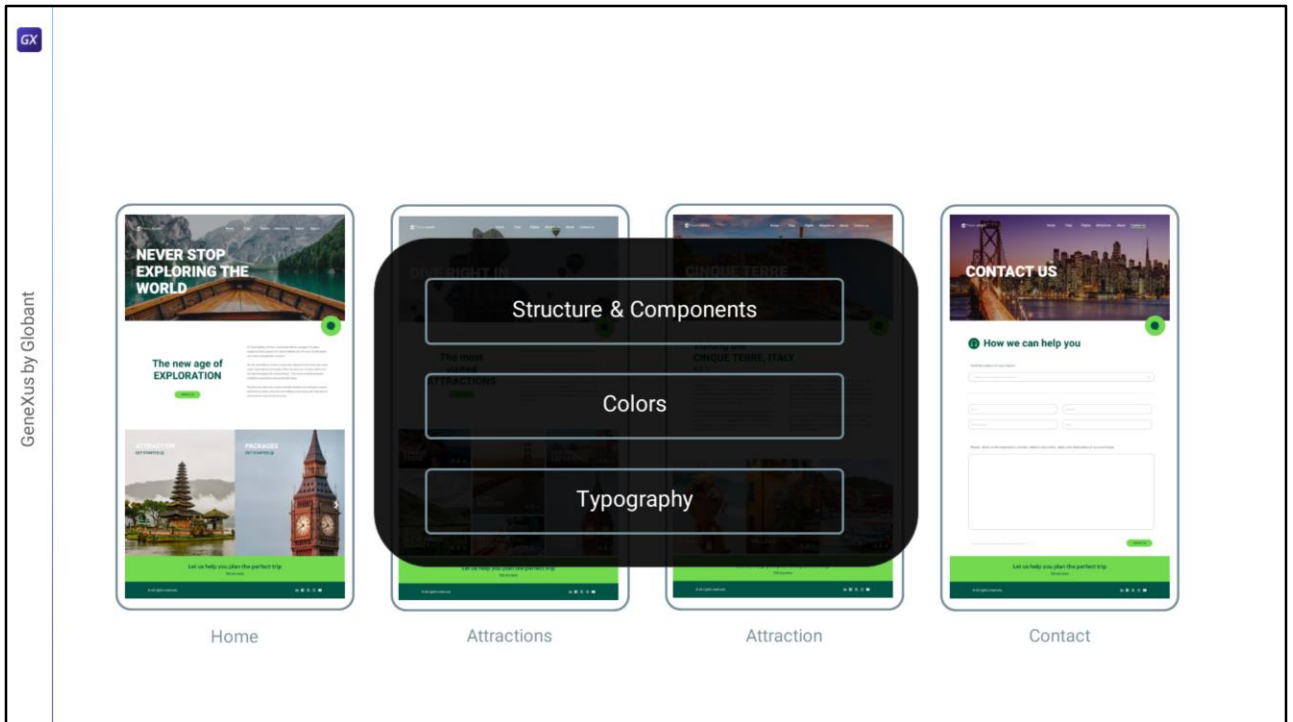


Design in GeneXus. Players

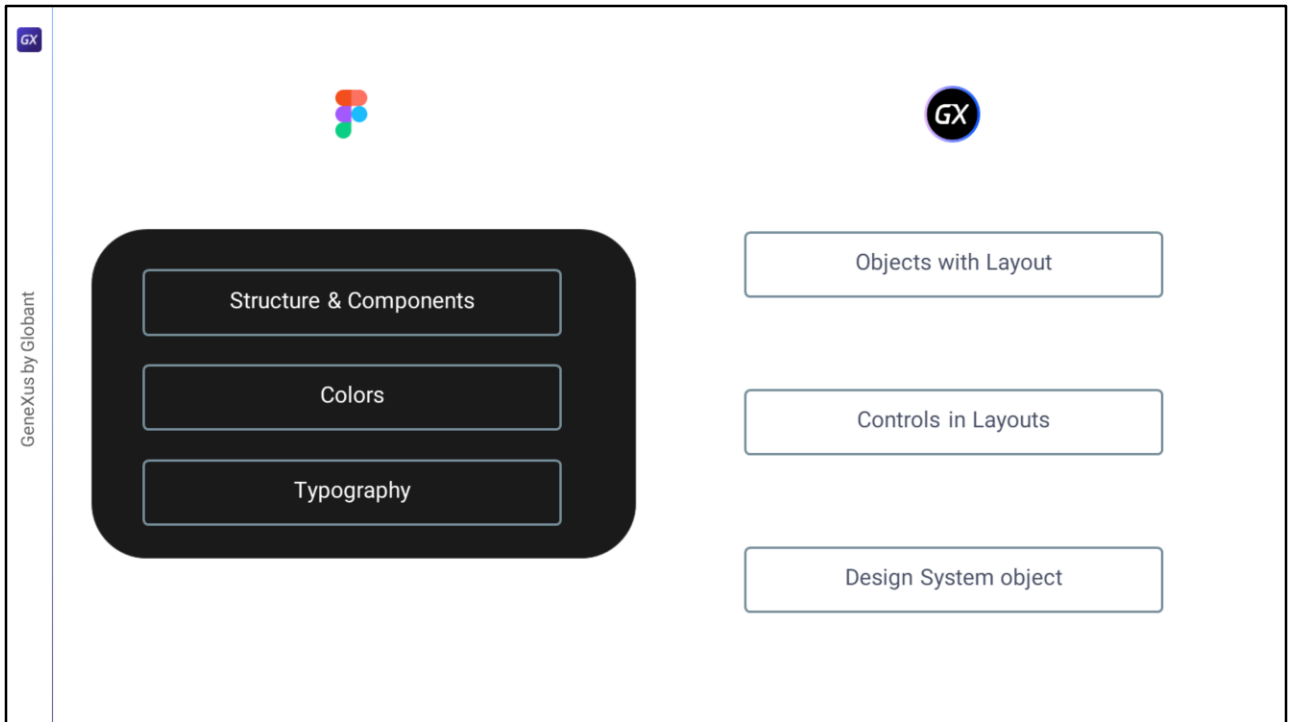


Cecilia Fernández



Na aula anterior, Chechu, a nossa designer principal, analisou estas dimensões do sistema de design que montou para implementar o design justamente da aplicação Travel Agency que queríamos desenvolver.

A questão que vai reger não só essa aula, que é uma espécie de revisão, mas no geral todo o curso, será como refletir tudo isso, o que ela estava falando conosco, que tem a ver com a montagem do sistema de design, como levá-lo ao GeneXus.



Então, vamos começar analisando esses pontos como uma revisão para nos introduzir no curso.

O primeiro ponto, a estrutura e os componentes, que têm a ver com a forma que são estruturadas as telas da aplicação e são componentizadas, terá a ver no GeneXus, em uma primeira dimensão, com a identificação de objetos com layout, que vão representar cada parte dessas telas.

E depois em uma segunda instância, já mais micro porque é a interna de cada objeto, teremos que ver como são organizados os controles dentro de cada layout, para refletir essa estrutura.

E então teremos que entrar a fundo no tema que tem a ver com o estilo propriamente dito, onde vão intervir as cores, a tipografia e outras questões que veremos e que serão dadas no GeneXus pelo objeto Design System.

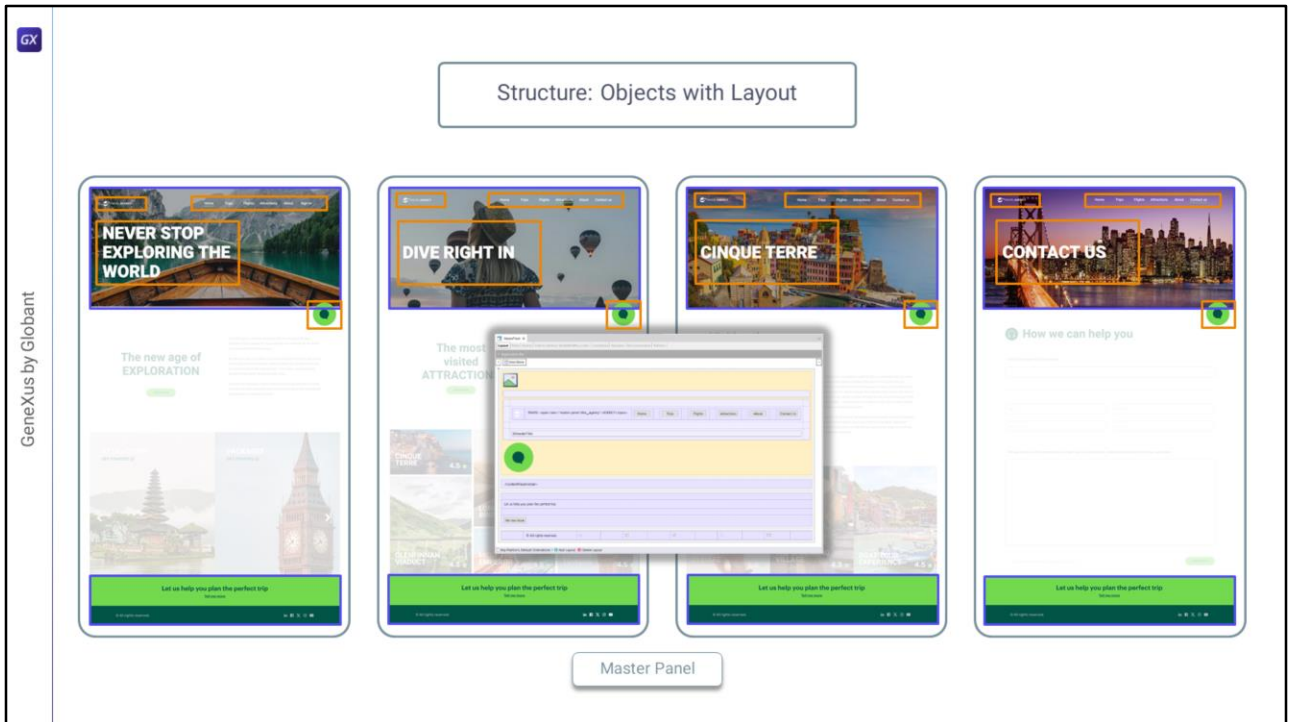
Então, vamos revisar nesta aula tudo isso antes de passar a aprofundar em cada coisa.

Então vamos começar visualizando um pouco a que nos referimos com a divisão de todas as telas em objetos com Layout.

Structure: Objects with Layout



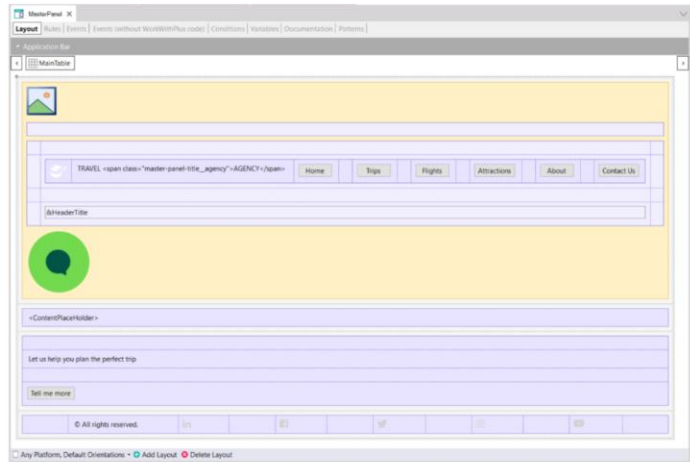
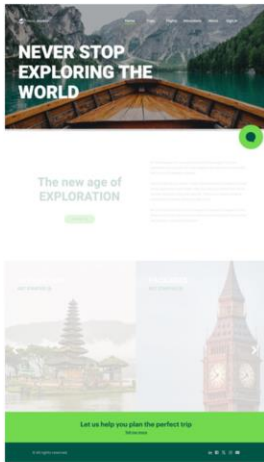
Se observamos as 4 telas que Chechu definiu...



...já vemos que tudo isso se repete em estrutura e estilo. Assim, o cabeçalho de cada página terá uma imagem de fundo que ocupa o mesmo espaço em cada tela (mesmo que a imagem em si varie, o tamanho e o tipo de imagem é igual); sobre a imagem serão sobrepostos o logotipo da empresa e o menu, um texto (que também irá variar de tela para tela, mas não em estilo ou em estrutura) e uma imagem para conversar. E por outro lado, também encontraremos repetido entre as telas o rodapé, que neste caso se repetirá igual não só em estilo, mas também em conteúdo.

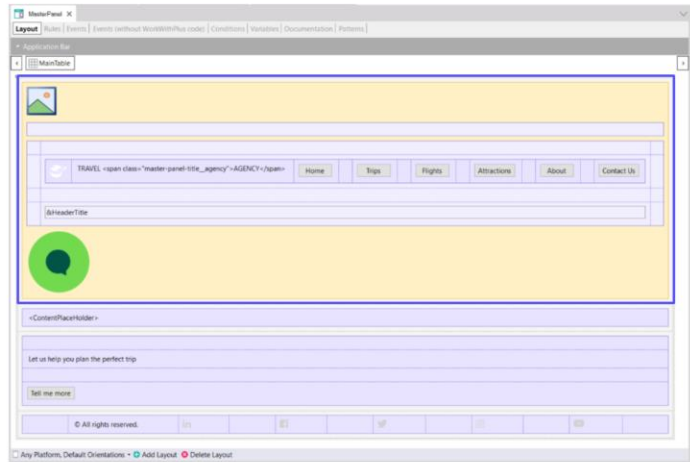
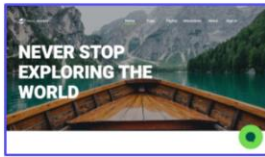
Sabemos que em GeneXus temos o objeto **Master Panel** que nos permite desenhar aparência e comportamento das partes que serão comuns a todas as telas (ou pelo menos a um conjunto delas: neste caso a estas primeiras 4 telas-).

Structure: Objects with Layout | Master Panel



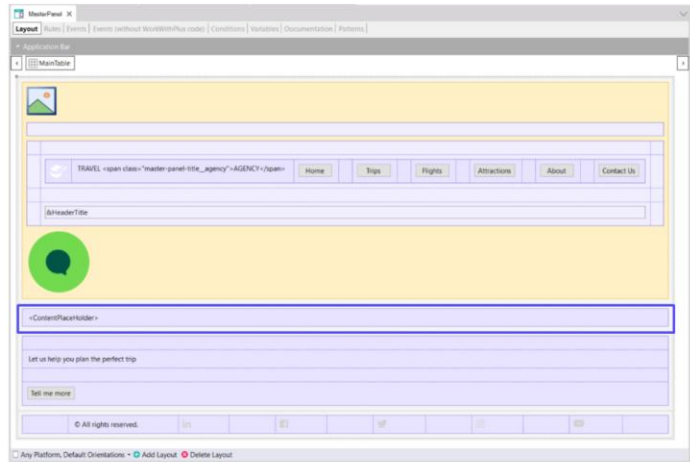
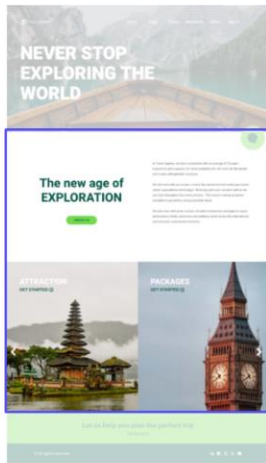
Então tudo isso que será repetido, vamos implementar em um objeto Master Panel.

Structure: Objects with Layout | Master Panel



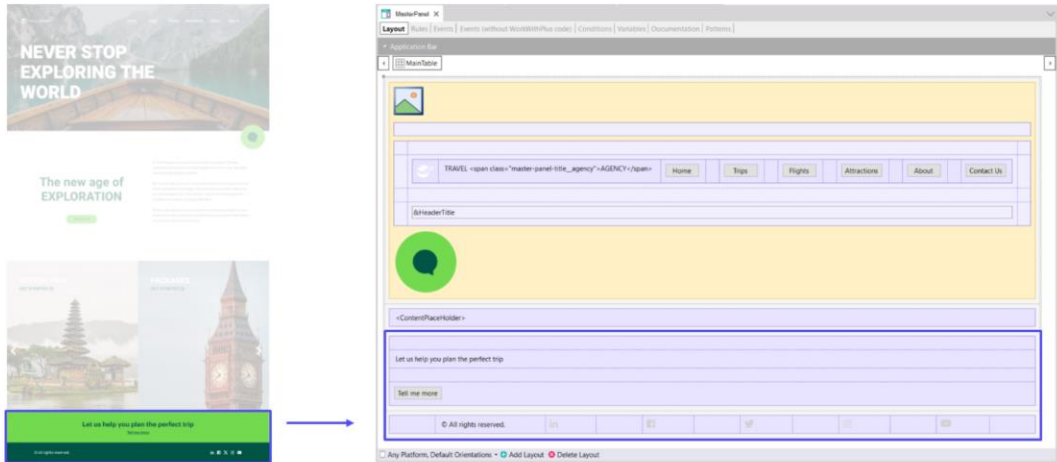
Onde na primeira linha da tabela do Master Panel vamos modelar o cabeçalho...

Structure: Objects with Layout | Master Panel

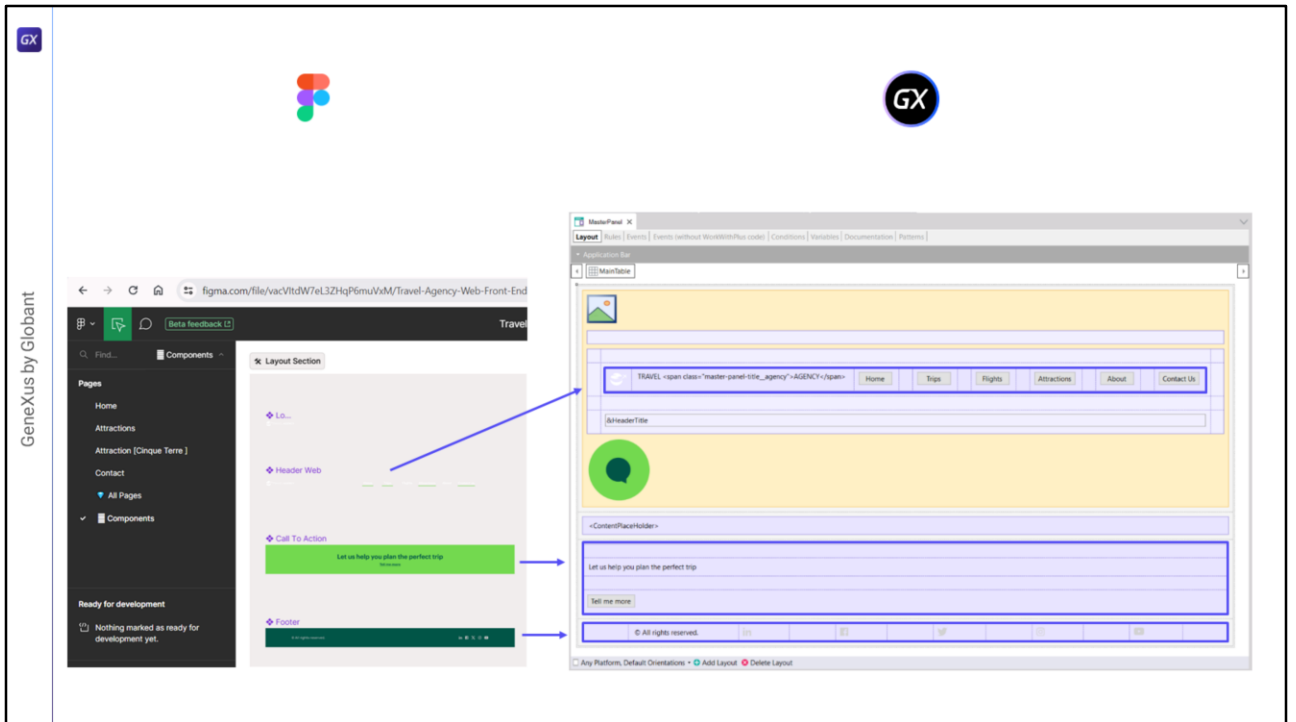


Na segunda linha teremos esse controle ContentPlaceholder que é específico do objeto Master Panel, que é justamente o que indica que ali dentro é onde será carregada a informação particular de cada página que tenha este, como seu Master Panel...

Structure: Objects with Layout | Master Panel

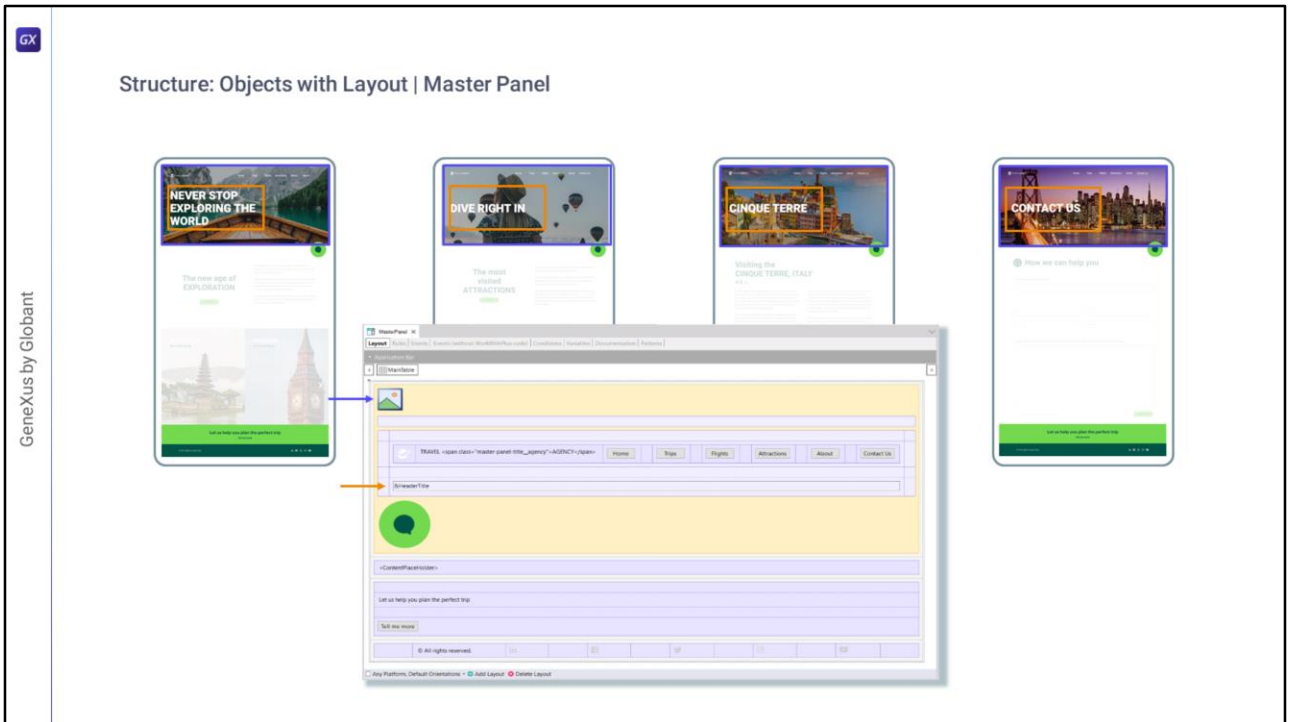


... e por último, próprio do Master Panel também, nas linhas 3 e 4 vamos modelar este rodapé.



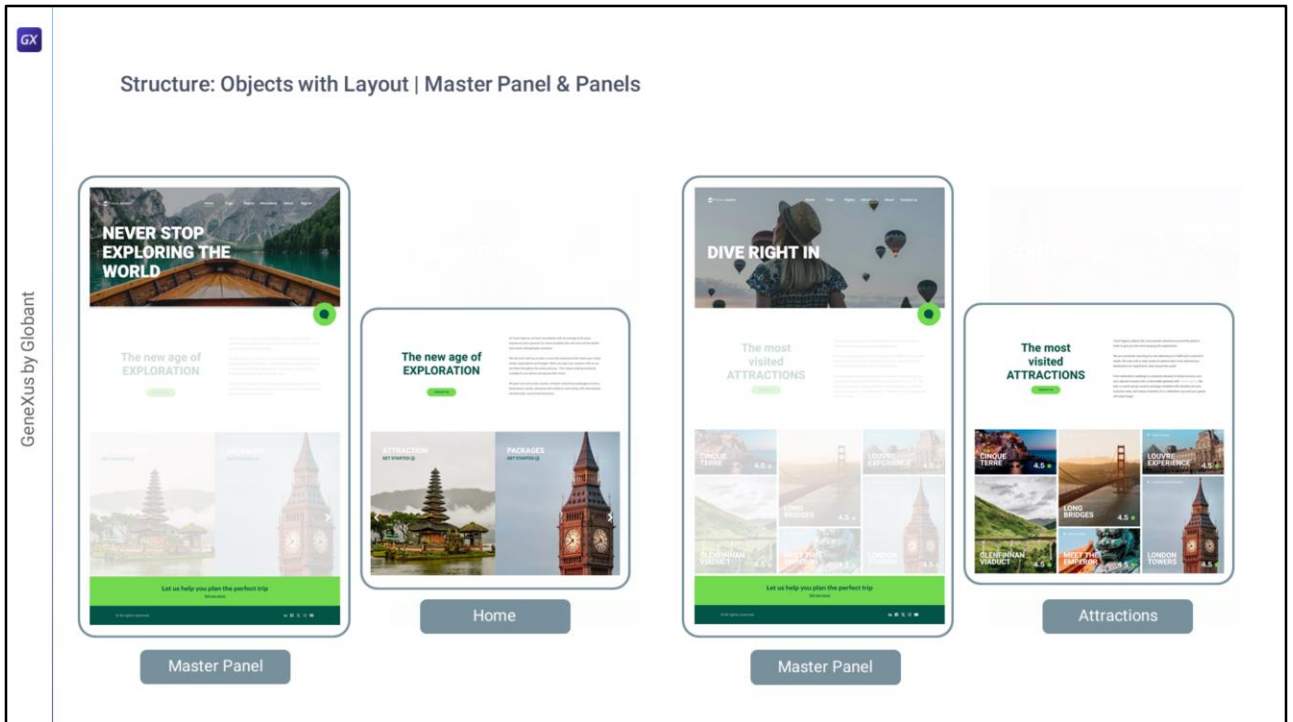
Bem. Se atendermos ao que modelou Chechu em Figma, ela não tinha como expressar isso. O máximo que conseguiu fazer foi modelar estas partes (o menu e essas partes do rodapé) como componentes, mas não consegue estabelecer uma relação entre eles porque não possui uma ferramenta que lhe permita expressar a mesma coisa que nós podemos expressar em GeneXus através de um objeto Master Panel.

É aqui que começamos a ver que as ferramentas de design e GeneXus não têm exatamente a mesma capacidade de expressão, por isso é necessário entender um pouco sobre ambas as ferramentas, em particular a ferramenta de design que está sendo utilizada, para poder fazer o melhor uso combinado das duas.



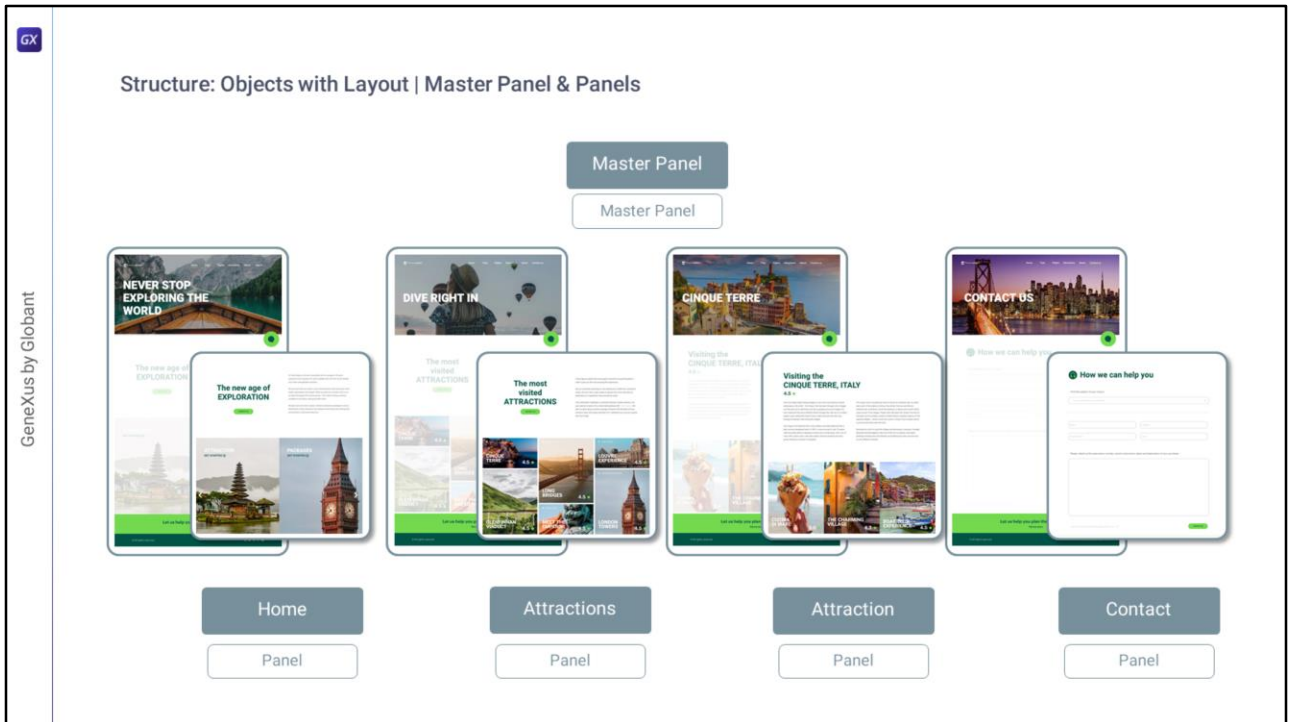
Bem, antes de passarmos para outra coisa. Talvez lhes tenha ficado uma dúvida rondando na cabeça em relação à parte que varia de conteúdo entre uma página e outra, mas dissemos que ainda iria ser implementada em um Master panel, que é justamente a imagem de fundo, que vemos que muda de tela para tela e o texto que é sobreposto a ela.

Bom, se no Master Panel os implementamos, esses dois elementos, com essa variável imagem e esse controle variável, então a resposta é fácil, virá fazendo variar o conteúdo desses controles de acordo com o objeto que está sendo carregado em cada oportunidade no ContentPlaceholder. E isso já pode ser imaginado que será por código na seção de eventos. Veremos isso mais adiante no curso.



Voltando... com o Master Panel podemos começar a realizar uma primeira estruturação: separando cada tela em duas: uma com tudo o que é repetido, neste caso header e footer, e outra com o conteúdo específico de cada página, que será modelada em um objeto Panel, que para esta tela chamei de "Home".

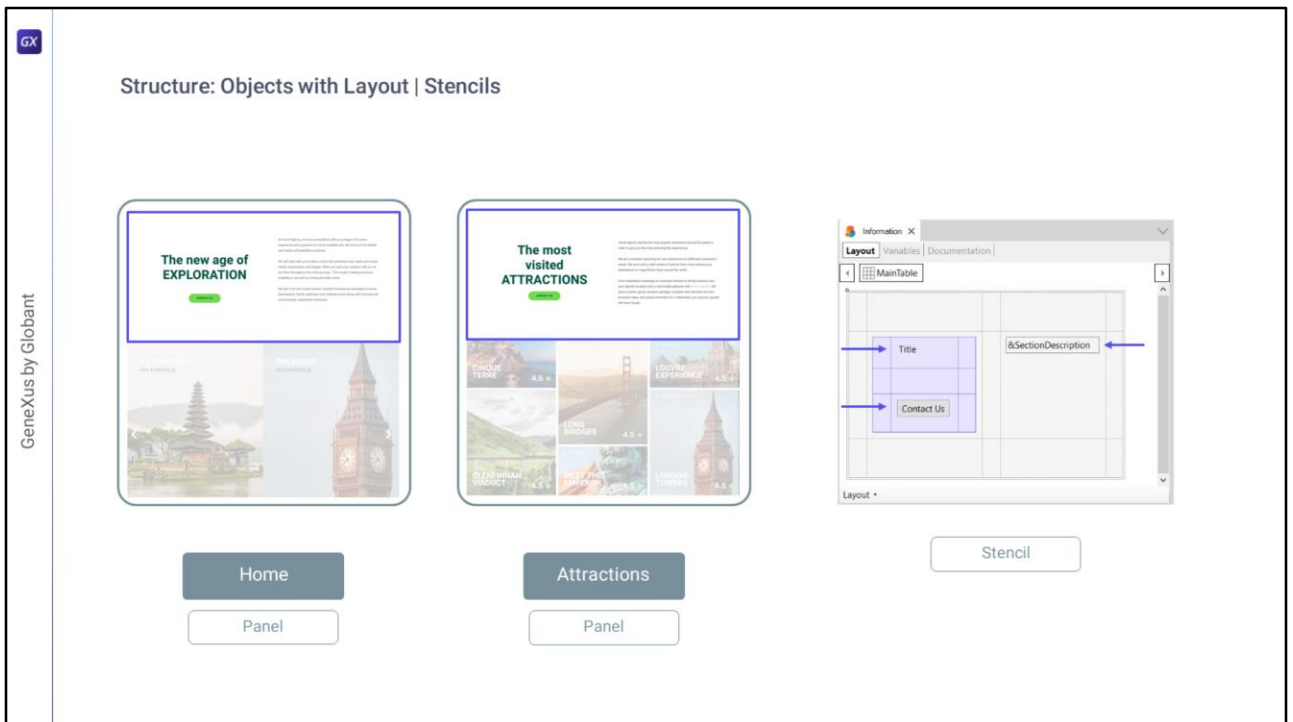
Para cada tela se repetirá a mesma coisa: a separação em Master Panel e Panel.



Assim então, ficará um objeto Master Panel, apenas um (aqui chamei o objeto de Master Panel, que é do tipo Master Panel; poderia ter chamado de TravelAgency, ou de qualquer outra maneira, não fui muito criativo), e 4 objetos panels, que serão carregados, cada um deles, no contentplaceholder do Master Panel.

Então fizemos esta primeira divisão das telas em Master Panel e Panels.

Mas não vamos parar por aqui, porque se continuarmos analisando as telas podemos notar que há mais estruturas que se repetem, embora o conteúdo varie, assim como vimos antes e, portanto...



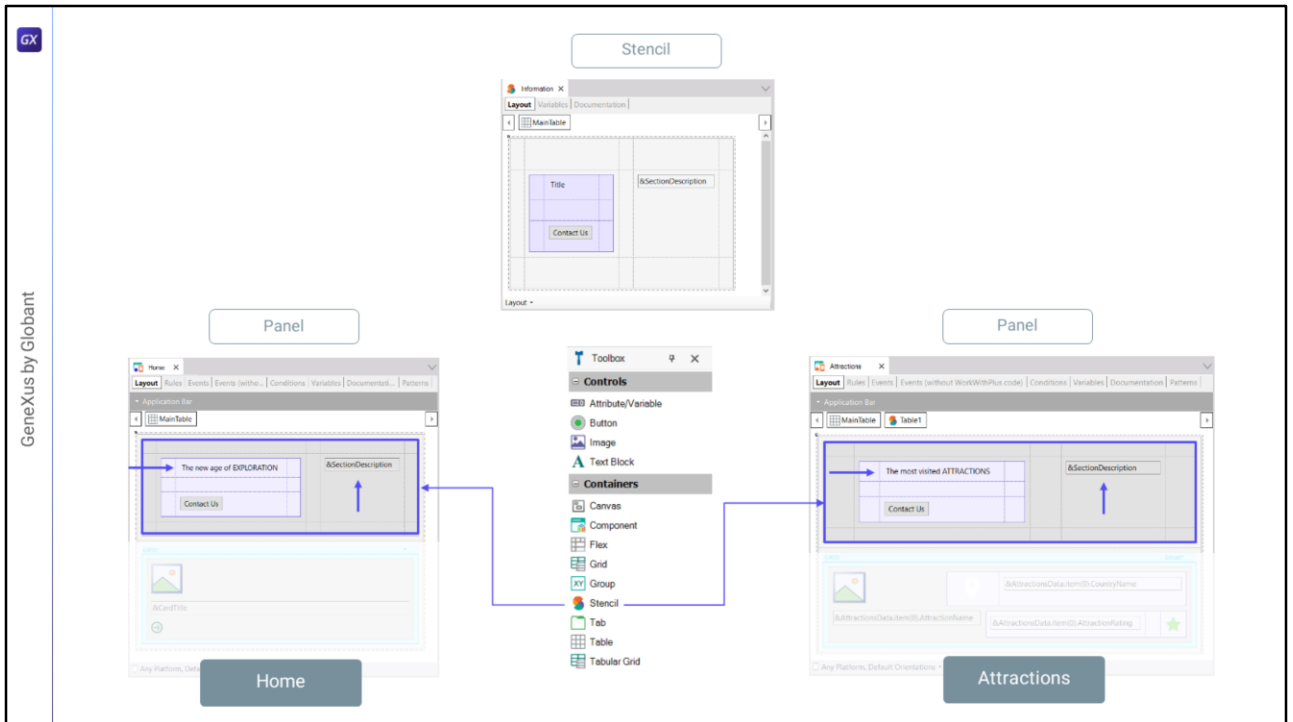
...essas partes podem ser isoladas para defini-las uma vez e reutilizá-las todas as vezes que forem necessárias. É o caso do que acontece entre esta parte do panel Home e esta de Attractions.

São iguais em estrutura visual e de comportamento (e digo de comportamento porque vemos ali um botão que chamará em ambos os casos exatamente a mesma tela), então posso implementar todas essas seções que se repetem com um objeto Stencil, que é utilizado justamente para isso.

Ou seja, é um objeto GeneXus com Layout, mas diferente dos outros que vimos, este não é executável por si só. Para ser utilizado e executado, deve ser instanciado.

Desta forma, o que faço é, no objeto stencil, definir a estrutura e estética sem me preocupar com o conteúdo, que será próprio de cada instância.

Por isso vemos ali um textblock, um botão e uma variável de texto. O textblock, vemos, tem um caption genérico, e a variável não tem nada carregado, é uma variável de texto, nada mais.

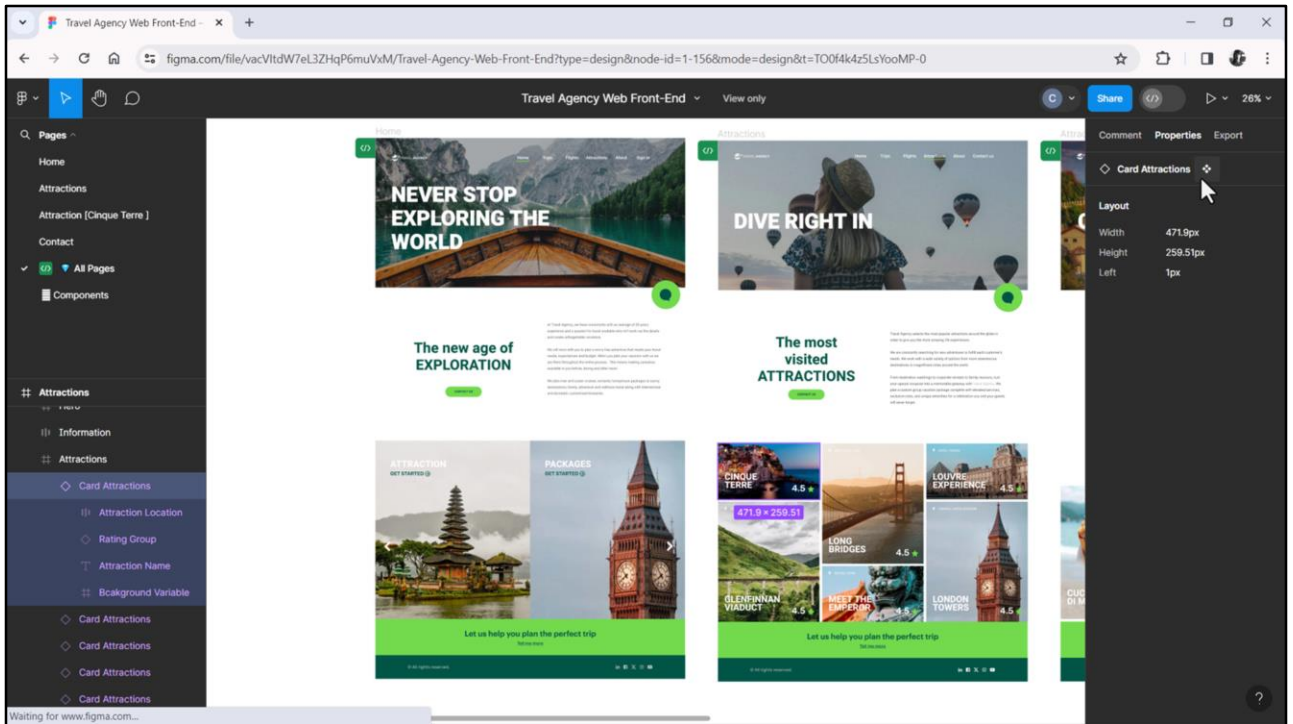


E depois o que faço é **instanciar** o stencil nos painéis onde preciso.

Assim, inserimos na primeira linha de ambos os painéis um **controle** stencil, no qual será carregado o conteúdo do **objeto** stencil, mas no qual podemos modificar aquelas questões que têm a ver com o conteúdo. Por exemplo, o caption do textblock modifico em cada um desses painéis para que adote a particularidade que preciso para cada um. E por exemplo, também vou ter esta variável aqui que vou carregar em execução nos eventos com o conteúdo particular e específico.

Então conseguimos outra forma de reutilização. Se precisarmos modificar algo na estrutura ou na estética, fazemos isso no objeto stencil, uma vez, e isso se aplica aos painéis onde esse stencil esteja instanciado.

Chechu, nossa designer, não identificou essa repetição...

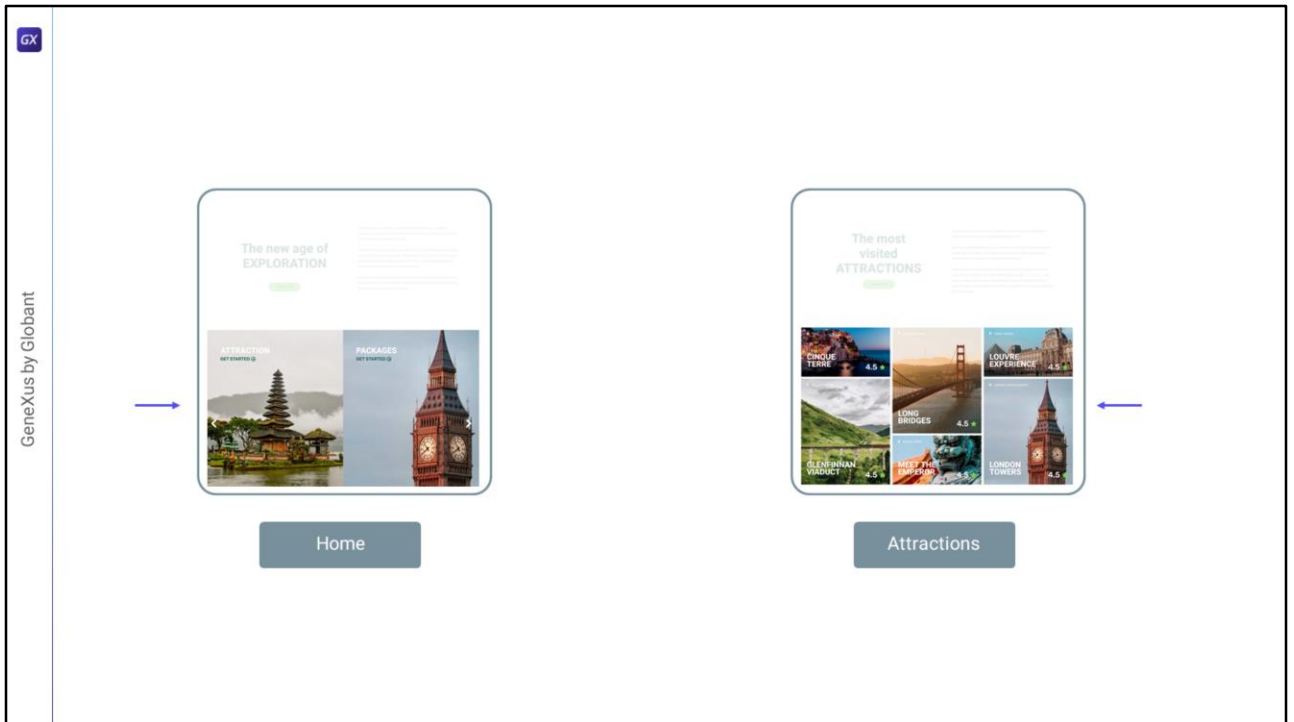


Se formos inspecionar as duas telas, vemos que possuem um grupo que foi chamado de Information, foi chamado igual em ambos os casos, mas não o componentizou... e por isso não aparece na página de componentes, como veremos, como acontecerá por exemplo com esses cards... Se formos ver aqui vemos que está indicando que é uma instância de um componente e ali acesso o componente, na verdade me leva à página de componentes e até poderia retornar à instância da qual parti. E aqui volto ao card, este dentro da página Home.

E a mesma coisa, se eu me posicionar nesse outro elemento vemos que é uma instância de um componente... que é esse aqui, que está na página de componentes.

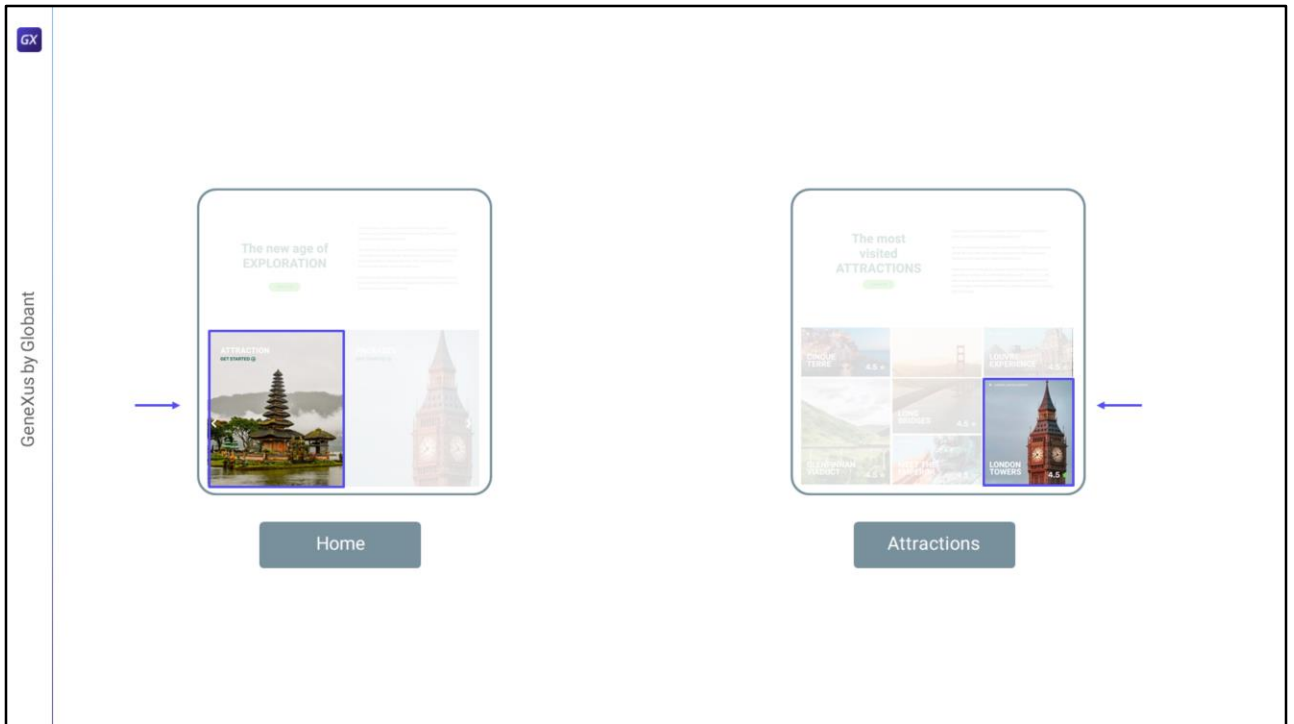
Chechu certamente não componentizou o grupo Information de que falávamos porque isso não lhe pareceu tão óbvio. Nós podemos fazer isso. Não pareceu óbvio para ela, para nós sim.

Mas por outro lado, era evidente para ela que tinha que componentizar esses cards, então, tendo identificado esses cards como componentes, nos dá esta pista de que possivelmente podemos componentizá-los também, no nosso caso criando stencils. Então... para poder reutilizá-los...

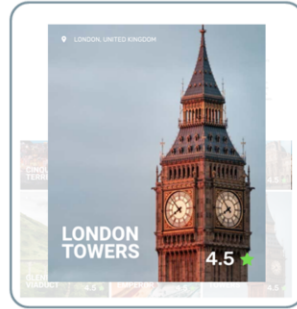


Se observarmos estes dois painéis, vemos que temos aqui estes dois carrosséis onde se repetem esses cards.

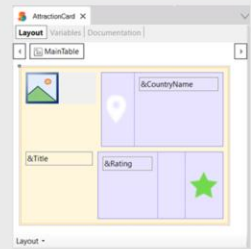
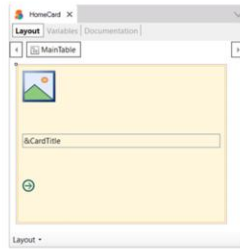
Ou seja, vamos precisar de uma estrutura repetitiva para modelar os carrosséis, que já podemos antecipar que serão grids de diferentes tipos: em um caso será horizontal, em outro caso será um grid do tipo flex...



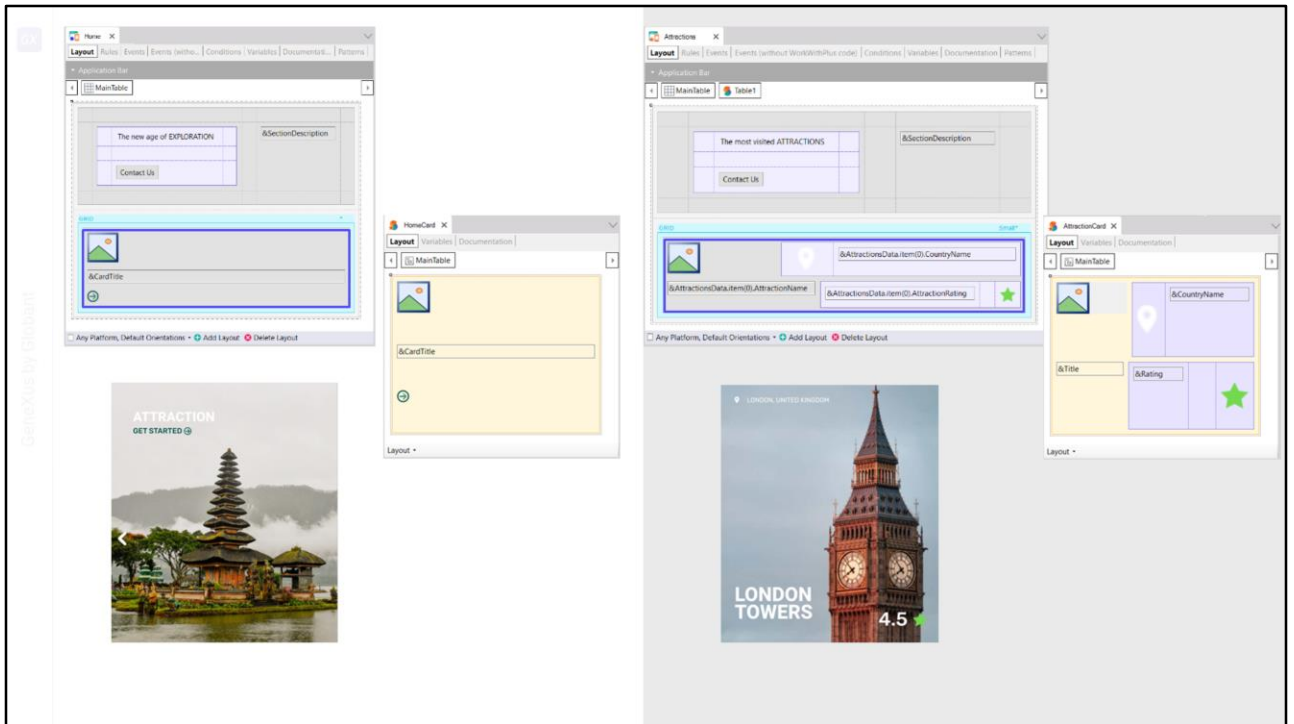
E onde o que vamos fazer é reutilizar... Será um grid composto por elementos que serão do tipo: estes cards. Então, como se repetem estilo e conteúdo, já podemos antecipar que seria interessante...



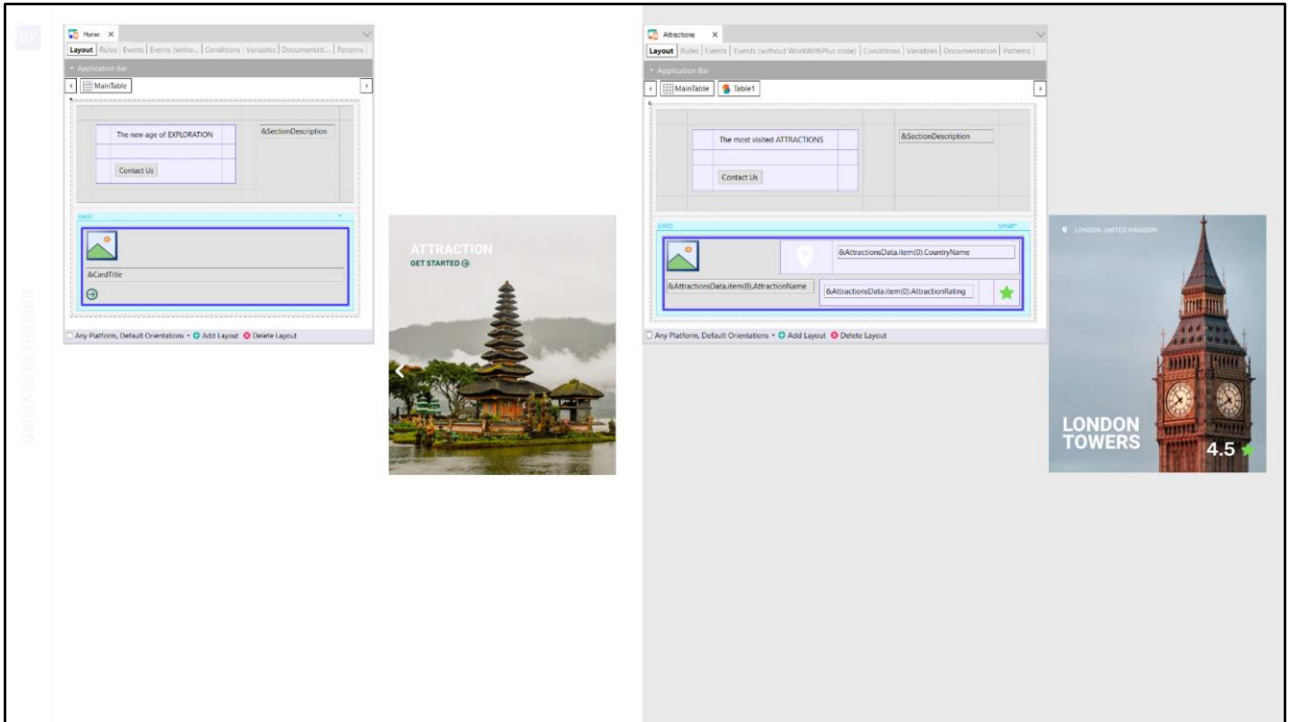
...modelá-los como stencils.



Então o que vamos fazer é criar um stencil para cada um, ali vamos modelar seu layout, e depois...



...simplesmente, como vimos antes, vamos instanciar cada stencil no panel, no primeiro caso no grid, como elementos de cada um desses grids.



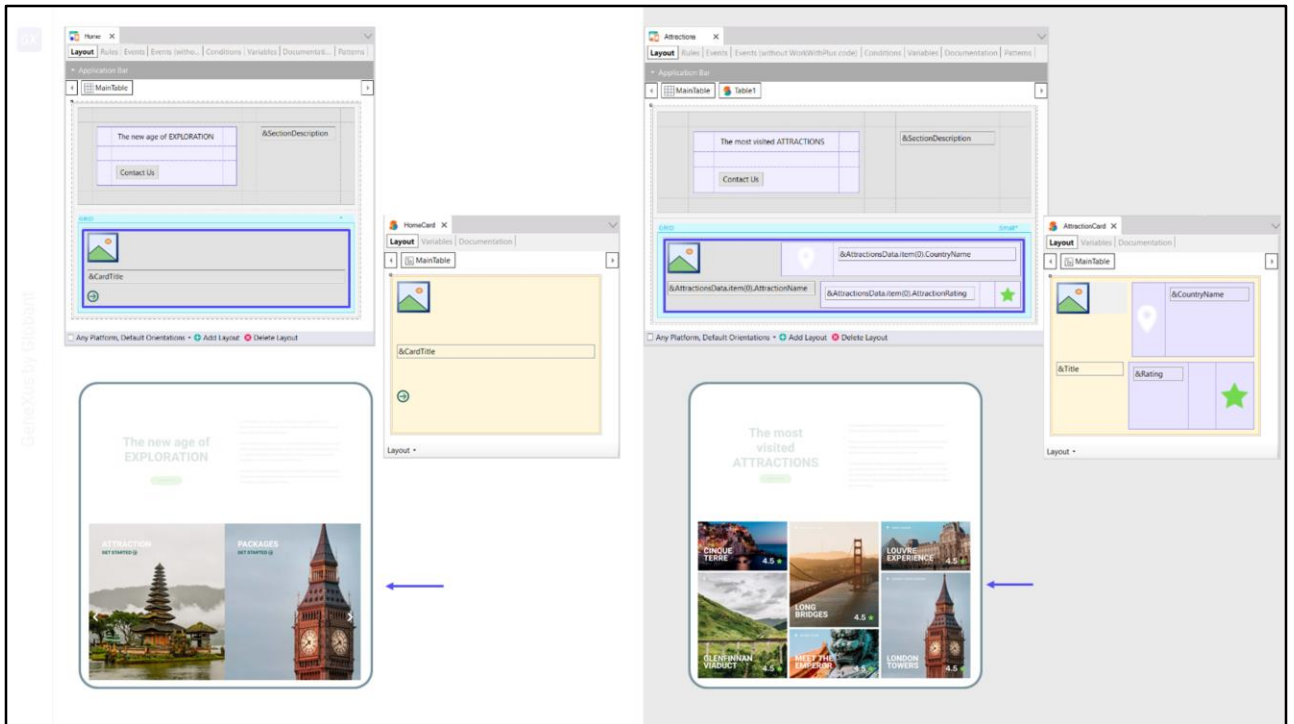
Então, ao fazer isso...

Home X
Layout Rules Events Events (with...) Conditions Variables Documentation Patterns
Application Bar
MainTable
The new age of EXPLORATION &SectionDescription
Contact Us
Title
&CardTitle
Any Platform, Default Orientations - Add Layout Delete Layout

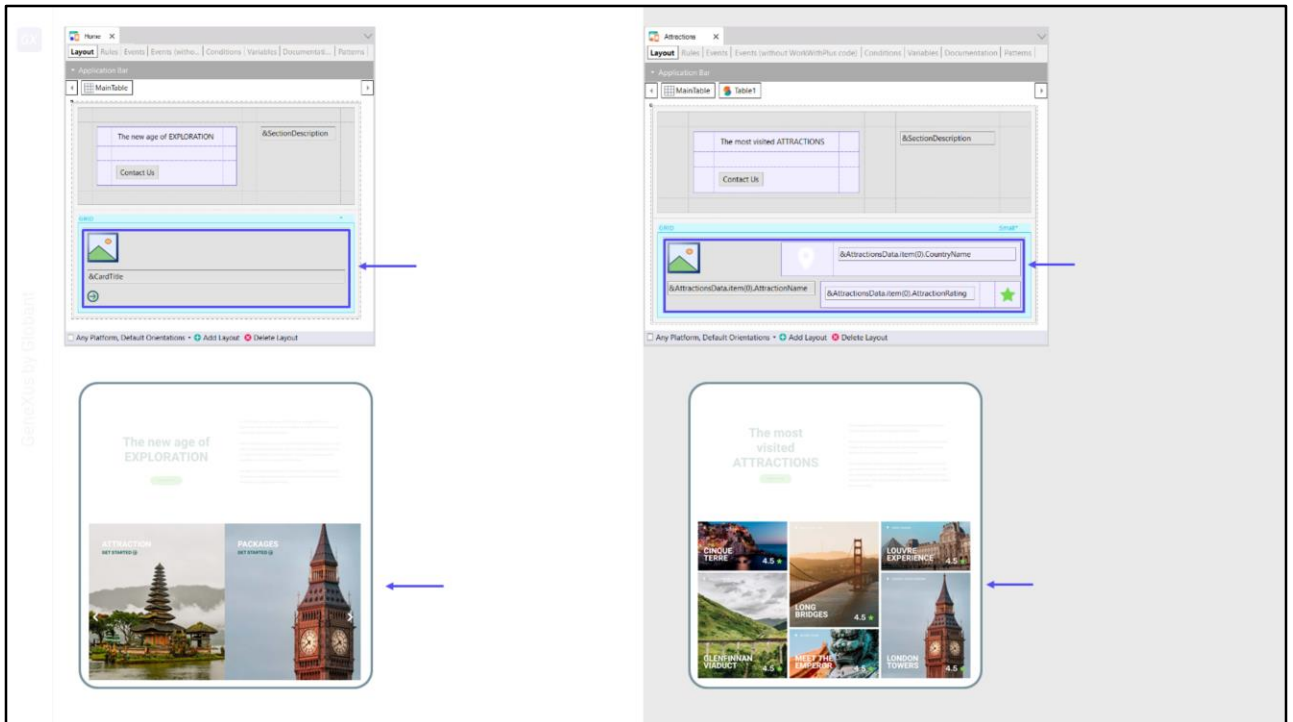
HomeCard X
Layout Variables Documentation
MainTable
&CardTitle
Layout

Attractions X
Layout Rules Events Events (without WorkWithPlus code) Conditions Variables Documentation Patterns
Application Bar
MainTable Table1
The most visited ATTRACTIONS &SectionDescription
Contact Us
Title
&AttractionsData.item(0).CountryName
&AttractionsData.item(0).AttractionName &AttractionsData.item(0).AttractionRating
Any Platform, Default Orientations - Add Layout Delete Layout

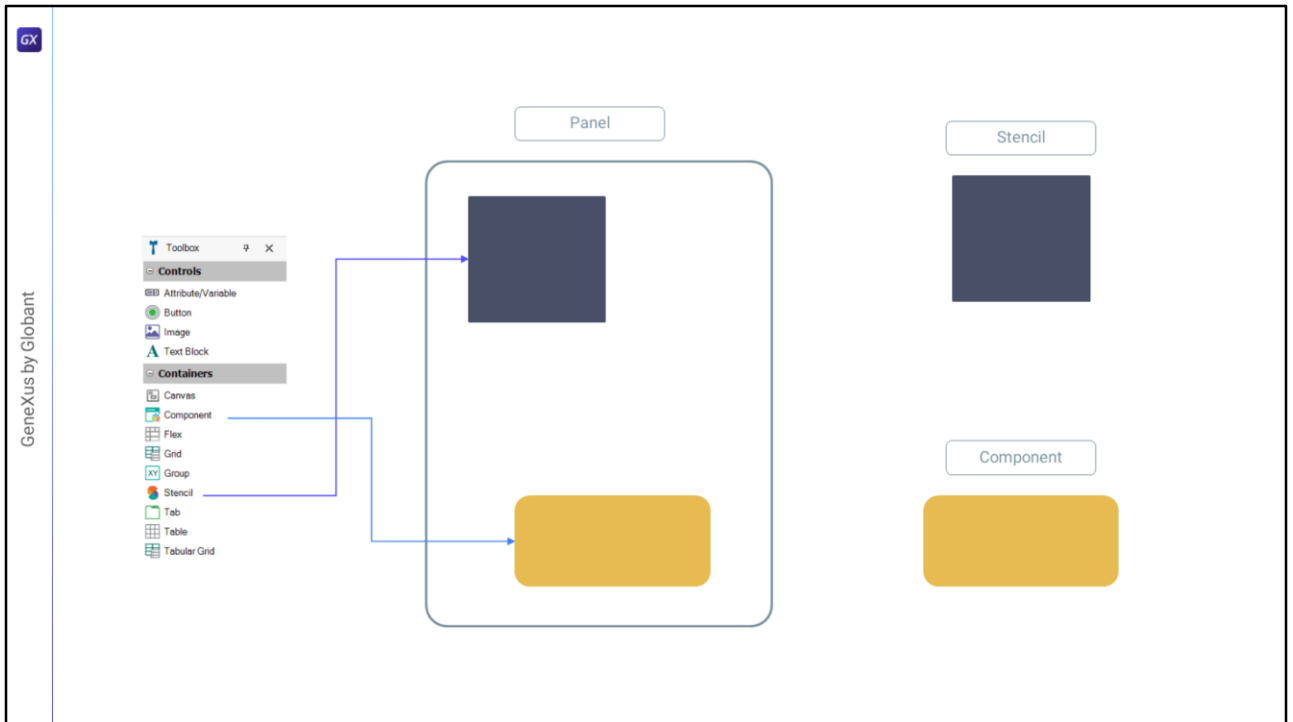
AttractionCard X
Layout Variables Documentation
MainTable
&CountryName
&Title &Rating
Layout



Tenho modelado o grid como uma repetição desses stencils...



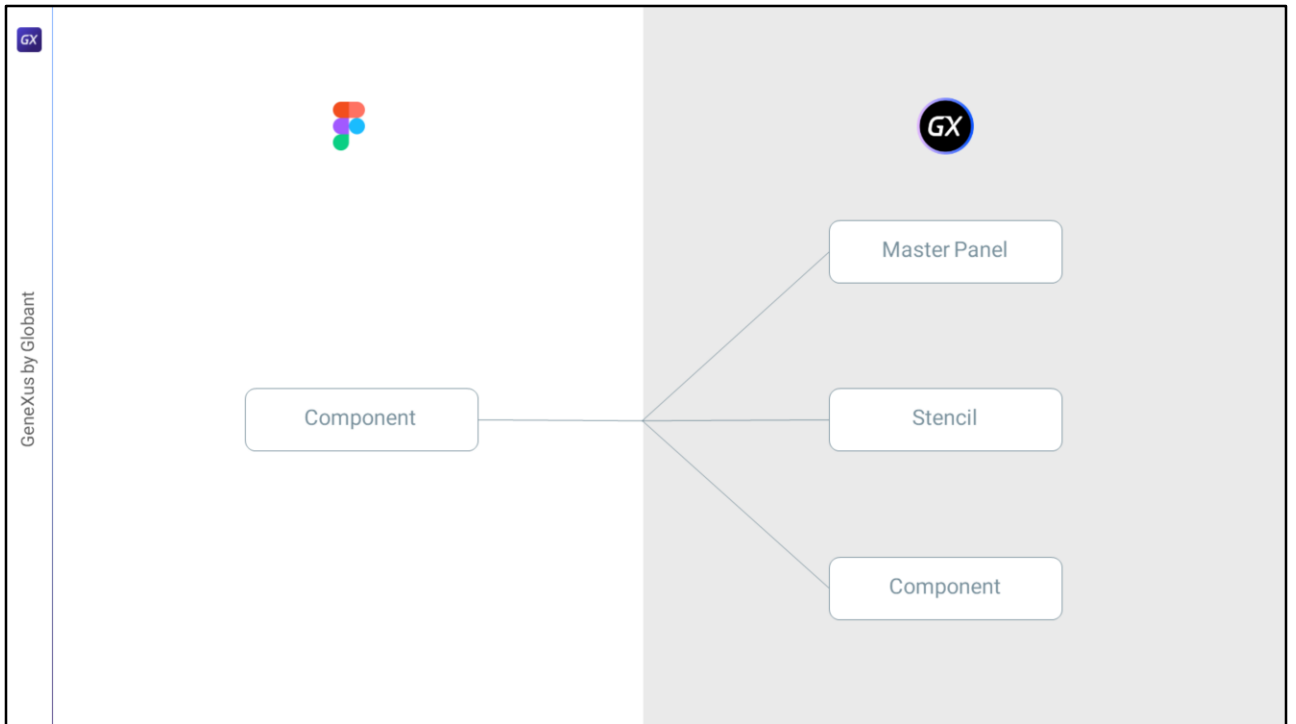
E depois o que vou precisar é simplesmente, nos eventos, carregar cada item, cada conteúdo, com a informação correspondente. Que no segundo caso terei que pegar da base de dados.



Bem, em GeneXus, além de poder componentizar uma seção de layout com estética definindo-a em um objeto stencil, como fizemos até agora, que depois instanciamos no layout do objeto que necessita, temos outra forma, que é utilizada sobretudo quando também implica comportamento (embora os stencils hoje permitem certo grau de comportamento)...

Se estivermos desenvolvendo no mundo dos Web Panels será com o objeto do tipo Web Component, e se estivermos desenvolvendo no mundo dos Panels será um panel comum e normal. Então aqui eu identifiquei isso como Componente mas vai ser implementado, dependendo do mundo: se for o mundo dos Web Panels com um Web Component, se for o mundo dos Panels com outro Panel. Que será inserido em qualquer um dos casos, será inserido no layout onde deseja utilizar através de um controle do tipo Component.

Em resumo, o que quero mostrar com isso? Que em GeneXus temos várias opções para fazer com que uma parte da tela seja isolada e modelada separadamente. Na verdade temos outra que é o uso de User Controls.

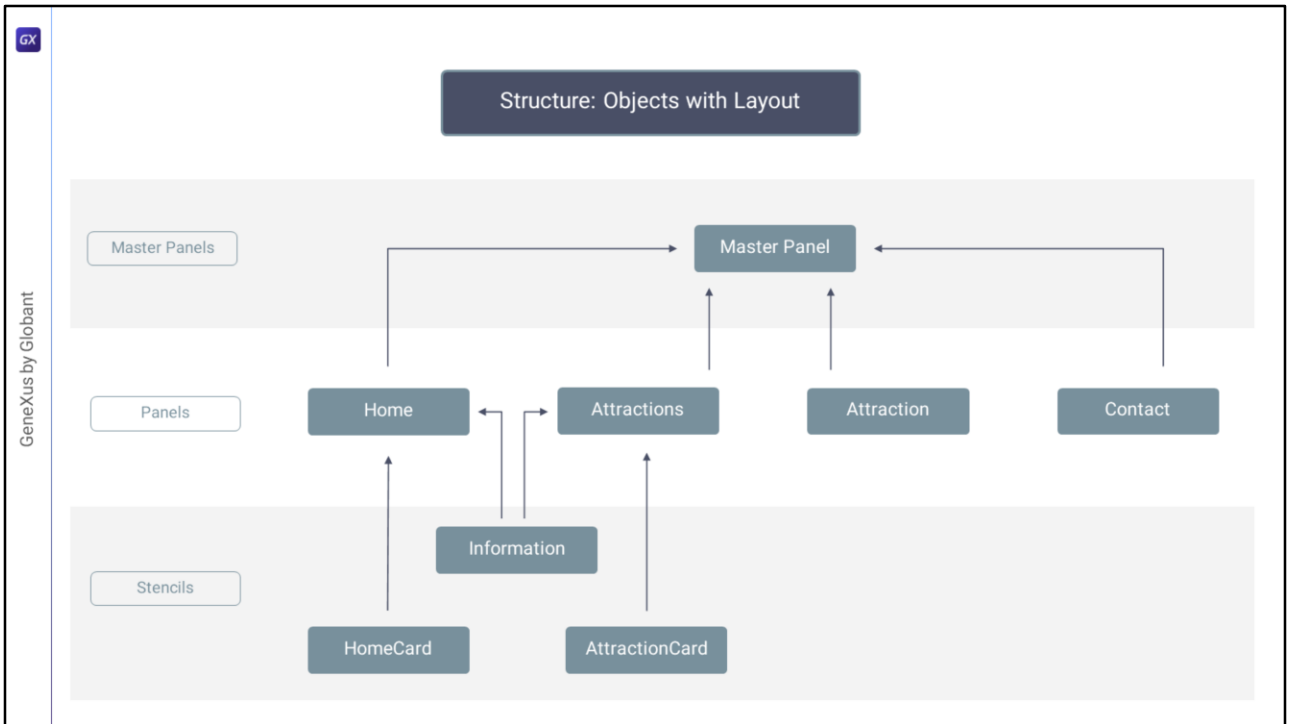


Então um componente que Chechu isolou no Figma pode ser representado no GeneXus de várias maneiras: tanto dentro de um Master Panel, como de um stencil, como de um componente. E também um User Control.

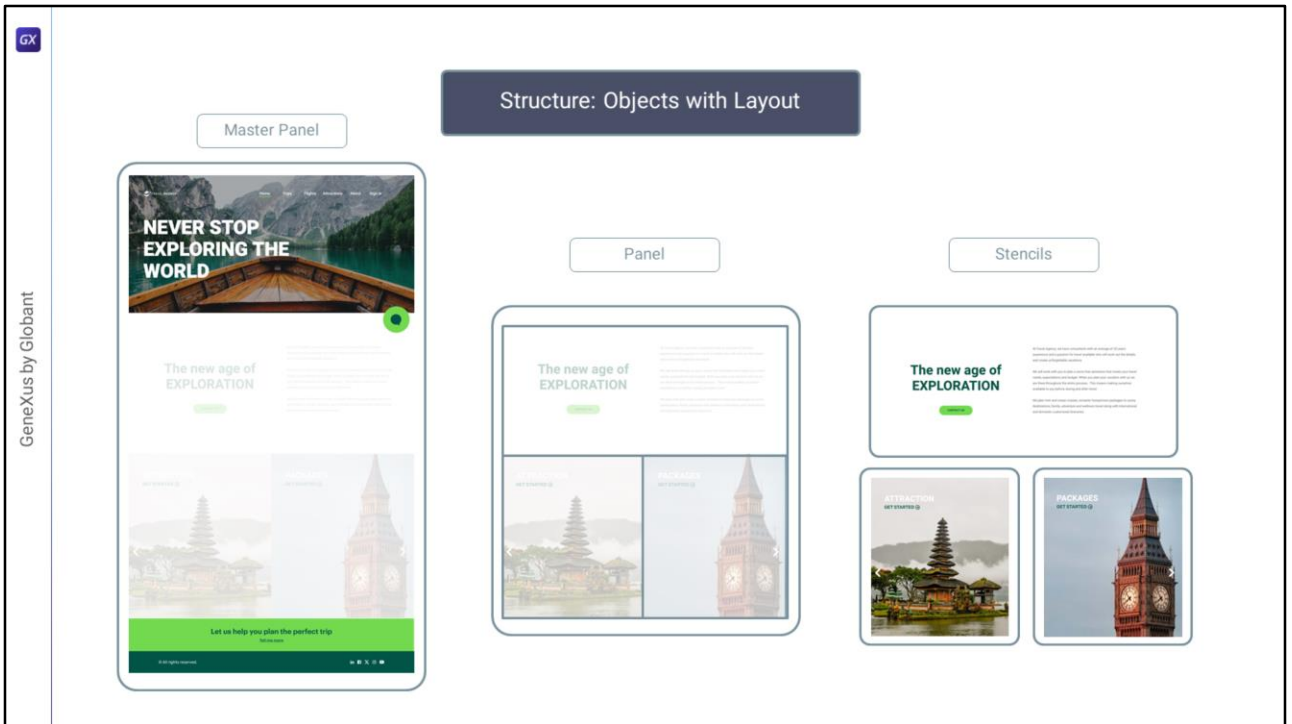
Structure: Objects with Layout



Em qualquer caso, com isso fica clara esta primeira divisão estrutural da aplicação em objetos com Layout, que estão inter-relacionados...

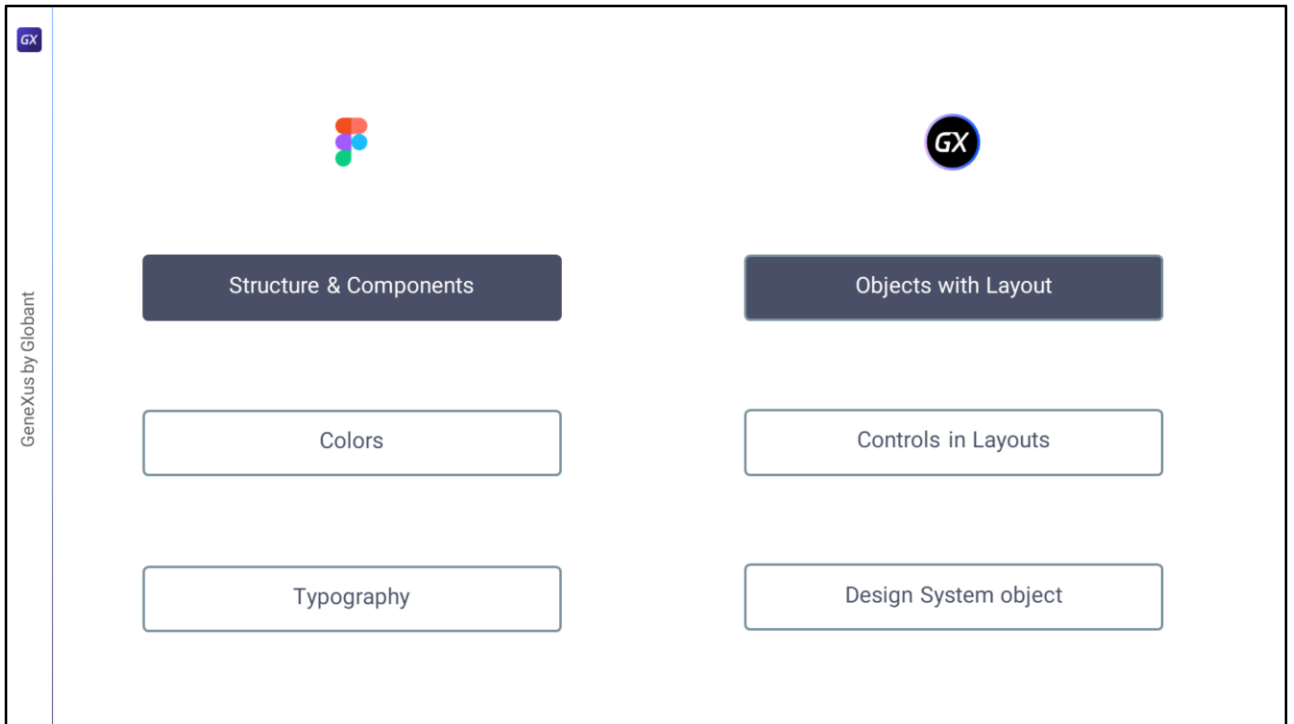


...como vemos neste esquema que fiz aqui, onde vemos o Master Panel, os quatro painéis que identificamos até agora, e aqueles três stencils que vimos.

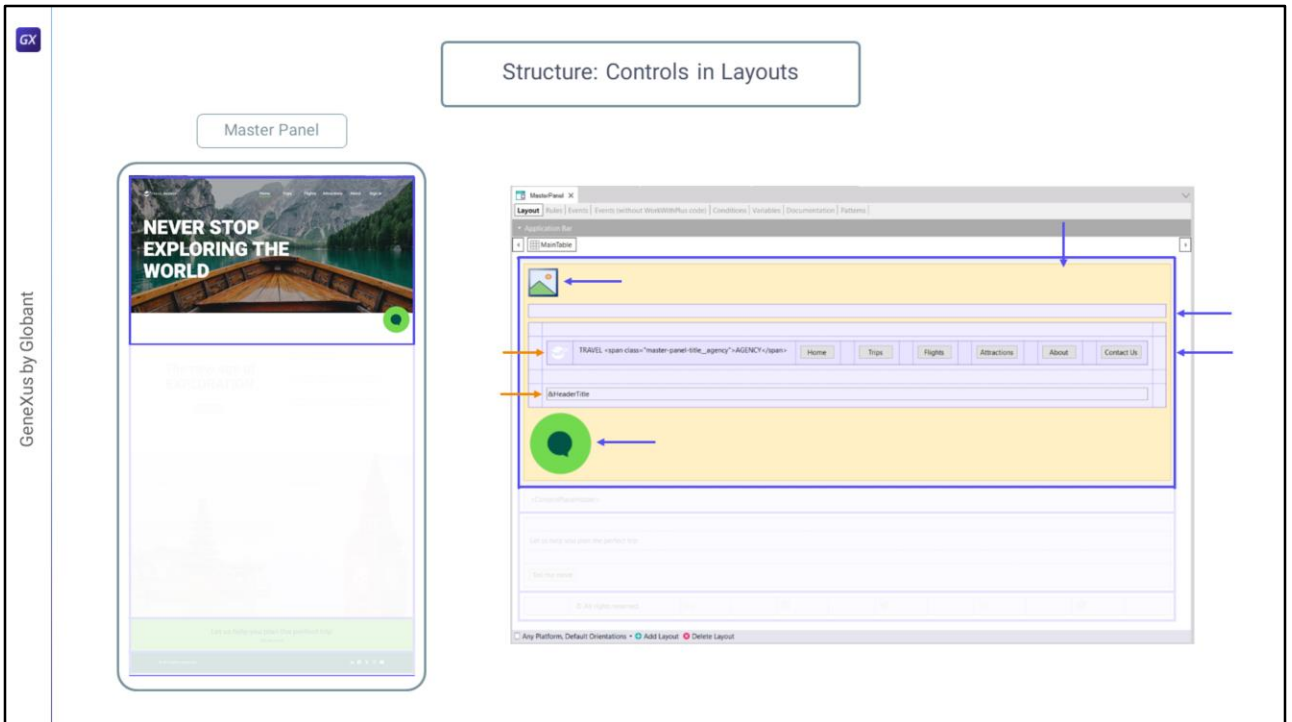


Então, se pegarmos a primeira tela... fica dividida primeiro em Master Panel e Panel; mas o Panel por sua vez fica também dividido em seus controles, como o grid por exemplo, e Stencils.

O problema, então, foi reduzido agora a um problema menor, mas de mesma complexidade, que é: como implemento o layout de cada um desses objetos?



E agora é que vamos passar para o segundo ponto, o dos controles nos layouts. Como organizamos os layouts para que a estrutura reflita o que será a estrutura das telas?



A pergunta então é como estruturamos o layout de cada objeto. Do Master Panel, do Panel e do Stencil em questão.

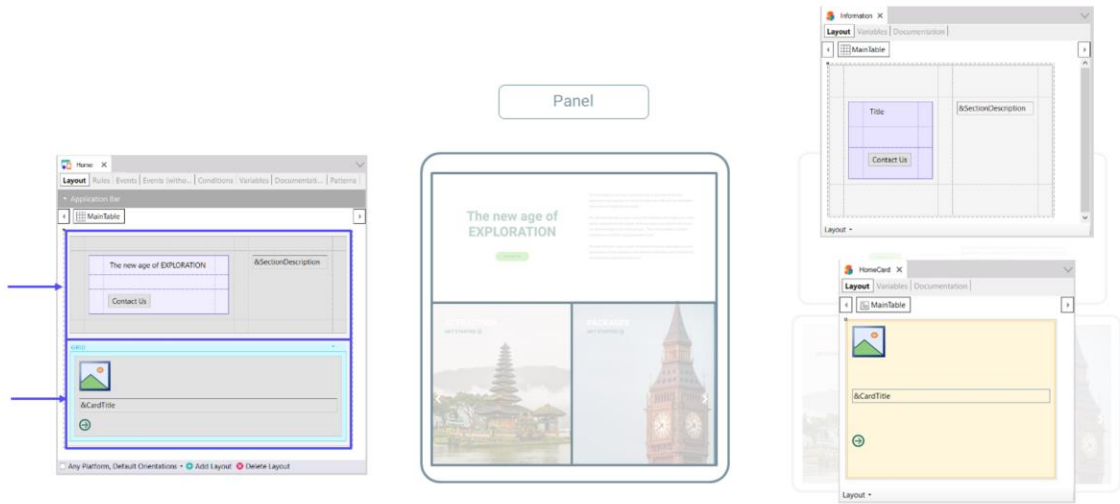
Por exemplo, se analisarmos o Master Panel: aqui teremos uma tabela main (principal) com 4 linhas...

Dentro de cada uma estão localizados controles que por sua vez podem conter outros controles... como nesta primeira linha onde temos uma tabela especial, canvas, (a tabela canvas é aquela que permite sobrepor controles), que contém por sua vez um controle de tipo variable imagem, duas tabelas e um controle de tipo imagem.

A segunda destas tabelas simultaneamente contém outra e um controle variable.

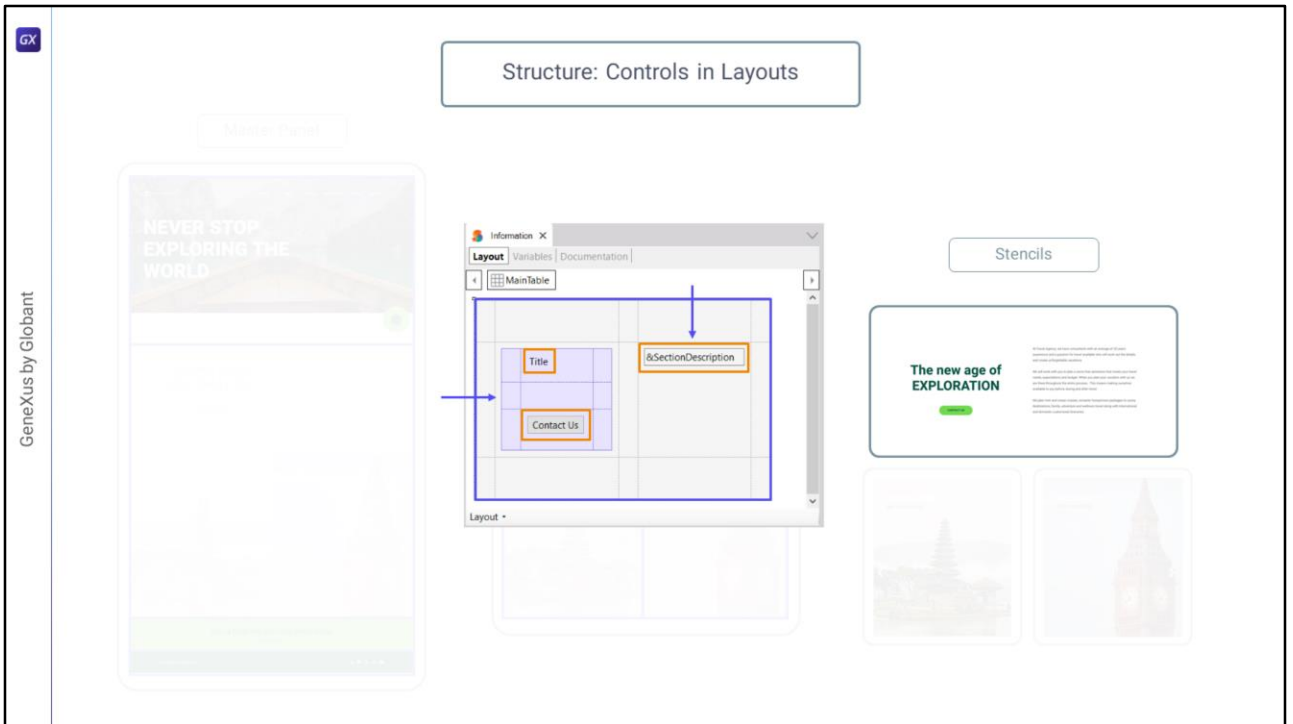
E assim sucessivamente... Ou seja, temos uma hierarquia de controles, onde o pai de todos é a tabela main.

Structure: Controls in Layouts



Analogamente, se analisarmos o panel Home ele também terá uma tabela main... desta vez com 2 linhas...

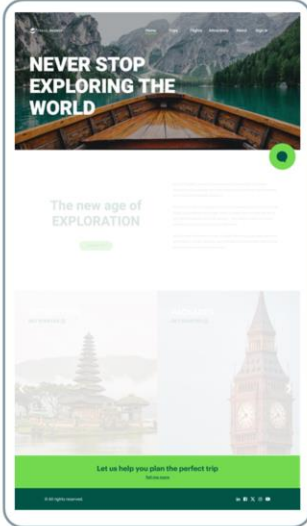
A primeira conterá um controle stencil e a segunda um grid que conterá outro controle stencil.



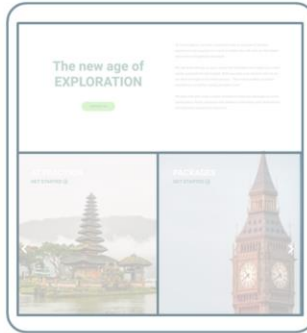
E por outro lado, se analisarmos, por exemplo, o primeiro dos stencils... outra vez... teremos uma tabela main com 3 linhas e 1, 2, 3, 4, 5 colunas. A célula da linha 2, coluna 2 terá por sua vez, outra tabela, que terá lá dentro um controle text block e um botão. E a célula na linha 2, coluna 4 terá um controle variable.

Structure: Controls in Layouts

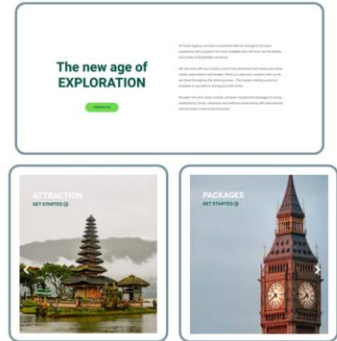
Master Panel

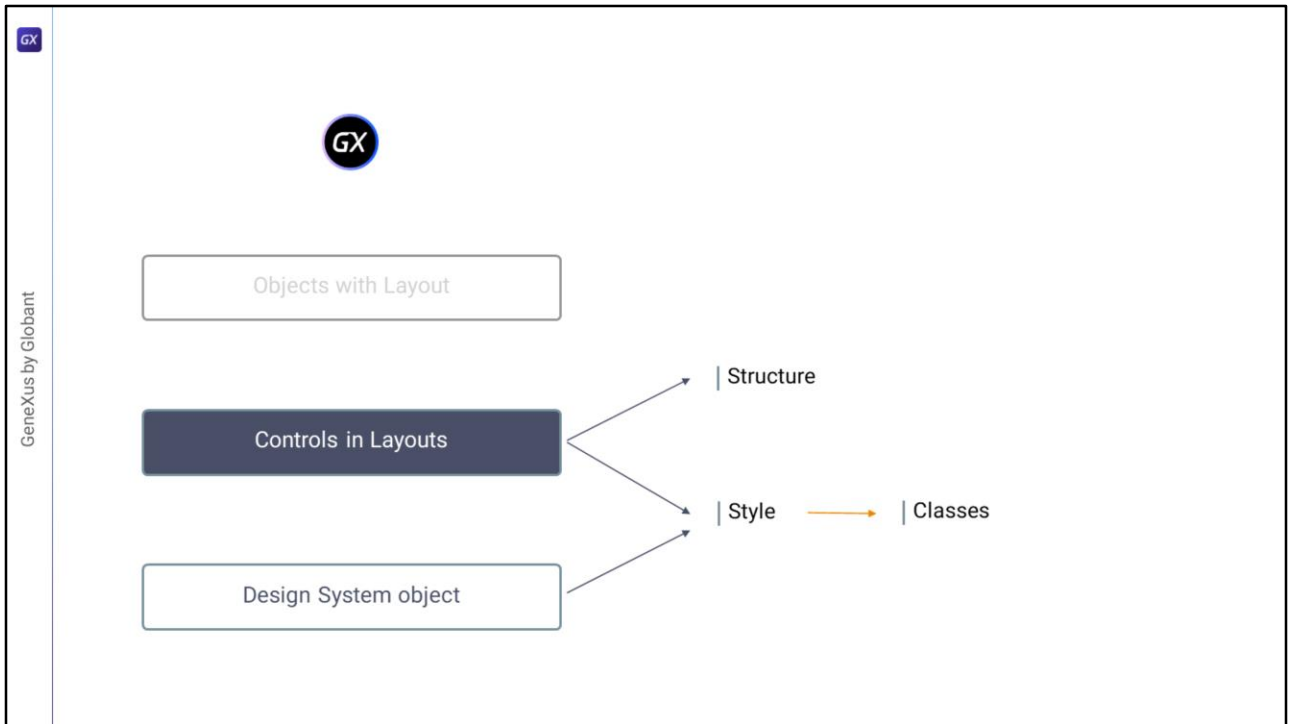


Panel

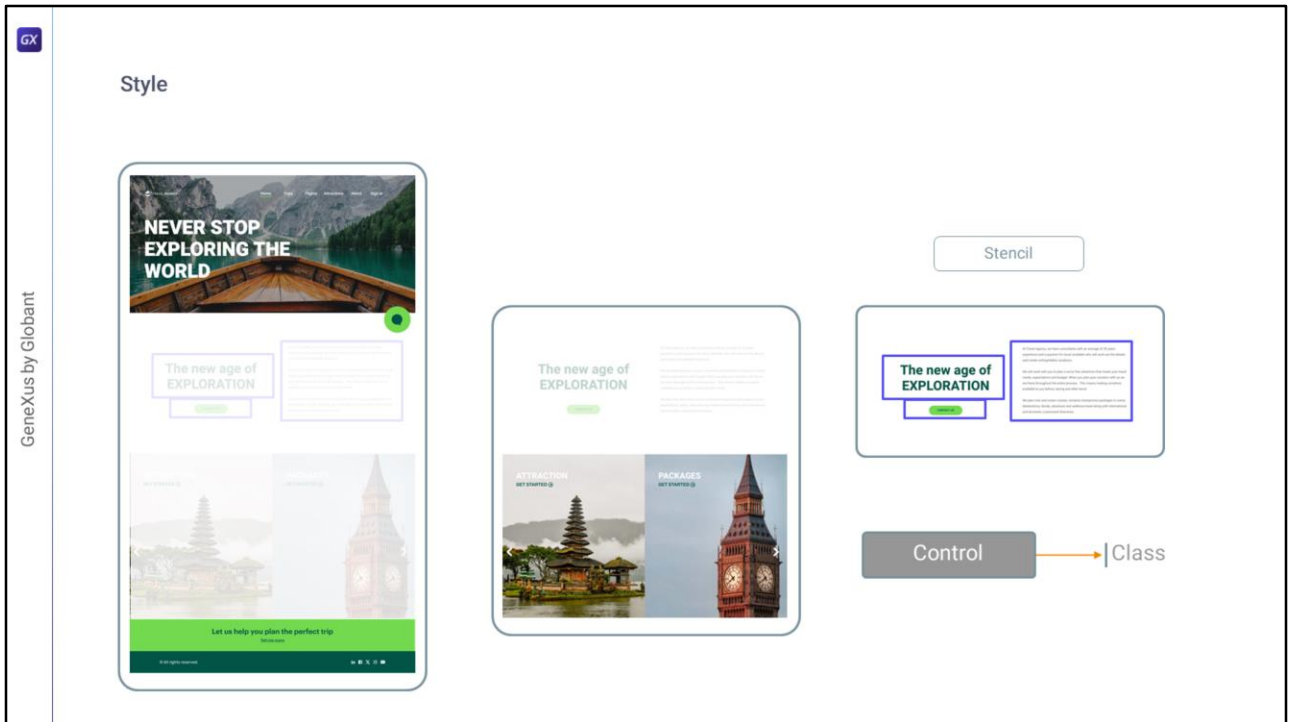


Stencils



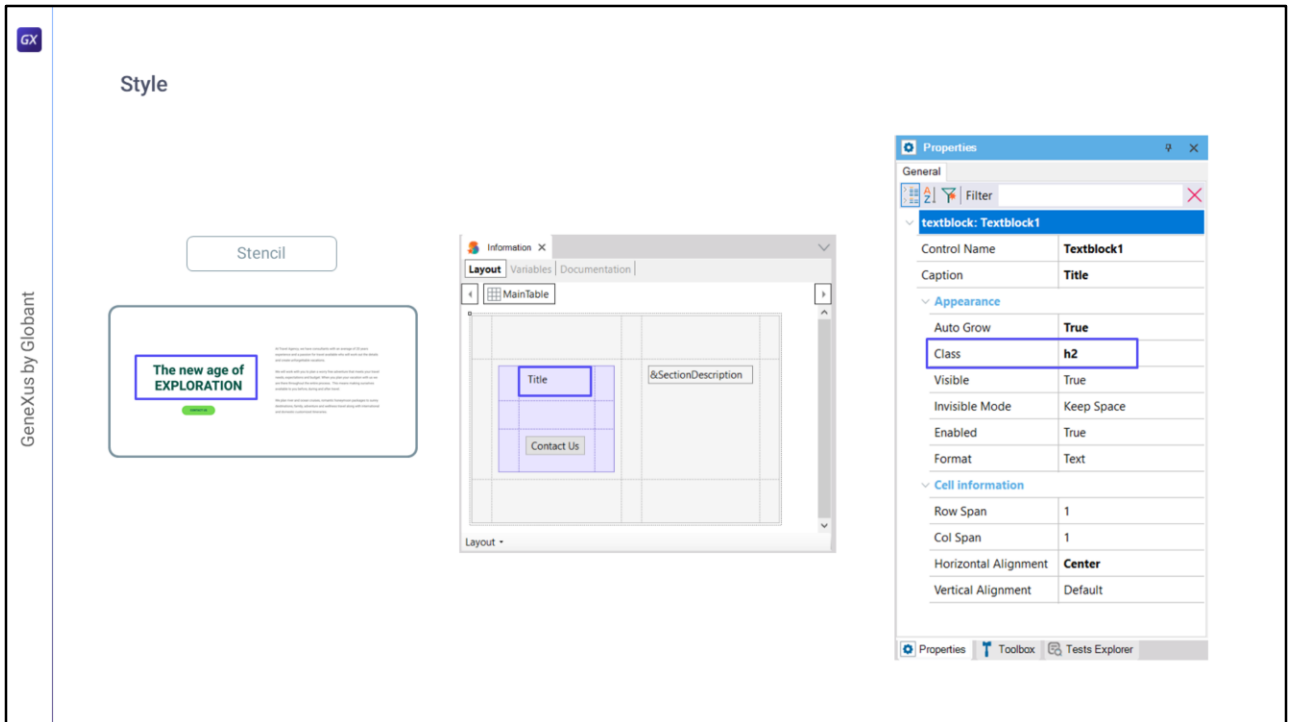


Bem, mas se dentro da estrutura temos os controles em sua organização hierárquica, para o estilo, ou seja, para que os elementos tenham uma determinada aparência, temos suas propriedades e, fundamentalmente, as de suas classes. E é aí que entra o Design System object.



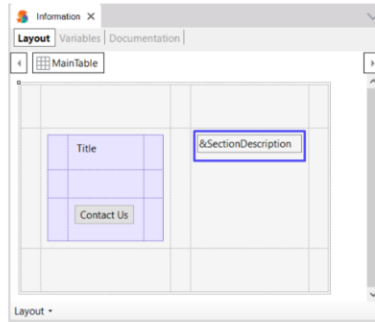
Então, por exemplo, para fazer com que este texto seja visto com esta tipografia e esta cor verde, este parágrafo seja visto com esta tipografia e esta cor, e este espaçamento e assim por diante, e que o botão seja visto desta maneira...

...no stencil onde isto está implementado, basta associar a cada controle uma classe...



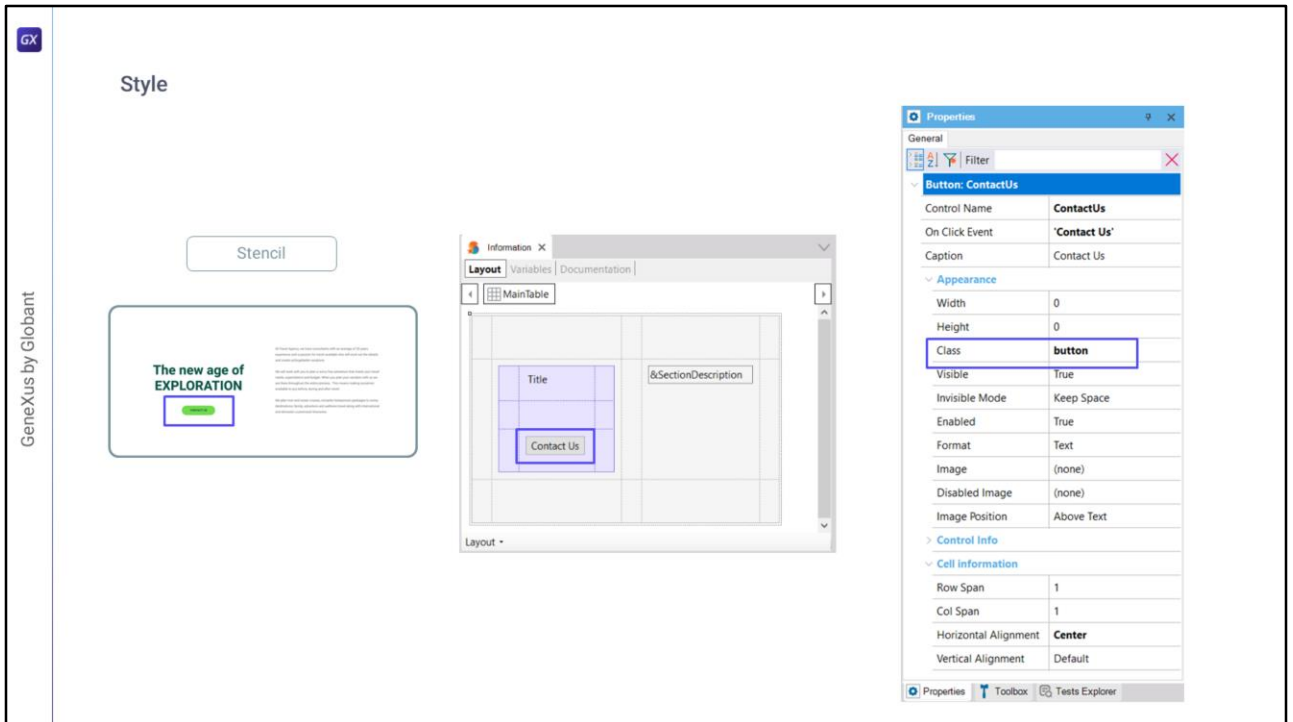
Assim, ao primeiro controle, a este text block, vou associar uma classe que vou chamar de h2 (veremos que não a inventei do nada, mas que Chechu já preparou o caminho para mim porque definiu no Figma um estilo h2 que será refletido como classe em GeneXus)...

Style

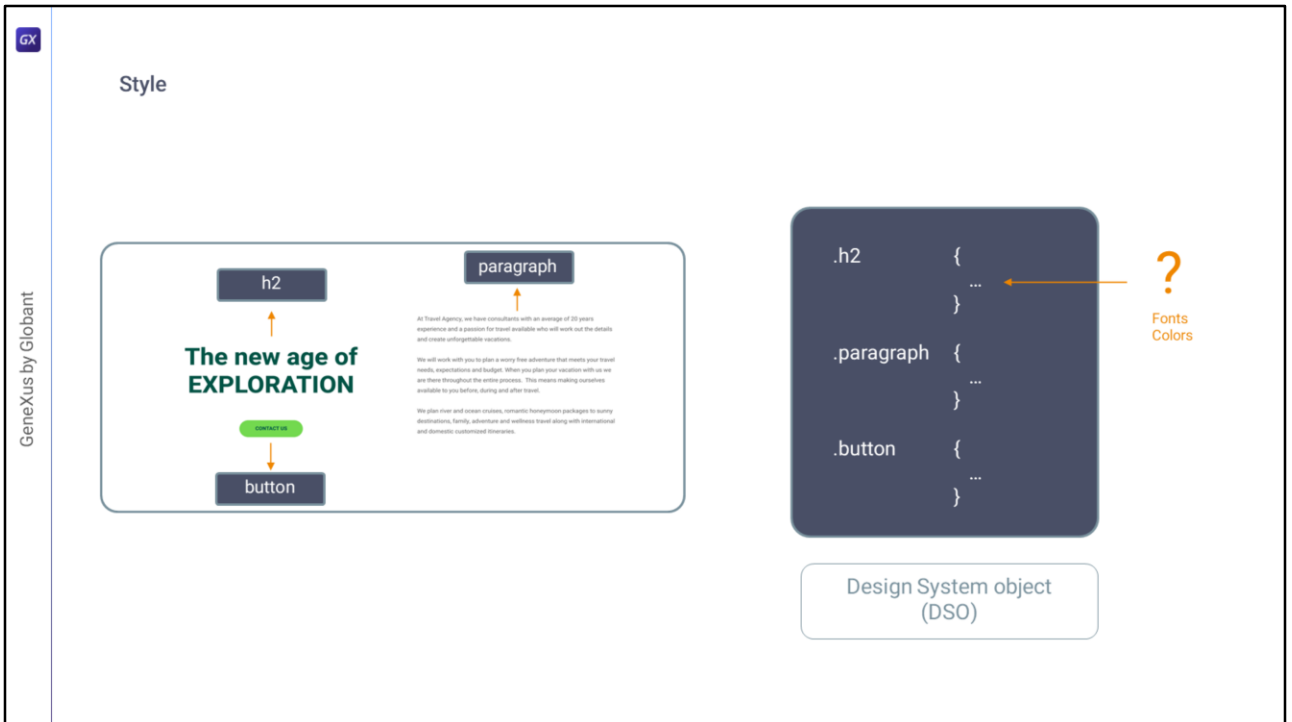


General	
Control Name	&SectionDescription
Attribute	&SectionDescription
Readonly	True
Appearance	
Label Position	None
Class	paragraph
Visible	True
Invisible Mode	Keep Space
Enabled	True
Format	Text
Invite Message	
Control Info	
Cell information	
Row Span	1
Col Span	1
Horizontal Alignment	Default
Vertical Alignment	Middle

... para esta variável vamos definir uma classe paragraph...

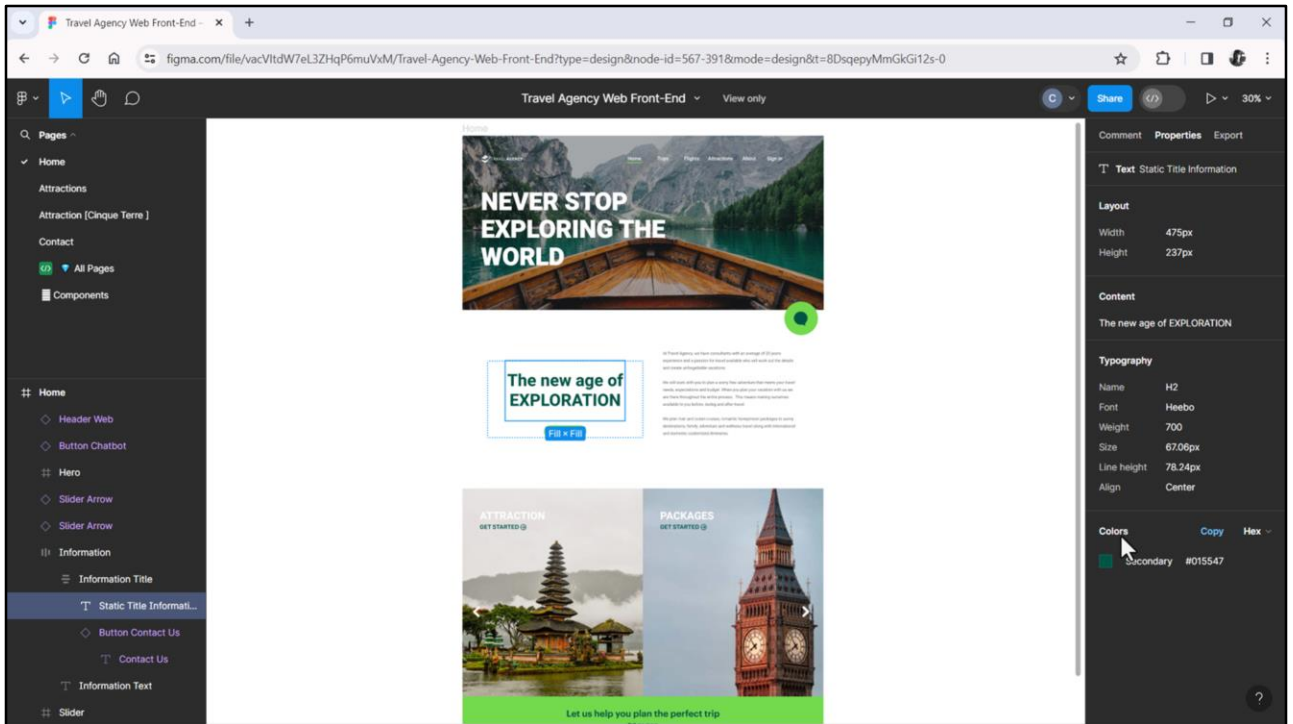


e a este botão associei como classe, a classe com este nome.



E depois então, uma vez que associei classes a cada um dos meus controles, será suficiente, no Design System object que será utilizado para dar estilo a todos os objetos, definir as características de cada uma dessas classes (como a fonte, a cor, espaçamento entre linhas, etc.).

Como já sabemos, estas são classes, em alguns casos idênticas às do CSS. E como já podemos imaginar, vamos defini-las a partir do arquivo de design em Figma que Chechu compartilhou comigo e que já nos mostrou.



Assim, se formos ao Figma em modo View... e nos posicionarmos sobre este texto, com a tecla Control, vemos que na janela de propriedades estão aparecendo todas essas propriedades que têm a ver com a tipografia do texto... propriedades às quais Chechu deu um nome, as chamou de h2, e isto será um estilo em Figma que Chechu criou, que no nosso caso será transformado na classe h2. E por outro lado temos esse estilo Secondary, com esse valor, que vai ser um estilo de cor. E essa é a cor que está assumindo, se virmos, o texto.

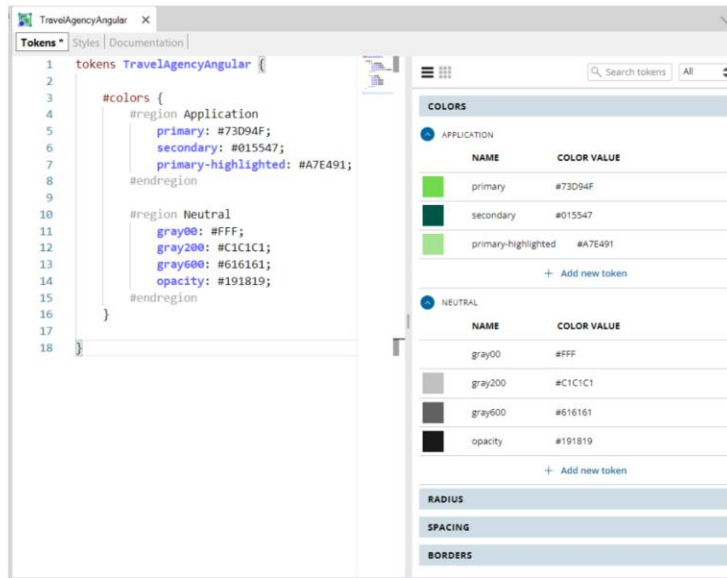
Se clicarmos com o botão direito sobre o texto / copy as / code / CSS, serão copiadas no clipboard as propriedades CSS que correspondem a esse texto. Então vou colá-las para que possamos vê-las neste Notepad... e vemos aqui o H2 com suas propriedades CSS, a da cor, a de Secondary e outras que veremos depois se vamos utilizar ou não e para que serviria...

O mesmo se nos posicionarmos sobre o parágrafo, temos o estilo que Chechu chamou de Paragraph, com essas características, essa cor e assim por diante.

A mesma coisa para o botão, posso obter a tipografia que foi chamada dessa maneira, com as características e assim sucessivamente.

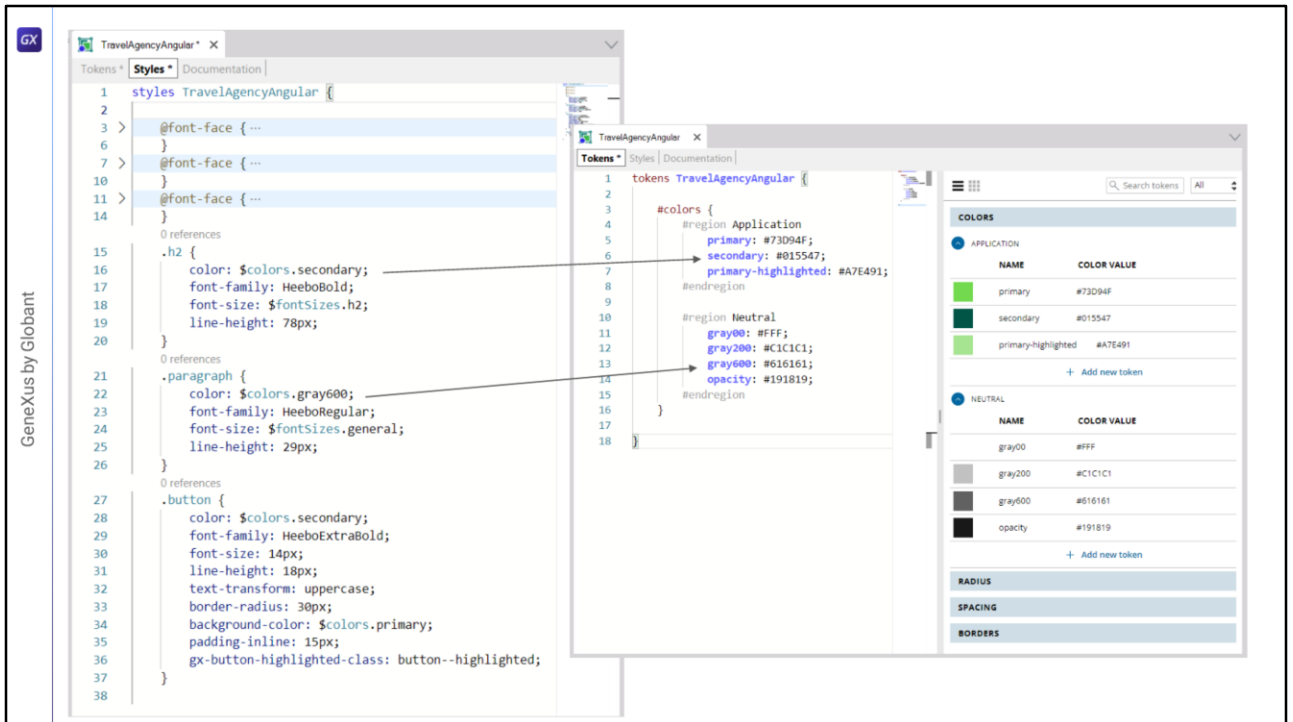
Por outro lado, por exemplo, vamos para a página All Pages, onde posicionados fora dos frames, vemos que na aba de propriedades estão aparecendo listados todos os estilos de texto que Chechu criou (aqui temos o H2 por exemplo, aqui o Paragraph, e outros, os demais), aqueles que estão criados para todas essas telas. E o mesmo vale para as cores que estão sendo utilizadas, com esses nomes que são nomes que Chechu criou em Figma e aos quais deu valores.

Bem, então com isso...

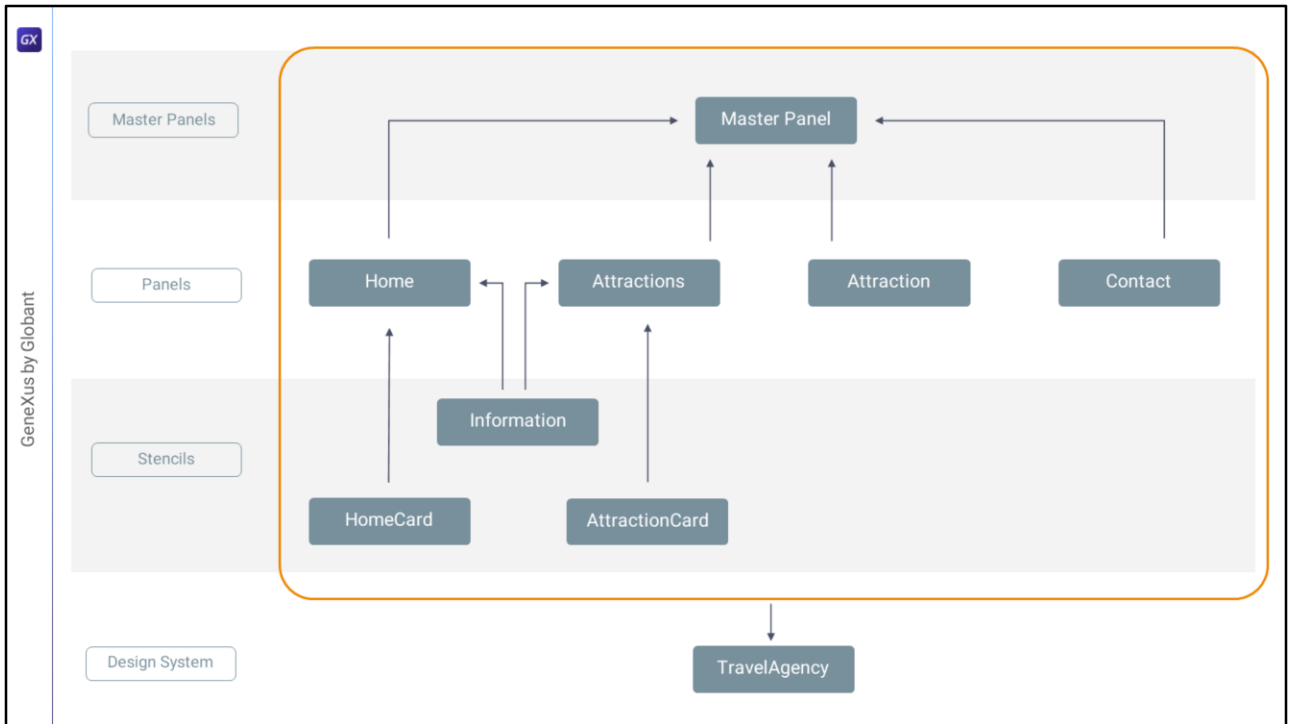


... vou ao GeneXus e já posso começar a construir meu objeto Design System.

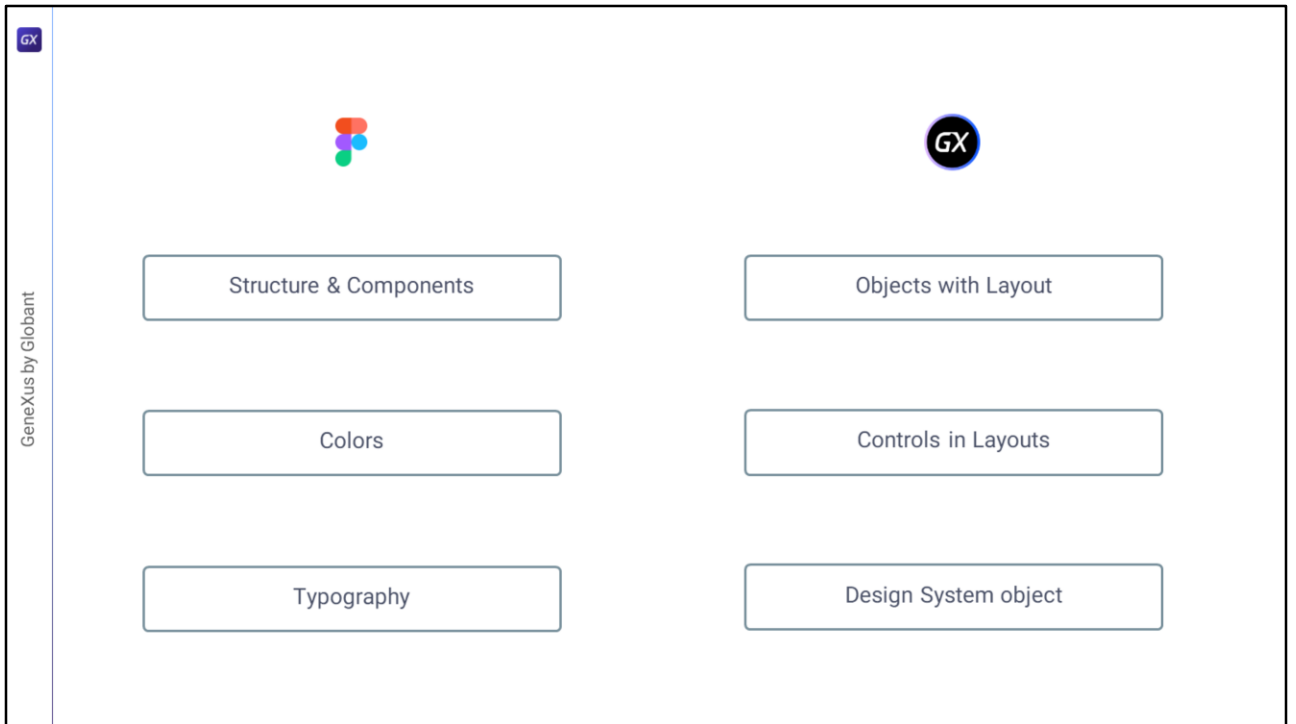
Aqui o que fiz foi, na aba de Tokens, que são as constantes, definir as constantes de cor, aquelas que acabamos de ver, que Chechu havia identificado e nomeado desta maneira, e então essas constantes de cor serão utilizadas...



...dentro da aba de estilos, dentro das classes. Então, por exemplo, para a classe h2 estou definindo a propriedade color que está utilizando o token que estou vendo lá. E assim então começamos a construir tudo que tem a ver com o estilo da aplicação, através da definição das classes.



Pois bem, então aqui vemos o esquema do que vimos até agora, separando toda a aplicação em objetos: Master Panel, Panels, Stencils e o Design System comandando o design e a estética de todos os objetos...



E com isso finalizamos uma primeira revisão de todos esses atores que vão intervir em GeneXus para refletir todas as decisões que Chechu toma em seu Design System. Então agora o que nos resta é começar a trabalhar.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com