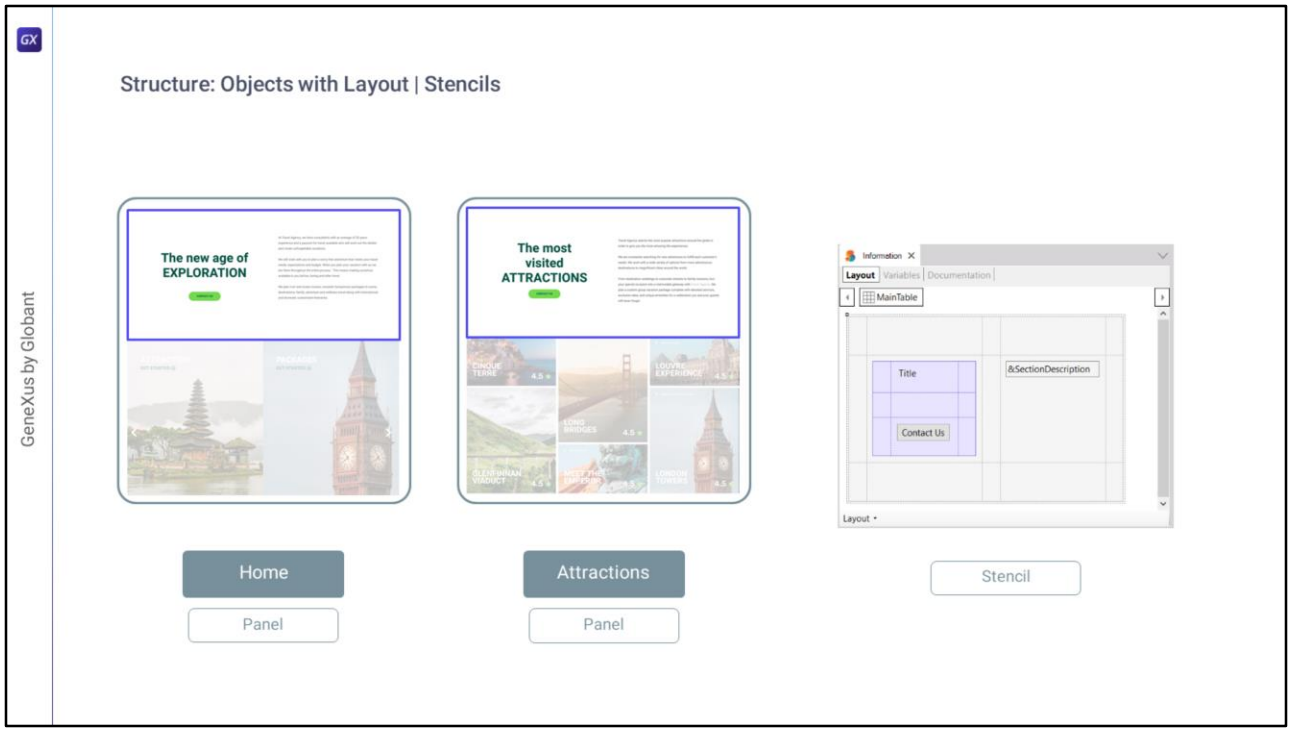
GeneXus by Globant

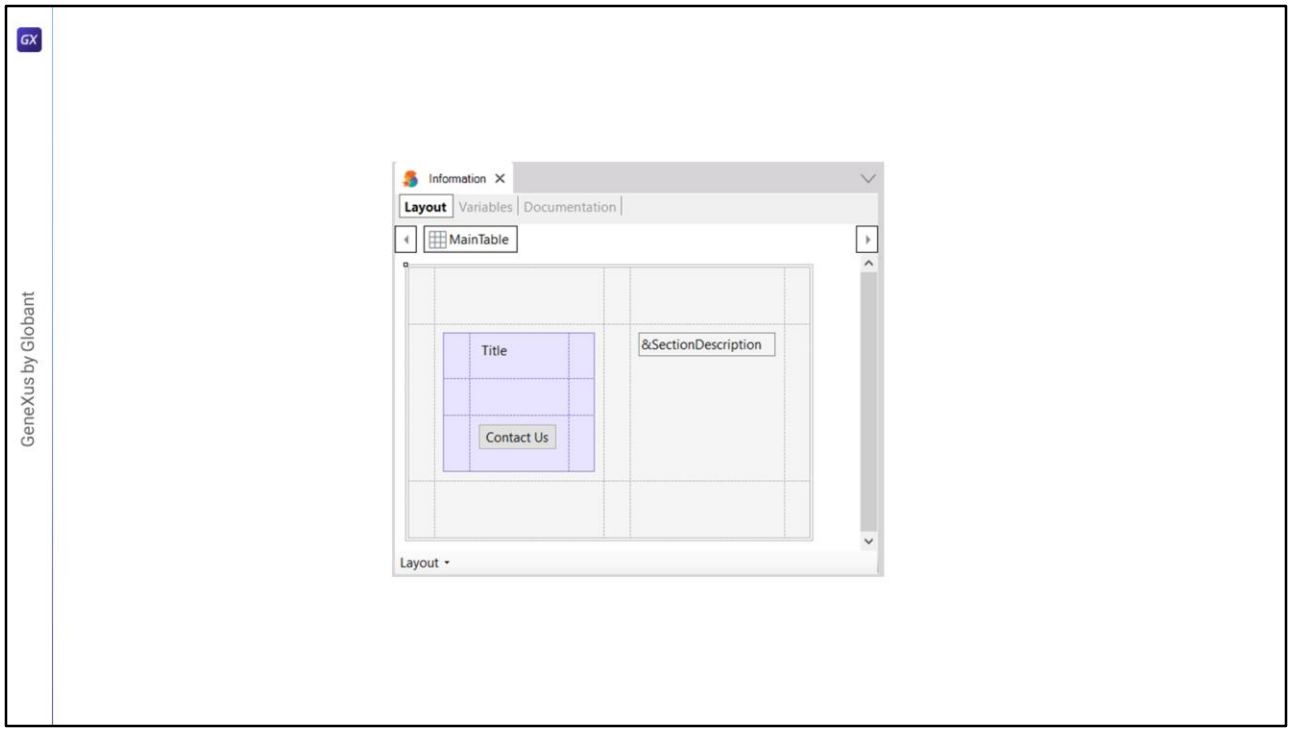
First Layout in GeneXus. Structure

Cecilia Fernández

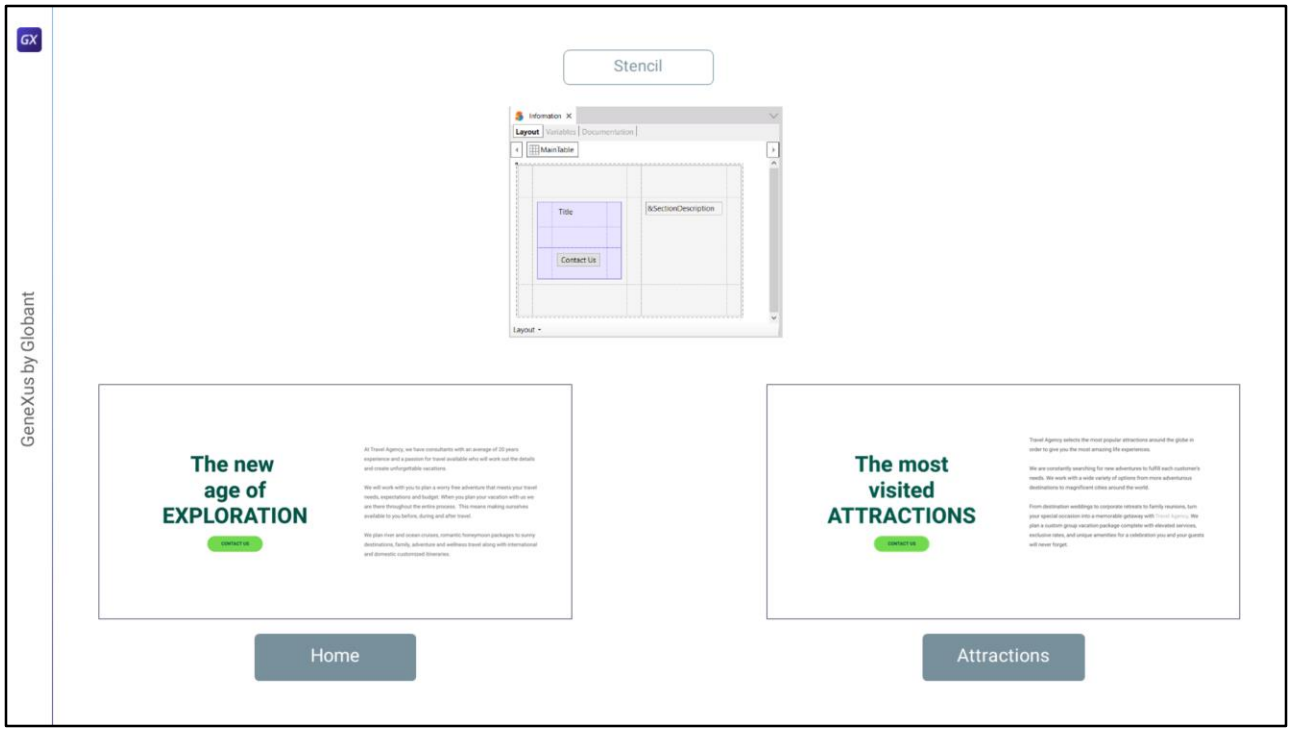


No módulo anterior havíamos identificado que essas duas partes das telas Home e Attractions eram idênticas em estrutura, então podíamos optar por modelá-las dentro de um objeto Stencil, justamente para fins de reutilização, e então…

...inseri-lo como controle em cada um desses painéis.

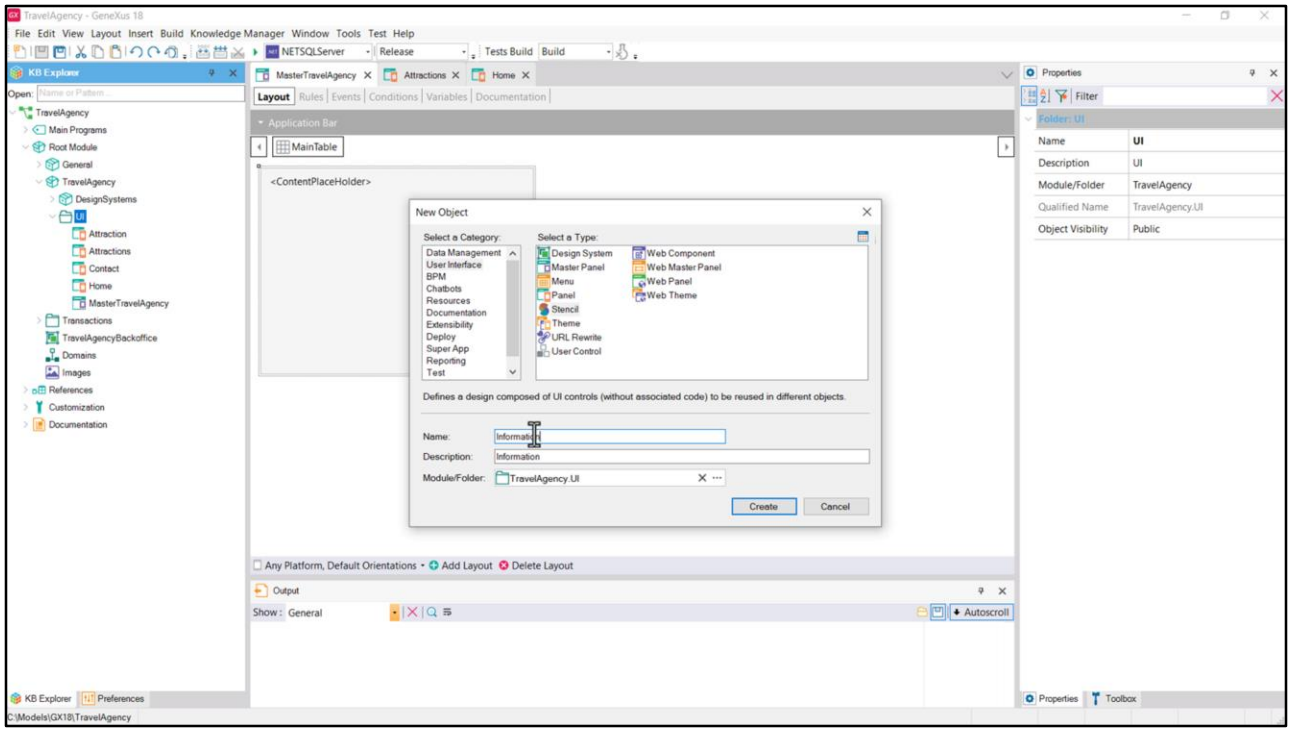


Nesta aula vamos nos dedicar a analisar qual hierarquia de controles devemos utilizar para implementar justamente a estrutura desse stencil...

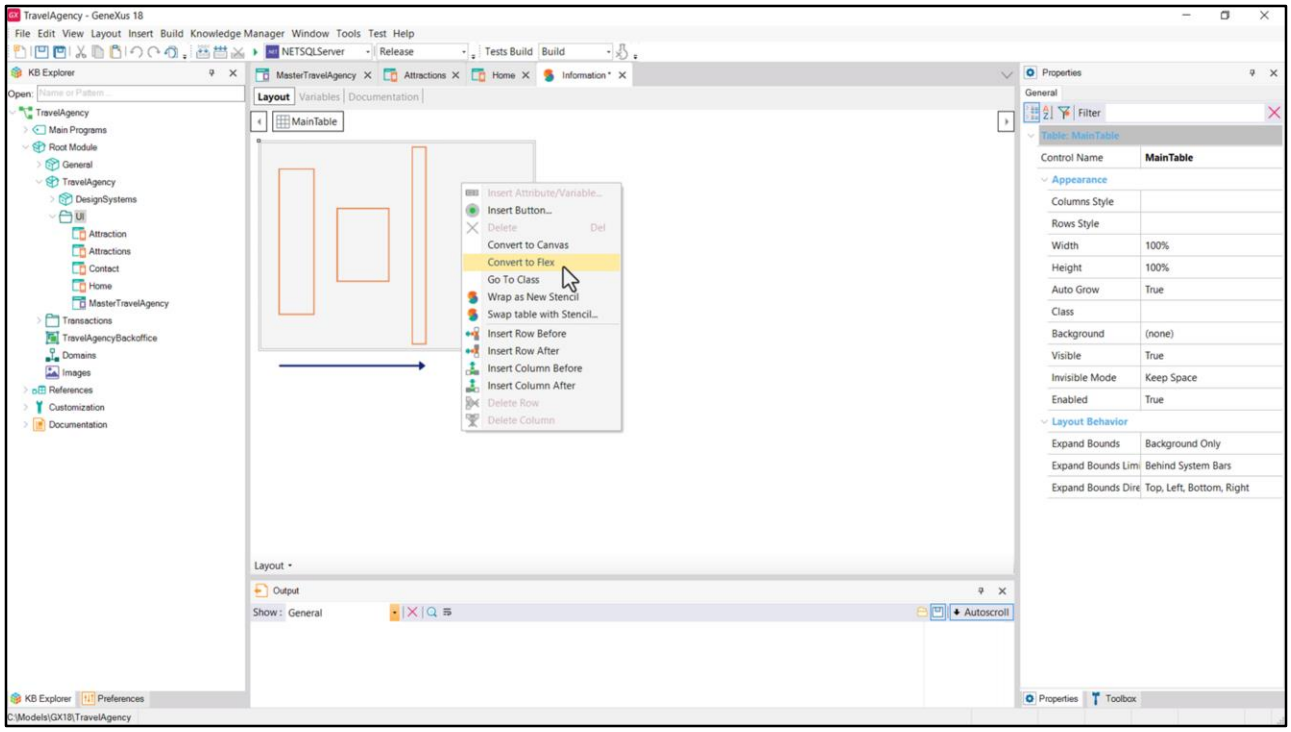


E então veremos nas próximas aulas como fazer com que essa estrutura fique da maneira definida por nossa designer no arquivo em Figma.

Então vamos começar criando em GeneXus o Stencil.

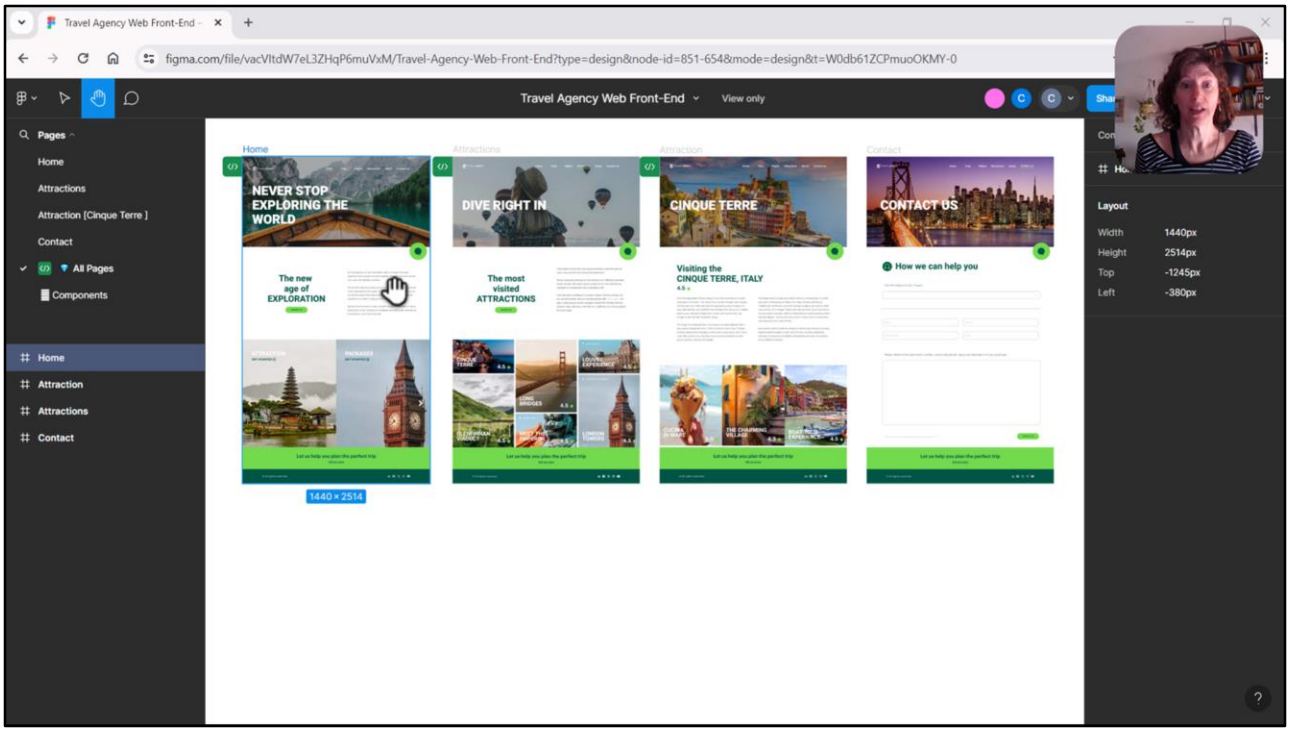


Para isso, posicionados nesta pasta, que é onde queremos criar o objeto Stencil, escolhemos o tipo de objeto Stencil e seu nome será Information.



Vemos que o layout é inicializado com uma tabela denominada Main Table e que, enquanto não colocarmos nenhum controle dentro dela, permanecerá vazia.

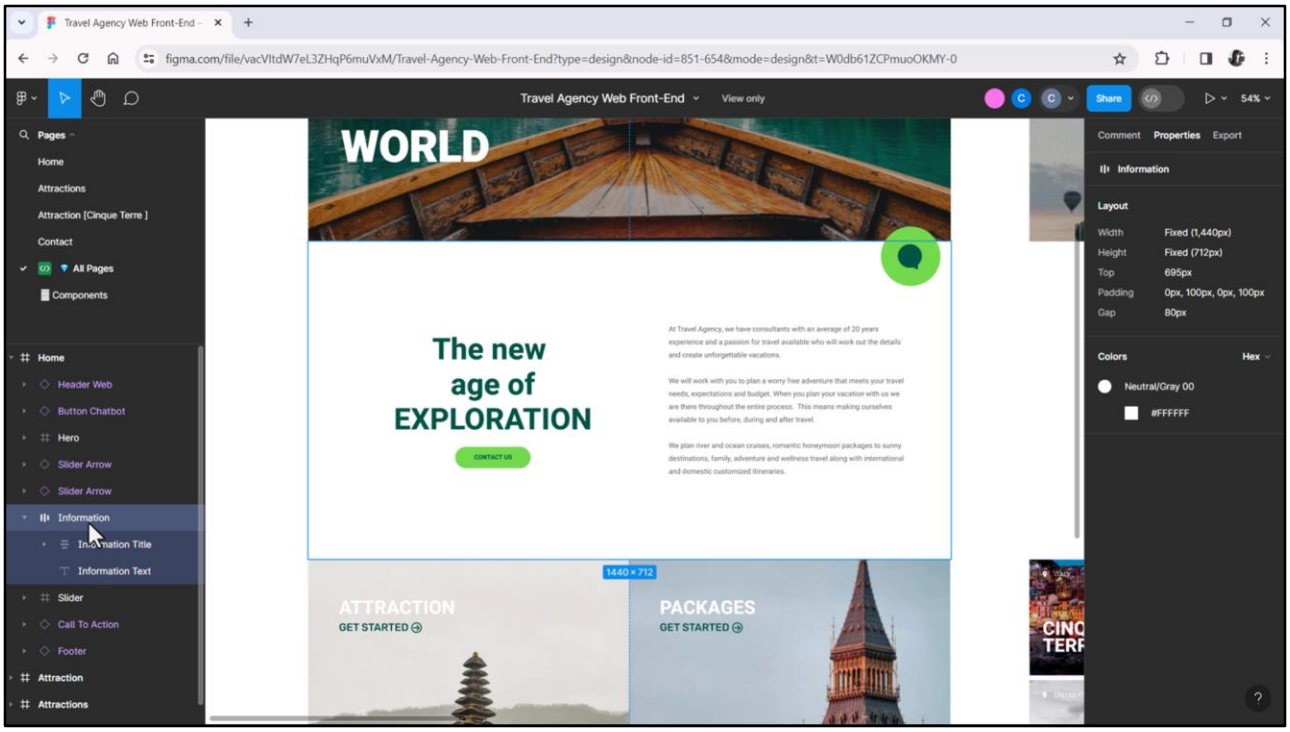
Se pressionarmos o botão direito, nos oferece a possibilidade de converter esta tabela em dois tipos de containers que possuem usos específicos: Canvas, que será o que utilizaremos posteriormente para implementar o Header do Master Panel, que nos permitirá, já adianto, sobrepor controles; e o Flex, que nos permitirá colocar os elementos da tabela em torno de uma direção, seja horizontal, ou seja, de linha, ou vertical, ou seja, de coluna.



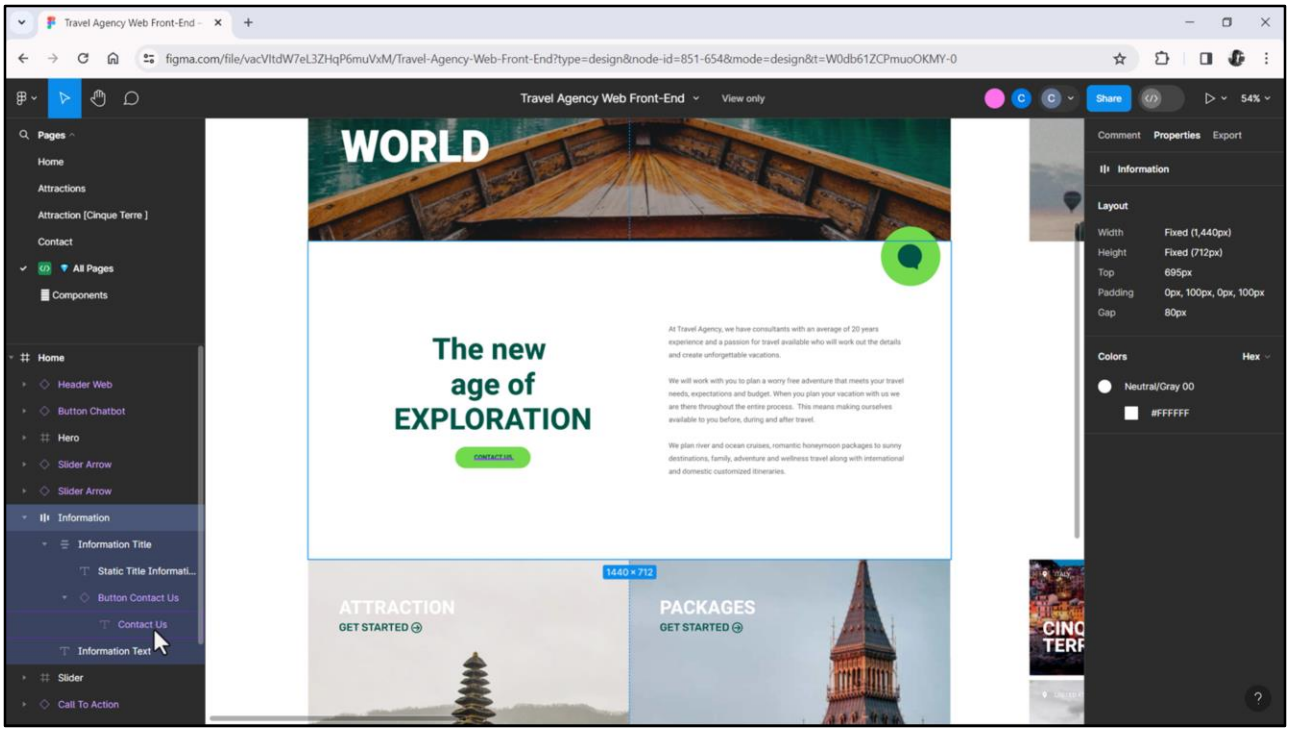
Bom, vamos analisar em Figma esse layout que é o que queremos implementar.

Primeiramente, observemos que aqui estamos posicionados no frame Home, que é aquele que desenha a página Home. Vemos que tem uma largura de 1440 pixels e uma altura de 2514. E todas serão mais ou menos iguais. O que isto significa? Que, se executarmos nossos objetos GeneXus, nossos painéis, em um navegador que tenha essas dimensões, as telas deverão ficar exatamente assim.
Mas, como nos disse Chechu, este desenho foi pensado de tal forma que pode expandir um pouco a largura ou contrair e que o design se adapta a essa nova largura. Como ela nos explicou, esse design corresponde a uma determinada largura e pequenas variações dessa largura, porque se o diminuirmos o suficiente, terá que montar novamente essas telas, redesenhá-las para acomodar a informação de outra forma, de acordo com essa largura de tela. Isso é o que se conhece como breakpoints, ou seja, são pontos de interrupção, onde o design deve necessariamente ser modificado. E isso acompanhará os tamanhos Tablet, Phone e pequenas variações desses tamanhos.

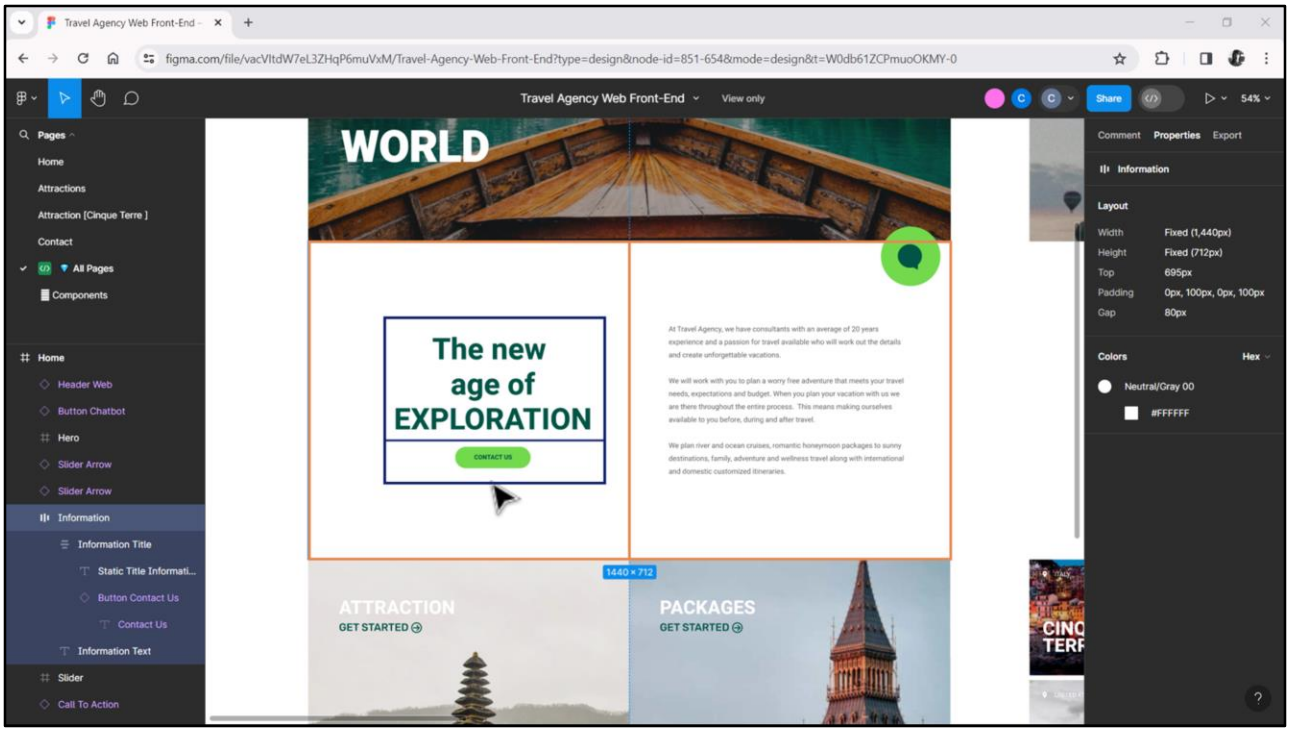
Aqui, vamos lembrar, estamos implementando a aplicação para tamanho deste, e suas variações.



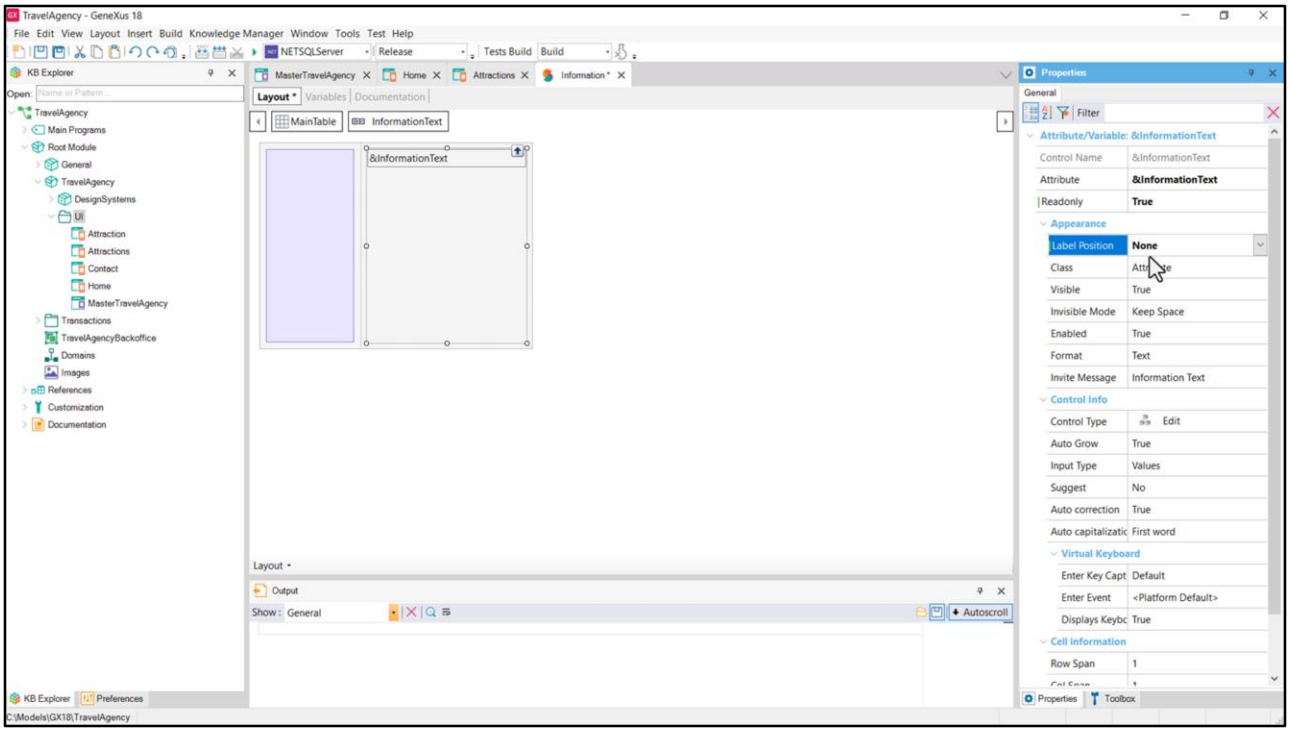
Vamos analisar agora o layout que nos interessa, aquele que queremos implementar com o objeto Stencil. Corresponde a este container aqui, o denominado Information, que, observemos, contém dois elementos: outro container, o denominado Information Title que vemos à esquerda na tela, e um texto, à sua direita.



O container da esquerda, por sua vez, contém outro texto, desta vez acima, e um botão abaixo, que como vemos, é uma instância de um componente; componente que terá duas camadas, uma para o fundo e outra para o texto do botão, depois iremos analisá-lo.



Então, a nível estrutural poderíamos organizar assim esta tela: uma tabela (nossa tabela main) com uma única linha e duas colunas, onde na primeira temos outra tabela, e na segunda o controle de texto, que em GeneXus poderia ser tanto um TextBlock como uma variável de texto do tipo edit, readonly. E a tabela interna seria uma com uma única coluna e duas linhas, onde na primeira colocaríamos o outro controle de texto (TextBlock ou variável, teremos que escolher) e na segunda o botão.



Se fizermos isso em GeneXus…

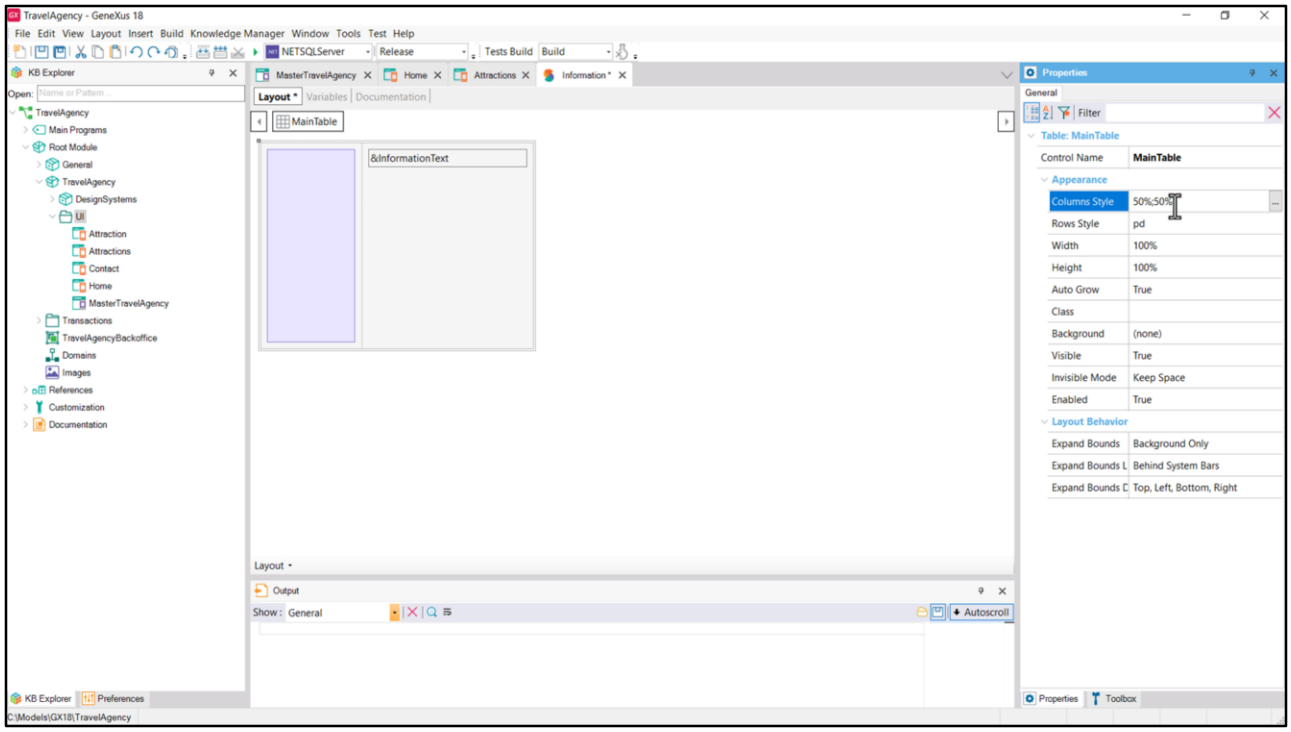
A tabela já temos, é a Main. Então vamos inserir o primeiro elemento, que é a outra tabela. Vamos mudar seu nome para que coincida com o nome que nossa designer lhe deu, será mais fácil para nós depois, de localizar. Bem, e agora vamos inserir o elemento da direita, que era o texto.
Podemos optar por implementá-lo com um controle de tipo variable ou com um textblock.

Vou optar pela variável. Eu vou arrastá-la. Vou criá-la, com o mesmo nome que nossa designer deu, Information Text. Vou defini-la com qualquer um dos tipos de dados que representam caracteres: pode ser Character, VarChar ou LongVarChar, que será o que vou escolher. Como veremos, e explicarei em breve, a escolha deste tipo de dados fará com que a propriedade Auto Grow automaticamente seja ativada, fique em True.

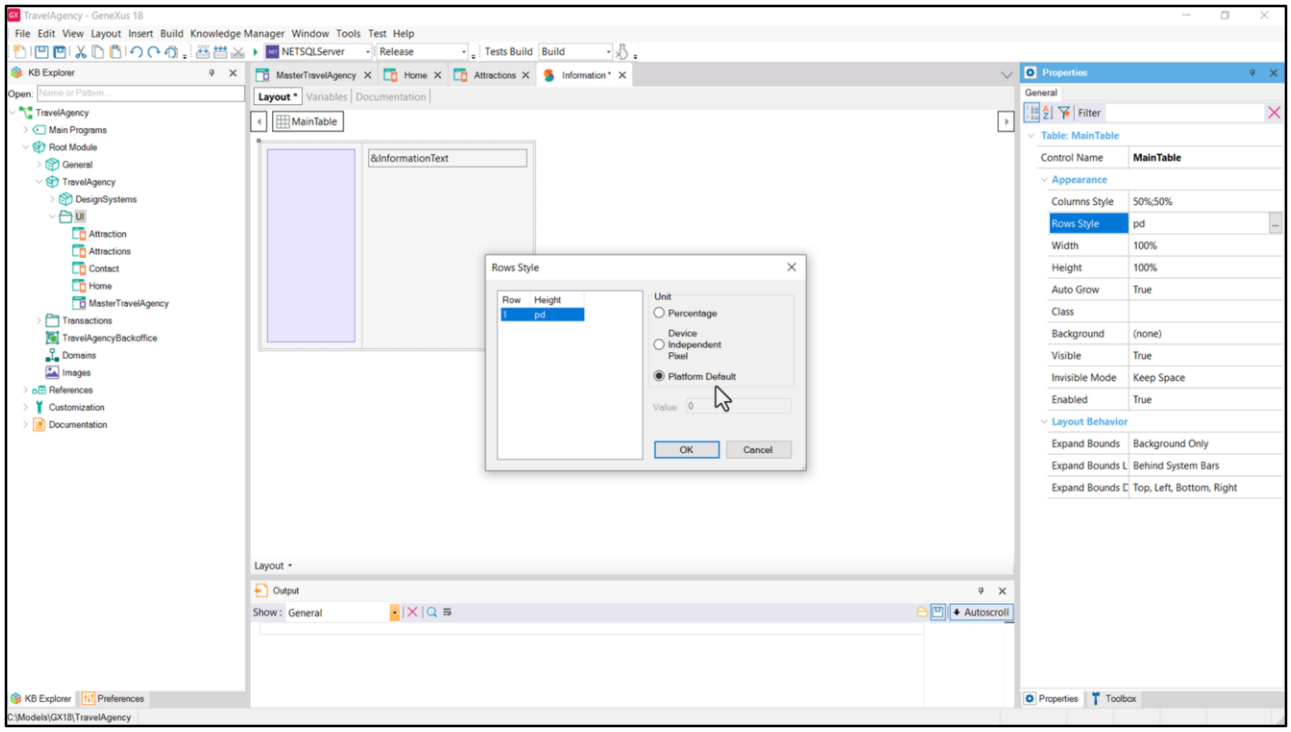
Uma das desvantagens das variáveis é que elas são por padrão de escrita. Então, vou alterar a propriedade Readonly para que seja somente de leitura.
E a outra desvantagem é que têm um rótulo, por padrão, então o que vou fazer é dizer que a posição do rótulo seja nenhuma, que não esteja em lugar nenhum, que esteja oculto, que não seja mostrado.

Eu disse então que poderia ter escolhido, em vez de escolher a variável, poderia ter escolhido representar essa mesma coisa com um Text Block. Escolho a variável porque é possível que essa informação, o conteúdo, venha do Backoffice, ou seja, o extraia da base de dados, e é mais fácil atribuir o conteúdo a uma variável, simplesmente com a atribuição direta usando o

sinal de igual, que fazer isso para o Text Block, para o qual terei que usar a propriedade Caption. Textblock ponto caption igual... e aí sim atribuir valor. Este justamente é um dos casos extremos, certo? Porque este texto não irá variar, portanto poderia muito bem ser um text block. É claramente o uso... estamos inclinados a usar uma variável, claramente, no caso em que o conteúdo mude.

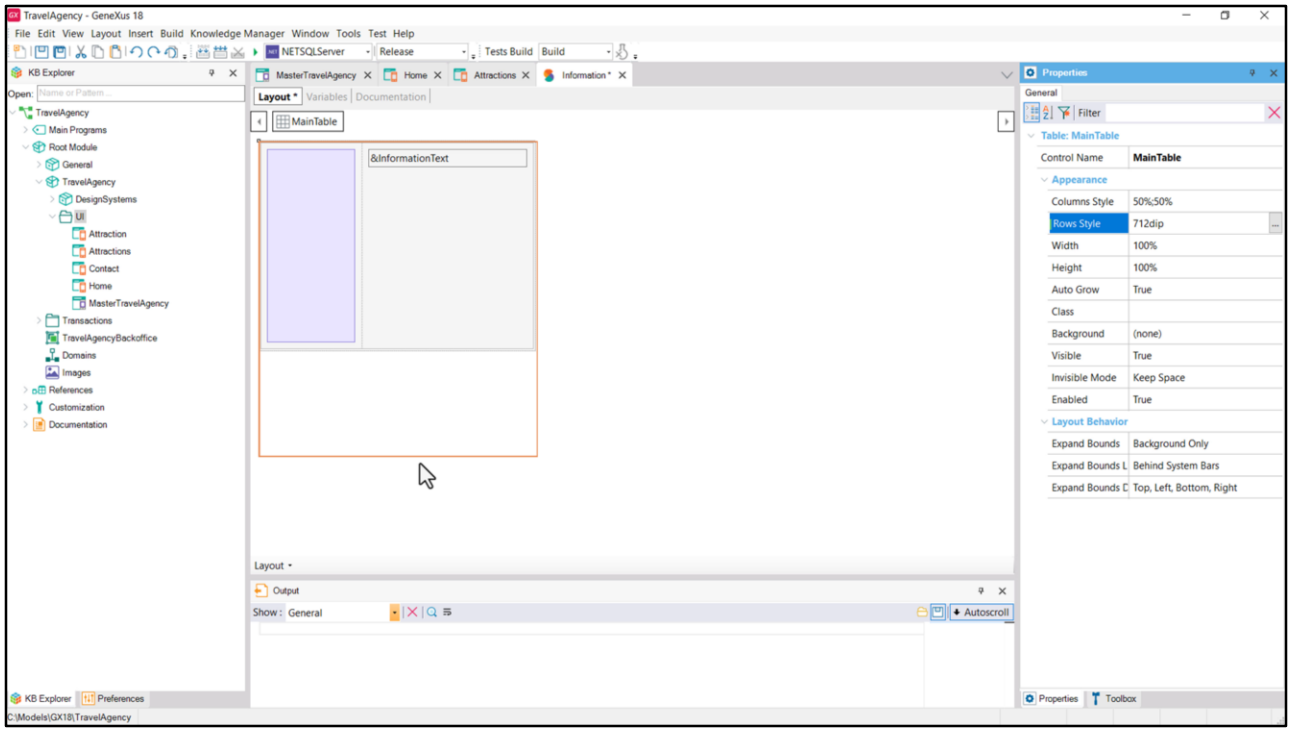


Agora, antes de adicionar os controles para esta tabela, observemos as propriedades da tabela main, em particular estas quatro.
A primeira está especificando qual será a largura de cada uma das duas colunas. E a segunda está especificando a altura da única linha que a tabela possui no momento.
O que significam esses valores de 50% para cada coluna? Que sua largura será 50% da largura total da tabela. E qual é a largura total da tabela? Aquela dada pela propriedade Width, que também é, neste caso, uma porcentagem, que está indicando que a largura da tabela corresponderá a 100% da largura do local onde se encontre.

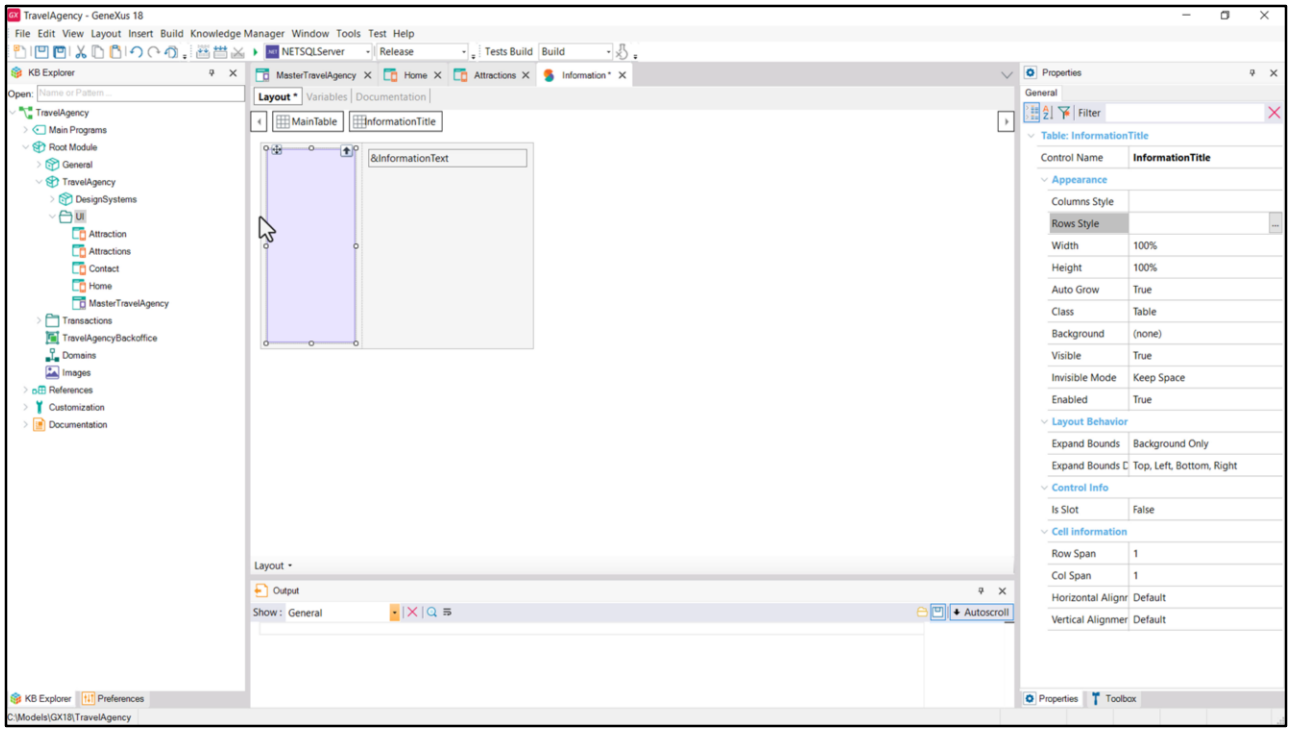


E o que significa o valor pd na propriedade que determina a altura das linhas? Aqui o indica claramente, é uma abreviatura de Platform Default. Dependerá de cada plataforma qual é esse default, mas é um valor fixo. Poderíamos especificar uma porcentagem, neste caso é relativa à altura da tabela, que aqui também vemos é 100%, ou poderíamos especificar um valor em dips, que é o mesmo que dizer pixels.

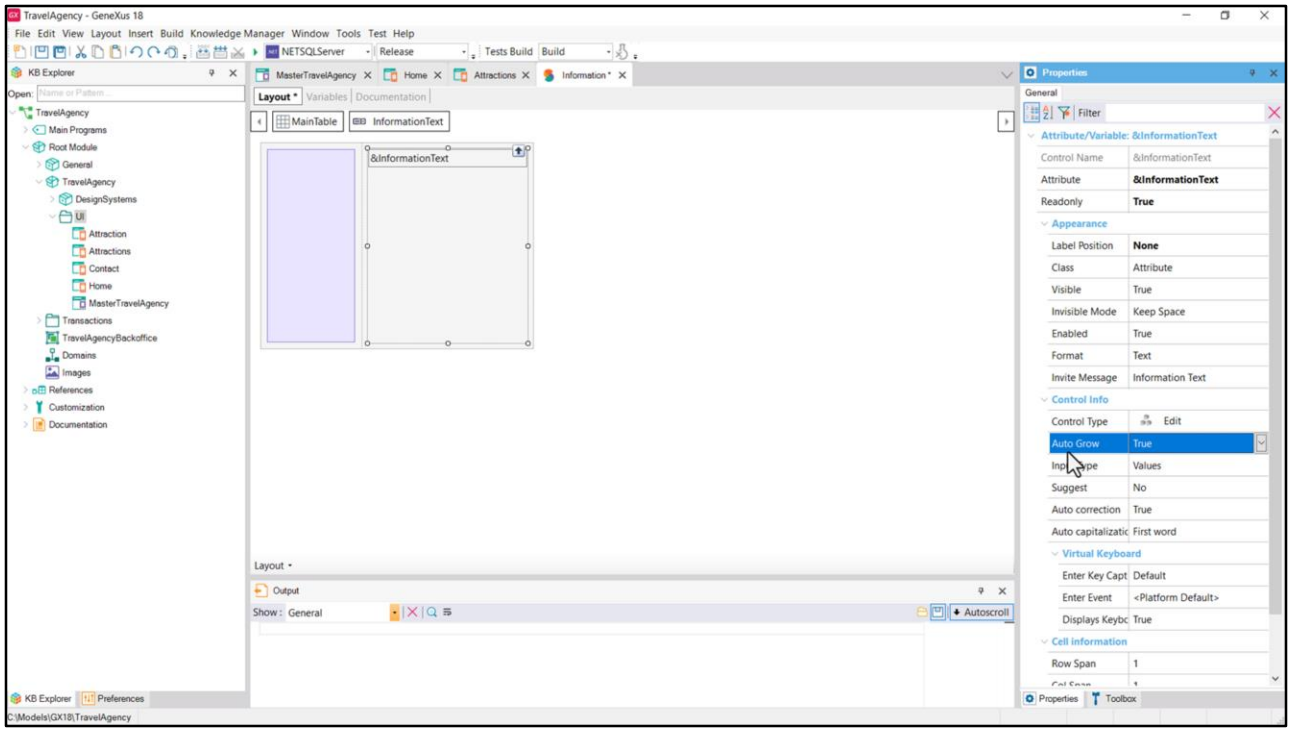
O dip é uma abstração do pixel que começou a ser utilizado no GeneXus quando começamos a implementar aplicações nativas. A equivalência é de um para um.



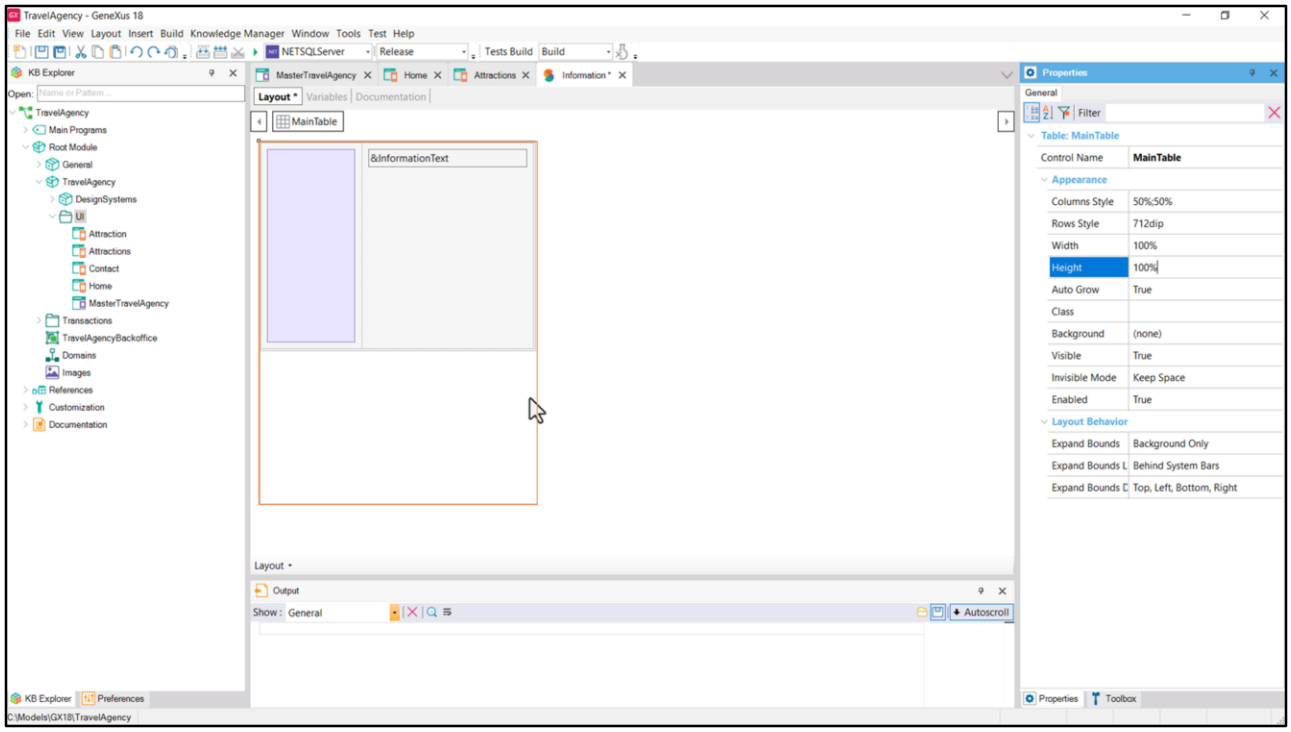
Coloquemos, então, aqui os 712 dips ou pixels que obtivemos inspecionando o frame em Figma.
Te incomoda que a altura desta linha seja desse tamanho, mas a altura da tabela (que só tem essa linha) seja 100%? Aqui você tem que ter a seguinte precaução: embora no editor não esteja sendo discriminada a borda da tabela da borda da linha, não são iguais. A borda da tabela se estenderá até ocupar 100%. Enquanto a borda da linha corresponderá a esses 712 dips.



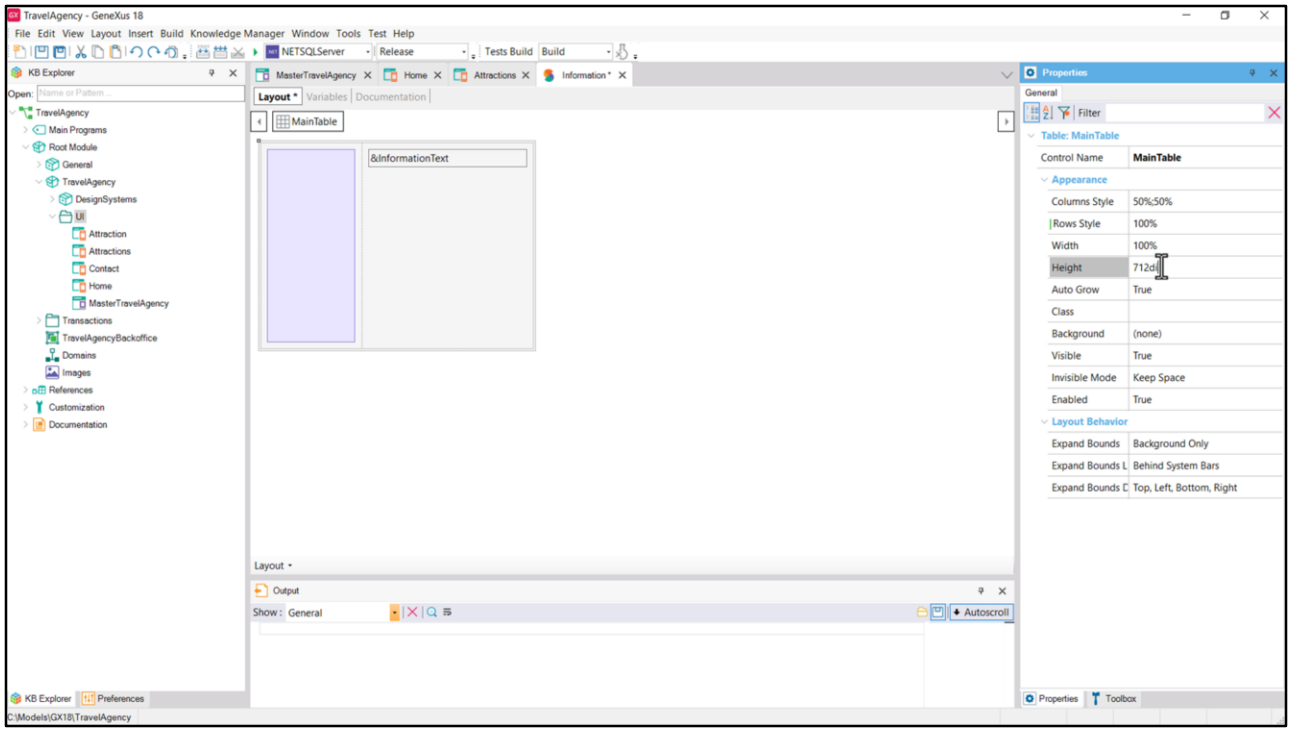
Os controles, por padrão, ocupam todo o espaço da célula em que se encontram. Assim, se observarmos as propriedades desta tabela, que por enquanto está vazia, vemos que sua largura e altura estão configuradas em 100%, ou seja, ocuparão toda a célula em que se encontram.



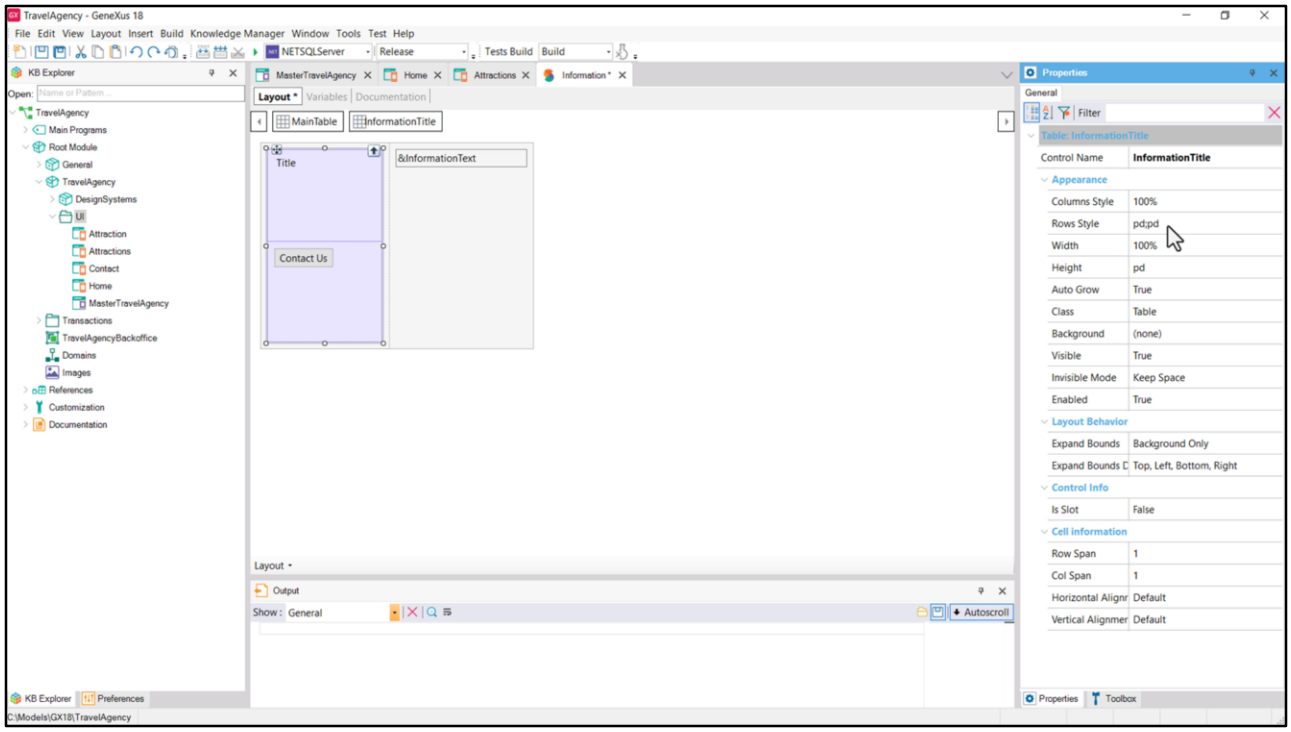
E o mesmo acontecerá com a variável, e você tem que saber disso porque não existe propriedade que mostre isso.
O que acontecerá se o conteúdo da variável for tão grande que não caiba nos 712 pixels de altura da célula? Como a propriedade Auto Grow está definida como True, assim como está a mesma propriedade no nível da tabela, então o que acontecerá é que a célula se expandirá para baixo para não deixar conteúdo de fora. Se a propriedade estivesse definida como false, então o conteúdo seria truncado em 712 pixels ou dips.



Agora, se deixarmos isto assim, então embora a linha terá 712 pixels ou dips de altura, como a altura da tabela ficou em 100%, se no local onde é colocado este stencil tiver uma altura maior que 712 pixels, então poderia ficar espaço vazio.



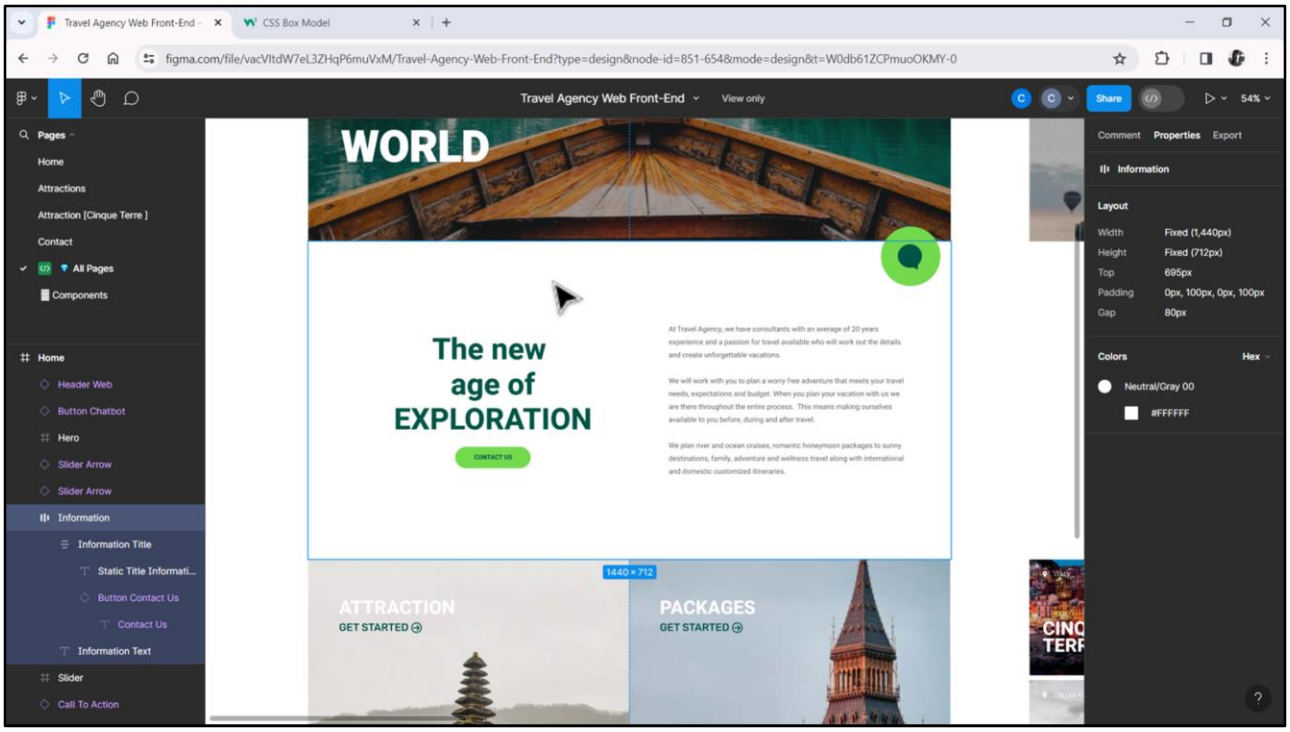
A outra alternativa seria que a altura da linha fosse 100%, ou seja, que ocupe 100% da altura da tabela, e definir na altura da tabela esses 712 dips ou pixels.



Agora vamos inserir o texto como TextBlock nesta tabela (sabemos que é um texto curto, que provavelmente não mudará), vamos chamá-lo da mesma forma que Chechu em Figma. E para Caption, por enquanto vou deixar Title, pois posteriormente será atualizado nos painéis Home e Attractions com o valor que deverá assumir em cada um.

E em outra linha vamos inserir o botão.

Se observarmos as propriedades da tabela, vemos agora que a coluna é apenas uma, 100% da largura, e cada linha ocupará o default da plataforma de altura. Claramente teremos que mudar isso, então vamos ao Figma para ver quais são os valores que necessitamos para as larguras e alturas.



Mas antes de analisar esses tamanhos, vamos analisar o caso do container externo.

O que quero colocar em foco agora é no espaçamento entre os elementos e a borda do container. Continuamos no próximo vídeo.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com