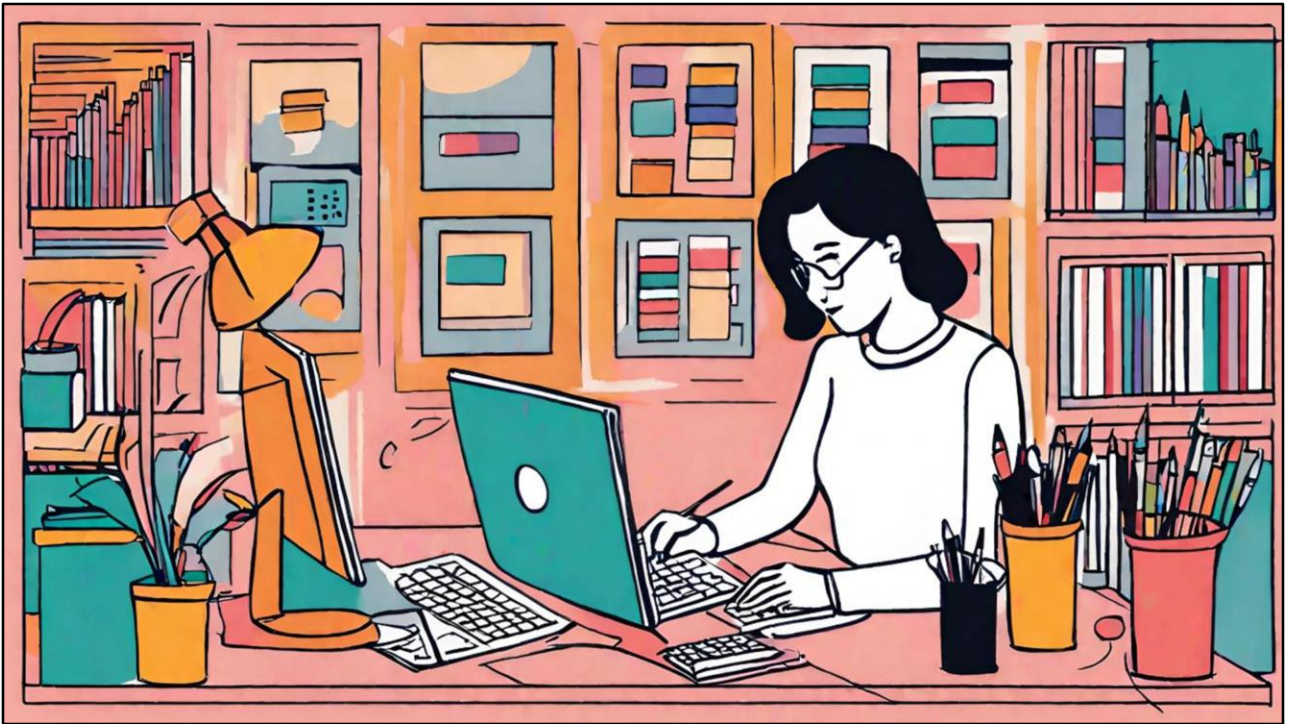
...como cor do **título sobre uma imagem**, tanto daquela de Hero quanto daquela dos Cards...



...assim como para outros textos sobre imagens, como os destes cards de Attractions. Ou seja, em pelo menos três funções diferentes.

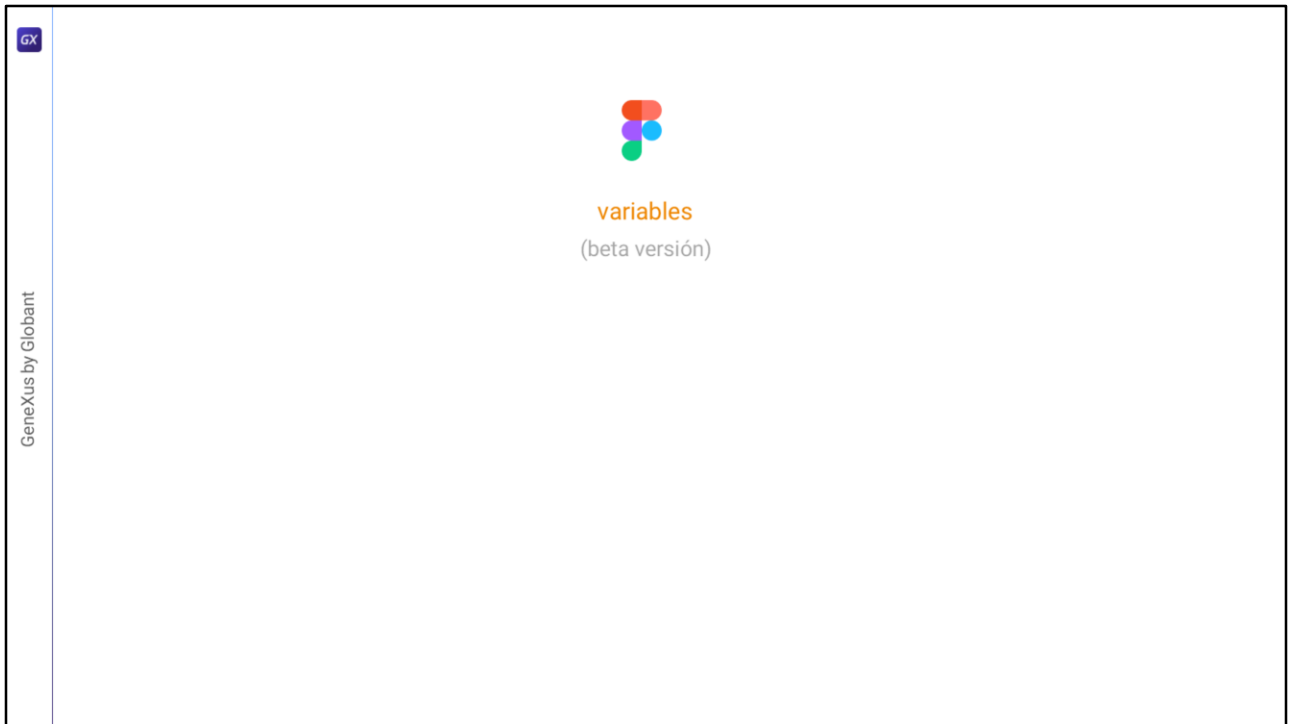
Se a estas funções: cor de **superfície**, cor de **títulos** sobre imagens, cor de **textos** sobre imagens, dermos um nome, estamos construindo um sistema de cores mais verdadeiramente sistêmico e, claro, mais semântico.

Dirá muito mais e melhor sobre o design da aplicação. Será então muito fácil mudar, por exemplo, a cor de fundo das telas. Ou a cor dos títulos sobre imagens, ou dos textos sobre imagens.



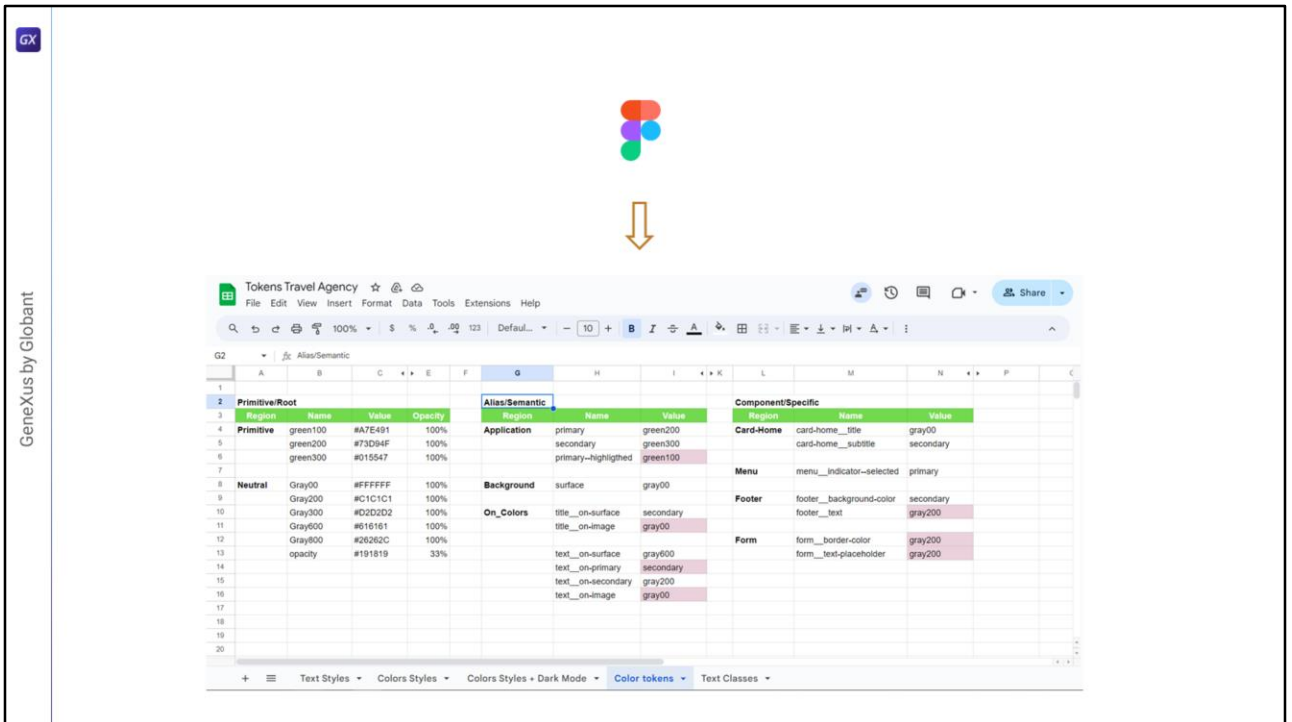
A tarefa de construir um bom **sistema** de cores, ou seja, de identificar e abstrair suas funções, não é tão fácil, mas é uma tarefa muito importante, que se adiada, degradará o sistema futuramente, por isso é conveniente desde o início dedicar toda a energia possível para resolvê-la, o mais rápido possível, e então tudo correrá bem. No máximo, será necessário fazer pequenos ajustes.

Em geral, essa tarefa fica nas mãos do designer, que é quem já trabalha mais ou menos conscientemente com essas abstrações.



Se a sua ferramenta de design não permite esse nível de expressividade (é o caso de Figma, que tem em beta ainda essa possibilidade)...





...poderá enviar ao frontend developer a modelagem do sistema em uma planilha, por exemplo.

Isso não impede que nós, frontenders, mesmo que não sejamos os que construímos o modelo, tenhamos que entendê-lo. Claro que se não temos um designer, ou o designer não é tão competente na sistematização, teremos que fazer esse trabalho nós mesmos.

Então vamos analisar este modelo que Chechu montou (e que eu fui completando) para o sistema de cores de nossa aplicação.

A primeira coisa que podemos ver é que está pensado em 3 níveis de abstração. O mais básico, este, corresponde à paleta de cores, onde o que fazemos é dar um nome, simplesmente, a cada cor da paleta e não muito mais.

O segundo nível é fundamental, é construído sobre esse primeiro, e é fundamental porque já corresponde à semântica global das cores na aplicação. Vai modelar, então, o sistema de cores da maneira mais geral possível.

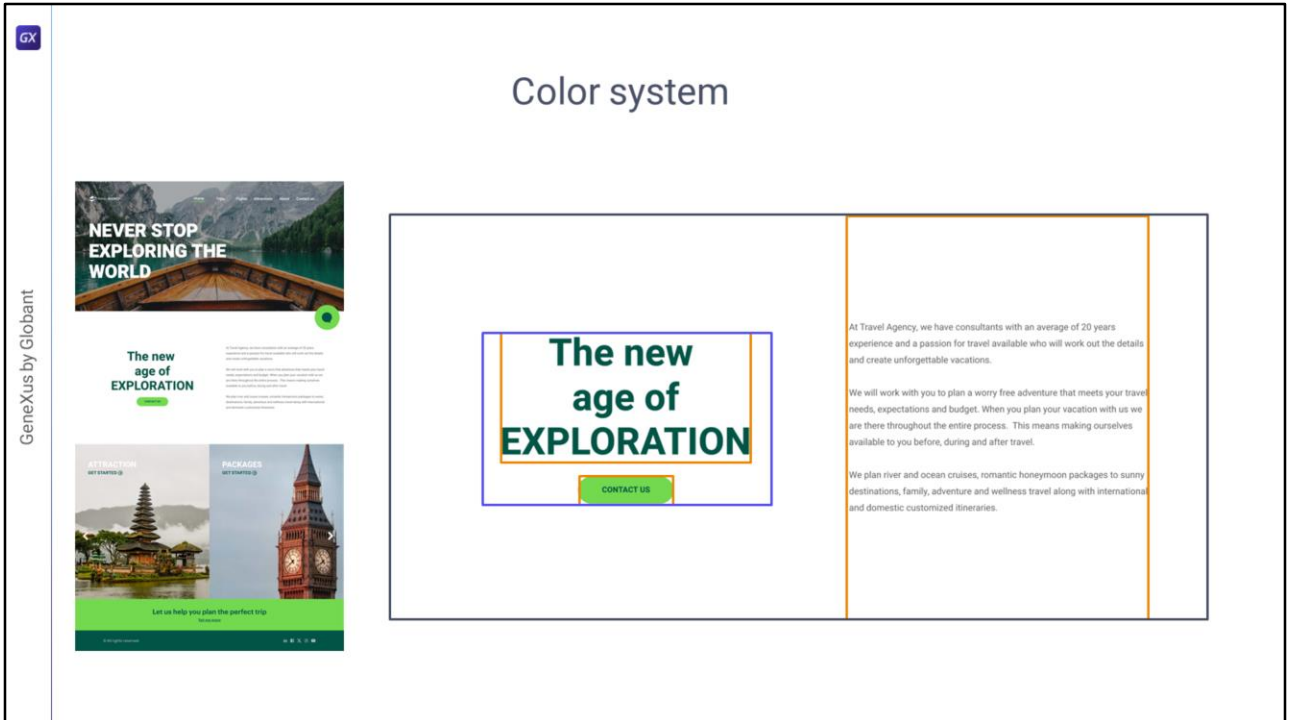
Com os estilos de cor que Chechu havia criado em Figma até agora, vocês se lembram, temos uma combinação muito básica desses dois níveis.

Depois pode haver ou não um terceiro nível, que já é para especificar os casos especiais que não podem ser modelados de acordo com a generalidade que oferece o segundo nível. Ou seja, o segundo nível é pensado como transversal a toda a aplicação, enquanto o terceiro já é

muito mais específico de um componente ou de uma parte do design, em particular.

Vamos analisar tudo isso um pouco para entender bem. Aqui vemos tokens para a cor primária, para a secundária, para a primária highlighteada (esses já tínhamos incorporado antes) e estamos adicionando um token para a cor de fundo das telas: este; e depois vemos que estão sendo adicionados tokens para representar as cores dos elementos que serão colocados **sobre** cores de fundo.

Para entender, vamos ver com exemplos.



É que todo elemento de um layout tem uma cor de fundo, a background-color, que eventualmente pode ser transparente.

Por exemplo, estes dois textos têm cor de fundo transparente, mas este botão não.

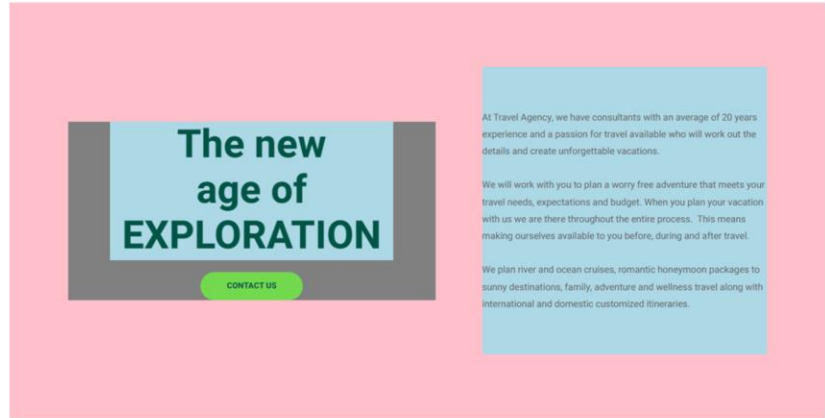
Mas estes dois controles estão por sua vez dentro de outro, tabela ou flex, não importa, que também pode ter cor de fundo. Ou, ser transparente.

E este e este, por sua vez, também estão dentro de uma tabela ou flex que pode ter ou não cor de fundo.

## Color system

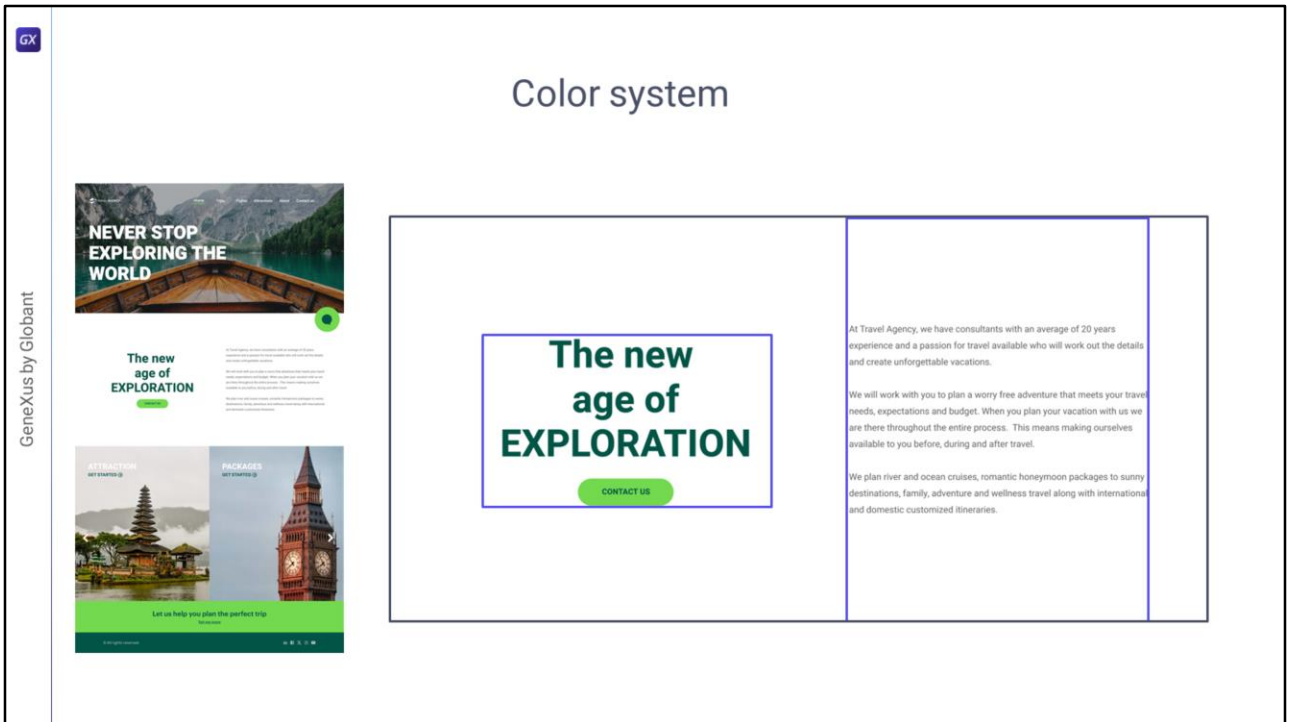


The new  
age of  
EXPLORATION



Por exemplo, vejam o que acontece se eu fornecer background-colors diferentes para vários desses elementos.

Em resumo, existe em todo layout uma hierarquia de controles, uns dentro de outros, cada um com cor de fundo, que pode (ou não) ser transparente.

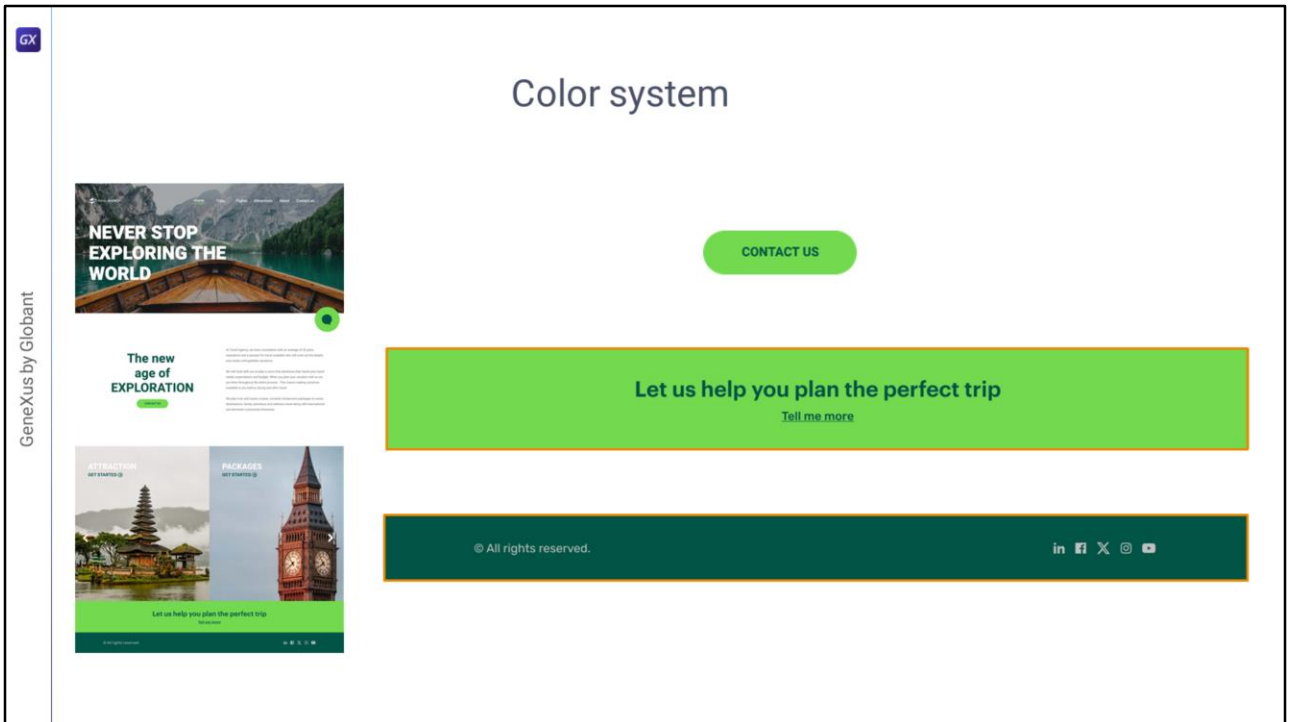


No exemplo do nosso design, a única background-color que não é transparente (na verdade é branca) é a do container. Bem, exceto pelo botão, que tem sua background-color verde claro, óbvio.

Bem, também não é necessário o que eu disse há um momento: também poderia ser transparente a background-color desta tabela, se aquela que por sua vez a contém tiver a background-color branca.

Na verdade, em nosso caso, lembrem-se de que tínhamos colocado a cor de fundo no nível da classe Application no DSO para que valesse universalmente para todos os layouts? Bem, se não se lembram, não se preocupem, porque esse agora não é o ponto.

No que quero que reparemos é no contraste necessário de fundo e frente.



Assim, por exemplo, esta pode ser uma tabela ou uma célula de uma tabela, e esta, outra. E essa tabela ou célula tem uma background-color. Em um caso, o verde claro, no outro, o verde mais escuro.

E os controles que estão dentro têm background-color transparente para que se localizem **sobre** essa superfície que é a do fundo.

Mas também têm uma cor própria, nesses casos de textos são a **cor do texto**, que é a que tem que contrastar sobre essa **superfície de fundo**, e isso é tudo o que interessa ao designer: conseguir um bom contraste, então, entre os elementos sobrepostos, para que possam ser bem visualizados e não fiquem confusos.

O botão também pode ser pensado assim, como tendo uma background-color e o texto do botão **contrastado sobre** ele.

## Color system



NEVER STOP  
EXPLORING THE  
WORLD

The new  
age of  
EXPLORATION

CONTACT US



EXPERIENCES  
LET STARTS

PACKAGES  
LET STARTS

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.

in f X @



CONTACT US

How we can help you

What is your name?

What is your email?

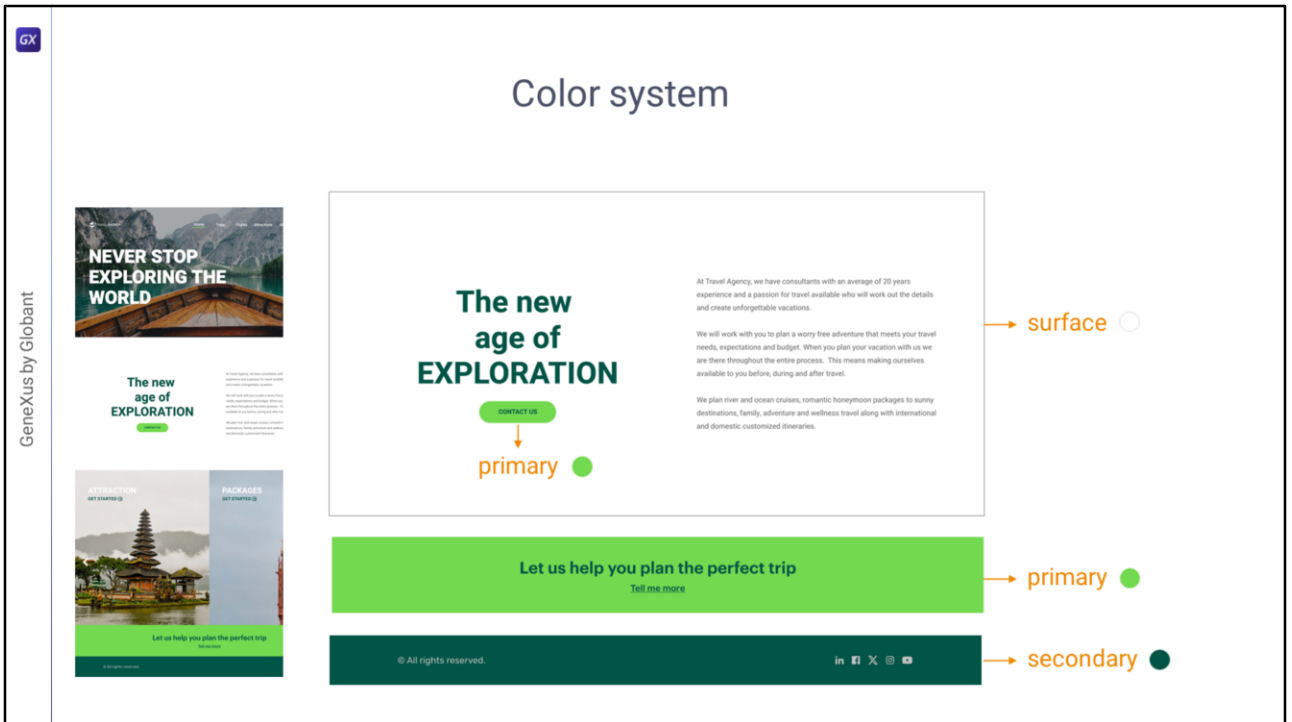
What is your phone number?

What is your message?

Let us help you plan the perfect trip

CONTACT US

Assim, se analisarmos o sistema com essa perspectiva, podemos ver em funcionamento 3 cores de fundo.



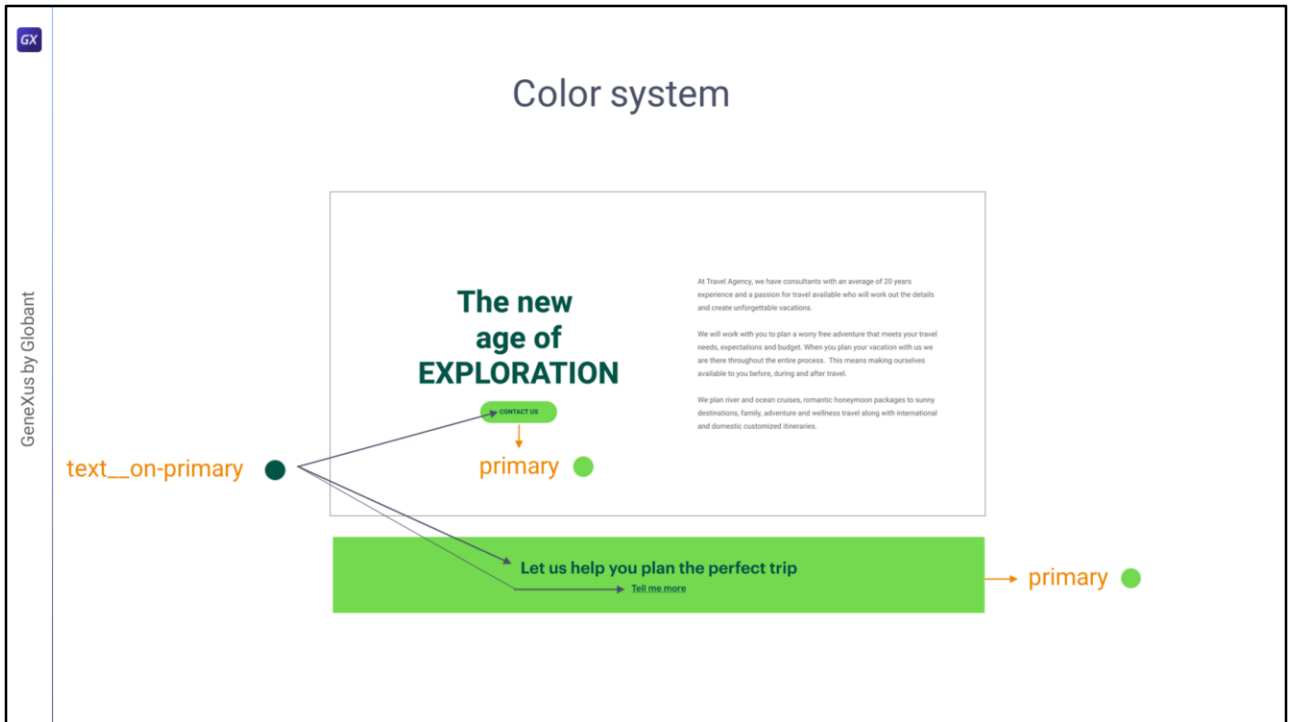
Esta, que chamaremos de **surface**, e corresponderá ao branco da paleta de cores.

Esta, à qual chamamos de **primary**, que é um verde da paleta de cores (o verde clarinho).

E esta outra, à qual chamamos de **secondary**, que corresponde ao outro verde.

Para o botão também temos a cor de fundo **primary**.



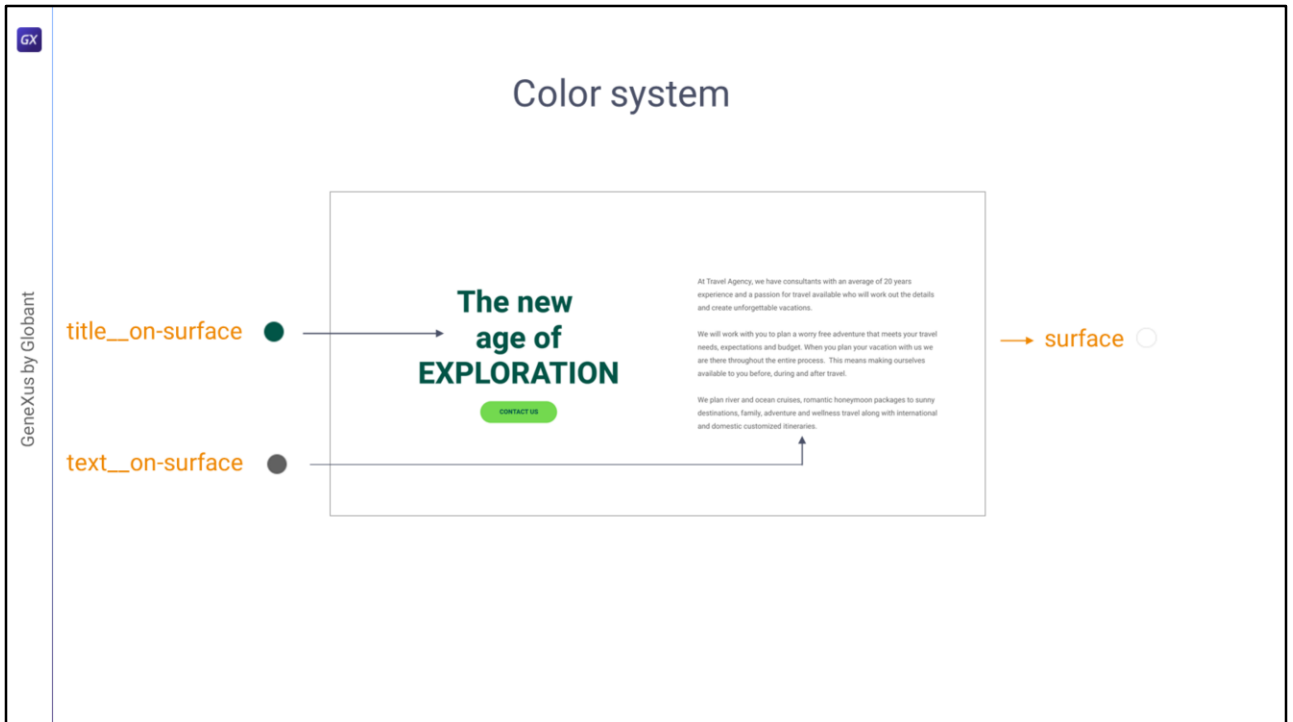


Bem, mas, como dizíamos, é fundamental para conseguir um bom design o bom contraste das cores que se localizam **sobre** essas **cores de fundo**.

Se observarmos os dois elementos que têm **fundo primary** (estes dois), vemos que os textos que se localizam **sobre** essa cor **primary** correspondem a uma mesma cor.

Por isso, podemos pensar em um token que especifique qual é a cor dos textos que se localizam sobre um fundo **primary**. E esse token é que chamamos de **text\_on-primary**. (Notem que aproveitei e já estou utilizando a nomenclatura BEM para nomear esse token).

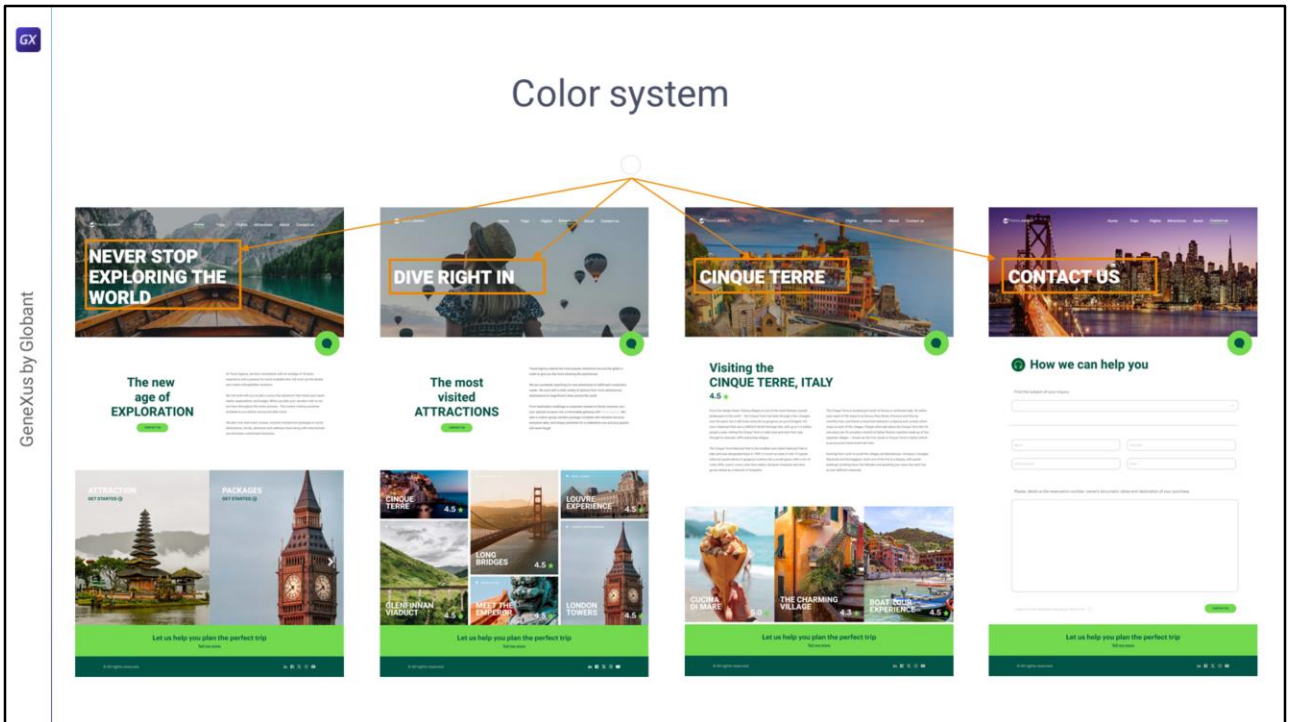
Neste caso, o valor que daremos ao token é o de outro token, o **secondary**, mas, além disso, não importa que valor lhe damos, o que importa é que acabamos de isolar, de identificar, um token que vai cumprir esta função: de ser texto que se localiza **sobre** uma superfície **primary**.



E já que estamos aqui, também podemos pensar em quais cores terão os textos que se localizam **sobre** a superfície **surface**. Vemos dois tipos de elementos diferenciados:

A **cor do título** e a **cor de um texto** comum. Aqui não sei se percebe bem que são duas cores diferentes. Mas mesmo que não fossem, é bom diferenciá-las porque suas funções são diferentes, embora para o design atual se concretizem, que não é o caso, porque são duas cores diferentes, mas mesmo que se concretizassem com o mesmo valor, ainda seria bom separá-lo em dois elementos. Porque poderiam mudar mais adiante e um adquirir uma cor e outro adquirir outra cor, porque correspondem a conceitos diferentes.

Então podemos dar-lhes nomes que signifiquem exatamente os dois conceitos diferentes: `title__on-surface`, um **título** sobre a superfície e um **texto** sobre a superfície.



Por outro lado, podemos pensar que as **imagens** também funcionam como **fundos**, quando são sobrepostos elementos em cima delas. Tanto as imagens Hero dos cabeçalhos das páginas, quanto as de todos esses carrosséis têm textos sobrepostos.

De que tipos? Temos dois casos: o dos **títulos** sobre as Heros...

## Color system



The new age of EXPLORATION



The most visited ATTRACTIONS



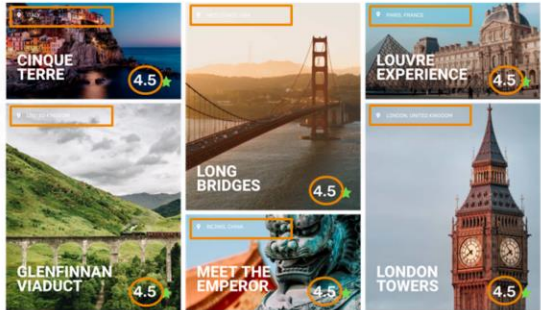
Visiting the CINQUE TERRE, ITALY



title\_\_on-image

...e sobre os cards, por um lado. Trata-se da mesma cor em ambos os casos, e seguramente a designer pensou nisso como conceito "**título sobre imagem**" (então podemos chamar esse conceito de cor **assim** e não separá-lo em dois, pelo menos inicialmente)

## Color system



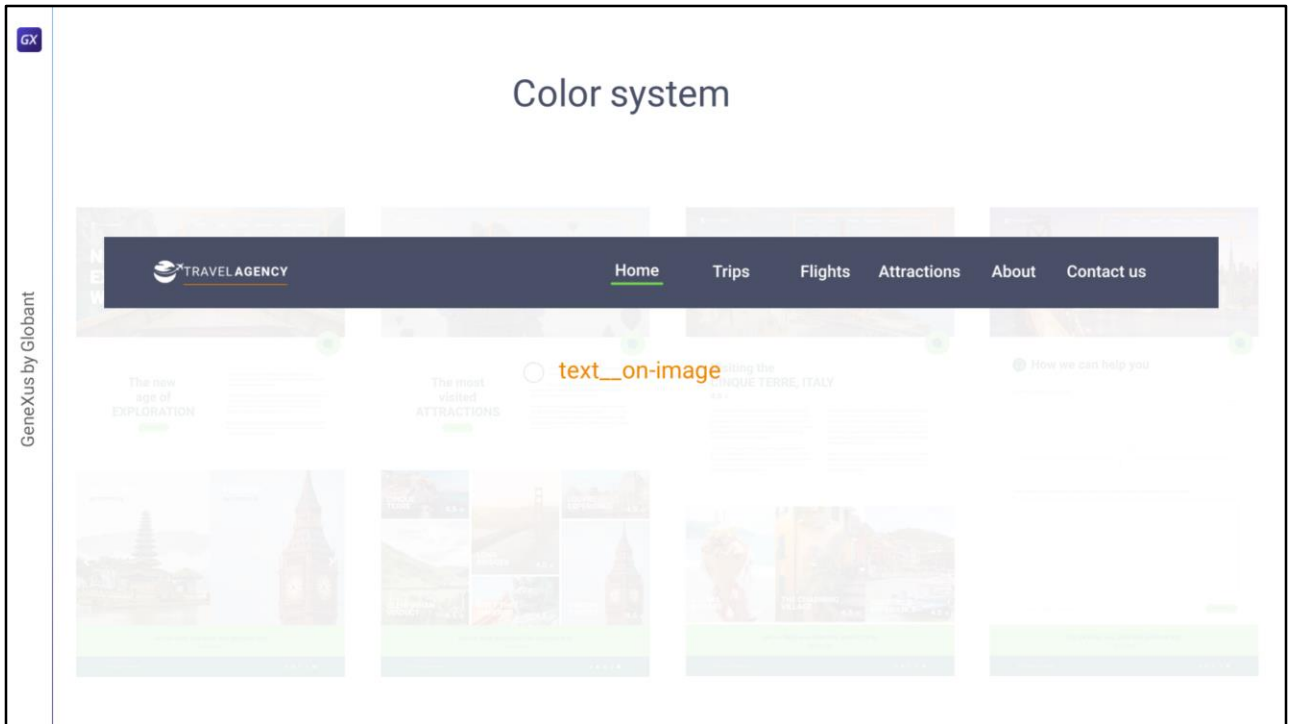
title\_on-image

Rating the  
Cinque Terre, Italy



text\_on-image

E, por outro lado, o segundo caso que podemos individualizar é o conceito “**texto sobre imagem**”, para todos esses outros casos: os do rating sobre esses cards e essas outras, e também para a localização da atração turística.



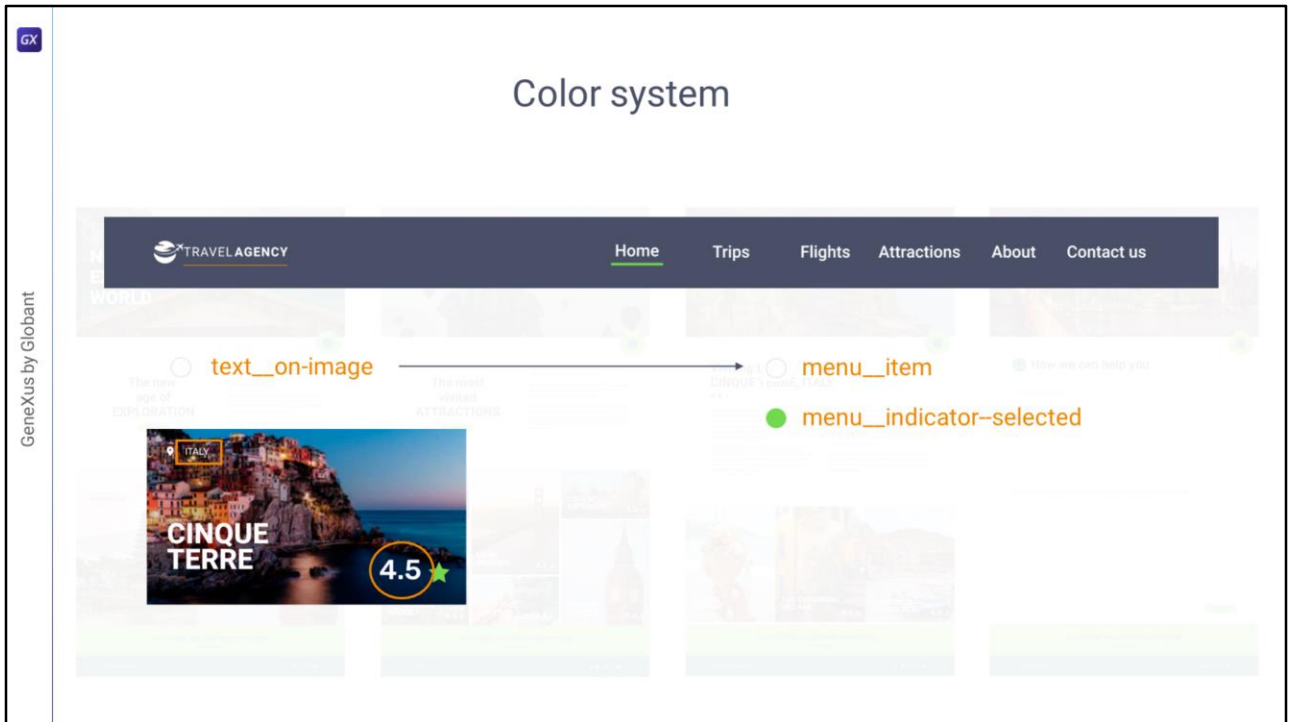
E para os itens do menu e estas duas palavras do logo? Poderíamos utilizar, em princípio, esse mesmo conceito.

## Color system

Primitive/Root				Alias/Semantic		
Region	Name	Value	Opacity	Region	Name	Value
<b>Primitive</b>	green100	#A7E491	100%	<b>Application</b>	primary	green200
	green200	#73D94F	100%		secondary	green300
	green300	#015547	100%		primary--highlighted	green100
<b>Neutral</b>	Gray00	#FFFFFF	100%	<b>Background</b>	surface	gray00
	Gray200	#C1C1C1	100%		<b>On_Colors</b>	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	

Com tudo isso que analisamos, já temos o nível semântico mínimo.

Precisaremos adicionar um terceiro nível, específico? E se sim, quão específico? ou será que com isso já modelamos todo o sistema?



Bem, por exemplo, como dissemos, inicialmente poderíamos utilizar para os itens do menu a mesma cor que para qualquer outro texto comum colocado sobre uma imagem, como estes outros. Ou seja, o `text_on-image`.

Mas também poderíamos pensar que a cor de um item do menu tem uma especificidade tal que pode justificar a atribuição de um conceito de cor próprio. Pensemos que neste design estamos implementando o menu sempre sobre a imagem Hero, mas pode ser que mudemos essa decisão mais adiante. Ou mesmo que se queira dar outra cor ao menu, por isso precisaremos tê-lo independente do resto dos usos de “textos sobre imagem”.

Este será um conceito específico do componente menu. Inclusive deveria adicionar a cor da barra que aparecerá indicando a opção selecionada, como vemos aqui, que vemos que assume esta cor verde, a primária.



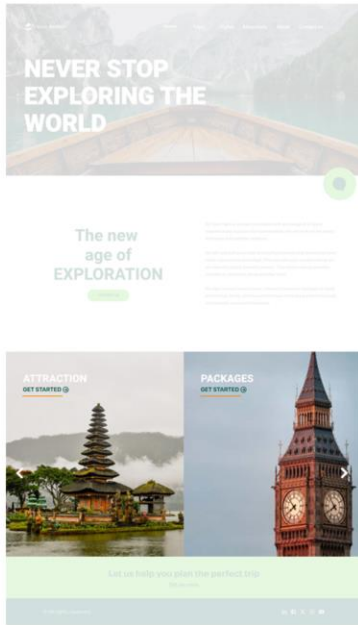
## Color system

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					

Então, adicionaríamos o terceiro nível, porque, embora essas cores só se aplicarão a um componente, por isso não seria tão indispensável isolá-las como tokens, fazê-lo melhora a manutenção do sistema e sua consistência.

Deixei o primeiro marcado porque, na verdade, observemos que atribuí a ele o mesmo valor que o do token `text_on-image`, o que está indicando claramente que, em princípio, não seria necessário tê-lo tornado independente.

São o tipo de decisões que devem ser tomadas, pensando em um equilíbrio. Para um sistema pequeno como o nosso, não se vê nenhuma desvantagem em princípio nesta especialização, mas à medida que o sistema cresce, a proliferação de casos particulares claramente complica e degrada. Então, sugestão, equilíbrio!



## 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			

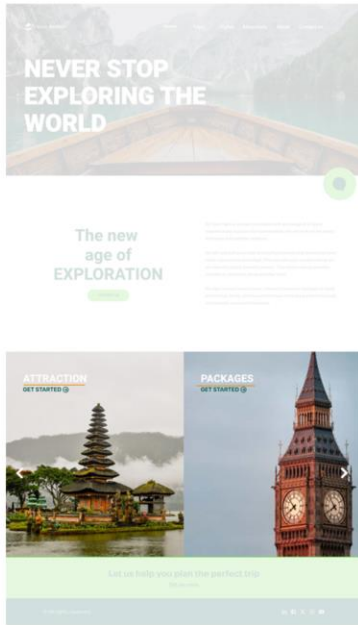
● text-alt\_\_on-image

● card-home\_\_subtitle

Por exemplo, não modelamos a cor deste texto. É um texto sobre uma imagem, mas não o branco que tínhamos para os outros casos de texto sobre imagem.

Portanto, poderíamos definir um segundo token, alternativo, para texto sobre imagem, no nível 2, semântico. Ou poderíamos defini-lo especificamente como no nível do componente onde o encontraremos, ao qual chamarei de card-home. Como nesse componente faz de subtítulo, o chamarei assim.

Tomei essa decisão porque o token geral não faz muito sentido, pois não parece ser geral de forma alguma, mas ser completamente específico para este caso.



## Color system

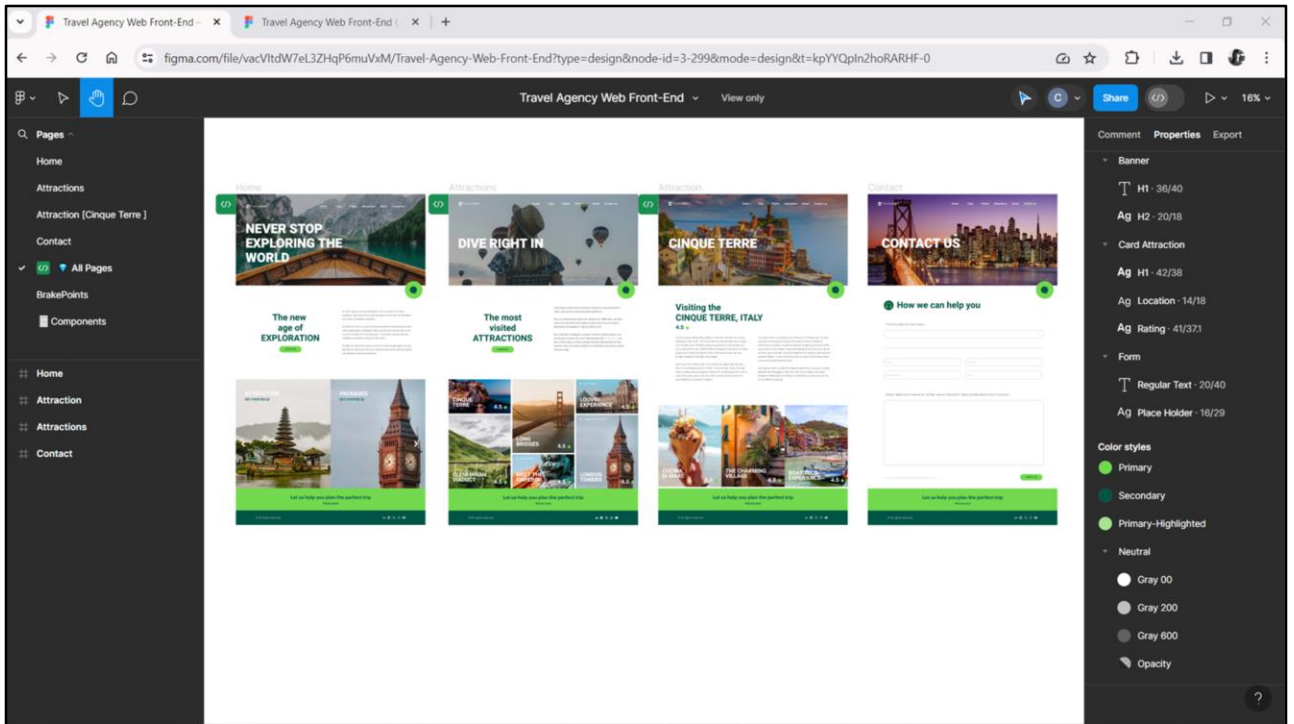
Alias/Semantic			Component/Specific		
Region	Name	Value	Region	Name	Value
Application	primary	green200	Menu	menu_item	text_on-image
	secondary	green300		menu_indicator--selected	primary
	primary--highlighed	green100	Card-Home	card-home_title	gray00
Background	surface	gray00		card-home_subtitle	secondary
	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

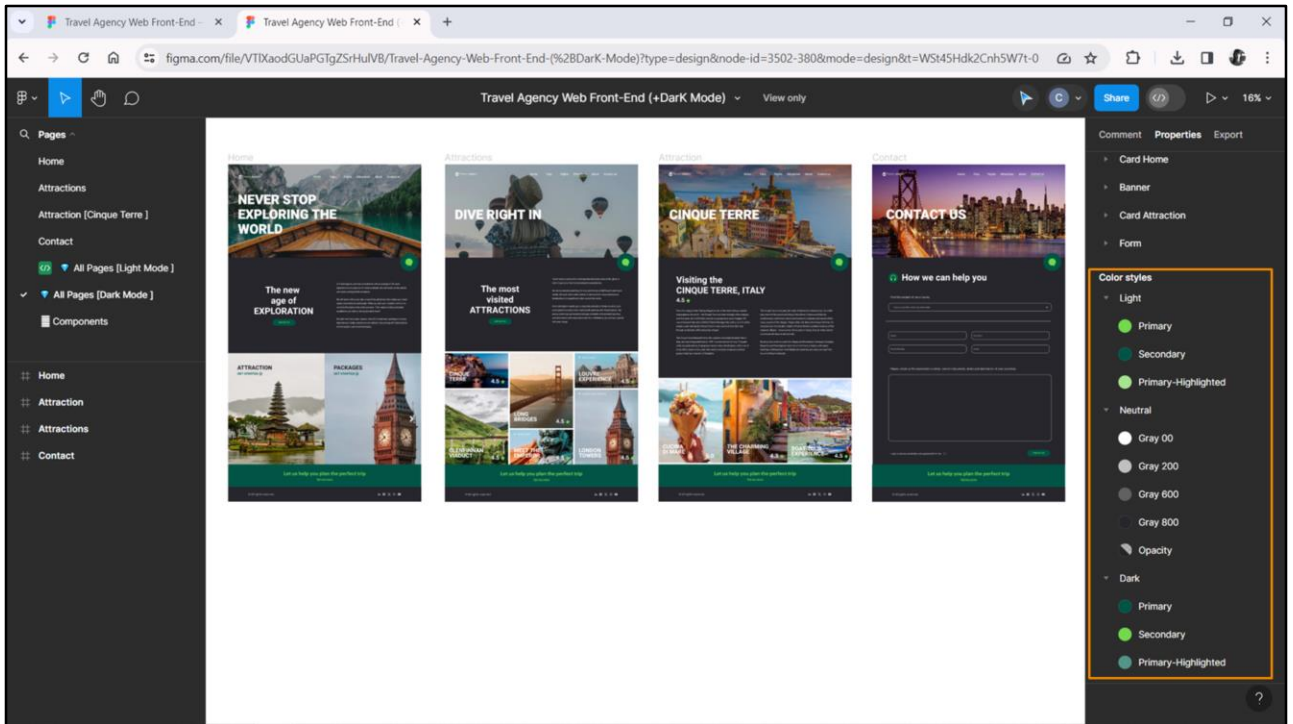
Então lá o adicionei, com o valor do token secondary.

Poderão se perguntar, com muito bom senso, por outro lado, por que para este elemento especifiquei um token, card-home\_\_title, se já tínhamos visto que me servia o geral, title\_\_on-image.

Poderia parecer que isso se deveu ao fato de que, como tive que definir um específico para o subtítulo, então por consistência seria boa ideia definir um para o título. No entanto... foi por algo muito mais forte do que isso... e que tem a ver com algo que até o momento não tínhamos considerado.



O que é isso... Este é o modo Light...



Este é o modo Dark.

Chechu montou outro arquivo para mim, diferente daquele com o qual estávamos trabalhando, onde adicionou este modo.

Se clicarmos aqui embaixo e nos posicionarmos sobre as propriedades, vemos sobre os estilos de cor que tínhamos que aparecem agora aqueles do modo Dark.

Vemos que, embora este seja o Primary Light, o Primary Dark é o que neste outro é o secundário. E para o que é em modo Dark secundário vai ser o que em modo Light é o primário.

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				

Então na planilha, para cada token vou ter que especificar outra coluna para o valor que assume o token para o modo Dark.

A que tínhamos, que dizia Value (agora mudei para Light) correspondia ao modo Light (que era no qual estávamos trabalhando), e temos que fazer essas associações, para cada token especificar qual vai ser o seu valor no modo Dark também.

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		

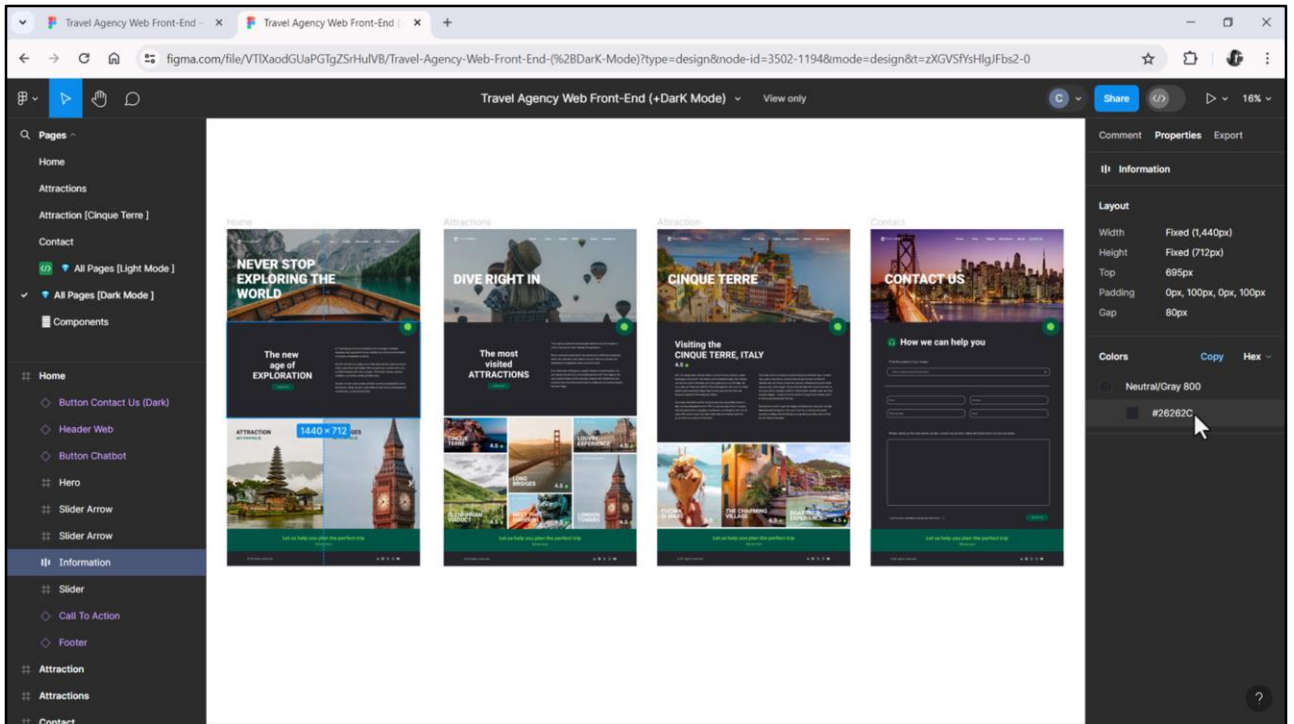
Então, por que eu estava dizendo isso? Porque se viéssemos aqui... teríamos que inspecionar tudo isso agora para ver como varia a cor de acordo com o token. Isso que deixei marcado é quando é o mesmo valor em ambos os modos.

Então, o que não está marcado é o que varia entre um modo e o outro.

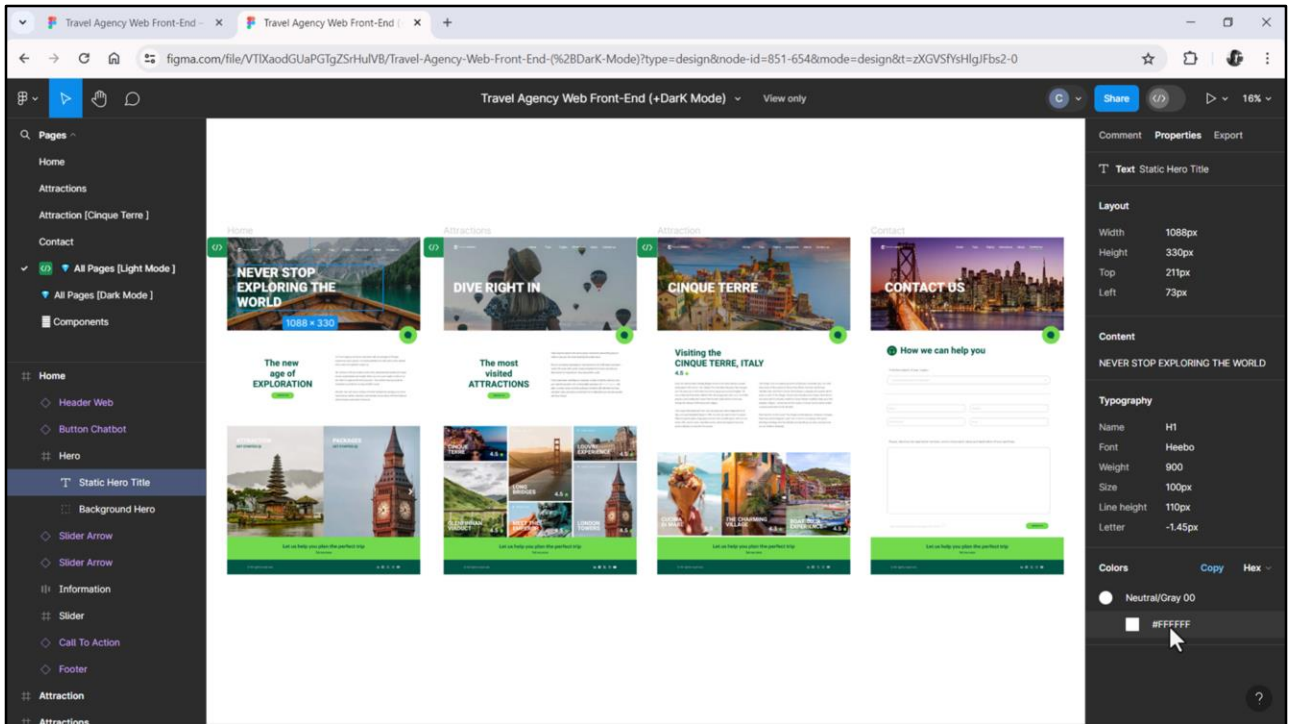
Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	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%						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				

Por exemplo, vamos começar pela **surface**, a superfície: enquanto no modo Light é este branco, o gray00, que temos aqui, para o Dark será este gray800.



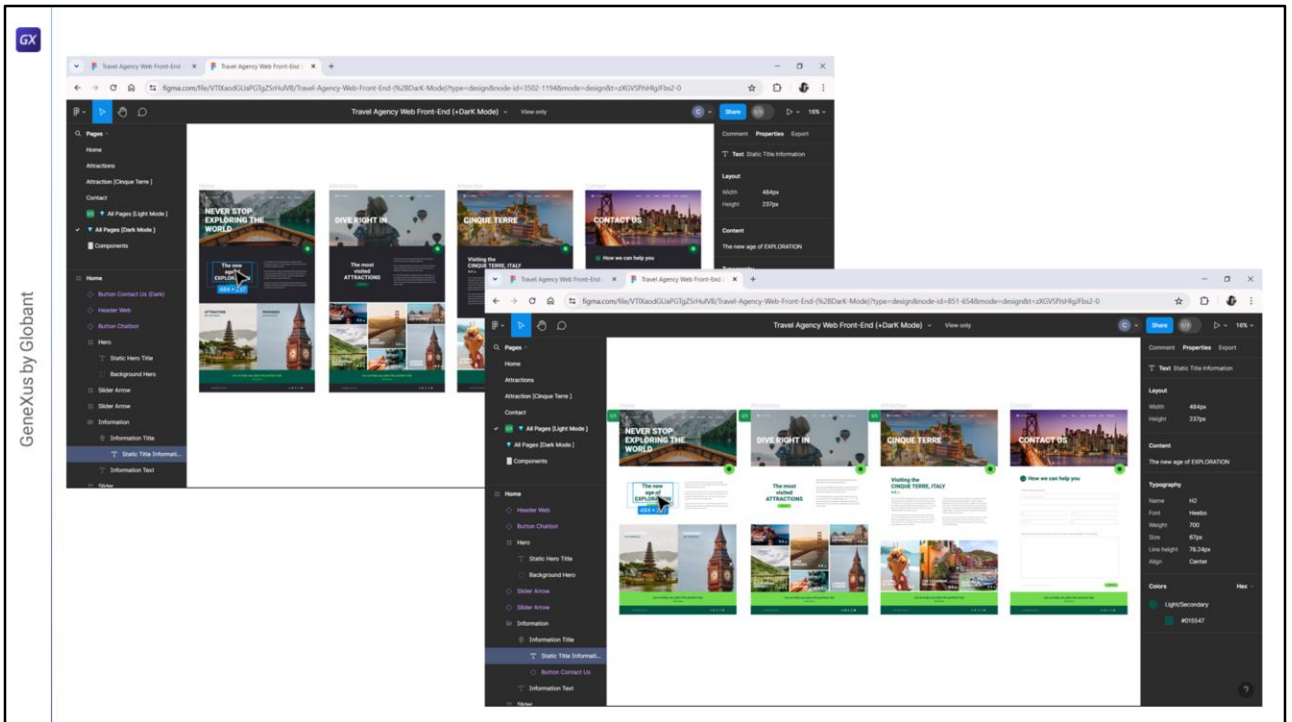


E podemos ver claramente posicionando-nos lá... e vemos: gray800... enquanto para o modo Light já sabíamos que era o gray00.



Como eu estava dizendo, nem tudo muda.

Por exemplo, a cor do título sobre imagem que correspondia ao nosso token `title__on-image` é um `gray00` para a versão Light ...e para a Dark também. Não se modifica.



Em vez disso, vejamos o que acontece com o título sobre surface, title\_\_on-surface. Aqui, no modo Dark, podemos ver que é branco, é o gray00, enquanto no modo Light era o secondary.

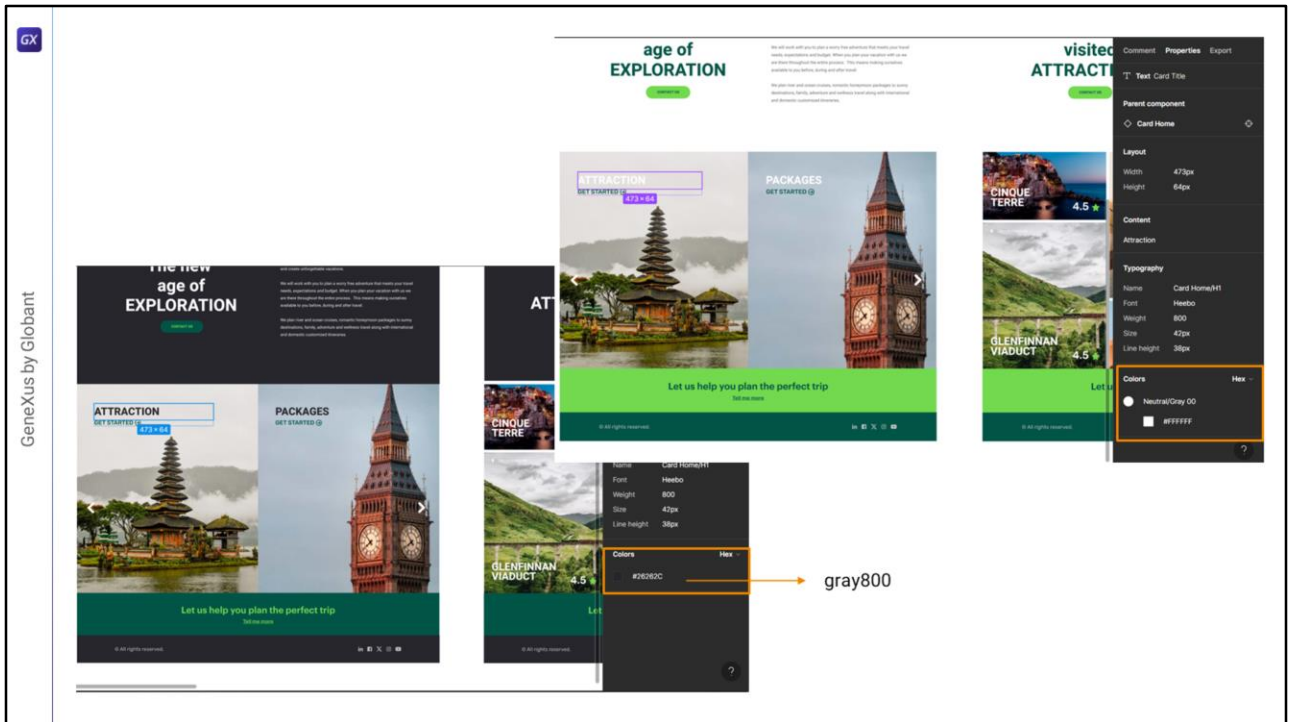
Primitive/Root					Alias/Semantic				Component/Specific		
Region	Name	Light Value	Value Dark	Opacity	Region	Name	Light	Dark	Region	Name	Light
Primitive	green100	#A7E491	#54958A	100%	Application	primary	green200	green300	Menu	menu_item	text_on
	green200	#73D94F		100%		secondary	green300	green200		menu_indicator--selected	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				

Então, vemos essas coisas aqui: surface, title\_\_on-surface, secondary, gray00.

Enquanto title\_\_on-image, o primeiro que vimos, mantém a mesma cor branca em ambos os casos.

Value Dark	Opacity	Alias/Semantic				Component/Specific			
		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--highlighed	green100	green100				
	100%	Background	surface	gray00	gray800	Card-Home	card-home__title	gray00	gray800
	100%						card-home__subtitle	secondary	primary
	100%	On_Colors	title__on-surface	secondary	gray00				
	100%		title__on-image	gray00	gray00				
	33%		text__on-surface	gray600	gray00				
			text__on-primary	secondary	secondary				
			text__on-secondary	gray200	primary?				
			text__on-image	gray00	gray00				

Agora, vamos ao que eu perguntava, que foi o que fez surgir essa introdução do modo Dark: por que isolou também um token para o card-home\_\_title? Em vez de utilizar, por exemplo, o title\_\_on-image.



Porque enquanto este title\_\_on-image, e até mesmo este title\_\_on-image não varia (vemos que continua sendo o mesmo branco) o que acontece com este?

Aqui vemos que é um tipo de cor cinza escuro... bem, observemos que aqui Chechu não está utilizando o estilo de cor. Estas são inconsistências que, claro, podem acontecer em qualquer projeto, certo? Mas claramente esta cor é a mesma que a da superfície... então é a gray800.

Bem, mas dizíamos, então, aqui é gray800, enquanto no modo Light é este branco.

Então vemos que esta cor, a cor deste título, tem um comportamento diferente da cor dos títulos sobre as imagens que são usadas para a generalidade da aplicação. Então é um caso particular. E é um caso particular que só se aplica aos cards do Home, não a outra coisa, e é por isso que tomo essa decisão...

Value		Dark	Opacity	Alias/Semantic		Region	Name	Light	Dark	Component/Specific			
34958A	100%			<b>Application</b>	primary			green200	green300	<b>Menu</b>	menu_item	text_on-image	text-on-image
	100%				secondary			green300	green200		menu_indicator--selected	primary	secondary
	100%				primary--highlighted			green100	green100	<b>Card-Home</b>	card-home_title	gray00	gray800
				<b>Background</b>	surface			gray00	gray800		card-home_subtl	secondary	primary
	100%			<b>On_Colors</b>	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				

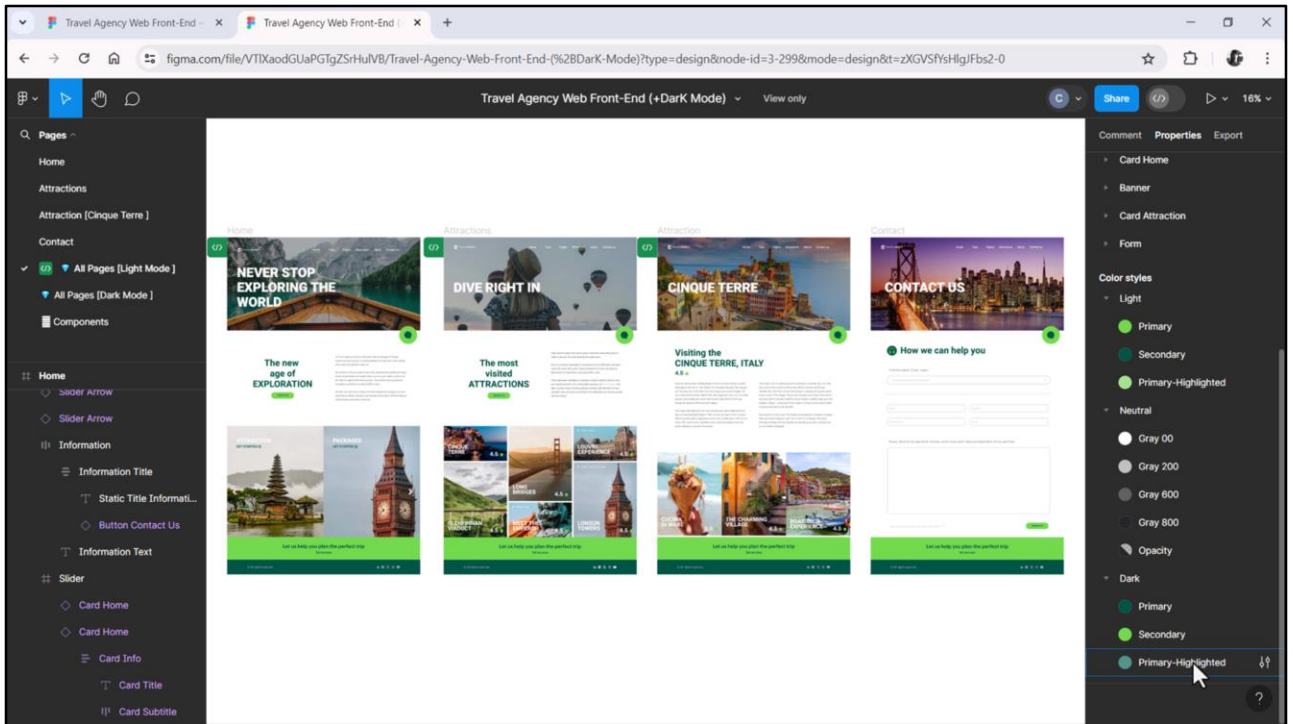
...de definir como um token específico do componente.

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%						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%							
opacity	#191819		33%		text_on-surface	gray600	gray00				
						text_on-primary	secondary	secondary			
						text_on-secondary	gray200	primary?			
						text_on-image	gray00	gray00			

Vamos ver também isso que eu estava dizendo antes, que o token primary vai ser um tipo de verde, o verde claro, o verde 200, que é o verde claro, para o modo Light; e o verde escuro para o Dark. E vice-versa, se invertem, para o token secondary.

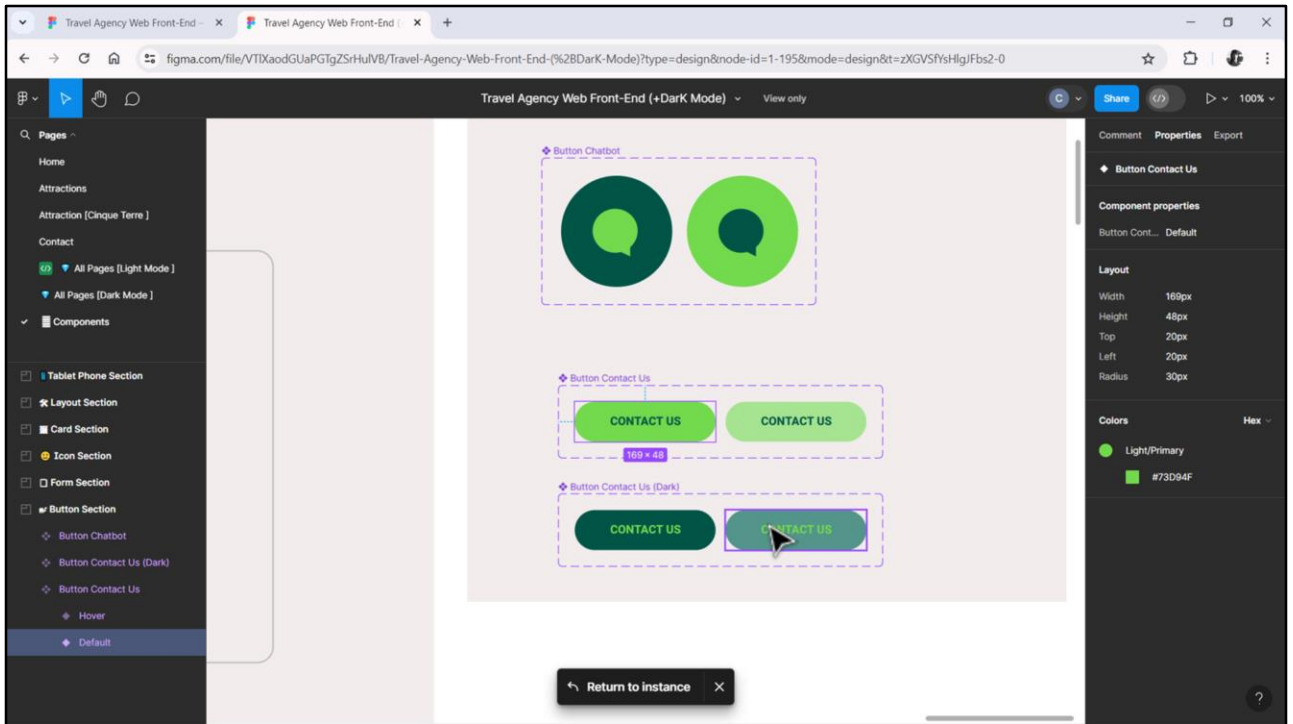
Aqui, observemos que o primary--highlighted que está utilizando o mesmo token primitivo green100, eu chamei de green100... estes não variam por modo Light e Dark, todos terão o mesmo valor para os dois modos, exceto este primeiro que eu faço variar por modo Light e Dark. E este, a que cor corresponde, que cor é esta?





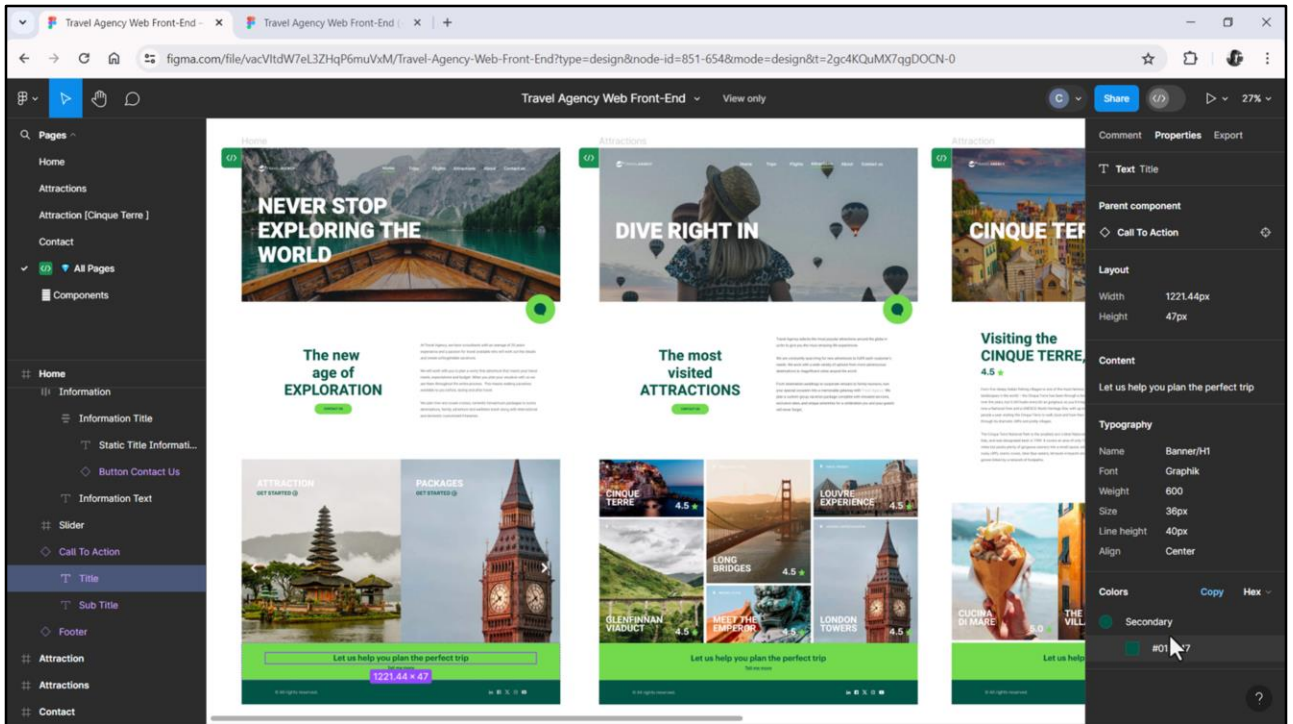
O primary-highlighted. Vejam que tenho este primary-highlighted para o modo Light, e para o modo Dark é este outro, que é mais claro que o primary, em ambos os casos.

Que era o que usávamos, justamente, para que o botão...



...vamos ao botão para mostrar a vocês... para fazer as versões das variantes do botão quando é feito hover sobre ele. No outro arquivo, só tínhamos isso, e neste temos as duas versões: a do Light e a do Dark.

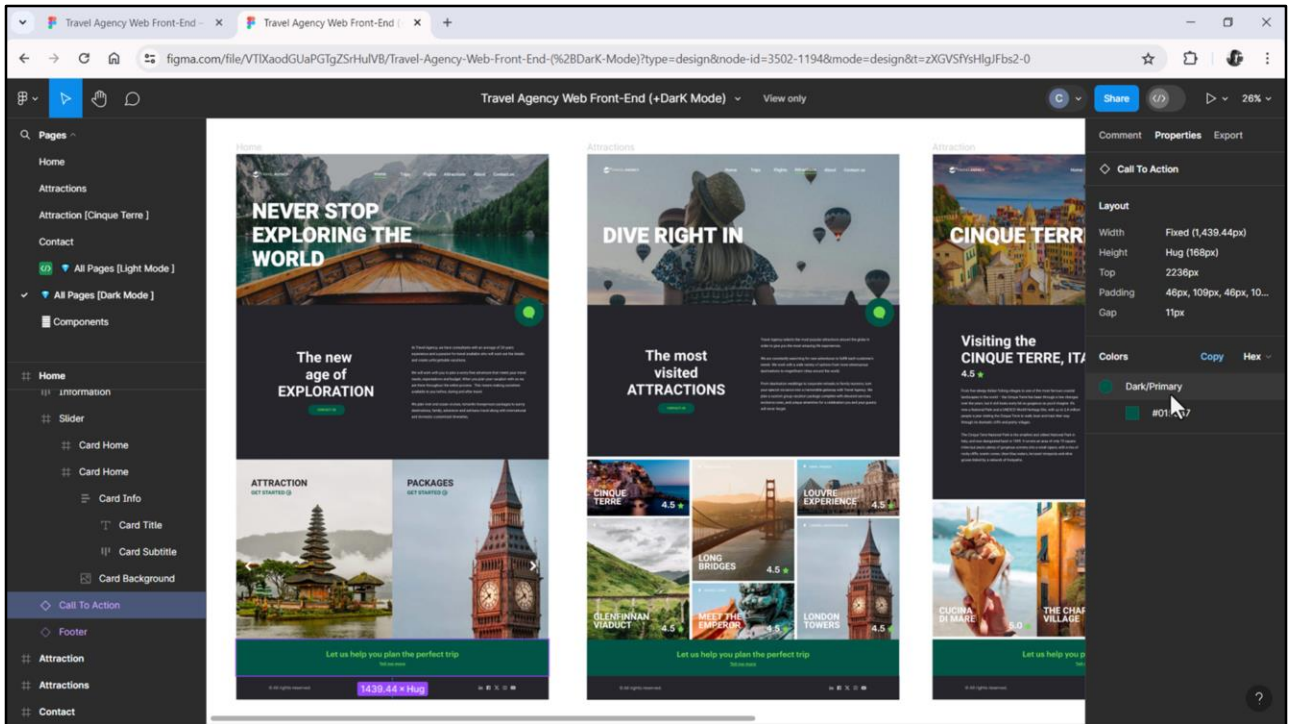
Assim como para o chatbot (que também tinha contado no vídeo em que falamos das imagens, mas ali ainda não tinha apresentado a vocês porque ainda não tinha este arquivo, não queria mostrar ainda este arquivo que tinha me passado a designer).



Para facilitar a comparação, vou ter os dois arquivos ao mesmo tempo. Este é o anterior, que tem o modo Light unicamente, e este é o que tem os dois modos, e vou ficar posicionado no modo Dark, para poder comparar facilmente.

Bem, o que deveríamos ver? Que com os tokens que temos, seja suficiente, para ver se não precisamos criar outros tokens, como aconteceu com o token que tínhamos associado ao card-home para o título.

Então, por exemplo, quero ver o que acontece com o primary como fundo. Tínhamos que para o modo Light definimos que com o primary, com este verde, depois definíamos um token `text_on-primary`, que servia tanto para o caso do botão quanto para o banner. Porque a cor do que se sobrepunha a essa cor primary, ou seja, `text_on-primary`, era sempre a mesma. Que neste caso, em modo Light, sabíamos que o primary era este verde e o on-primary correspondia ao secondary.

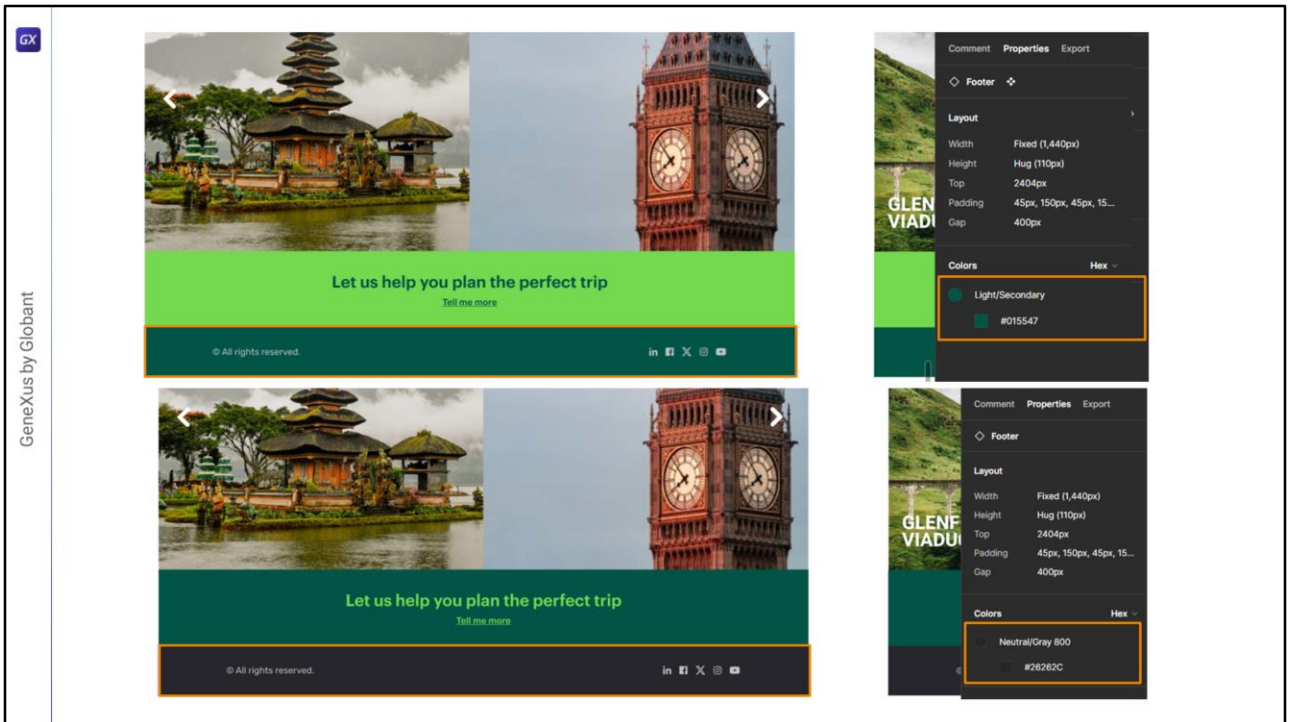


O que acontece com o modo Dark? Bem, vemos que se invertem: o que no outro caso era primary aqui é secondary e vice-versa. E isso funciona, veem? Porque então, o que temos que colocar aqui?

Sabemos que neste caso o primary é este, e o secondary é o que no outro caso era o 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	
Application	secondary	green300	green200	Menu	menu_indicator--selected	primary	secondary	
Application	primary--highlighted	green100	green100	Card-Home	card-home_title	gray00	gray800	
Background	surface	gray00	gray800	Card-Home	card-home_subtitle	secondary	primary	
On_Colors	title_on-surface	secondary	gray00					
On_Colors	title_on-image	gray00	gray00					
On_Colors	text_on-surface	gray600	gray00					
On_Colors	text_on-primary	secondary	secondary					
On_Colors	text_on-secondary	gray200	primary?					
On_Colors	text_on-image	gray00	gray00					

Parece um trava-línguas, mas aqui é onde se entende por que colocamos o mesmo valor, quando na realidade não é o mesmo. O que acontece é que este secondary está se referindo a este secondary que na verdade varia entre um modo e o outro.



Agora, vamos ver que há uma exceção a esta conversão de primary para secondary e de secondary para primary entre os modos Light e Dark.

Se olharmos aqui, vemos por exemplo no chatbot que o que é primário e secundário se converte na inversão. O mesmo para o botão, tínhamos visto, o mesmo para o banner.

Agora, o que acontece com o footer? Aqui não está sendo cumprida essa inversão. Porque a background-color do footer que é este verde do secondary não está se convertendo no verde primary do modo Light, ou seja, neste verde claro... se fosse seguida a mesma regra, teria que ser esta cor. Mas está sendo esta outra.

Então podemos ver claramente que se trata de uma exceção a esta regra (que se cumprida deveria indicar que o secondary daqui deveria ser o secondary daqui...) Que o secondary daqui, qual é? Este. E isso não está acontecendo.

Que cor está assumindo em vez disso? A 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

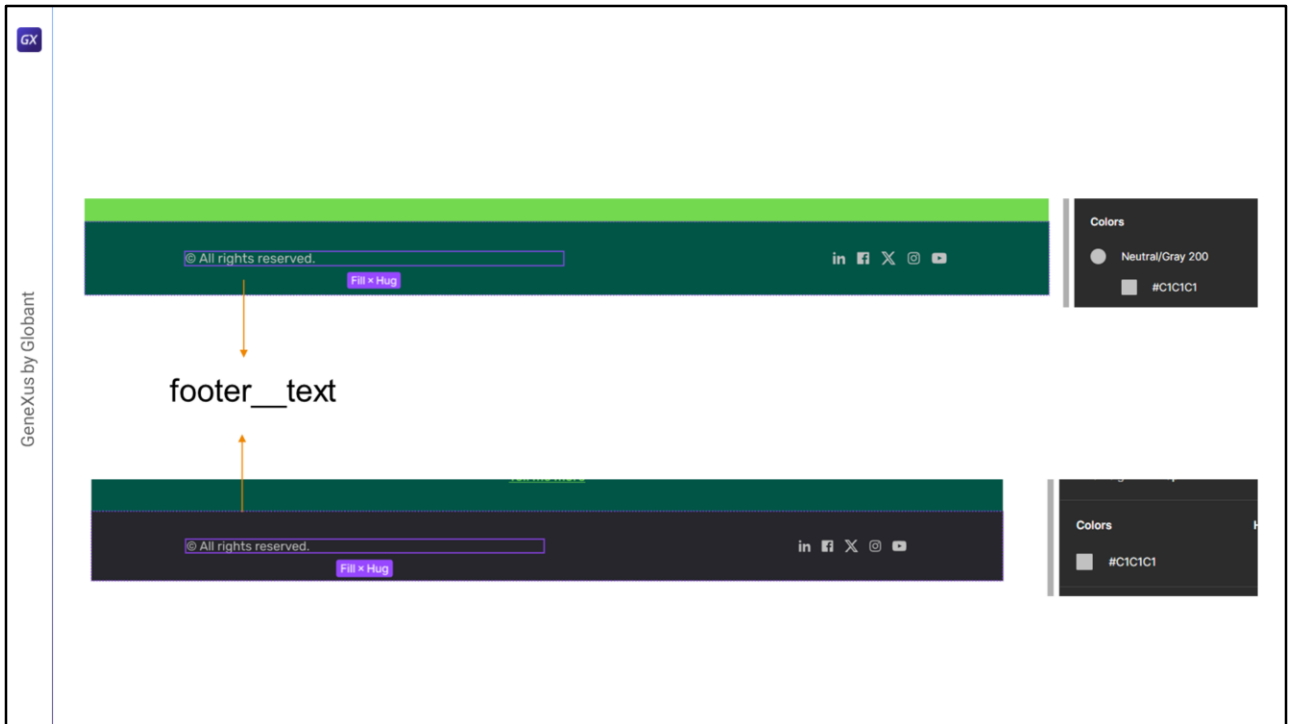
O10 gray800

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

maybe better would be: surface

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

Então crio um token especial, footer\_\_background-color, justamente para essa exceção. Vai assumir a cor secondary do modo Light. E para o modo Dark vai assumir este gray800.



E o que acontece com o texto sobre esse fundo?

Vemos que se trata dessa cor... se vamos ao modo Light é a mesma, é a gray200. E, portanto, adiciono-a como token a este, footer\_\_text.

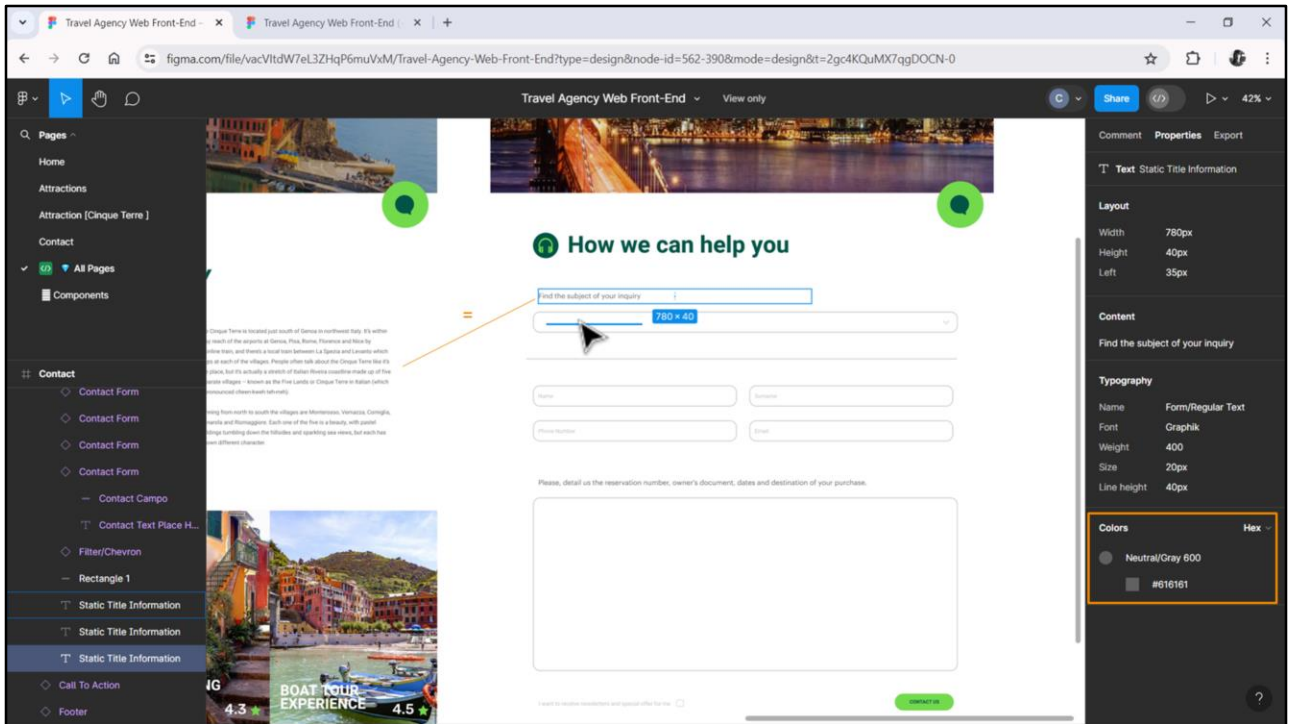


Alias/Semantic			Component/Specific				
Region	Name	Light	Dark	Region	Name	Light	Dark
Application	primary	green200	green300	Menu	menu_item	text_on-image	text-on-image
	secondary	green300	green200		menu_indicator--selected	primary	secondary
	primary--highlighted	green100	green100				
Background	surface	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 não vai corresponder a text\_\_on-secondary, porque não vai ser, por exemplo, para o modo Dark secondary o fundo, vai ser este outro, vai ser esta cor de fundo. Portanto, preciso adicionar um token especial.

De fato, o text\_\_on-secondary ficou assim... isto é para pensar porque não estamos usando na verdade. Um a priori, com esta inversão de cores que falávamos, que o primário de um se torna o secundário do outro modo e vice-versa, pensaria que na verdade um text\_\_on-secondary deveria ser primary em ambos os casos, se seguisse essa regra.

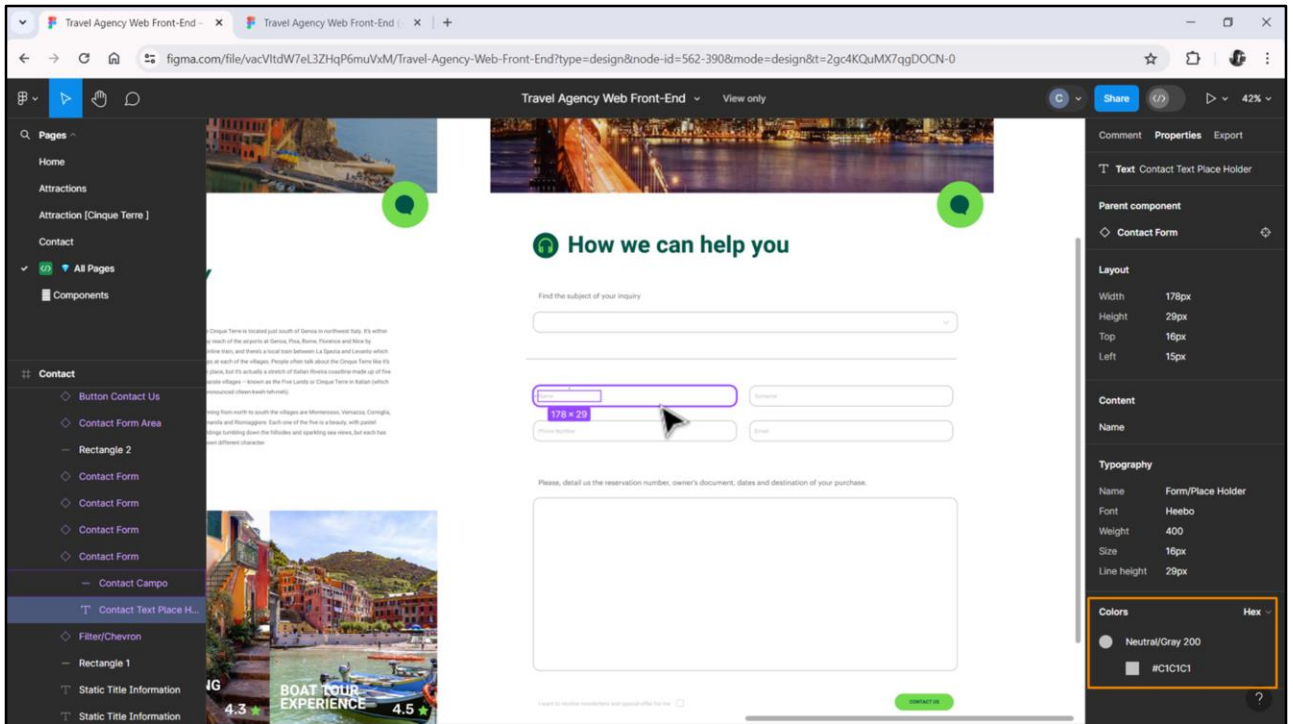
Vou deixar os dois casos com ponto de interrogação pelo que lhes disse... o único caso por enquanto de secondary como fundo é este, o do footer. Não temos outro caso.



Para terminar de expressar todo o nosso sistema de cores, nos falta modelar somente uma parte. Vamos observá-la aqui: tem a ver com o panel de Contato, que é o único panel que tem campos de entrada. Todos os outros panels têm apenas campos de saída.

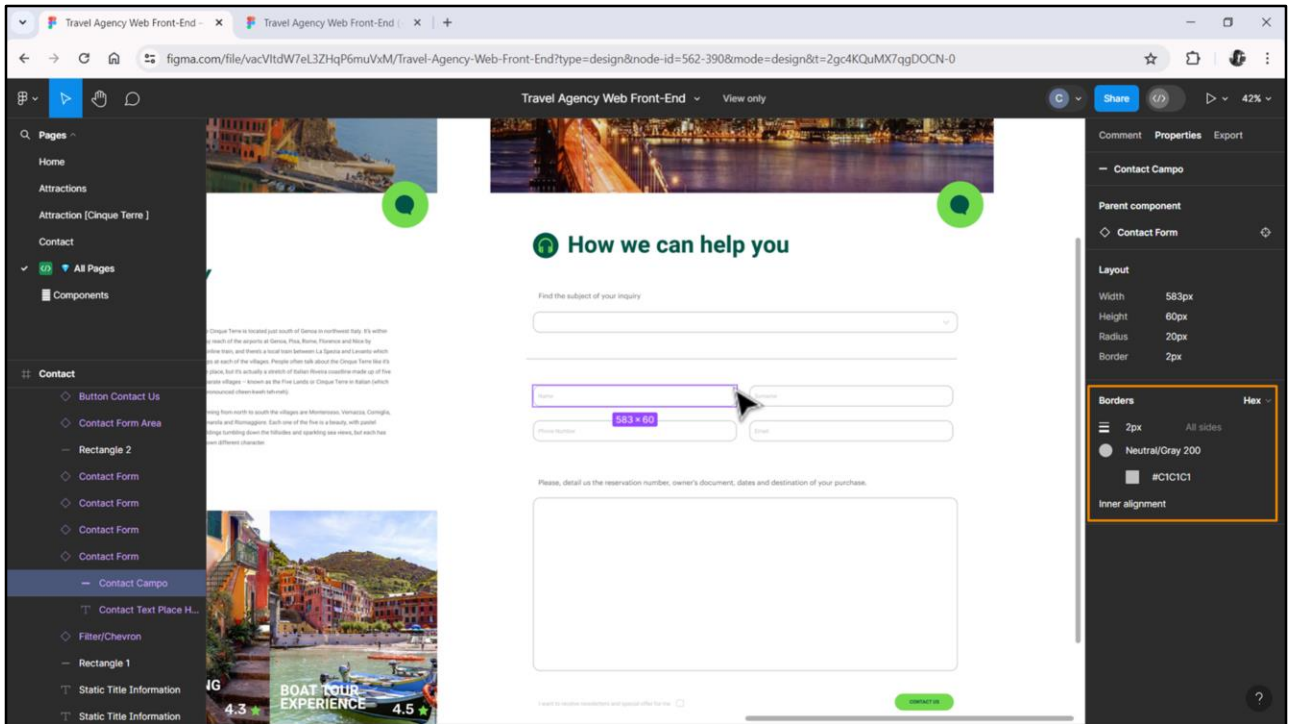
Aqui o usuário terá que inserir informação e pressionar o botão.

Se observarmos (vou ampliar um pouco mais), vemos que este texto que está descrevendo o que o usuário tem que inserir no campo seguinte, assume como cor, esta, gray600, que, neste caso, é o mesmo que assume qualquer texto sobre a superfície de fundo. Então, colocaríamos como cor a do mesmo token que para este. Aqui não há nada que especializar.



No entanto, observemos o que acontece com o texto que aparece aqui dentro, que é um texto de sugestão, ou seja, não é a descrição, mas corresponde a indicações para o usuário, que assim que começa a digitar desaparecem, ao contrário deste que é uma descrição que nunca desaparece.

Bem, se observarmos, esta é uma cor mais clara, não é a gray600, é esta gray200. Corresponde a um conceito diferente, por isso essa diferença de cor.

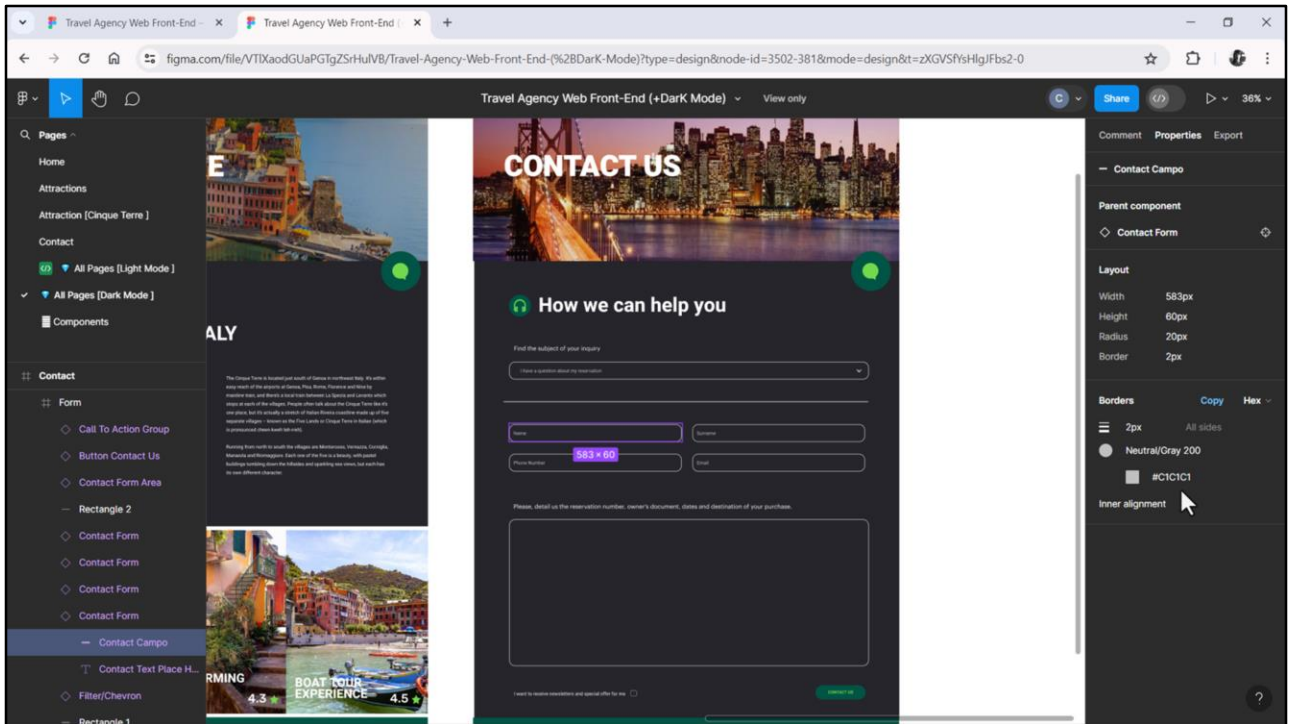


E além disso, aqui aparece outro tipo de cor, que é a das bordas desses campos. Os outros campos não tinham bordas. Este sim. E é uma gray200 também.

Então, está faltando modelar dois tokens: um para essas bordas e outro para o texto nestes 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					

É por isso que construo então dois tokens dentro de um componente ao qual vou chamar de form, porque é específico, é bem específico. A um chamei de border-color e ao outro text-placeholder, é o texto que vai dentro do placeholder.



Vejam que vai assumir para o modo Dark, não mostrei, o mesmo valor gray200. Esse é para a borda, e para o texto interno também.

	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					
7										Card-Home	card-home_title	gray00	gray800
8	FFFF		100%		Background	surface	gray00	gray800			card-home_subtitle	secondary	primary
9	C1C1		100%										
10	D2D2		100%		On_Colors	title_on-surface	secondary	gray00		Footer	footer_background-color	secondary	gray800
11	6161		100%			title_on-image	gray00	gray00			footer_text	gray200	gray200
12	262C		100%										
13	1819		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	secondary	secondary
24											banner_text	secondary	secondary
25													
26													

Com isso, teríamos a expressão mínima necessária do sistema de cores da aplicação.

Como disse antes, para um sistema tão pequeno quanto o nosso, poderíamos criar também tokens para esses outros componentes, o que torna muito mais fácil entender o sistema.

Por exemplo, temos bem individualizado o título que vai sobre a imagem Hero, a cor, portanto, se quisermos mudar essa cor, que inicialmente estava usando o mesmo token title\_\_on-image, podemos mudá-lo aqui diretamente e fica separado o que é o token de title\_\_on-image do que é o hero\_\_title, ou seja, podemos começar a torná-los independentes. De tal forma que, por exemplo, esta cor possa ser diferente desta cor daqui, desta daqui, desta daqui.

Bem, e o mesmo para os demais. Para o banner, por exemplo, que não era necessário, para os cards de atrações, que também não eram necessários.

Deixei marcado aqui tudo o que realmente em princípio não seria necessário.

E aqui o que apenas nestas duas linhas... as que repetem exatamente o mesmo valor. E depois vamos ver por que isso pode nos interessar.

Bem, agora temos que ver como levar tudo isso para GeneXus, mas como este vídeo já ficou muito longo, vou parar por aqui e continuamos no próximo.

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

[training.genexus.com](https://training.genexus.com)