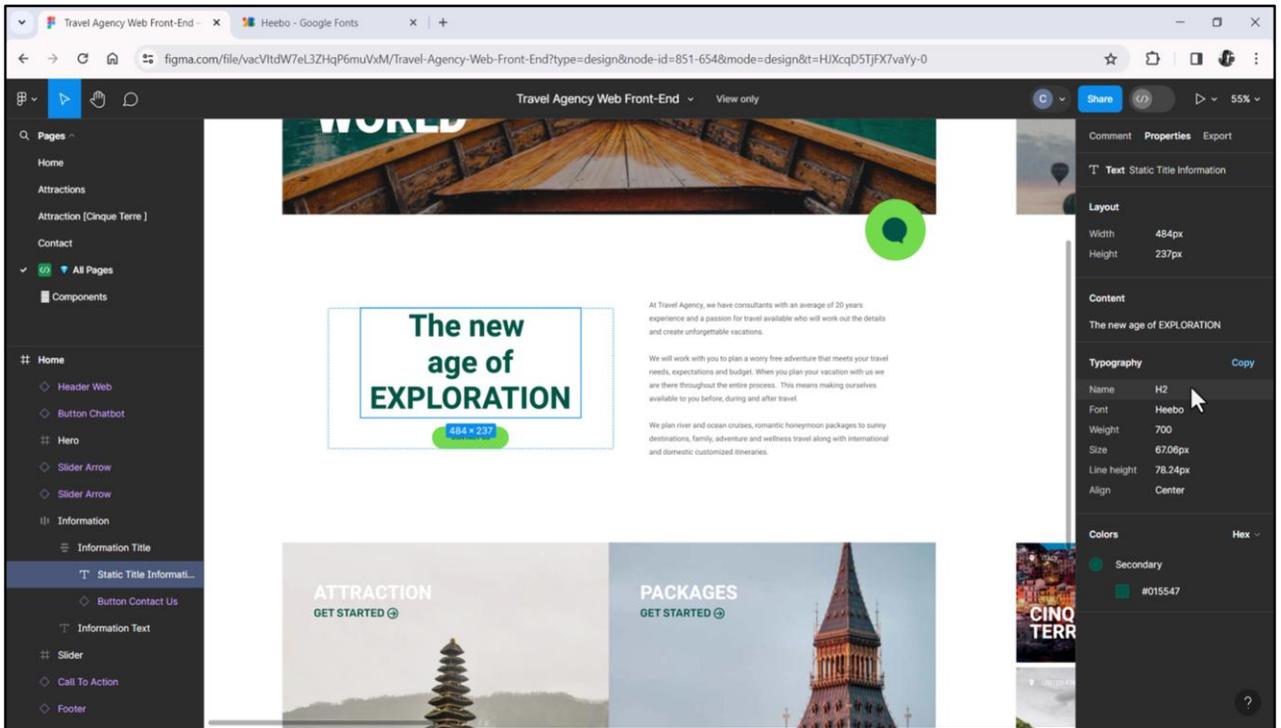


# First Layout in GeneXus. Style



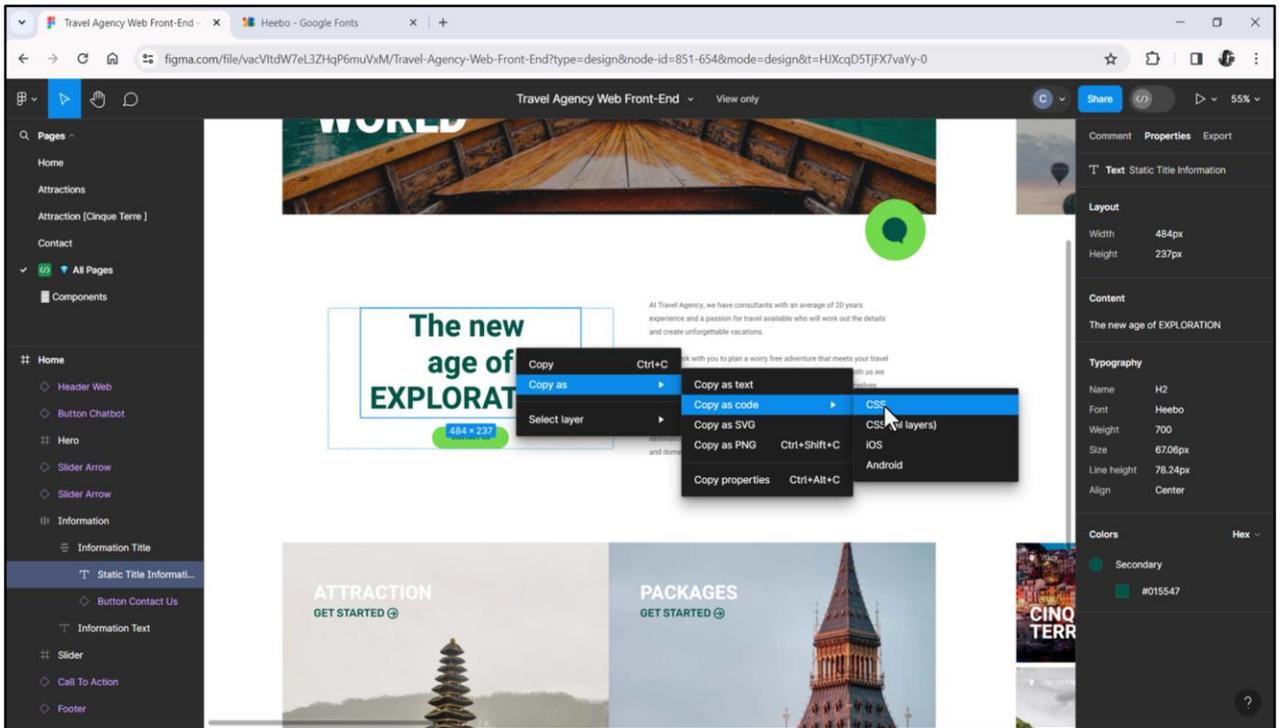
Cecilia Fernández

Agora que já temos a estrutura do layout do Stencil, passaremos a trabalhar sobre o Design System object para conseguir que esse layout do stencil pareça com o que queremos.

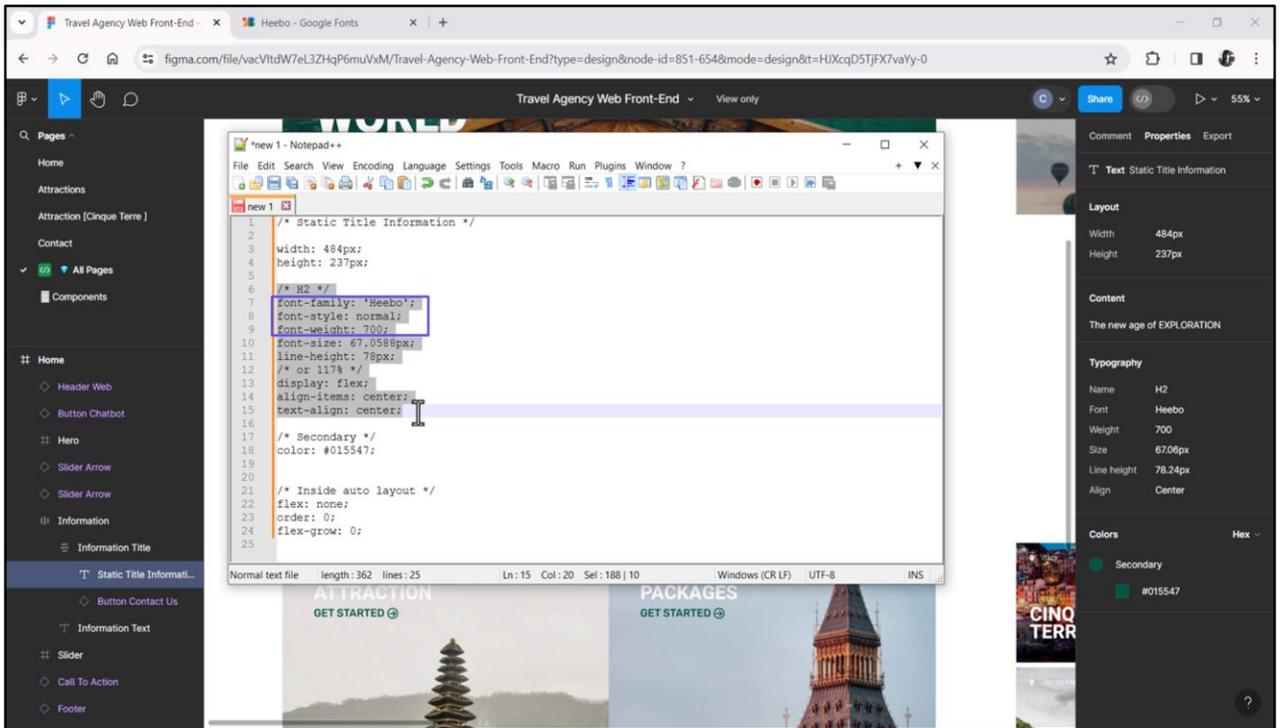


Comecemos por este texto. Se vamos inspecionar suas propriedades, aqui temos as relacionadas aos tamanhos no layout, já vimos elas nos vídeos anteriores. Aqui temos o conteúdo, e aqui nos mostra as características relacionadas com a tipografia e com as cores.

Se observarmos as características tipográficas, vemos que nossa designer lhes deu um nome, H2. Isso, lembre-se do que Chechu nos disse, significa que criou um estilo tipográfico em Figma. Com estas características: esta família de fontes, com este peso, este tamanho, esta altura de linha e este alinhamento.



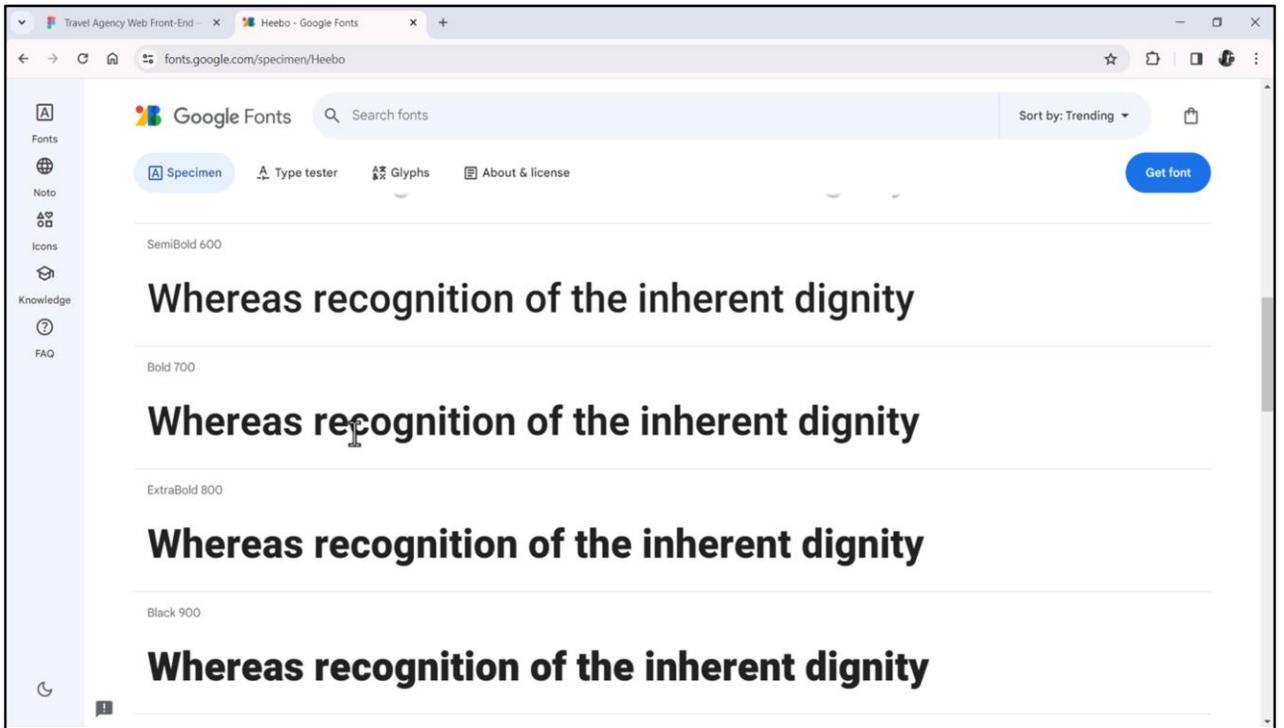
Vamos querer ter essas propriedades em formato CSS, pois é o que utilizaremos em GeneXus em nosso Design System Object. As copiamos...



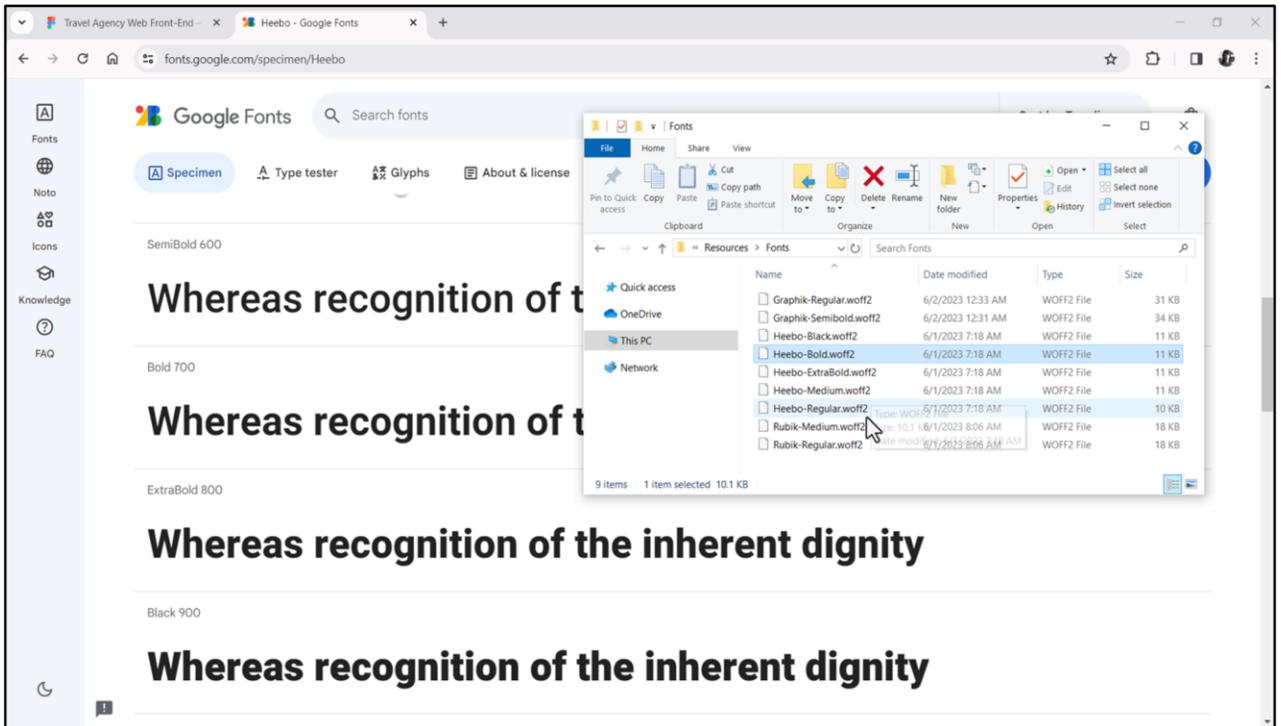
...e vamos colá-las para vê-las em um arquivo de texto.

Nos interessam, fundamentalmente, aquelas que têm a ver com o estilo H2.

Aqui vemos estas três primeiras que têm a ver com a família de fontes. Heebo é uma fonte web que melhora a legibilidade em telas digitais, mas não é uma fonte padrão dos navegadores, então teremos que incluí-la em nosso CSS.



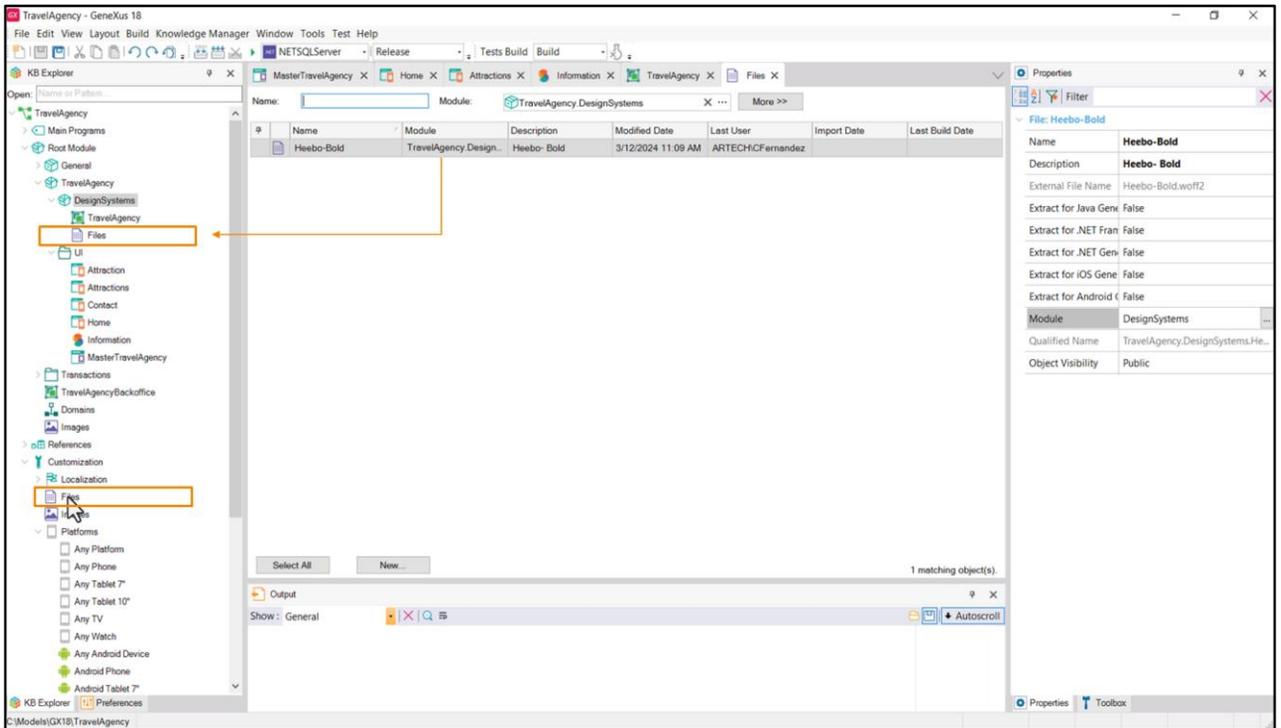
Você pode pesquisar no Google e baixá-la. Aqui a vê com seus diferentes pesos. A que precisamos para o estilo H2 é a de peso 700, ou seja, esta, que é a bold.



Eu baixei nesta pasta a Regular de peso 400; a Medium de peso 500; a Bold, como sabemos, de peso 700; a Extra Bold de peso 800; e a Black de peso 900. Além disso, baixei essas outras fontes que logo utilizaremos.

E as baixei em formato woff2, que é um formato moderno, suportado por Angular, e é o que ocupa menos espaço. A desvantagem é que não é suportado por Android ou iOS.

Bem, então a primeira coisa que faremos será inserir esses arquivos de fontes na KB.



Podemos inseri-los um por um, manualmente... dando-lhes um nome e aqui escolhendo o arquivo de nossa pasta...  
Vemos que ao inseri-lo dentro do módulo DesignSystems ele aparece separado dos demais arquivos da KB...

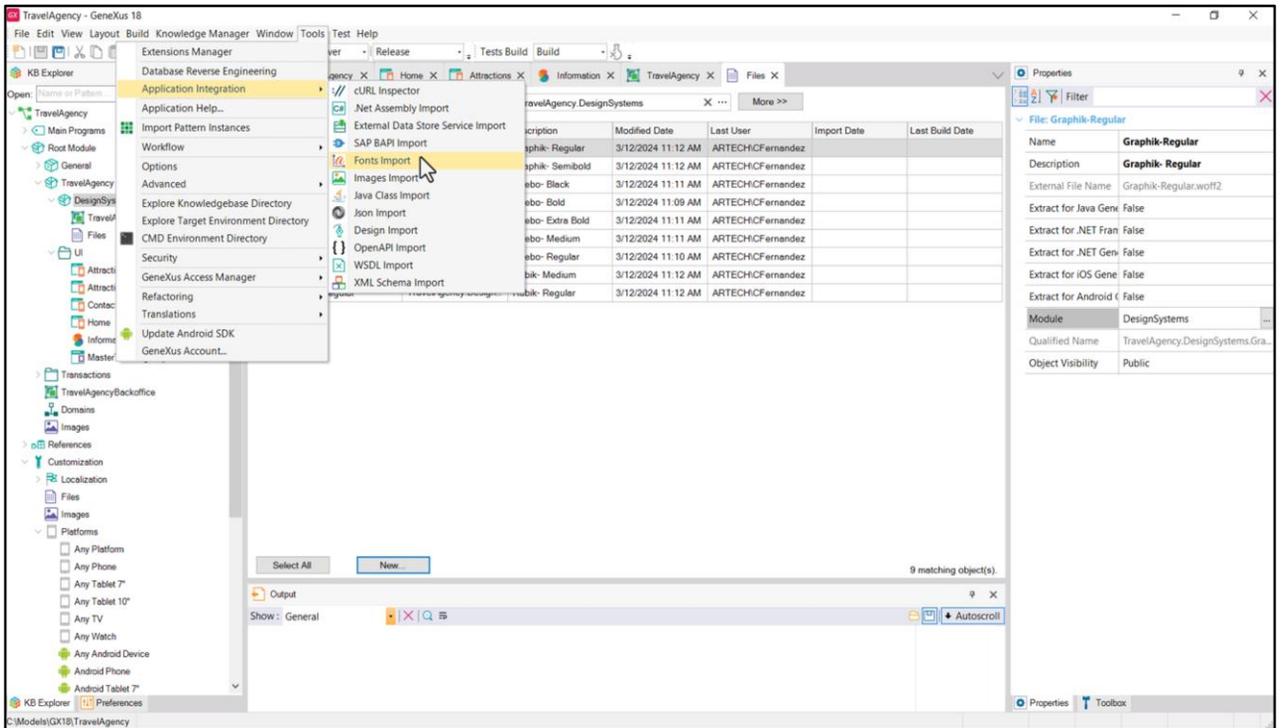
The screenshot displays the 'TravelAgency - Genius 18' application. The main window features a menu bar (File, Edit, View, Layout, Build, Knowledge Manager, Window, Tools, Test, Help) and a toolbar. On the left, the 'KB Explorer' pane shows a tree view of the project structure, including 'TravelAgency', 'Main Programs', 'Root Module', 'General', 'TravelAgency', 'DesignSystems', 'TravelAgency', 'Files', 'UI', 'Attraction', 'Attractions', 'Contact', 'Home', 'Information', 'MasterTravelAgency', 'Transactions', 'TravelAgencyBackoffice', 'Domains', 'Images', 'References', 'Customization', 'Localization', 'Files', 'Images', and 'Platforms'. The central area contains a table with columns: Name, Module, Description, Modified Date, Last User, Import Date, and Last Build Date. The table lists various font styles like Graphik-Regular, Graphik-Semibold, Heebo-Block, Heebo-Bold, Heebo-ExtraBold, Heebo-Medium, Heebo-Regular, Rubik-Medium, and Rubik-Regular. The 'Heebo-Block' row is selected. Below the table are 'Select All' and 'New' buttons, and a status bar indicating '9 matching object(s)'. On the right, the 'Properties' pane shows details for the selected 'Graphik-Regular' font, including its name, description, external file name, and extraction settings for different platforms. The bottom status bar shows 'C:\Models\GX18\TravelAgency'.

Name	Module	Description	Modified Date	Last User	Import Date	Last Build Date
Graphik-Regular	TravelAgency Design...	Graphik- Regular	3/12/2024 11:12 AM	ARTECHCFernandez		
Graphik-Semibold	TravelAgency Design...	Graphik- Semibold	3/12/2024 11:12 AM	ARTECHCFernandez		
Heebo-Block	TravelAgency Design...	Heebo- Block	3/12/2024 11:11 AM	ARTECHCFernandez		
Heebo-Bold	TravelAgency Design...	Heebo- Bold	3/12/2024 11:09 AM	ARTECHCFernandez		
Heebo-ExtraBold	TravelAgency Design...	Heebo- Extra Bold	3/12/2024 11:11 AM	ARTECHCFernandez		
Heebo-Medium	TravelAgency Design...	Heebo- Medium	3/12/2024 11:11 AM	ARTECHCFernandez		
Heebo-Regular	TravelAgency Design...	Heebo- Regular	3/12/2024 11:10 AM	ARTECHCFernandez		
Rubik-Medium	TravelAgency Design...	Rubik- Medium	3/12/2024 11:12 AM	ARTECHCFernandez		
Rubik-Regular	TravelAgency Design...	Rubik- Regular	3/12/2024 11:12 AM	ARTECHCFernandez		

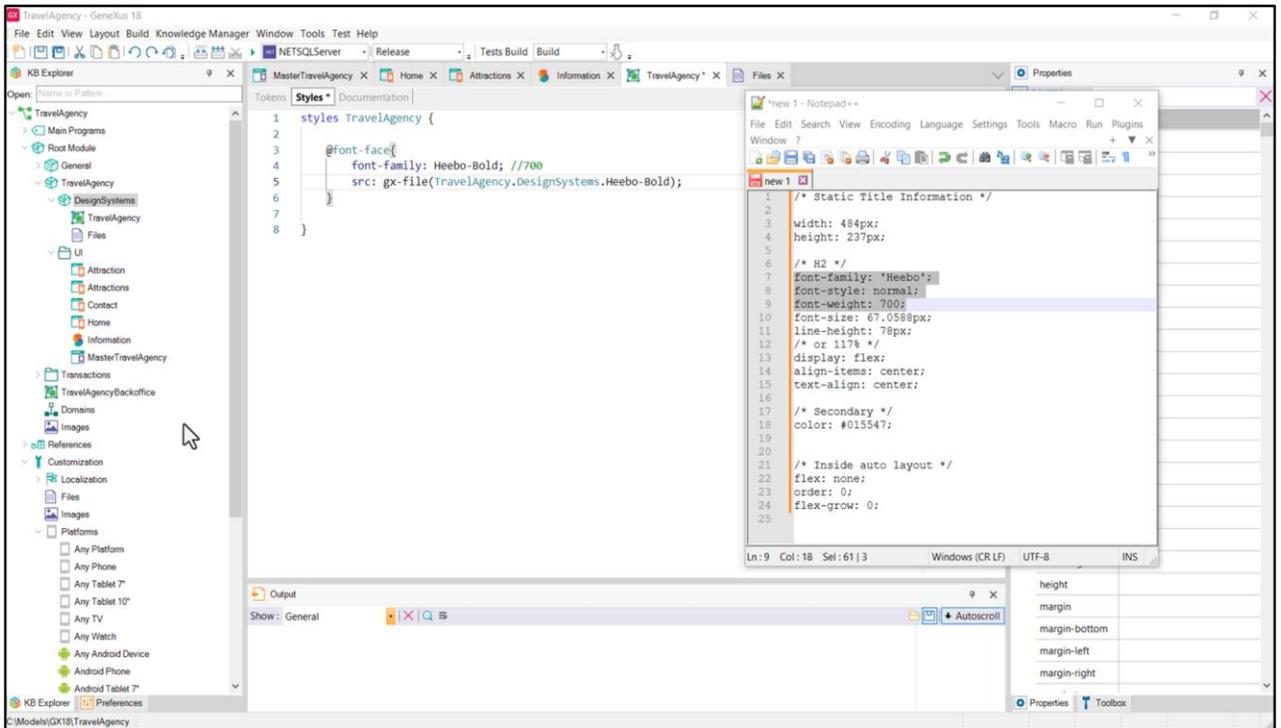
Properties for File: Graphik-Regular

Name	Graphik-Regular
Description	Graphik- Regular
External File Name	Graphik-Regular.woff2
Extract for Java Gen	False
Extract for .NET Fram	False
Extract for .NET Gen	False
Extract for iOS Gen	False
Extract for Android	False
Module	DesignSystems
Qualified Name	TravelAgency.DesignSystems.Gra...
Object Visibility	Public

E então podemos seguir assim, inserindo cada um separadamente... até completar todos...



...Ou a outra opção, muito mais simples, é inseri-los todos juntos desta maneira... aí indicamos o folder... e pressionando Ok seriam inseridos todos em uma única operação (e se ficam no módulo Root logo mudamos o módulo para cada um através da propriedade).

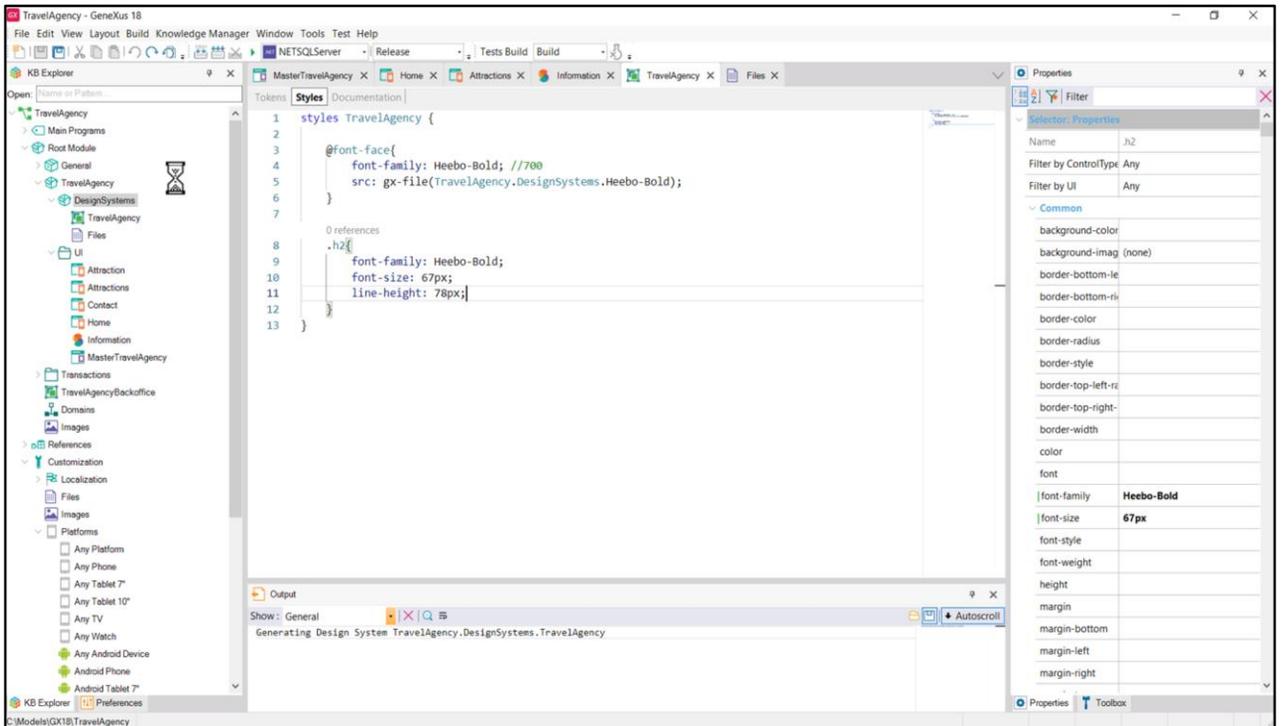


Bem, já temos os arquivos de fontes na KB. Começamos adicionando aquele que corresponde à fonte Heebo-Bold ao nosso Design System Object.

A primeira coisa será declarar o conjunto de estilos.

E agora sim, vamos escrever a regra font-face para adicionar justamente a fonte.

Com a propriedade font-family damos-lhe o nome com o qual nos referiremos a esta fonte em todo o DSO (vou escrever como comentário o peso correspondente, que é 700); e com esta outra propriedade indicamos o source, ou seja, onde está localizada. Para isso utilizamos a função de GeneXus “gx-file”... seu parâmetro é o nome do arquivo da fonte, qualificado com o módulo onde está localizada.

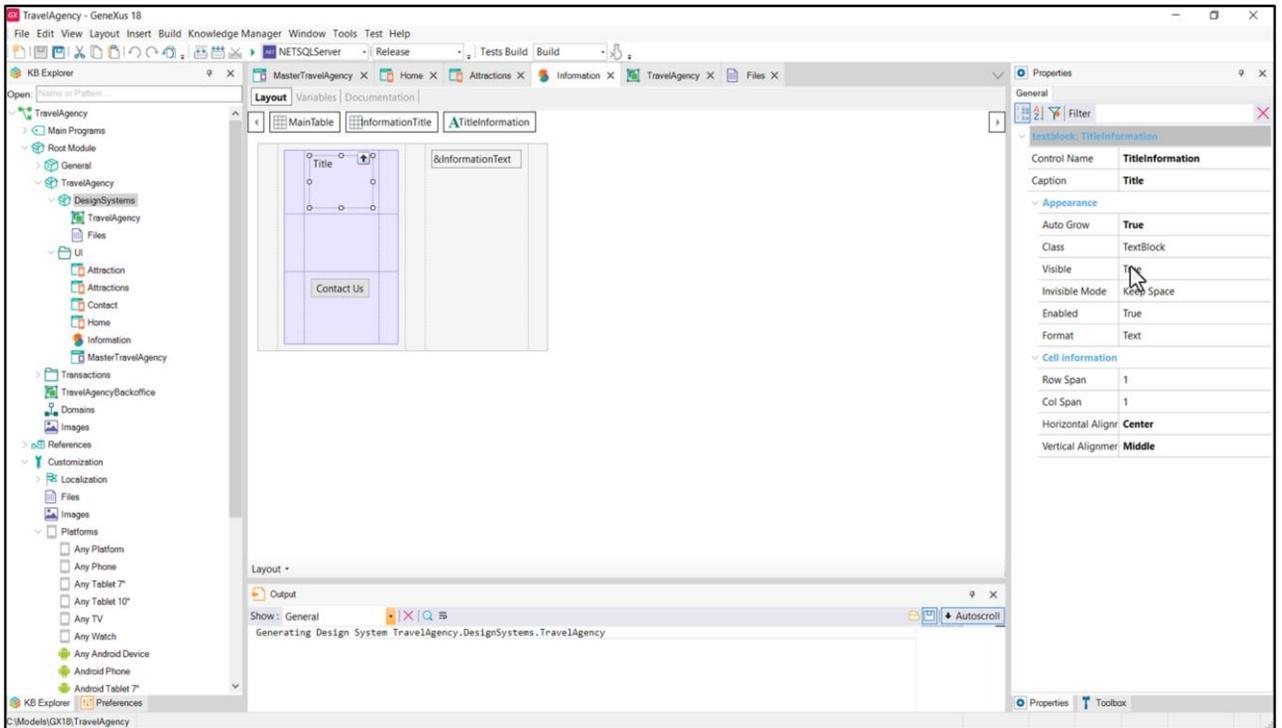


Já estamos em condições de especificar nossa primeira classe, a que chamaremos da mesma forma que nossa designer chamou o estilo tipográfico: h2.

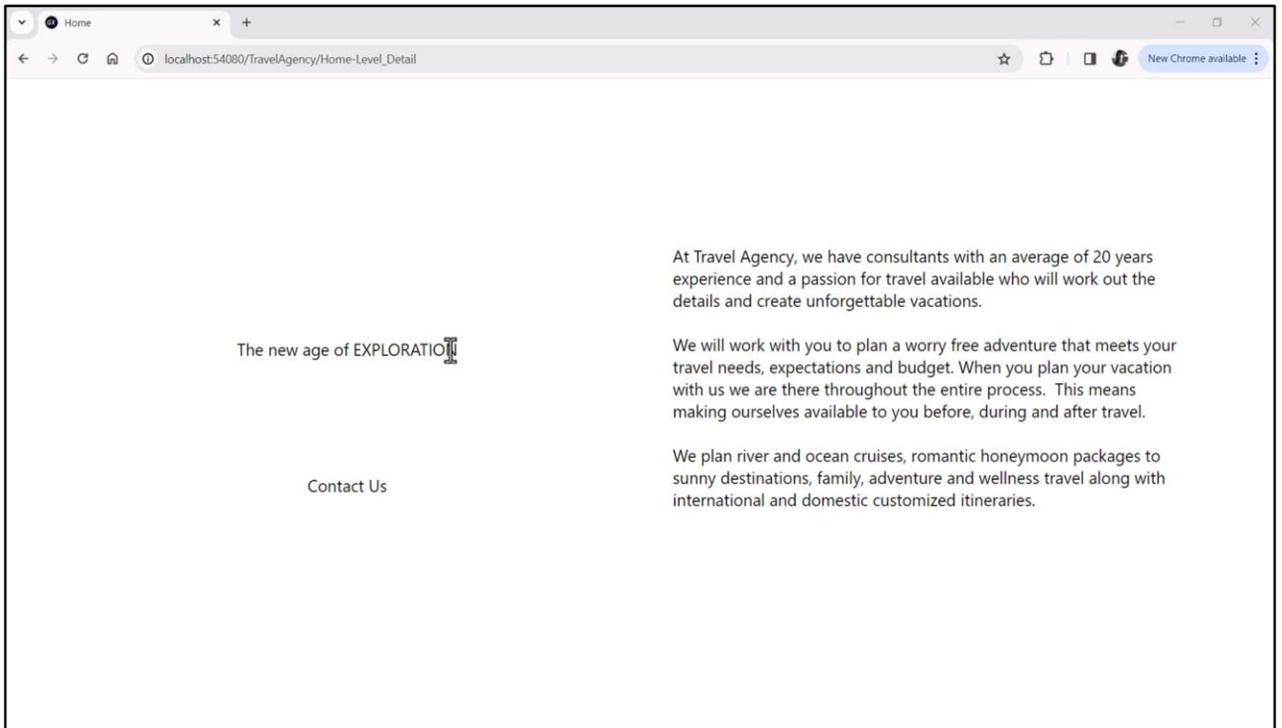
Lembremos que as classes de um DSO são extremamente parecidas com as de CSS, já que para o caso Web serão transformadas efetivamente em CSS. É por isso que para selecioná-las devemos primeiro escrever ponto e depois o nome da classe.

As primeiras três propriedades se transformarão em uma, a font-family, que aqui se referirá a esse nome, justamente. Não precisamos variar nem o estilo, nem o peso da fonte, porque estamos usando uma com exatamente esse estilo e peso. Sempre usaremos a fonte exata, para evitar deformações e é por isso que integramos um arquivo de fonte para cada peso.

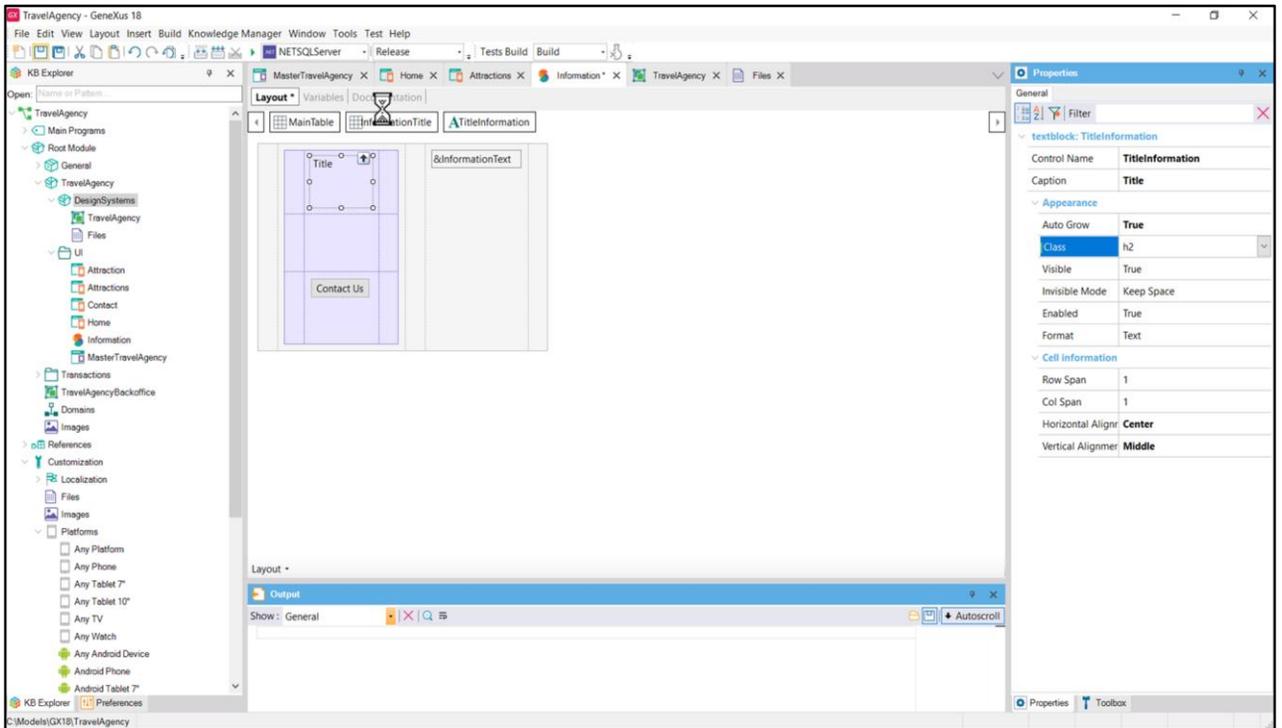
Em seguida utilizaremos como estão essas outras duas propriedades, removendo os decimais.



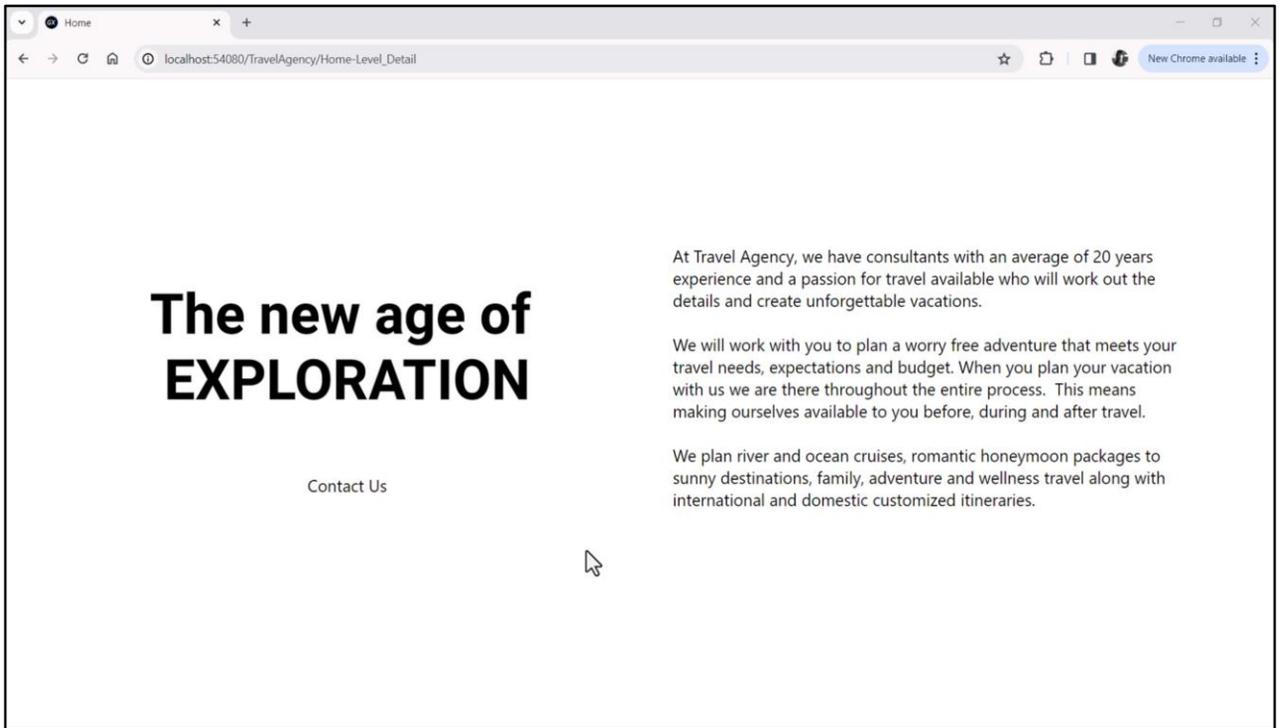
Gravamos e antes de testar, vamos ver qual classe tem associada esse Textblock. A classe deste nome.



Antes de mudá-la para a nova, vamos executar a aplicação para que possamos comparar.

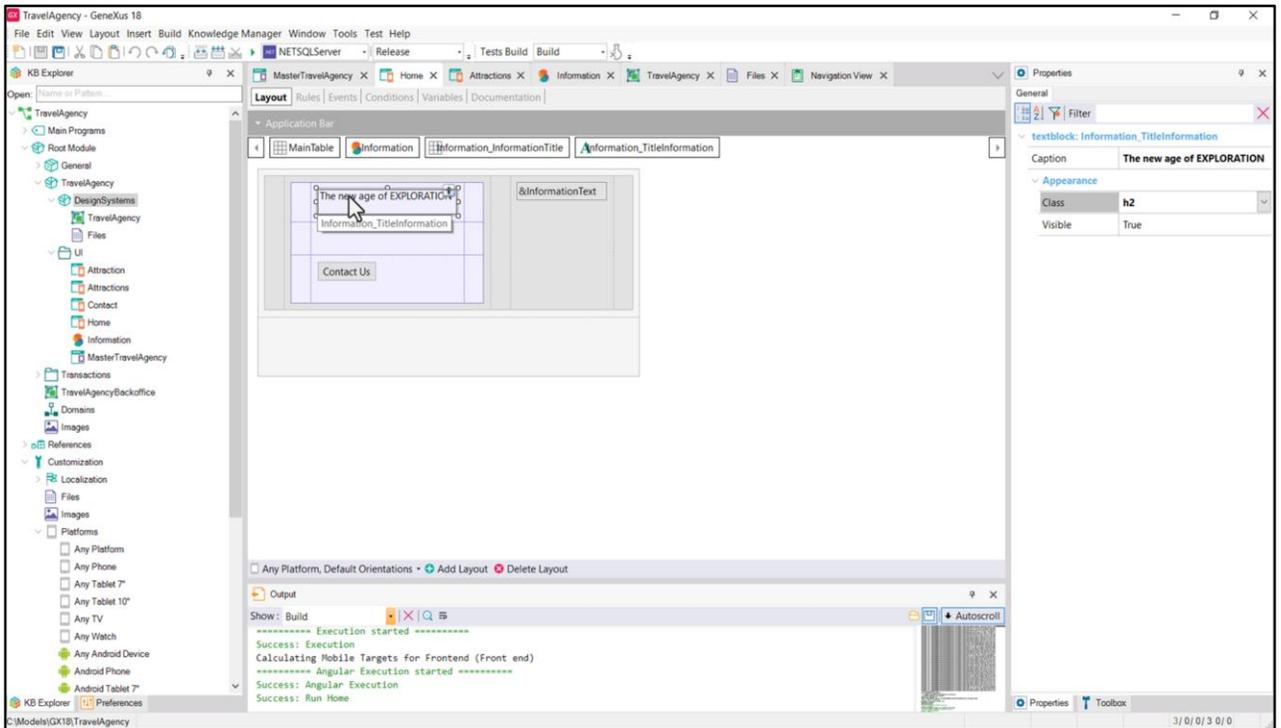


Agora sim, vamos para o stencil... e vamos mudar a classe do TextBlock para a nova, h2. Gravamos. E executamos, Run.



E aqui vemos claramente.

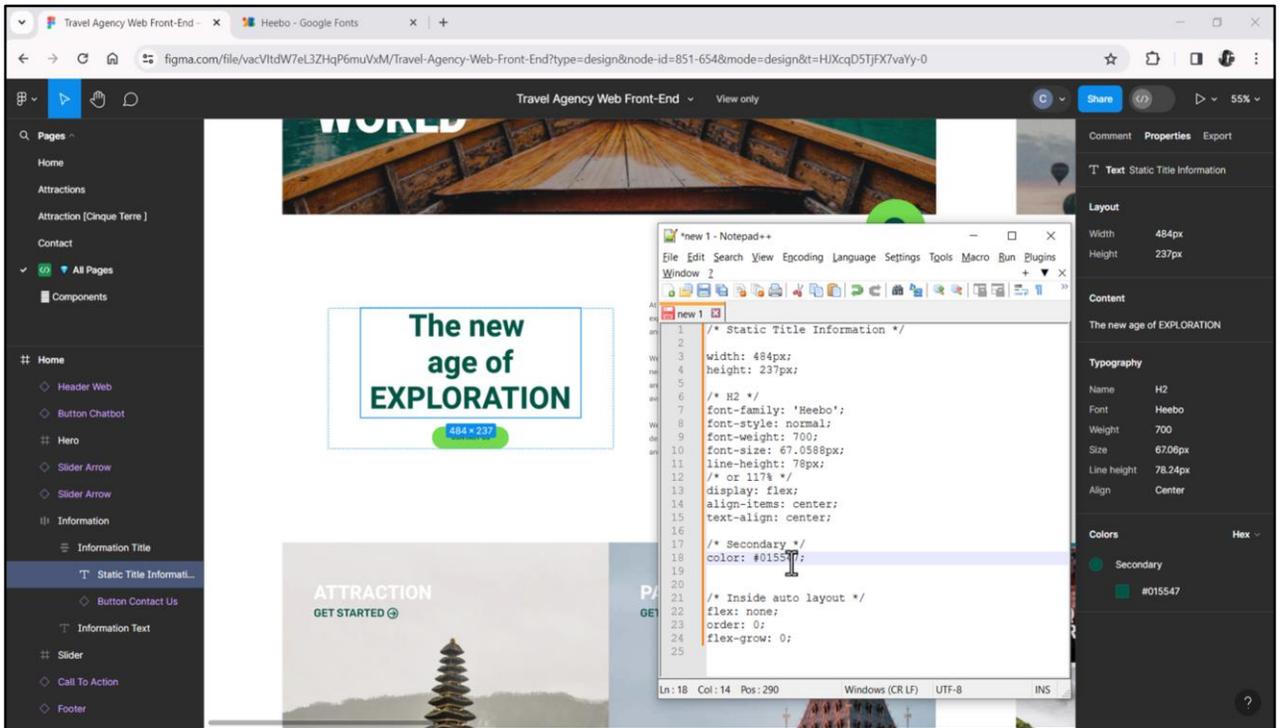
(Se vocês estão prototipando na nuvem, o que não é tão recomendado durante o tempo de desenvolvimento, e não veem a alteração em sua execução, voltem a carregar limpando o cache –no Windows com Ctrl+F5).



Algumas observações antes de continuar:

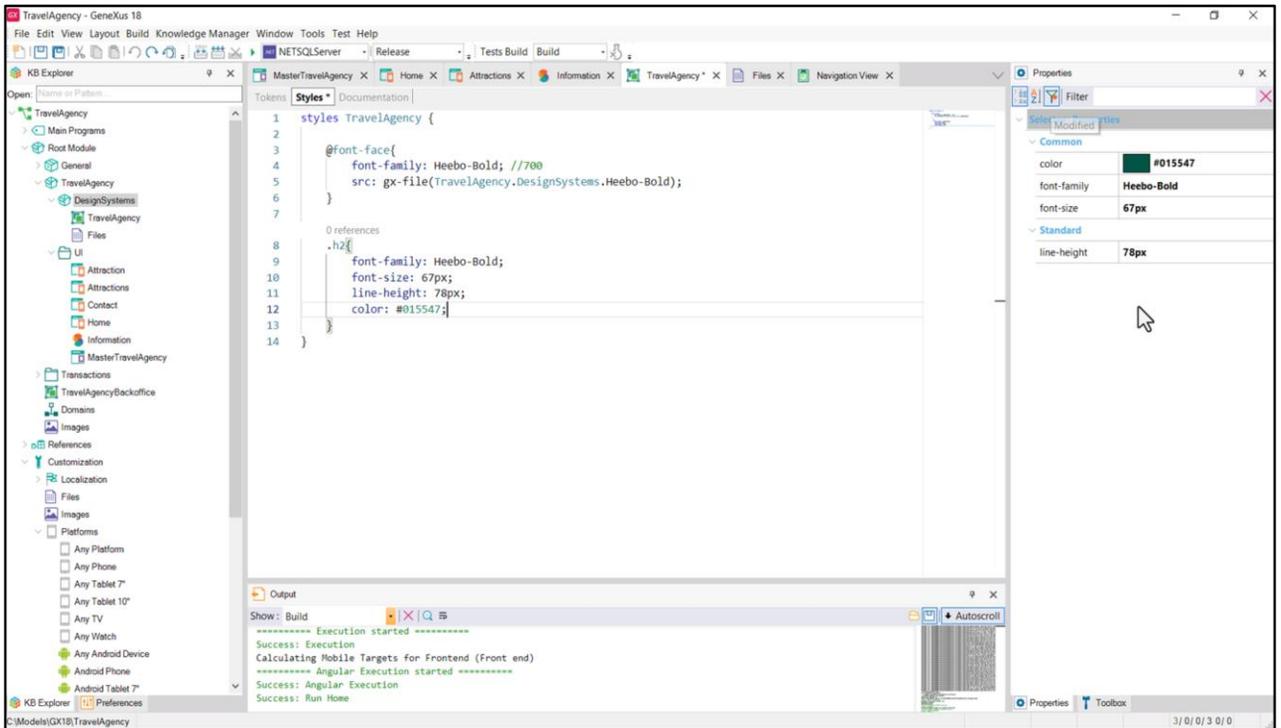
O textblock do Stencil tinha, antes dessa mudança, um nome de classe (TextBlock justamente) que não está especificada no DSO, e isso não causou nenhum problema. Pegou os defaults do navegador.

Por outro lado, tendo mudado a classe, automaticamente essa mudança foi efetuada nos dois painéis em que esse stencil está instanciado... Por isso o vemos aqui... e também o veremos neste outro caso.



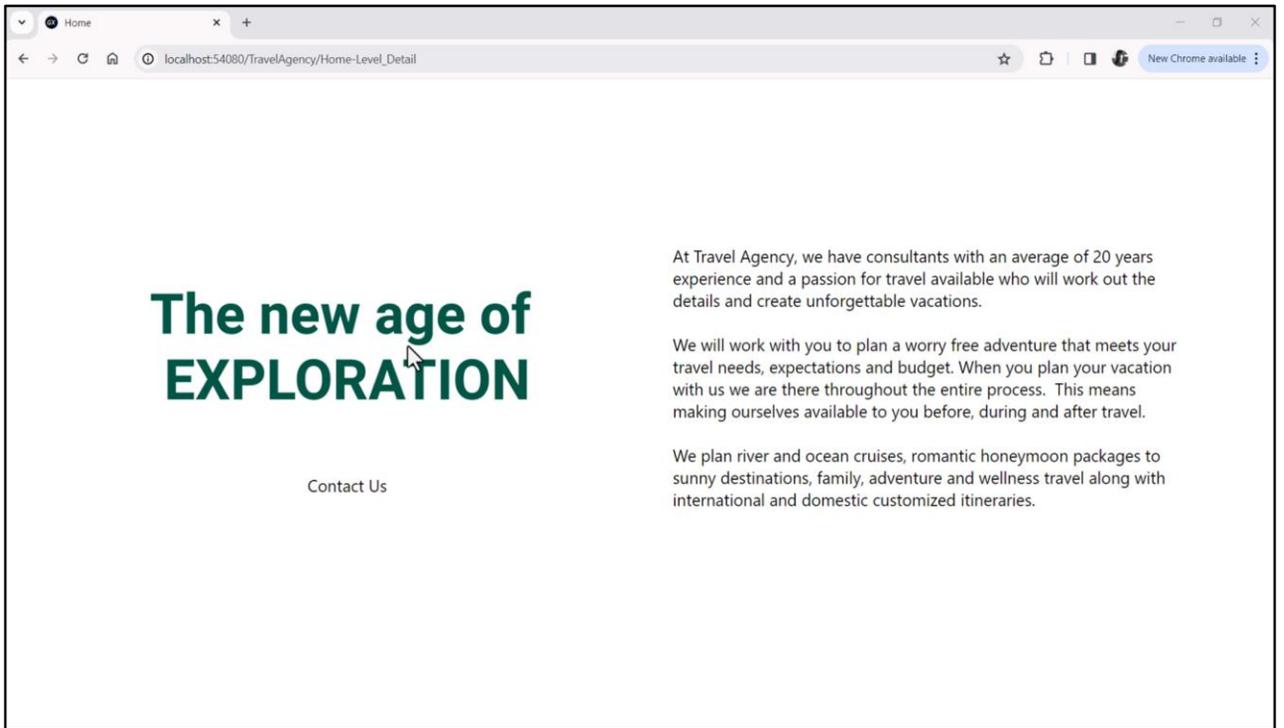
Bom, agora, o que acontece com a cor do texto? Vemos que será esse tipo de verde, o que tem esse valor hexadecimal, o que Chechu tinha denominado Secondary, criando um estilo com esse nome, um estilo de cor. Se virmos as propriedades CSS que havíamos copiado, temos justamente assim indicada a de cor, com o valor hexadecimal.

Poderíamos sem pensar muito, copiar esta propriedade CSS...

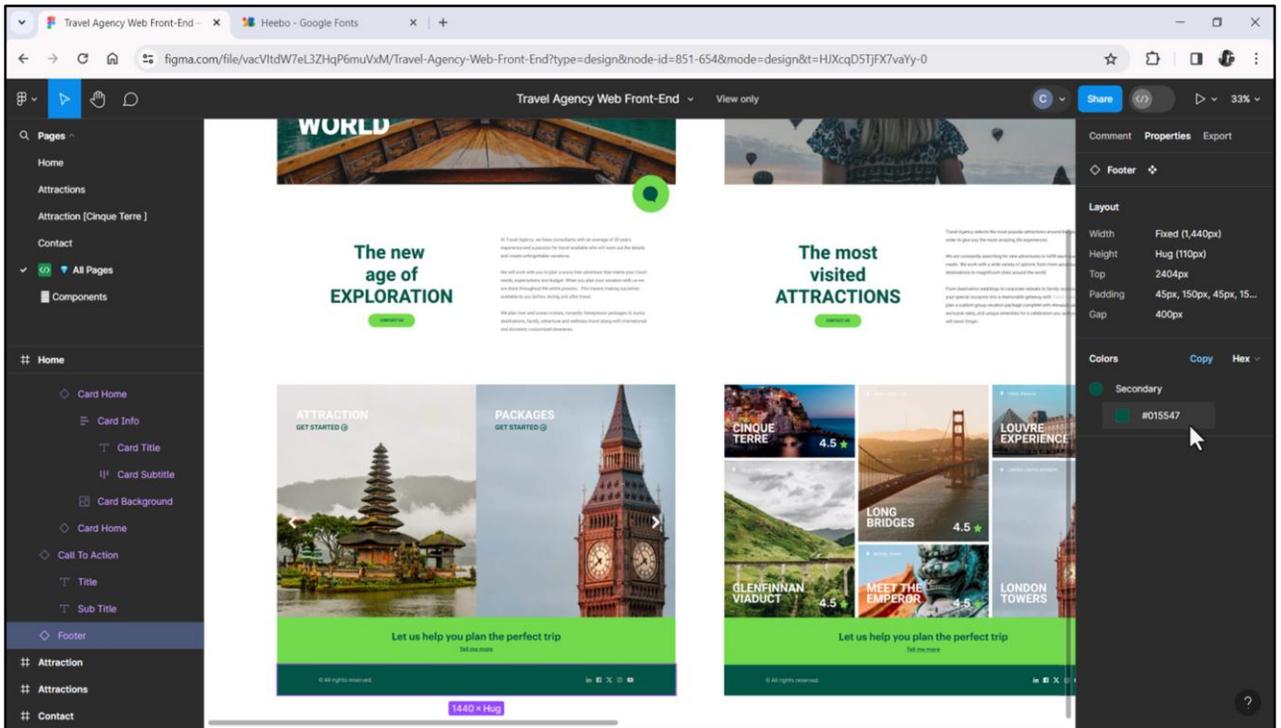


... e vir ao nosso DSO e colá-la. Vemos como na janela de propriedades mostra todas as propriedades possíveis para escrever dentro de uma classe e aqui estão sendo mostradas aquelas que utilizamos. Podemos filtrar por aquelas que modificamos para mantê-las à vista.

Vamos gravar e testar isso em execução.

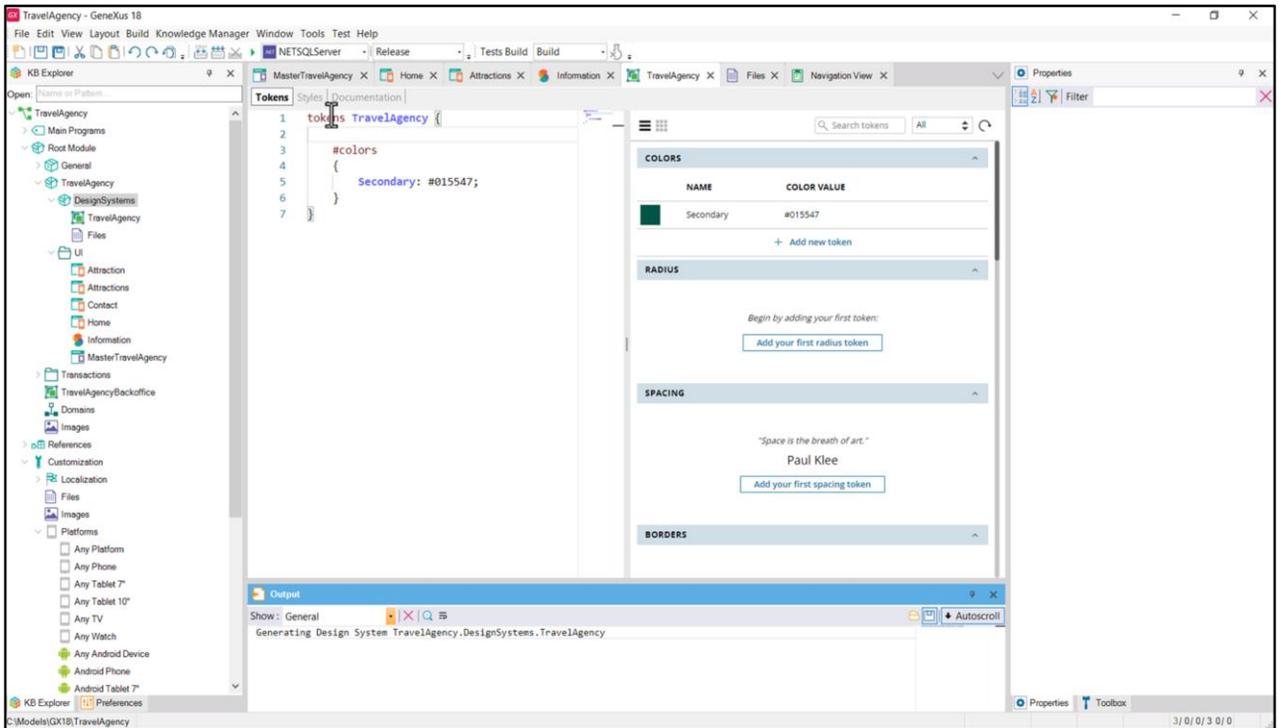


E aí o vemos, como esperávamos.



Agora, essa exata cor não será usada somente aqui para esse texto, mas para esse outro, para esse fundo, para esse outro texto, para esse outro... ou seja, ela será repetida em muitos lugares.

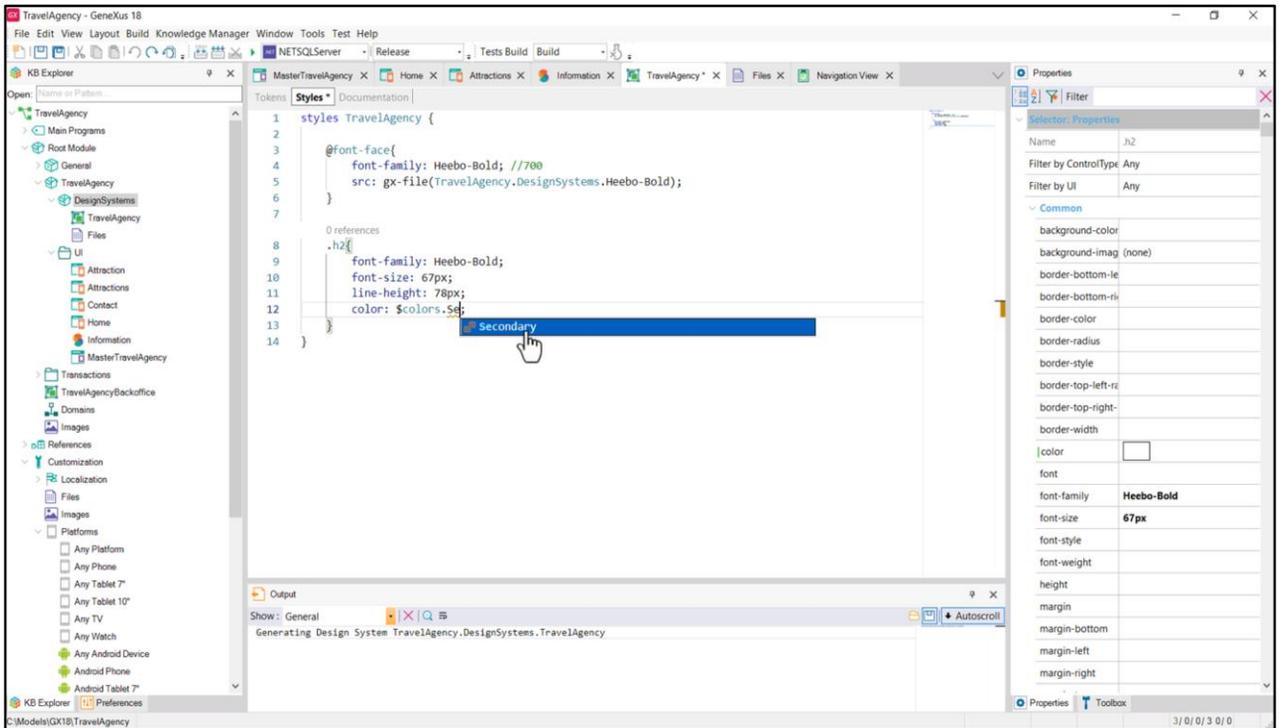
Acontece que esse valor responde a um conceito semântico, a uma cor semântica, é a cor secundária da aplicação, sendo a primária esse outro verde. É por isso que Chechu a definiu como um estilo de cor e o que nós faremos será defini-lo como um token em nosso DSO.



Será um token de cor... que chamaremos de Secondary... e seu valor será nem mais nem menos que esse aqui.

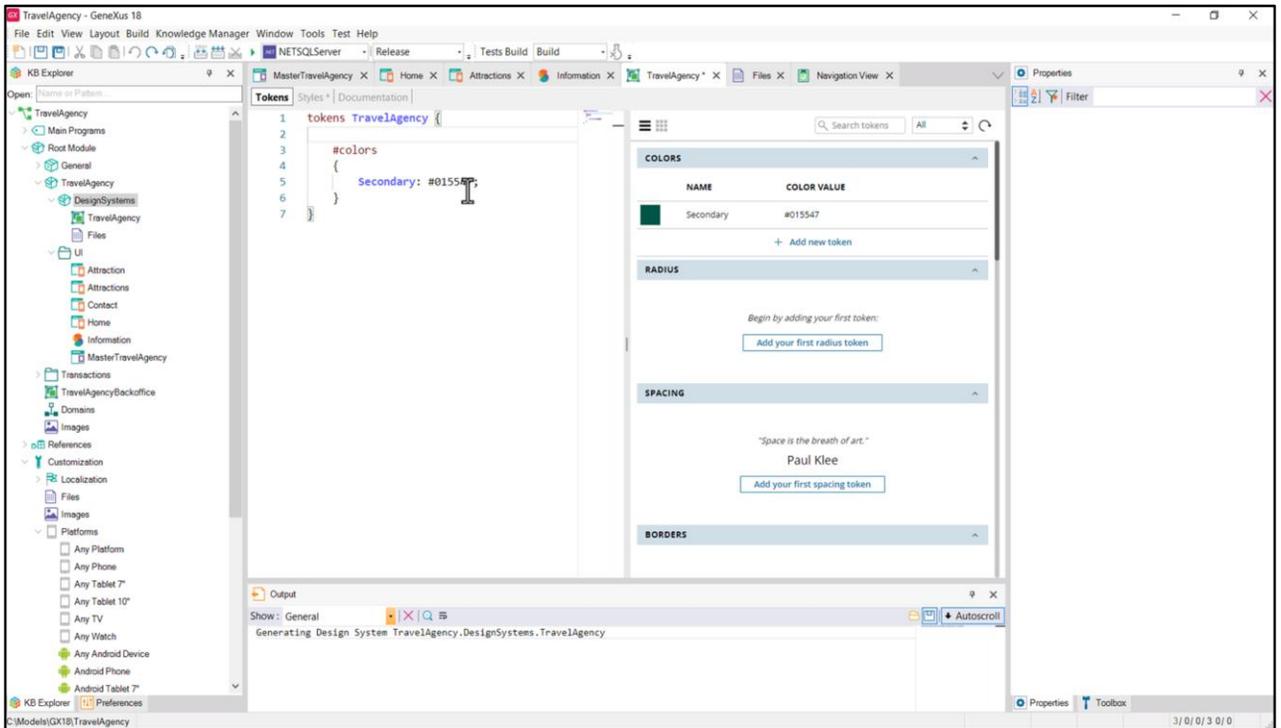
O editor de tokens é duplo, lembre-se. Podemos editá-lo diretamente como texto ou podemos editá-lo a partir do editor gráfico.

E agora vamos gravar...



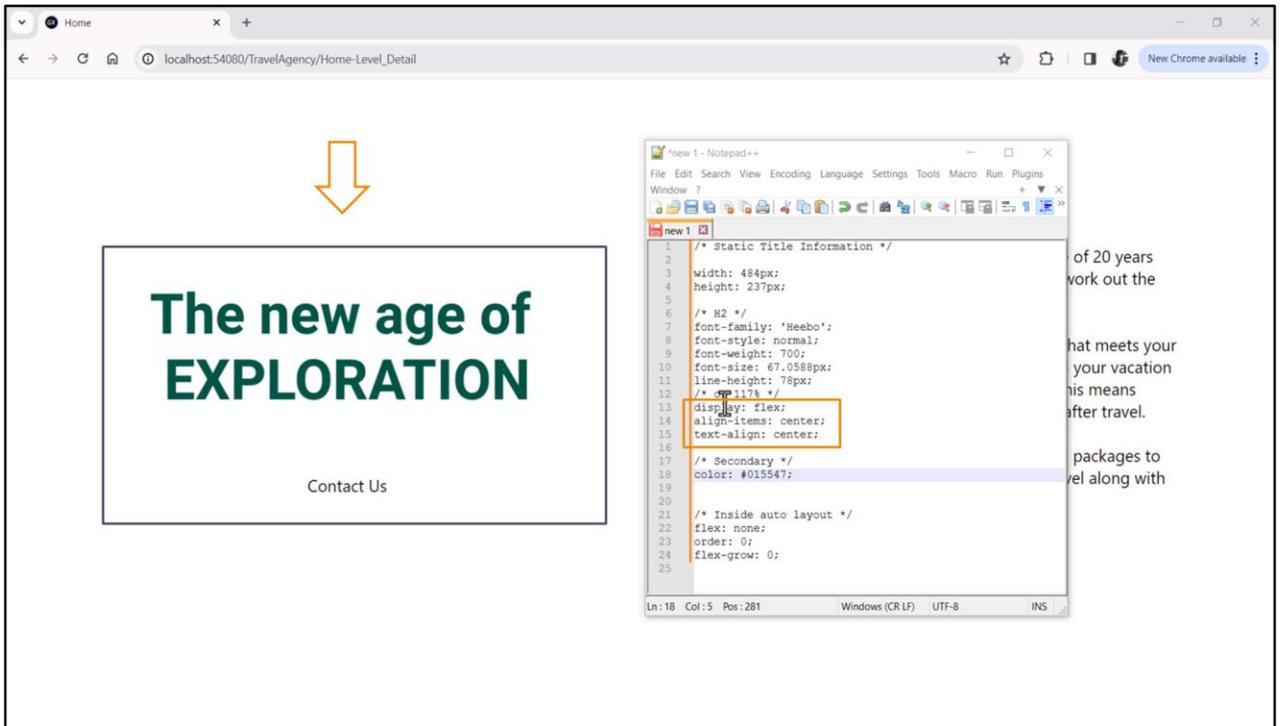
...e vamos para a aba de estilos e modificamos o valor pelo Token. Com o sinal de cifrão estamos dizendo que estamos querendo referenciar aqui um token, um token de cor... (aí está, tem que ficar assim), e ao colocar o ponto vemos que quando começamos a digitar as primeiras letras do token ele já me oferece aqueles que começam com essas duas letras.

Bom, desta maneira, então, estamos dizendo que a propriedade de cor obtenha seu valor a partir do token de cor com este nome, Secondary.



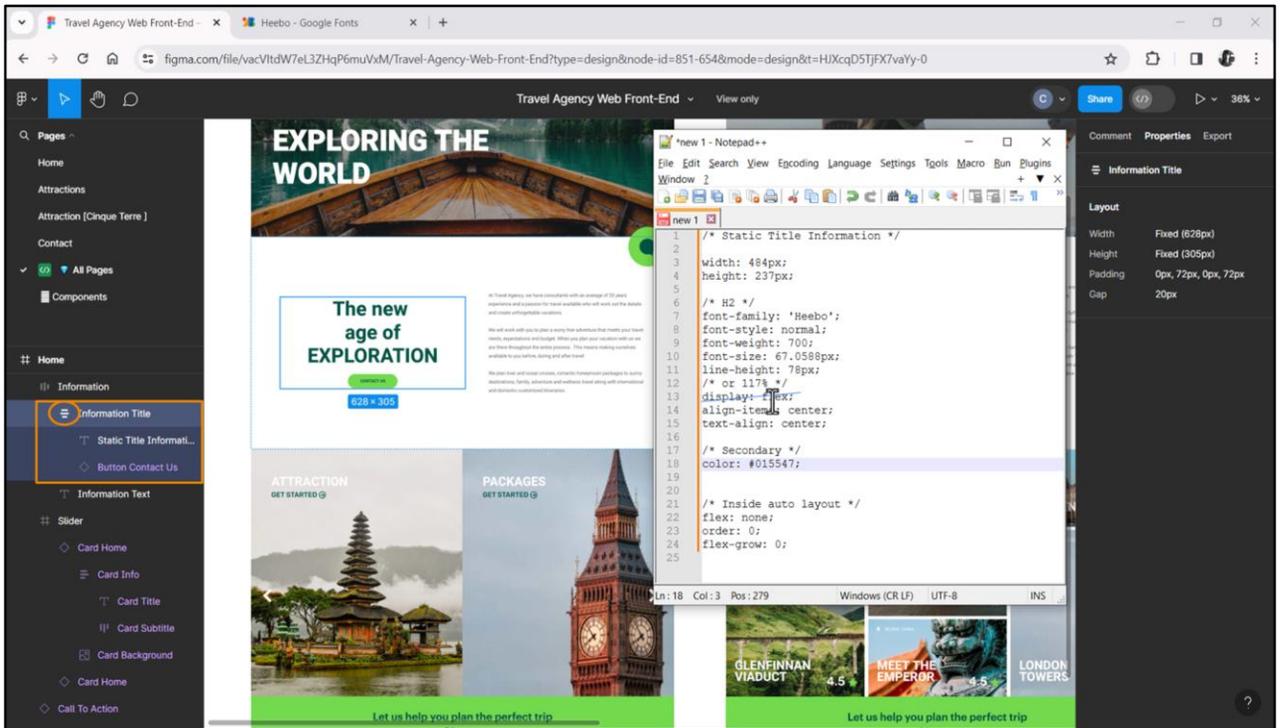
E isso fará com que, se depois, mais adiante, Chechu quiser mudar essa cor deste valor para qualquer outro valor, simplesmente viremos para alterá-lo aqui. E não teremos que mexer em absoluto na classe. Bem, vamos gravar.

E em execução não vamos notar absolutamente nenhuma mudança porque continuaremos vendo essa cor verde.



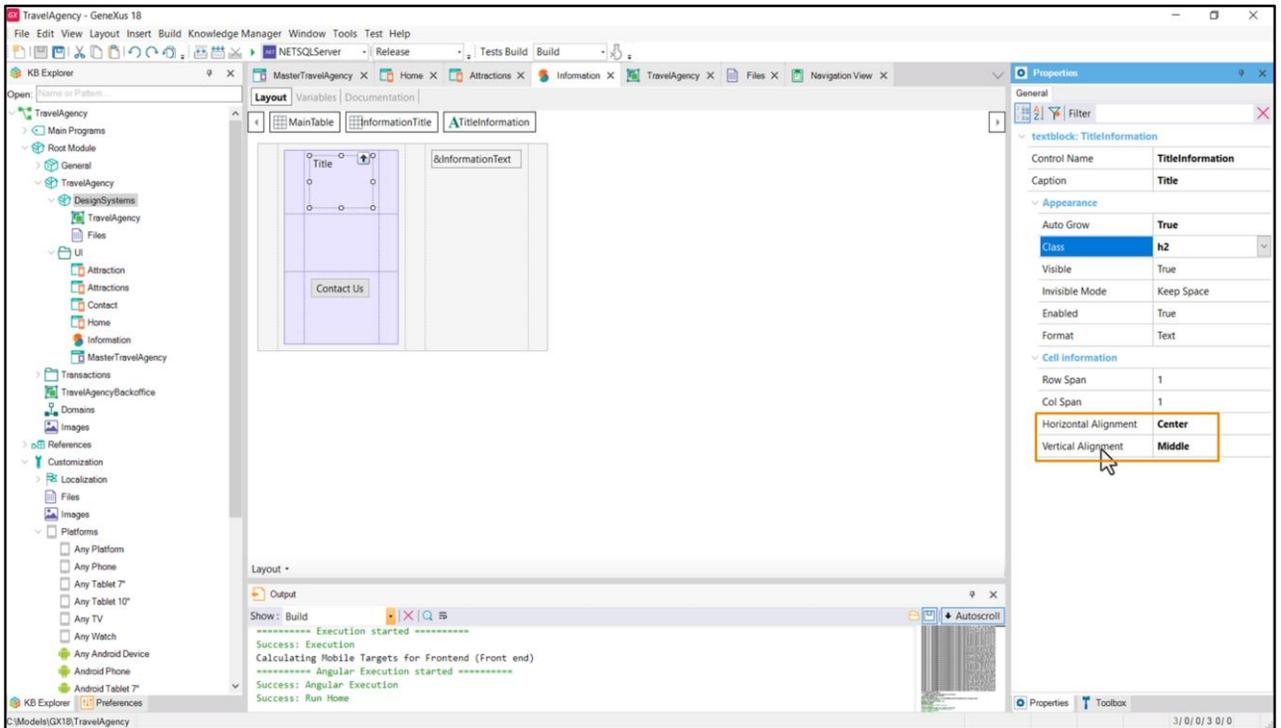
Bem, e o que acontece com essas três propriedades que não copiamos para nossa classe h2?

As duas primeiras têm a ver com o container que foi utilizado em Figma para colocar dentro esses dois textos. E em particular este.



Se trata de um grupo ou um frame que foi catalogado como Auto Layout e do qual falaremos mais tarde, mas que, já podemos adiantar, é implementado com um container Flex. Nós fomos por outro lado, que foi escolher uma tabela para implementar esse container.

Portanto, a primeira propriedade, display, não precisamos de forma alguma. E essas duas propriedades, implementamos...

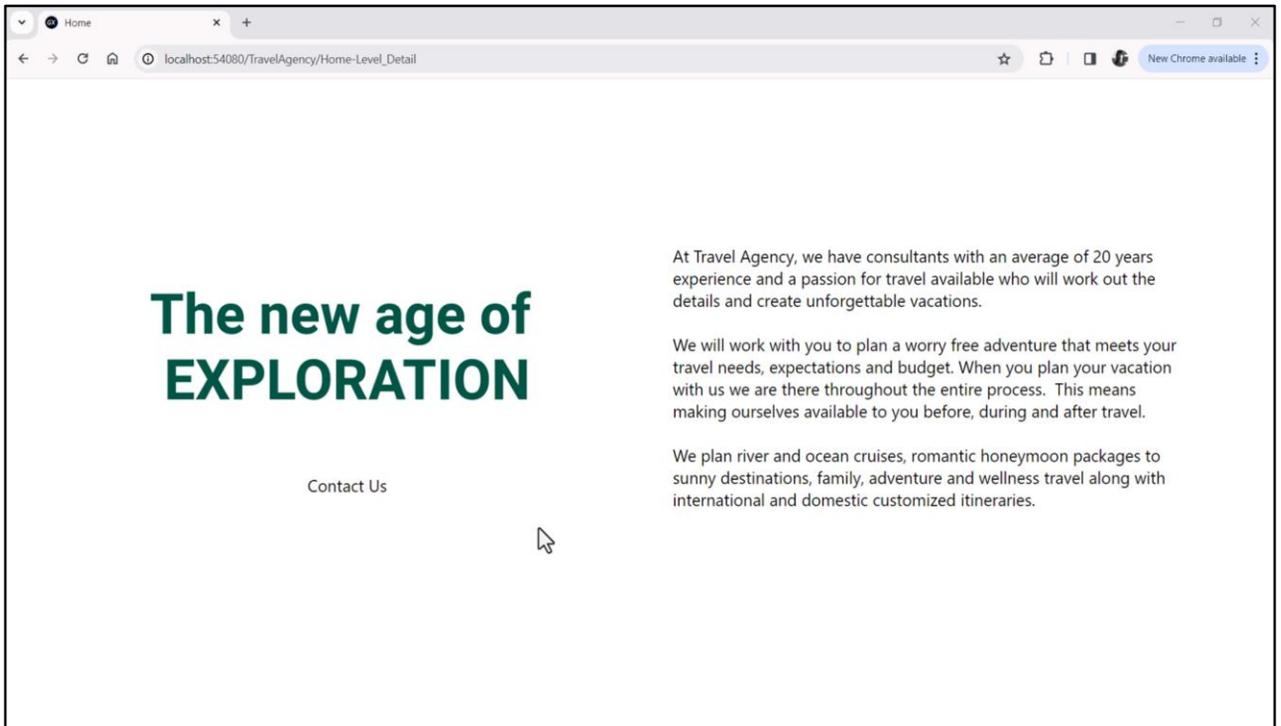


... através dos alinhamentos horizontal e vertical... dos dois elementos.

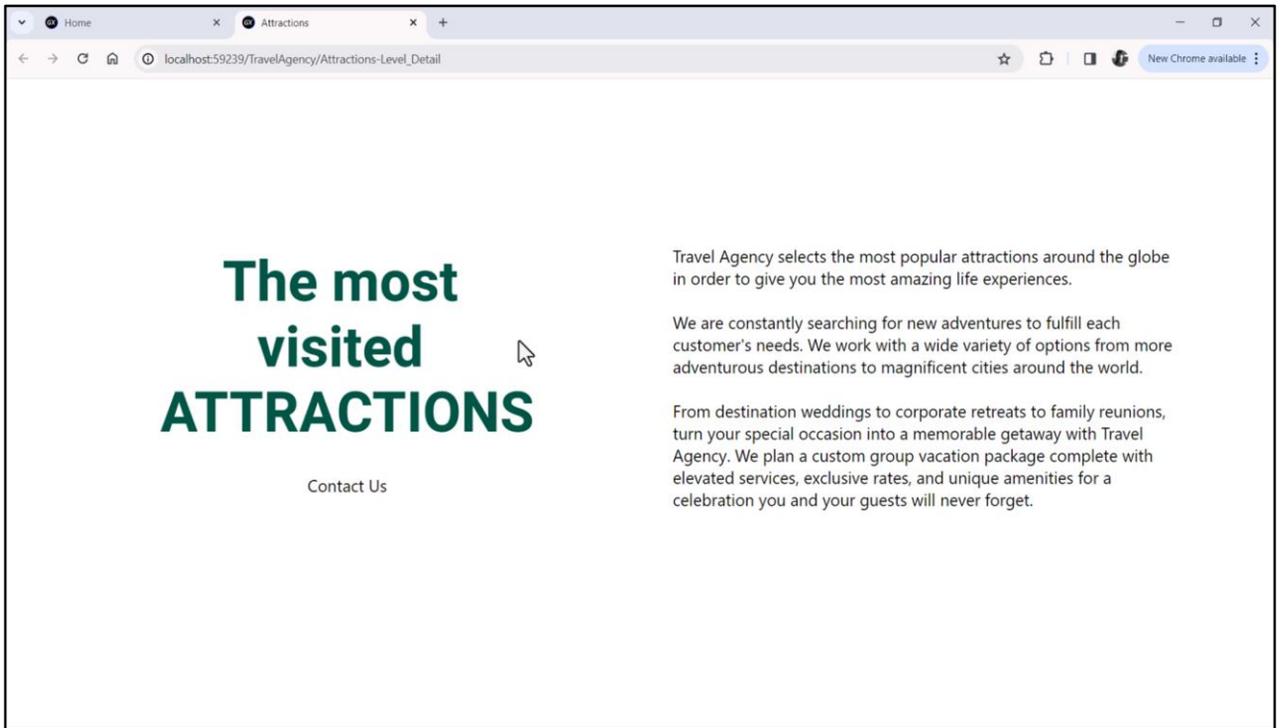
Assim, não precisamos copiar nenhuma dessas propriedades para nossa classe.

Desta maneira, ficou bem separado o que é o estilo propriamente do controle, do que é mais estrutural e tem a ver com o alinhamento do controle no layout. Esta é uma possibilidade, claro, e não necessariamente é a melhor.

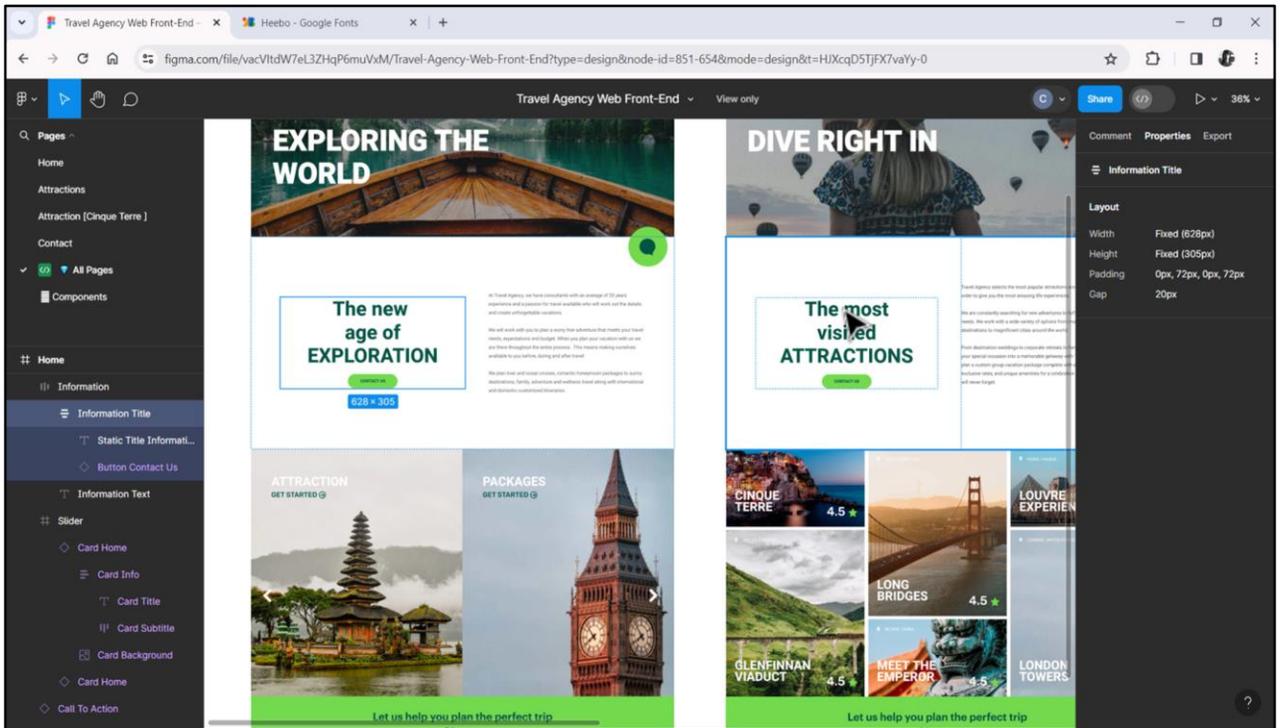
Na implementação que estamos realizando, a parte do espaçamento e dos alinhamentos, estamos resolvendo no próprio layout e o restante do estilo estamos manuseando no DSO. Um DSO é mais fácil de alterar do que um layout, portanto, tomar essas decisões tem seus prós e contras.



Uma última coisa antes de terminar, em execução o texto de Home, pelo seu tamanho, está sendo visto em duas linhas... vamos ver o que acontece com o de Attractions.

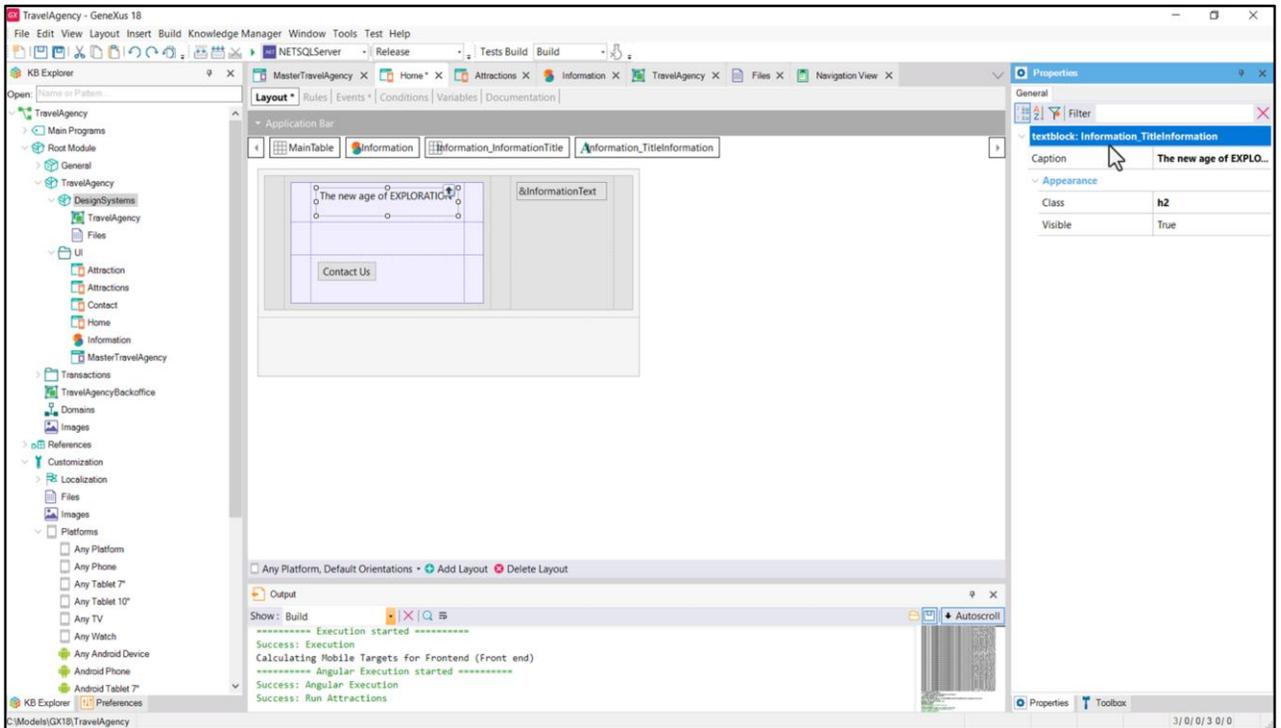


Este sim é visto em três...

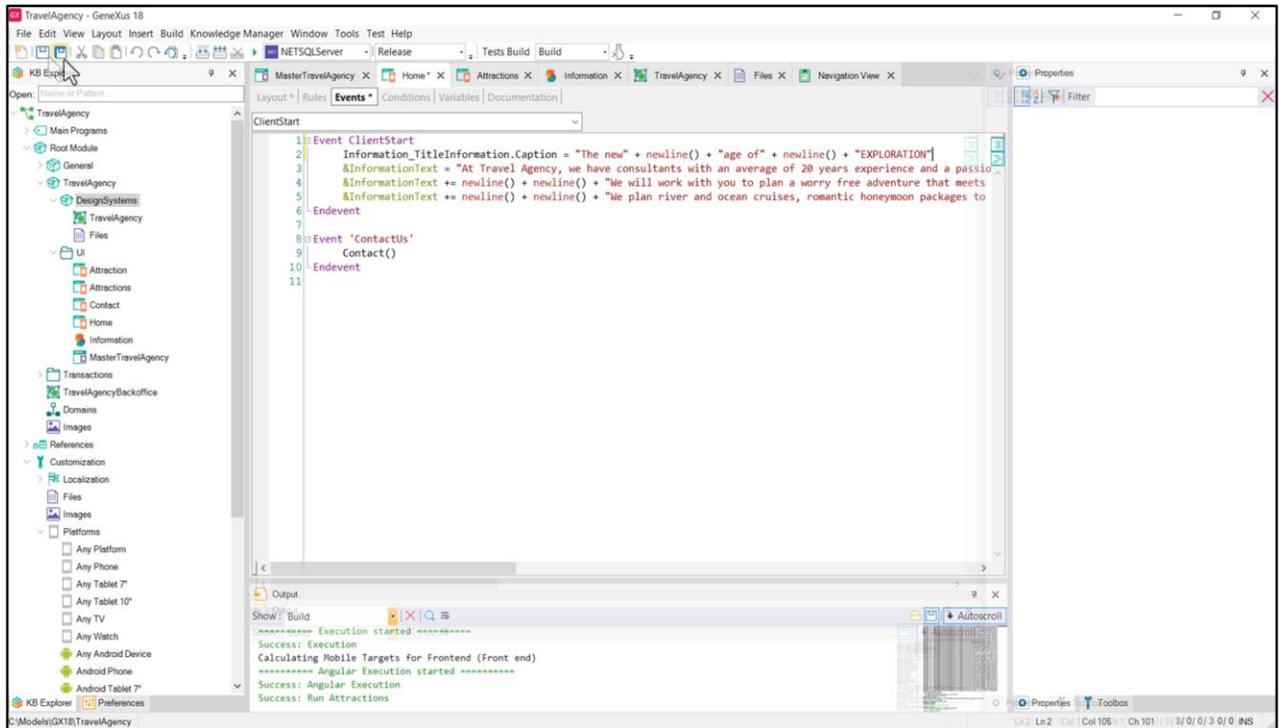


... do jeito que vemos no arquivo em Figma.

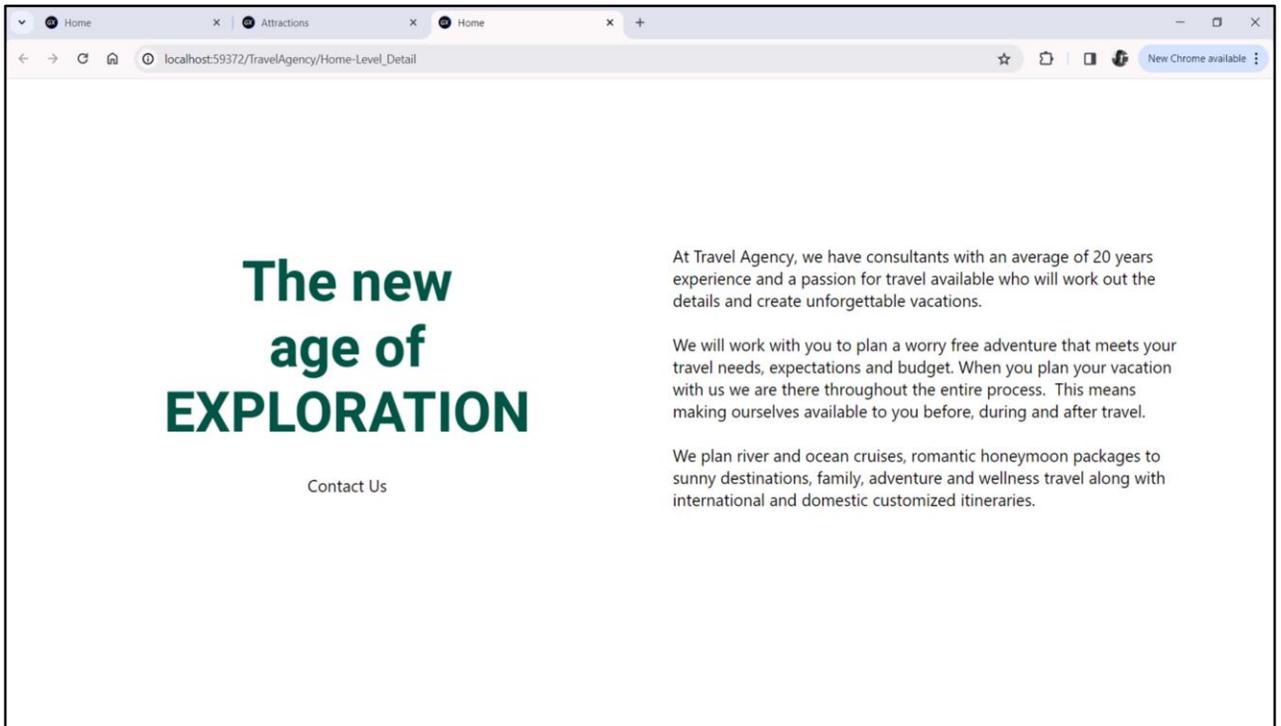
Então o que teríamos que fazer é separar este texto em três linhas.



Uma forma de fazer isso é: vemos aqui o nome do textblock, que é aquele que vem do stencil, Information, sublinhado e o nome do controle no stencil, TitleInformation.



Então, venho para Home e nos eventos, no ClientStart, para Information Title Information ponto Caption atribuo "The new" ...  
Gravo e executo.



E aqui o vemos.

Bem, agora vamos dar estilo ao parágrafo e ao botão... no vídeo seguinte.

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

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