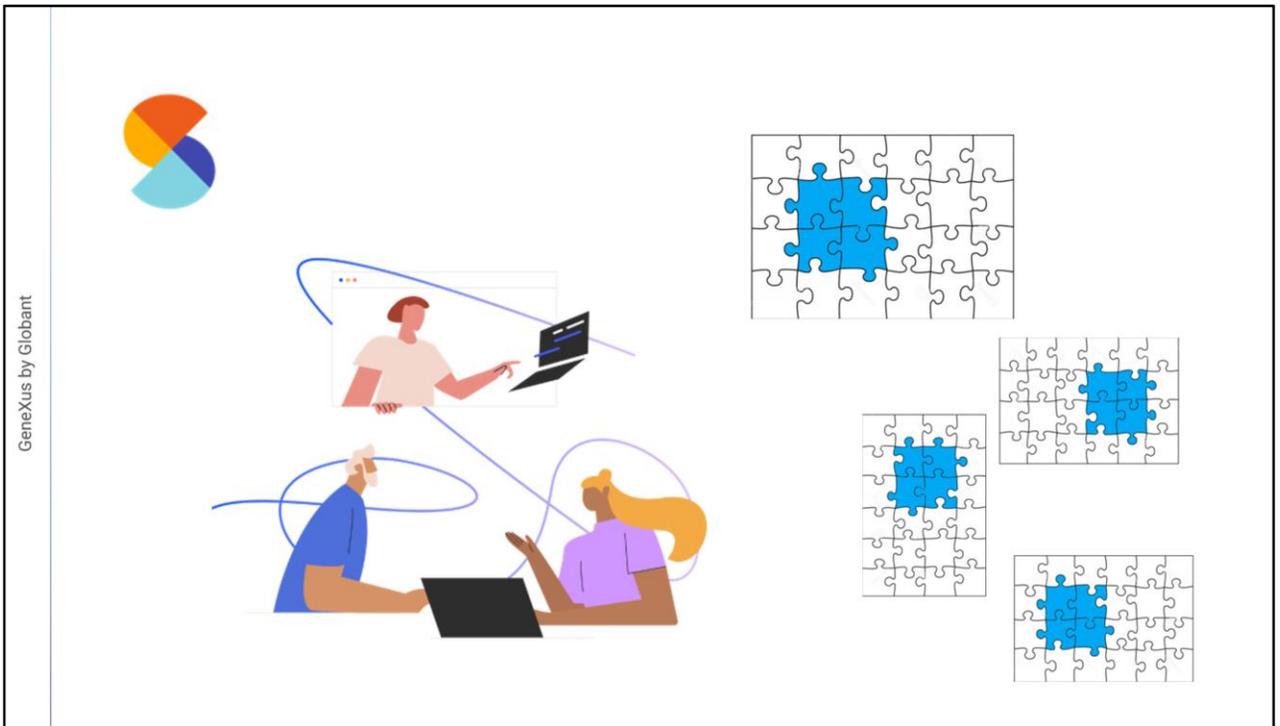


# Stencil object



Vanesa Fernández



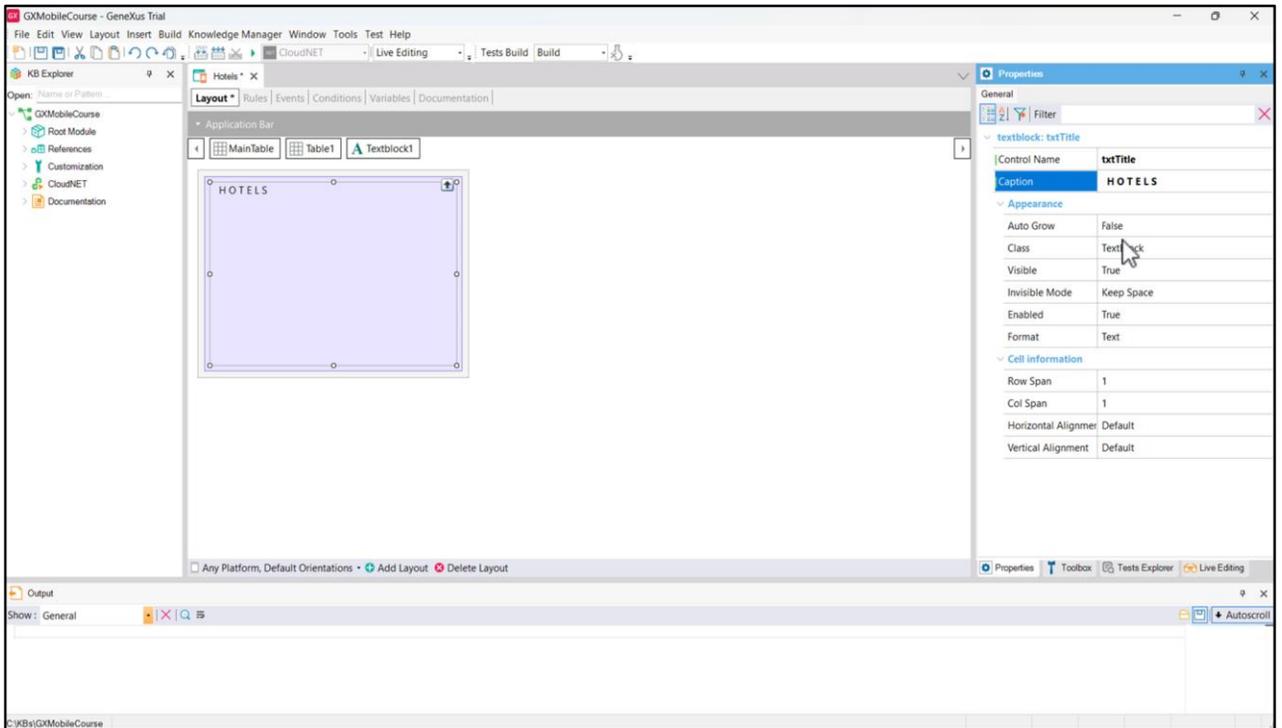
Os Stencils integram o trabalho de designers e desenvolvedores e permitem compartilhar designs compostos por controles da UI (sem código associado) e reutilizá-los "como uma unidade" em diferentes objetos.

É possível definir vários Stencils que possuem os mesmos controles, mas estão distribuídos ou apresentados de diferentes maneiras, permitindo manter o mesmo design em diferentes objetos, mas com variantes. Ajudam a definir padrões e adicionar consistência, simplificando a implementação de Sistemas de Design.

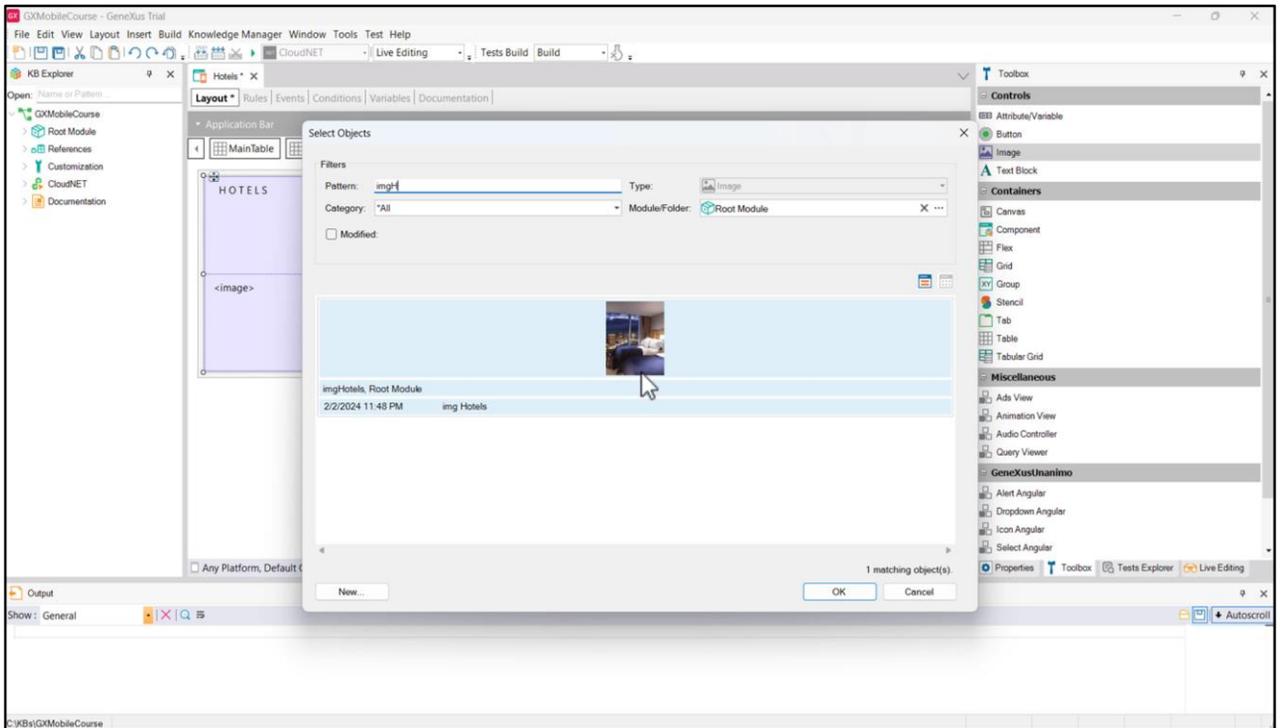
Os Stencils podem ajudar em diversos cenários: um deles é na reutilização do design visual de um controle de tabela em vários painéis, empacotando-o como um Stencil.



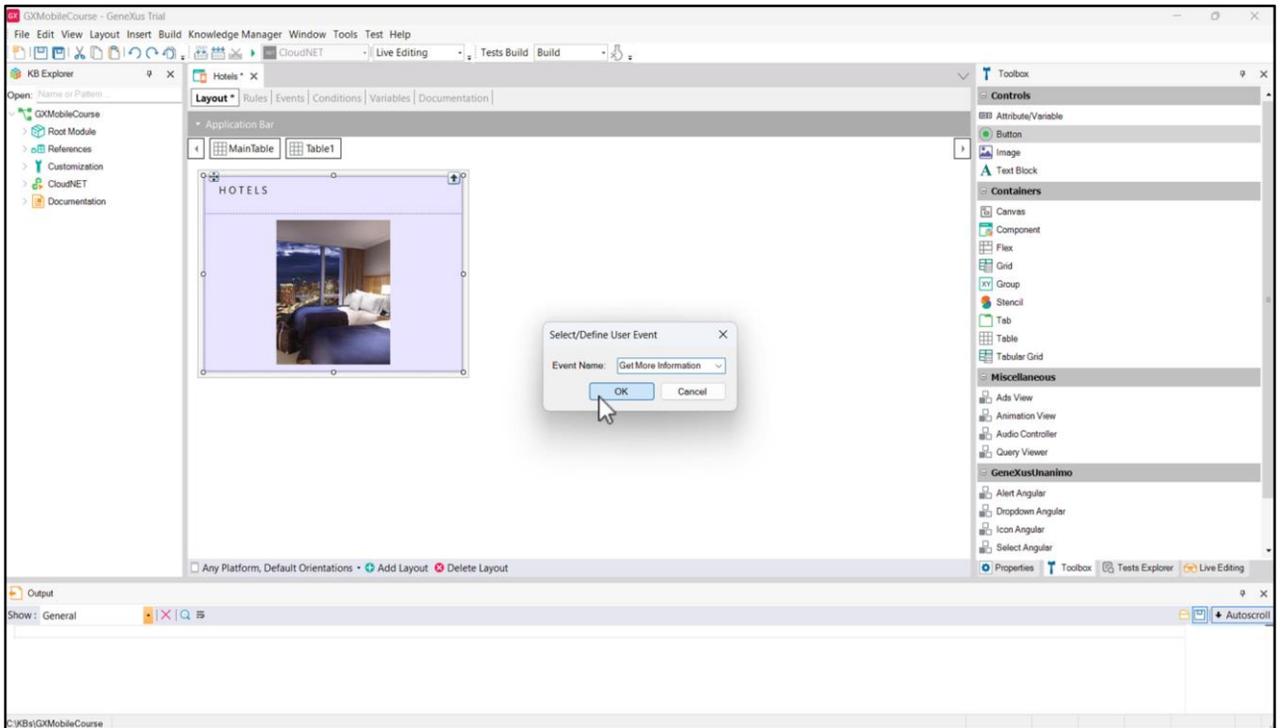
Suponhamos que a agência de viagens nos solicite 3 telas: uma para que sirva de porta de entrada para a informação dos Hotéis, outra para as Atrações e outra para os Pacotes.



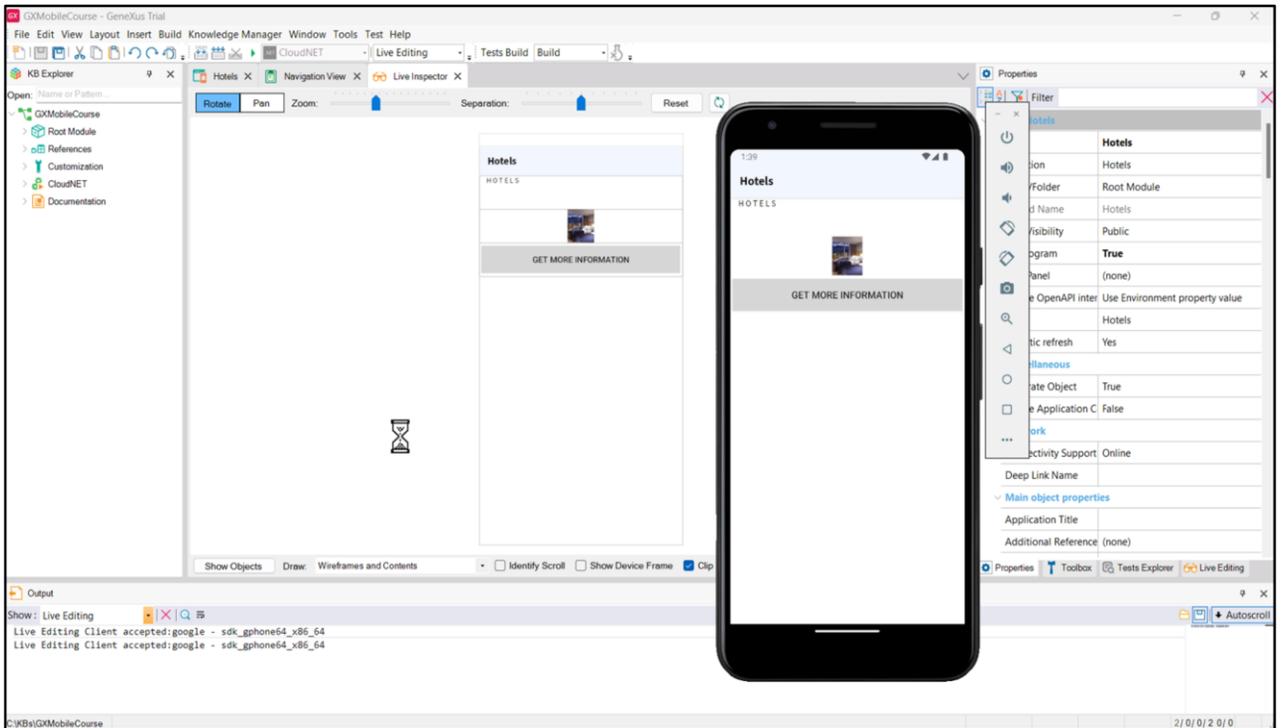
Vamos começar criando o primeiro Panel, que chamaremos de Hotels. Adicionemos uma tabela que usaremos para distribuir a informação e dentro dela arrastamos um Textblock chamado txtTitle, cujo rótulo será *HOTELS*...



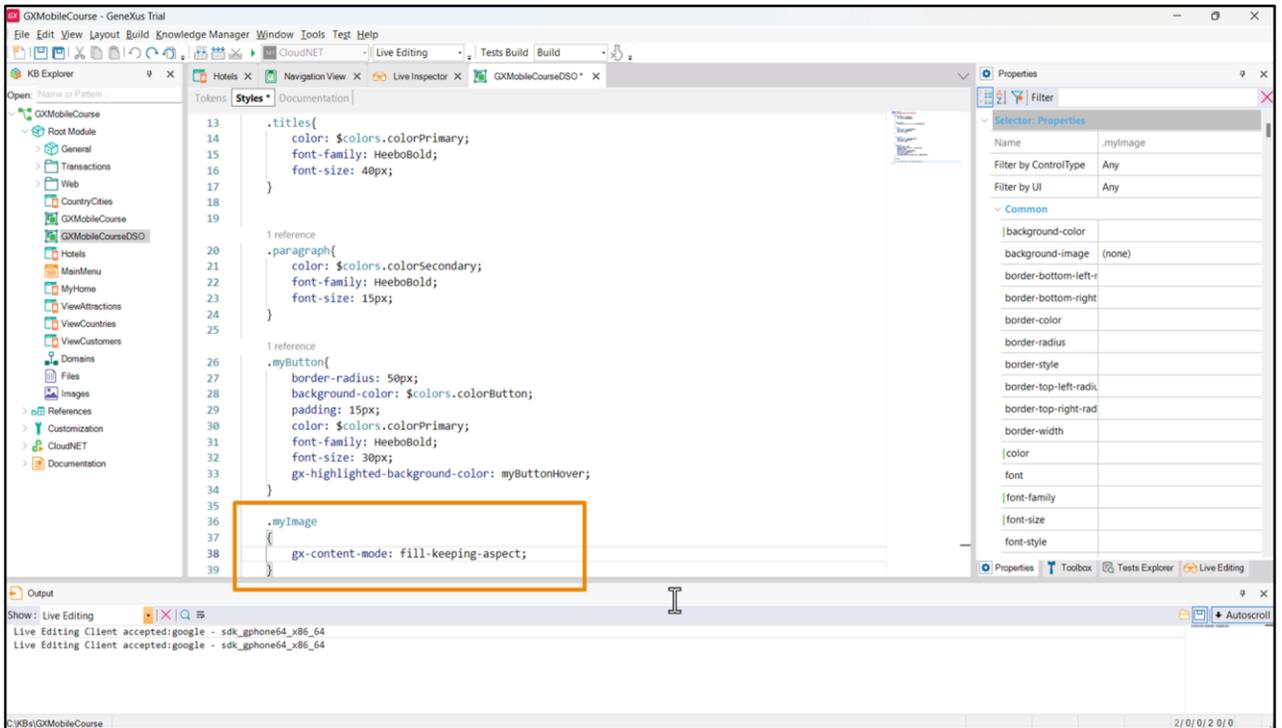
... uma imagem, para a qual carregaremos o arquivo imgHotels...



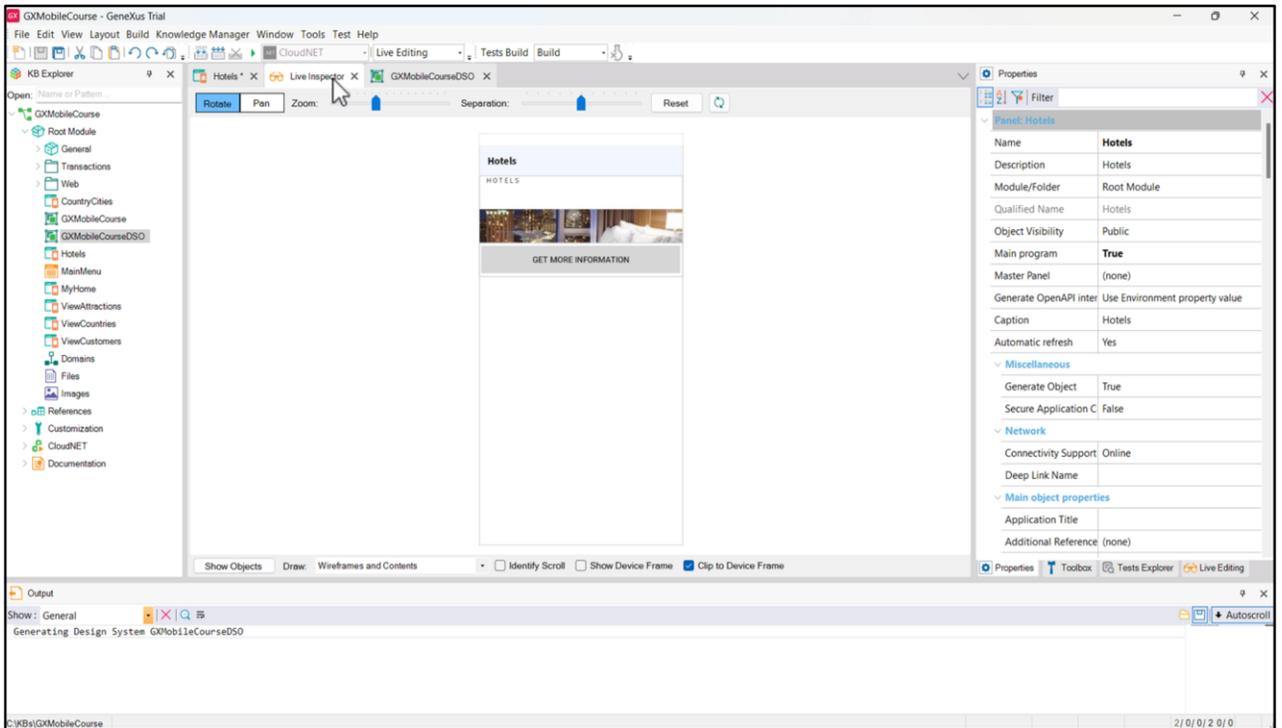
... e um botão, cujo evento associado e rótulo serão *Get More Information*.



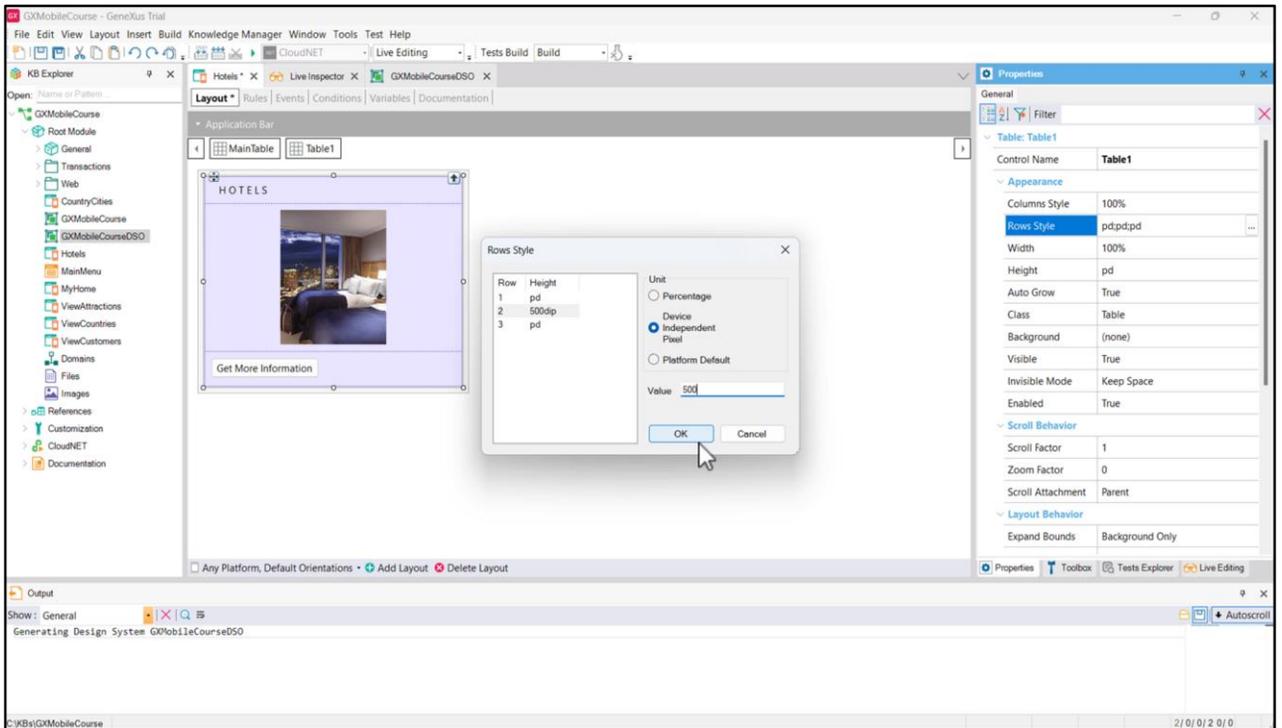
Vemos que obviamente não tem nada de design, então vamos a isso.



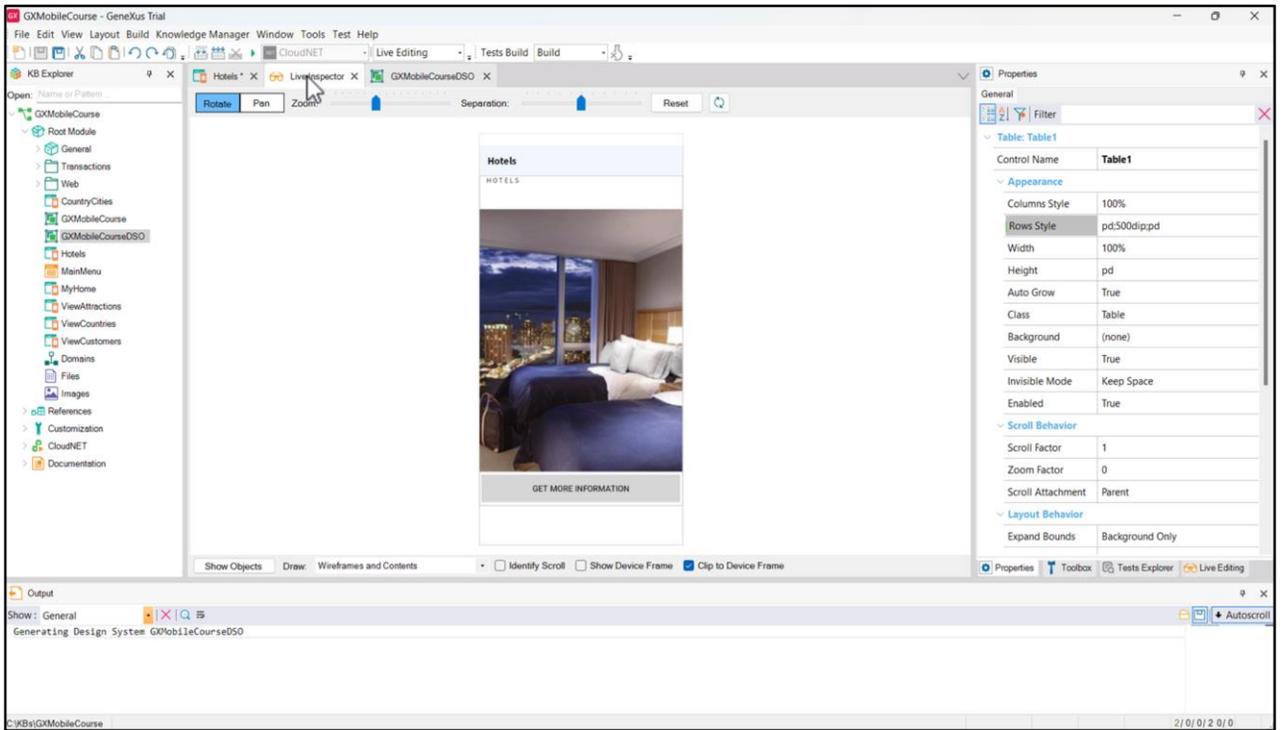
Vamos começar trabalhando com a imagem. Vamos criar uma classe no Design System Object que nos permita alterar seu modo de visualização, dimensionando-a para que ocupe todo o espaço disponível em seu contêiner. Para isso usaremos a `gx-property` `gx-content-mode`, e usaremos o valor `fill-keeping-aspect`;



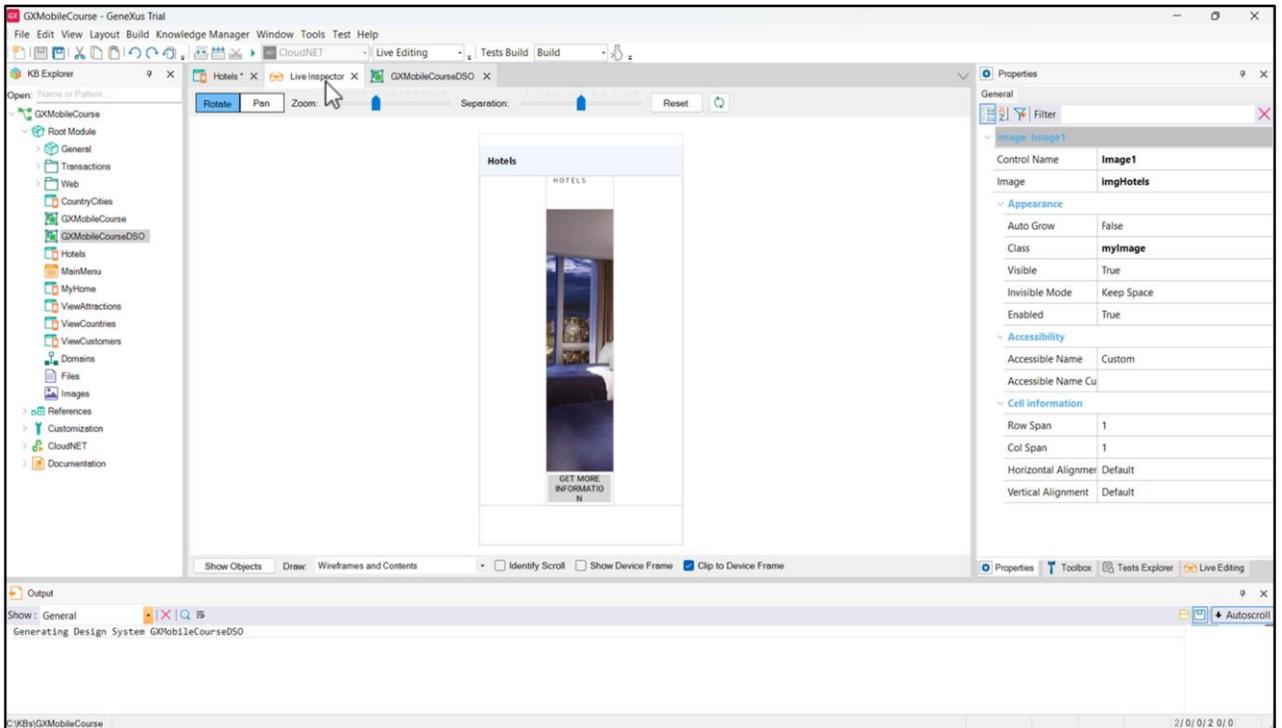
Vemos que a imagem foi dimensionada, mas ficou cortada, pois a largura da linha que a contém é insuficiente para que ela seja exibida inteira.



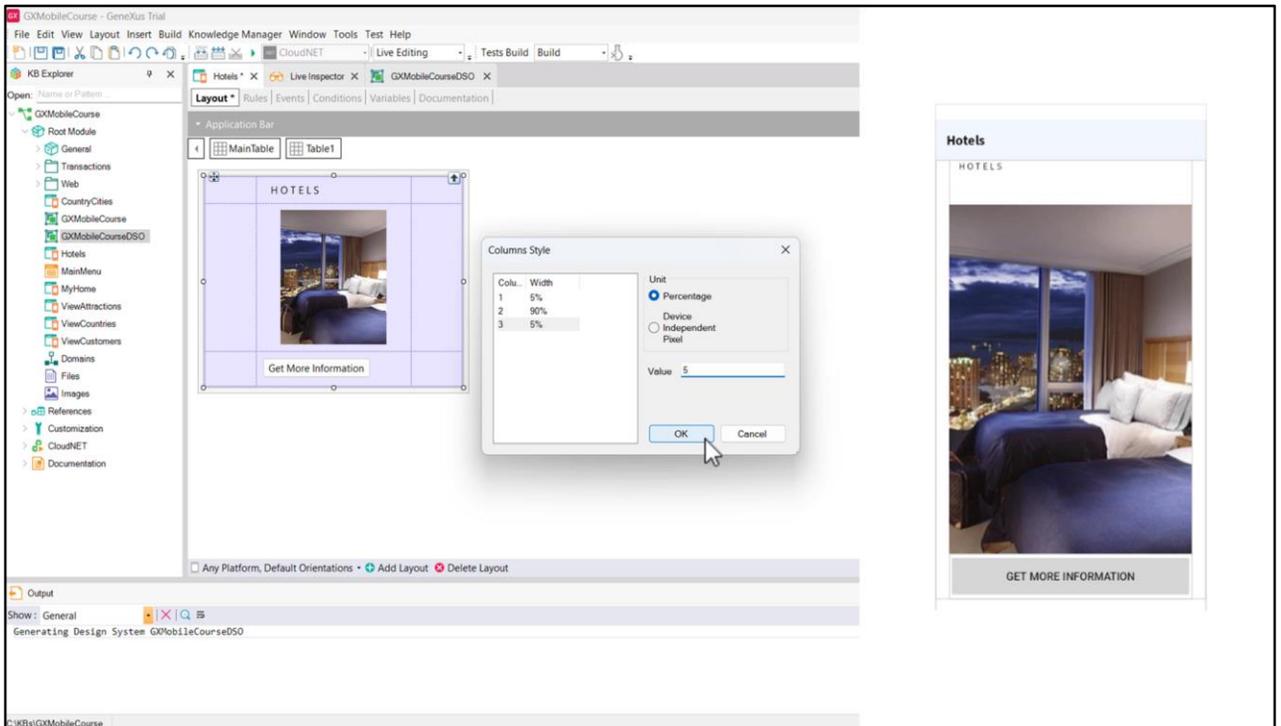
Alteramos o valor da propriedade Rows Style para a segunda linha da tabela para 500 dips.



Agora sim.

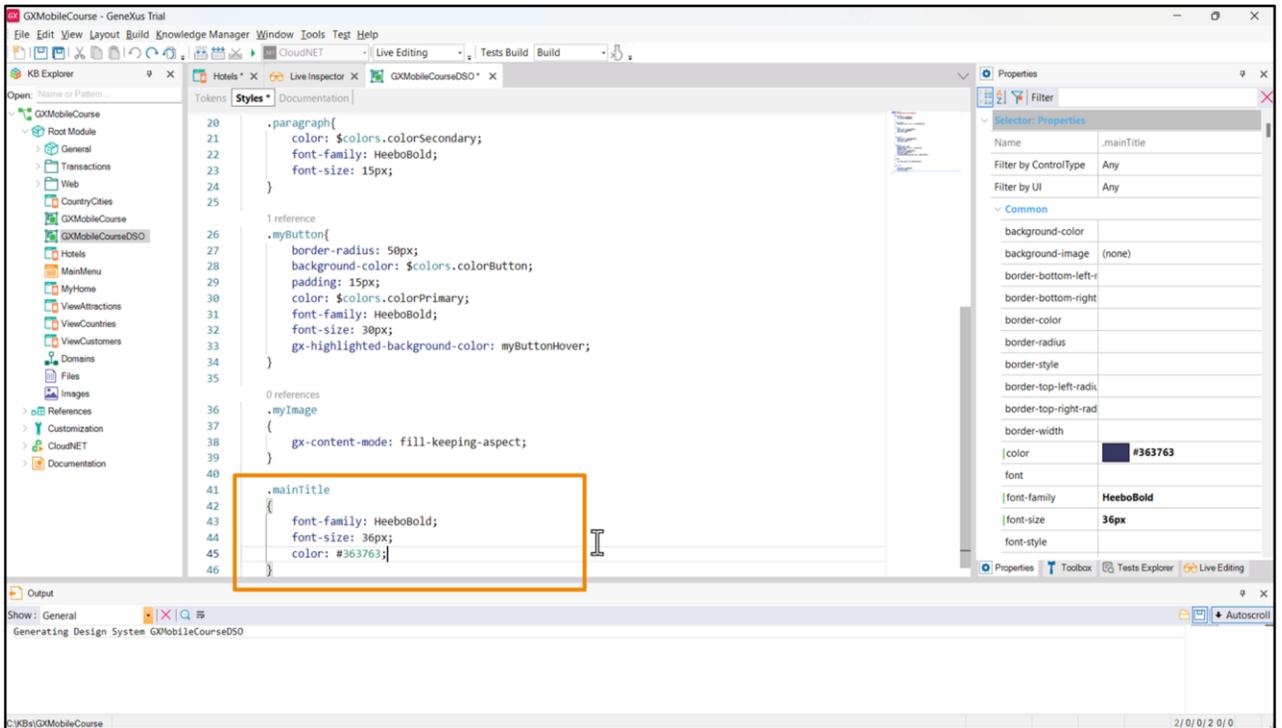


Adicionemos espaços à esquerda e à direita do conteúdo, para que a informação seja melhor apresentada. Para isso vamos até a tabela e adicionamos as duas colunas: vemos que automaticamente a tabela foi dividida em 3 colunas iguais, o que não nos serve, pois só queríamos um espaçamento mínimo para separar o conteúdo das bordas do Panel, mas a coluna do conteúdo deve ser aquela que ocupe o maior espaço.

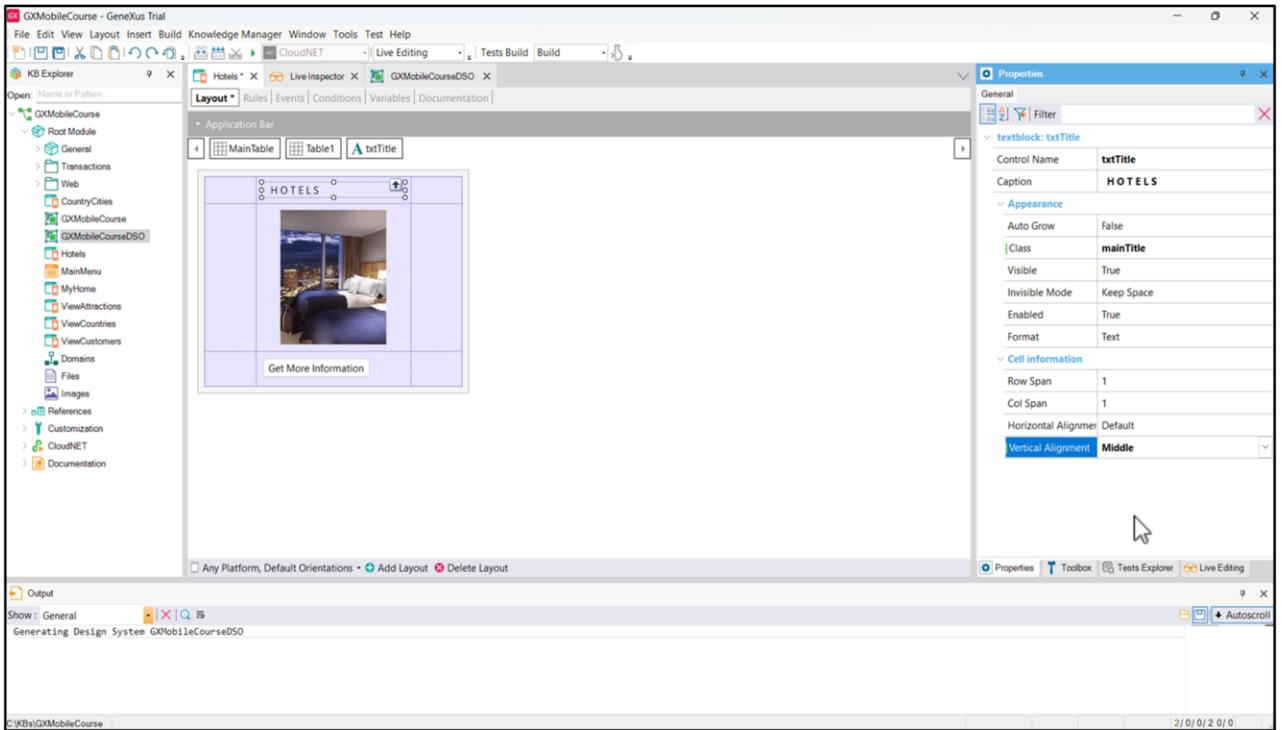


Vamos então configurar a propriedade Columns Style da tabela para que a primeira e a terceira colunas ocupem 5% do espaço total, e a do meio, que é a que tem o conteúdo, 90%.

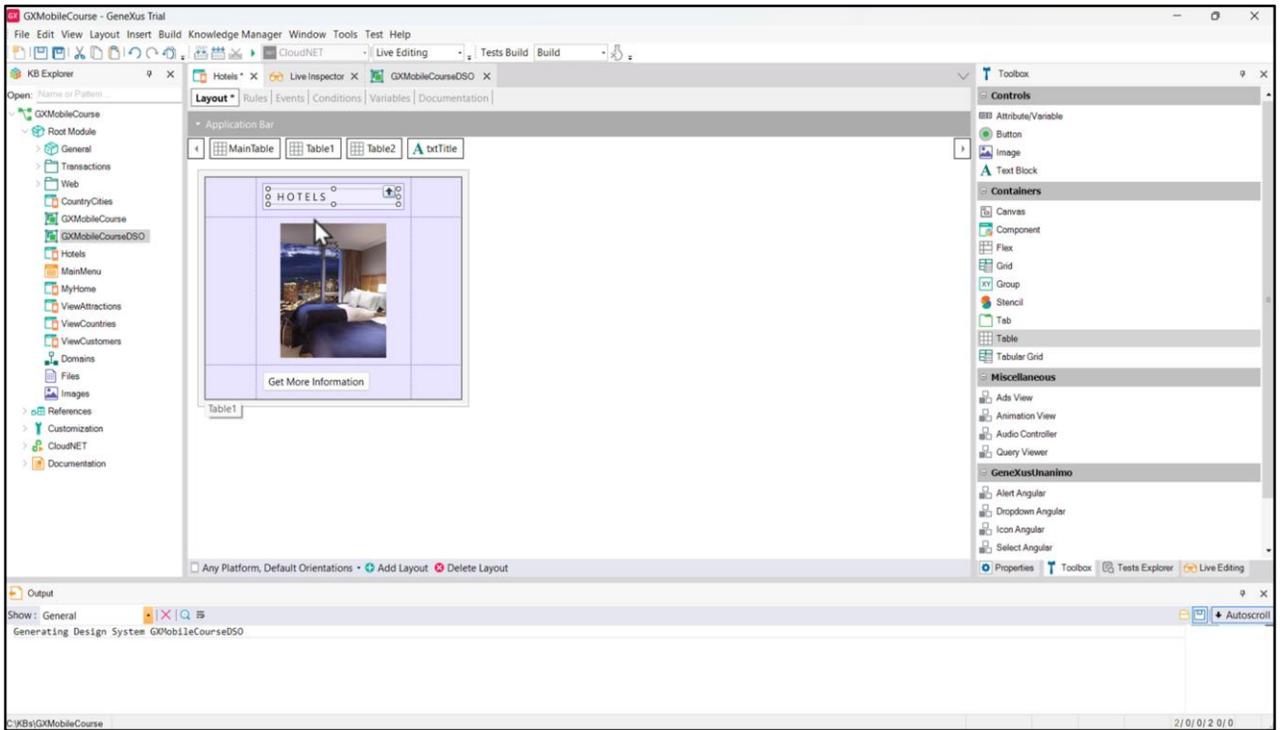
Vemos que nosso Panel está tomando forma.



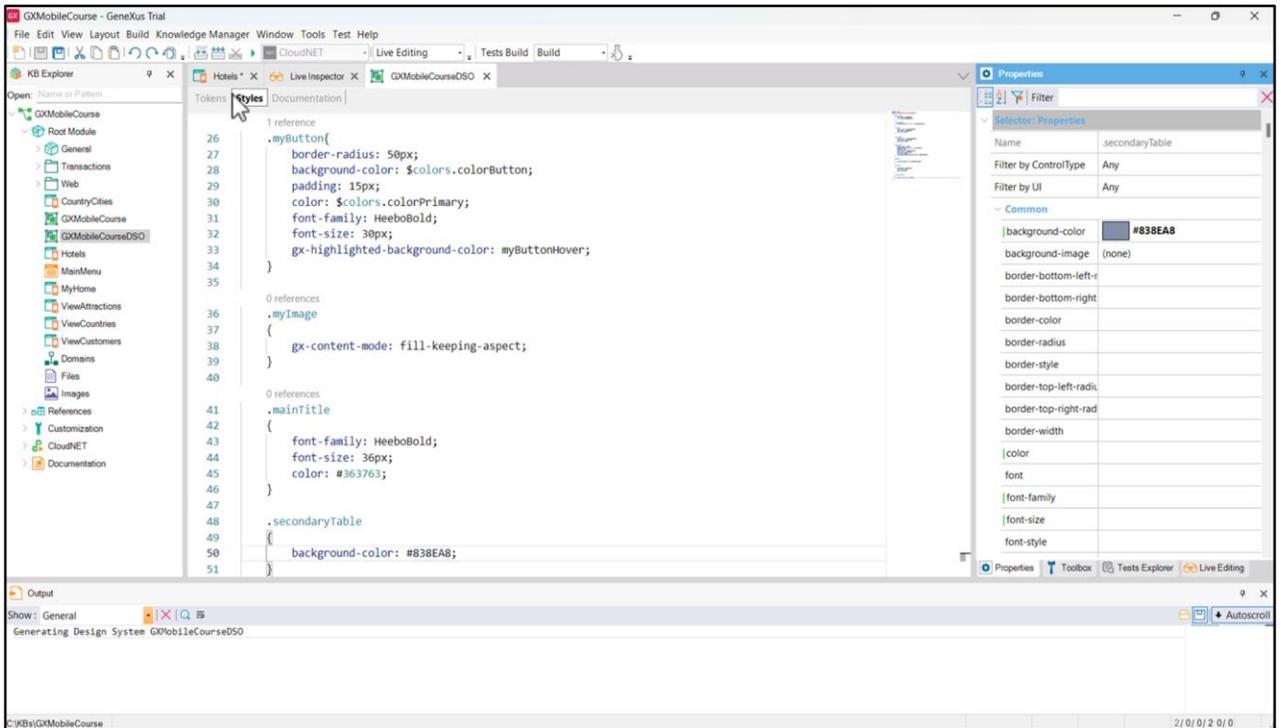
Criamos também uma classe para dar formato ao título: vamos chamá-la de *mainTitle*, ela terá a fonte HeeboBold, tamanho 36px, e esta cor azul escuro (#363763).



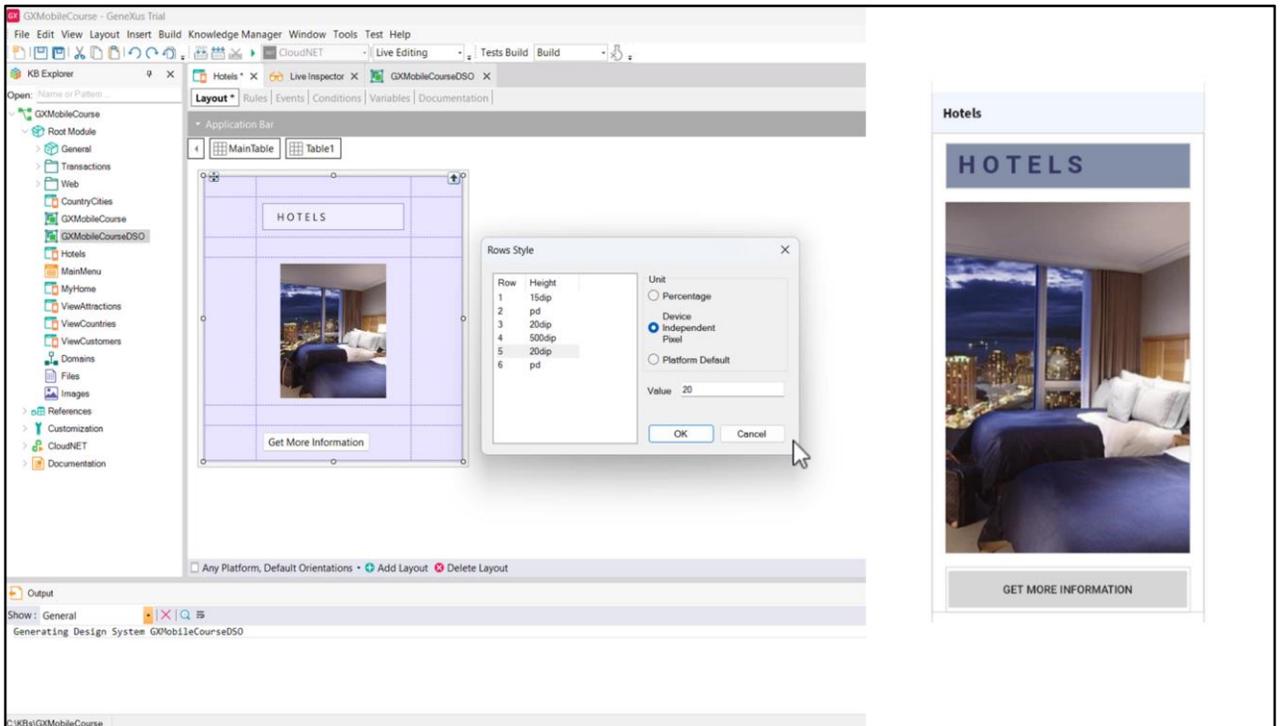
Vamos alterar a propriedade Vertical Alignment do textblock para Middle.



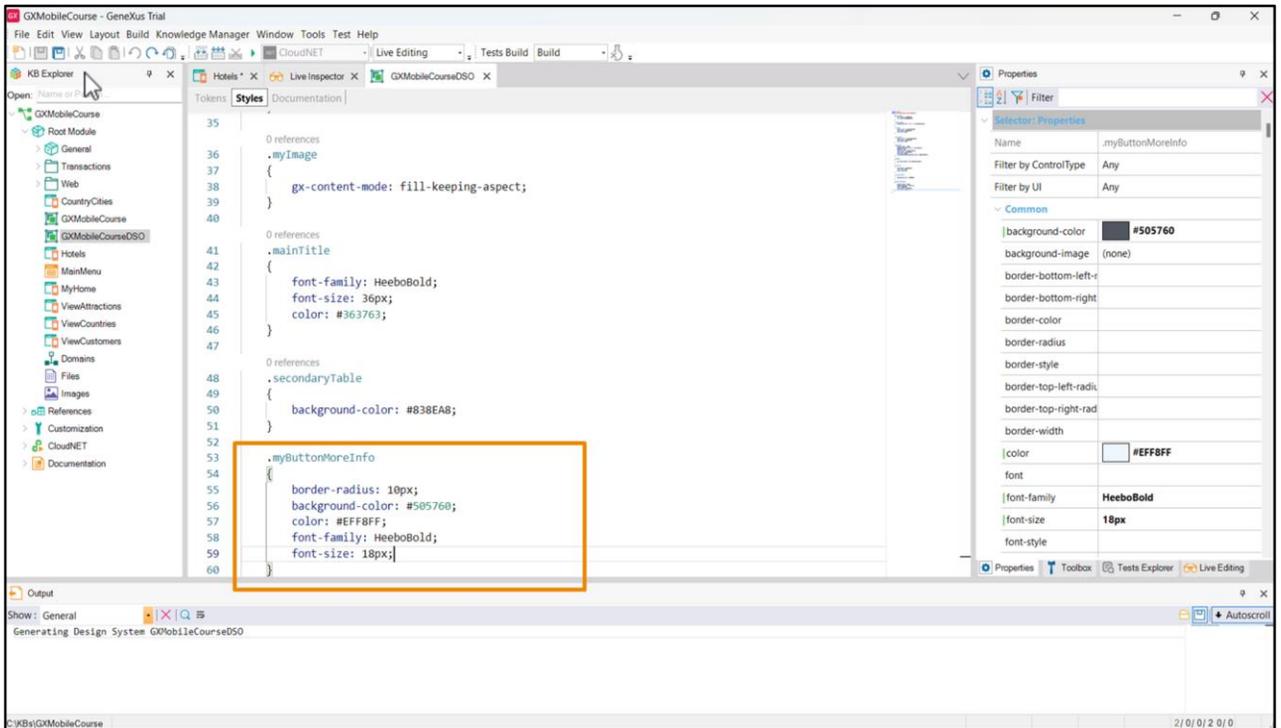
Adicionemos agora outra tabela para onde moveremos nosso controle Textblock...



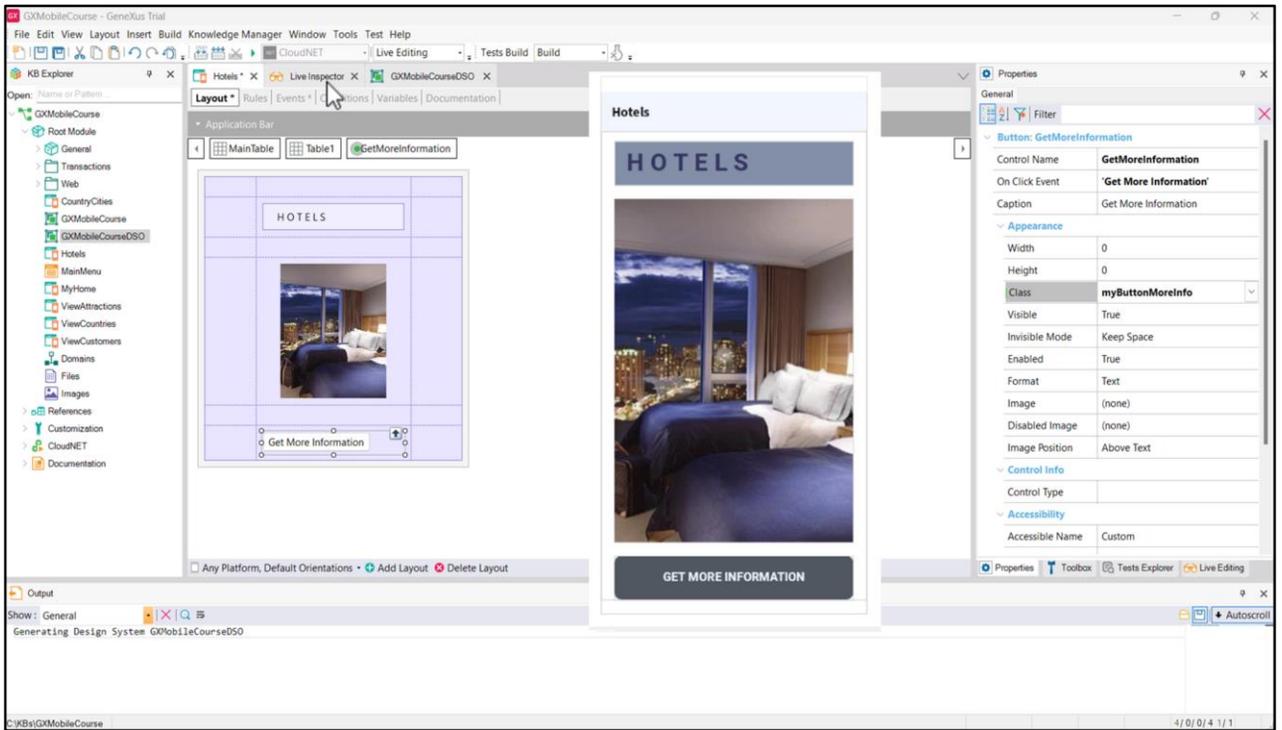
... e criamos uma classe para dar cor ao fundo da célula. Vamos chamá-la de *secondaryTable* e definimos a propriedade `background-color` com o valor hexadecimal correspondente a este azul (#838EA8).

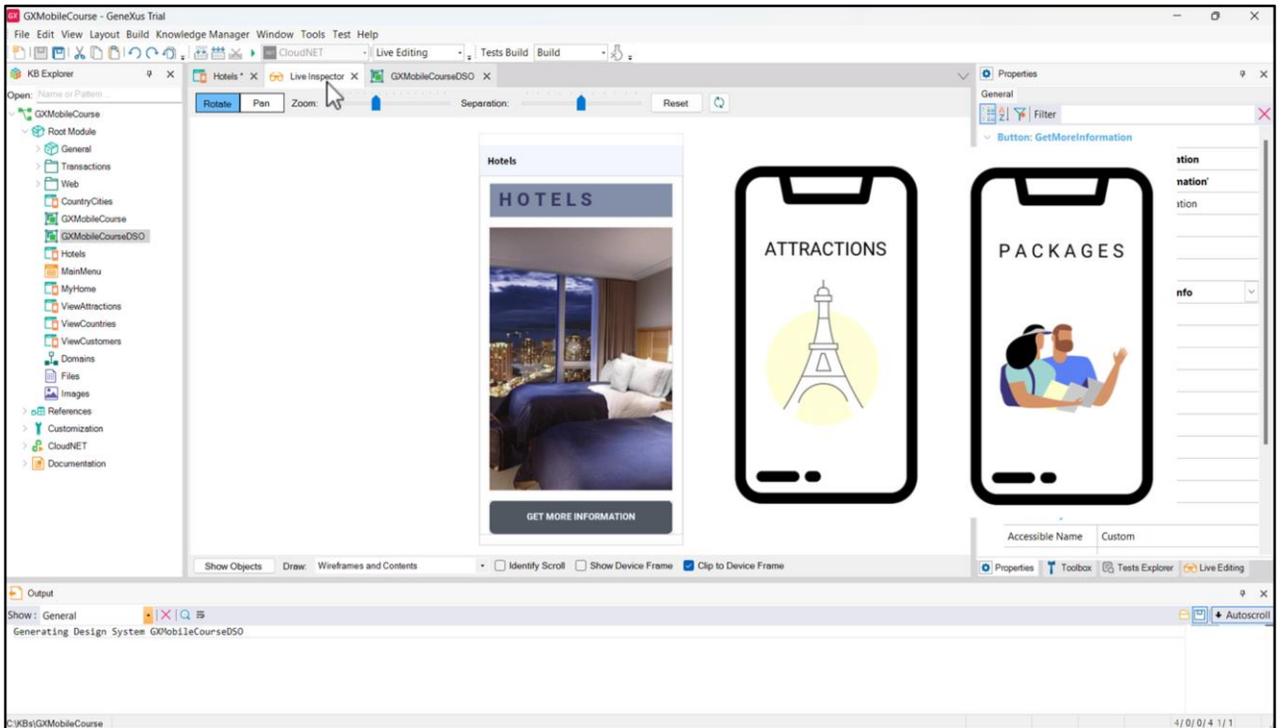


Agora, vamos adicionar mais 3 linhas à tabela para separar nossos elementos: uma antes do título, outra antes da imagem e outra antes do botão. Vamos ajustar a propriedade Rows Style para que a primeira linha ocupe apenas 15 dips, a terceira um pouquinho mais, 20 dips, e a quinta também 20 dips.



Vamos trabalhar com o botão. Criamos a classe `myButtonMoreInfo`, e adicionamos as propriedades para definir seu estilo: `border-radius` com 10px para arredondar minimamente suas bordas, `background-color` com esse cinza escuro (#505760) para definir a cor de fundo, `color` para definir a cor da letra (#EFF8FF) com este azul bem claro, quase branco, `font-family` para definir sua fonte HeeboBold e, por último `font-size` para que a fonte ocupe 18px.



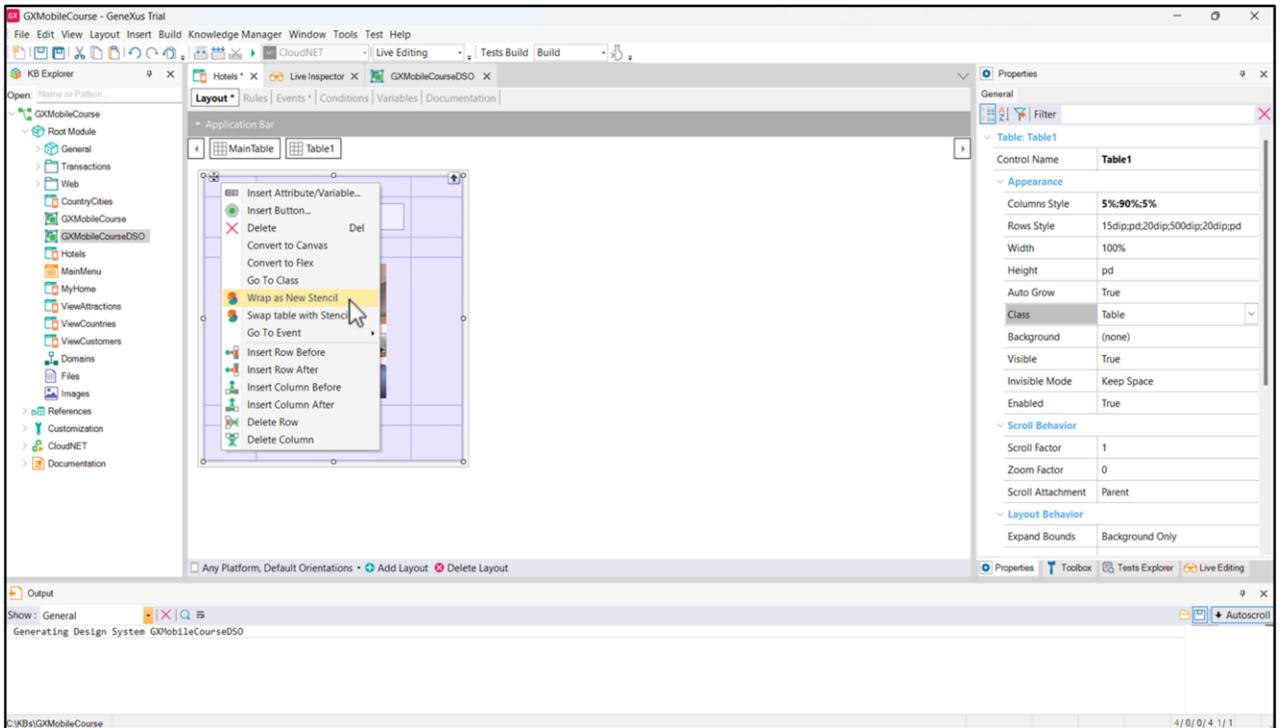


Suponhamos que estejamos satisfeitos com o design até aqui.

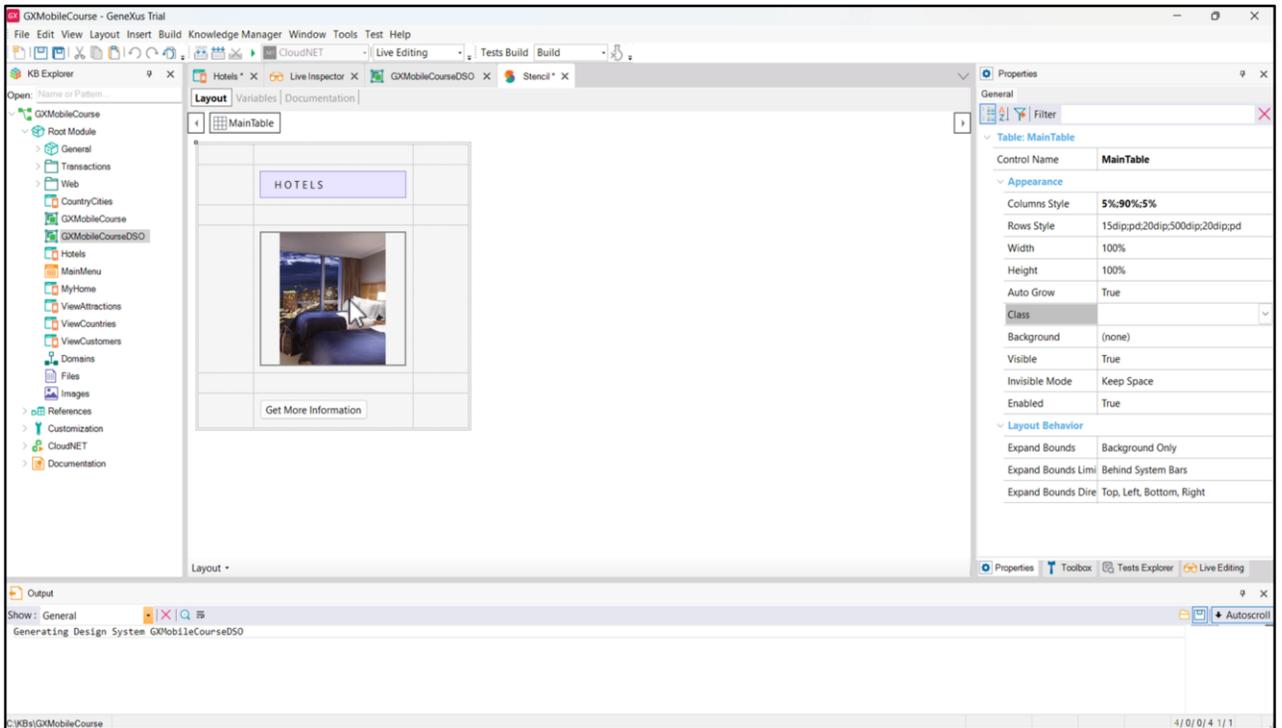
Lembremos que a agência de viagens também nos pediu uma tela para ver as atrações e outra para ver os pacotes.

É necessário que criemos cada Panel e façamos novamente todas essas configurações em cada um deles? Isso não faria sentido. Poderíamos pensar em fazer um Save as deste Panel para depois alterar o que corresponde. Mas o que aconteceria se fosse necessário alterar algo no design depois de tê-lo implementado? Deveríamos fazer a(s) alteração(es) em cada Panel. O uso de Stencils nos evita tudo isso. Podemos simplesmente “empacotar” a tabela principal que criamos como um Stencil e depois usar esse Stencil em cada Panel.

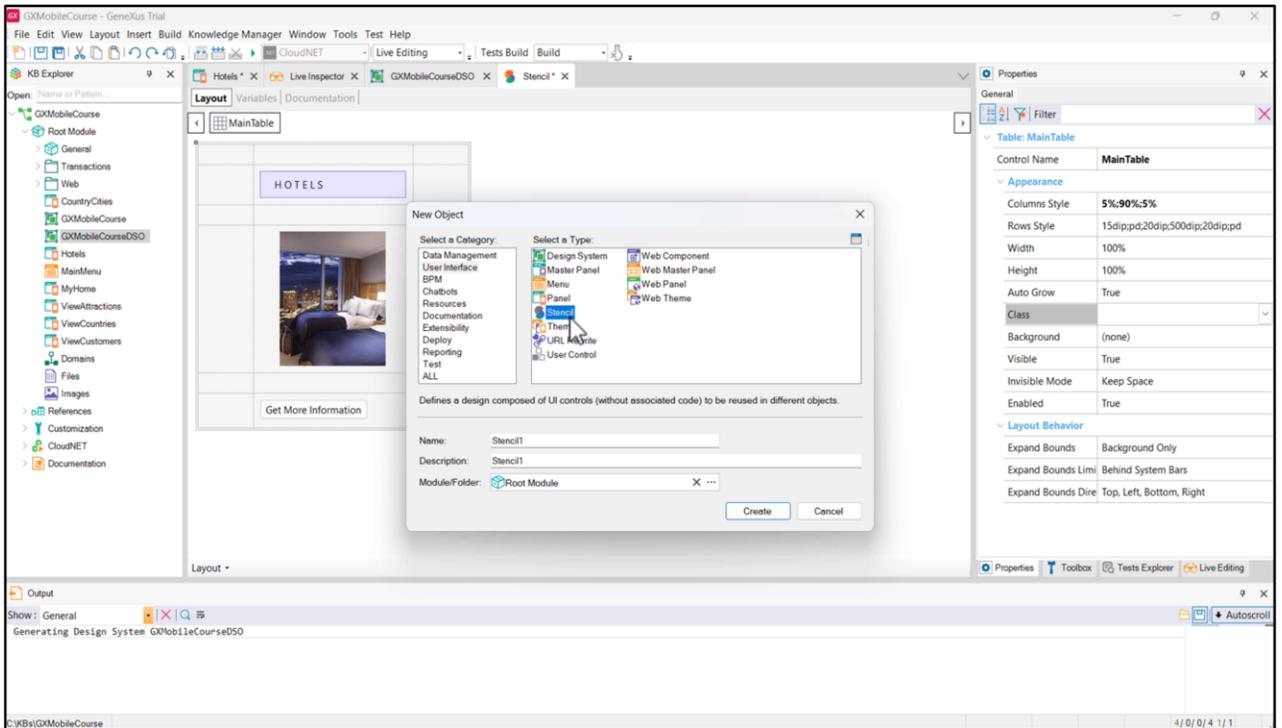
Vamos fazê-lo.



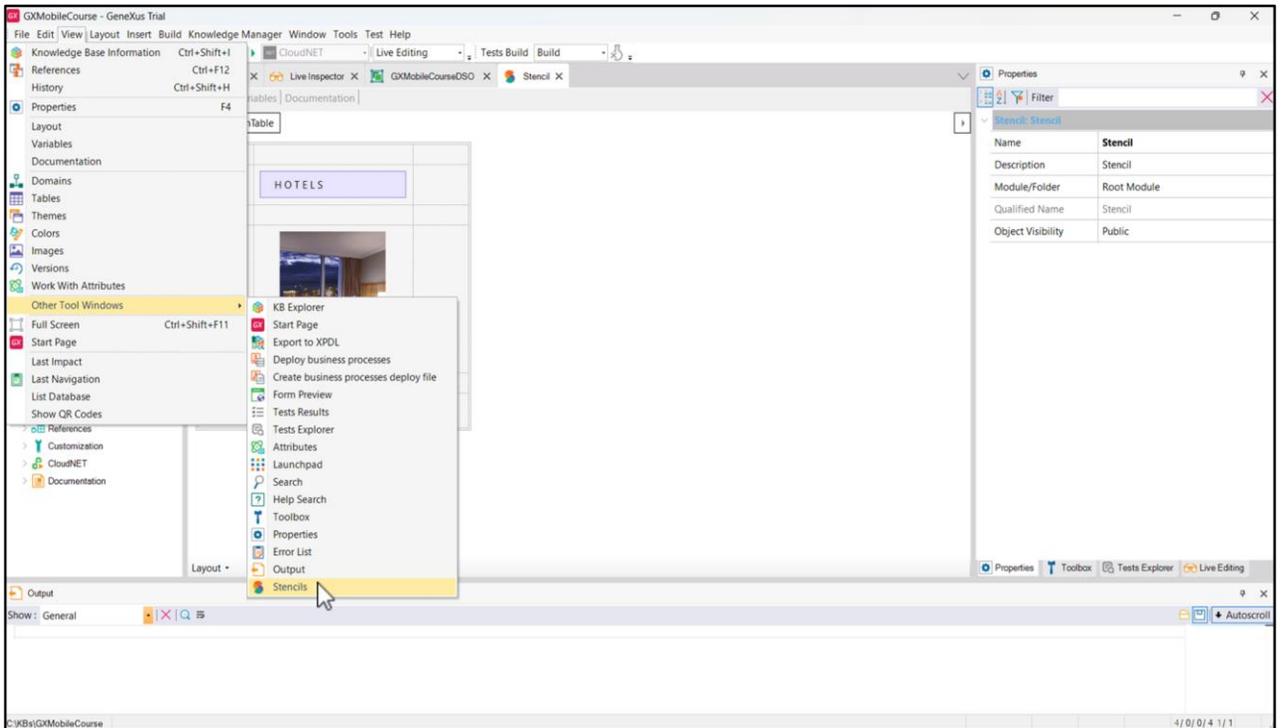
Se clicarmos com o botão direito sobre a tabela, veremos uma opção chamada *Wrap as New Stencil*.



É aberta a caixa de diálogo para inserir um nome e ao pressionar o botão Create, o Stencil será criado como um novo objeto, e a partir daqui poderemos fazer qualquer alteração em suas propriedades. Uma vez salvas, essas alterações serão refletidas no objeto que contém o Stencil (até agora apenas o Panel Hotels).

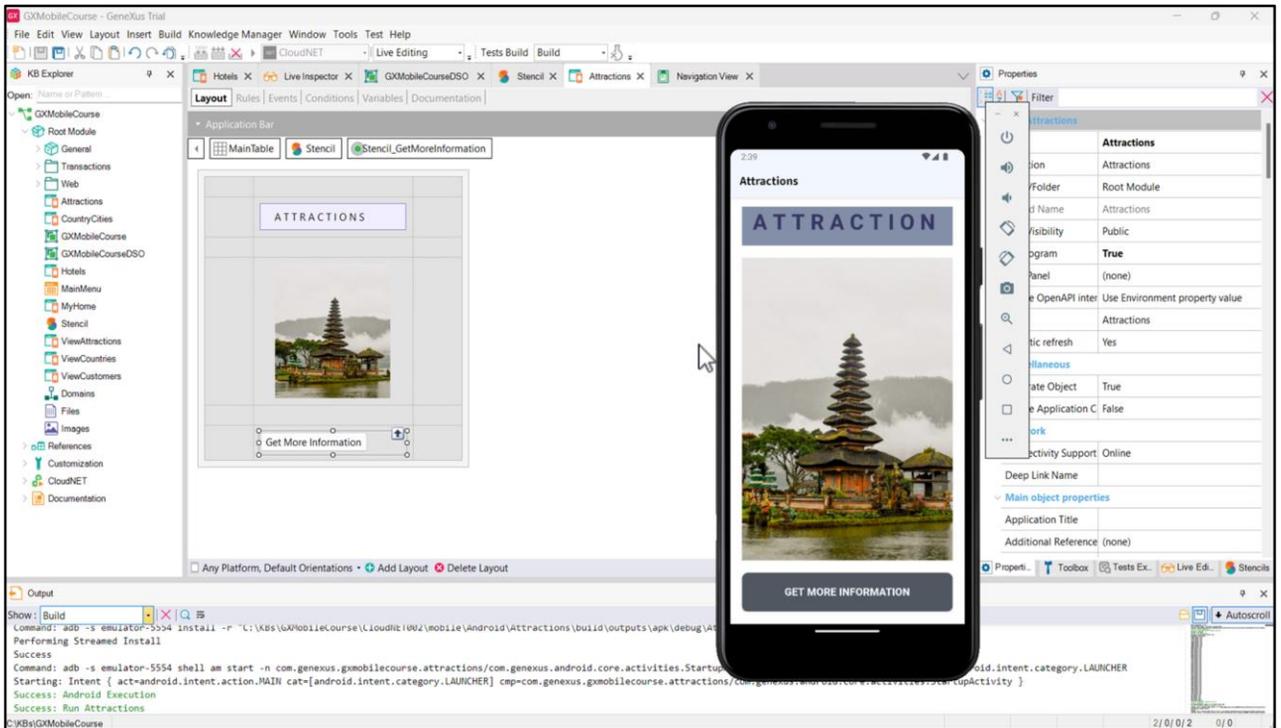


Outra possibilidade para criar um objeto Stencil é selecionando diretamente no Menu principal de GeneXus: File > New > Object.



Uma vez criado, para incluí-lo em algum objeto só precisamos arrastá-lo a partir do KB Explorer ou a partir da janela de Stencils, que podemos acessar a partir do menu View > Other Tool Windows > Stencils.

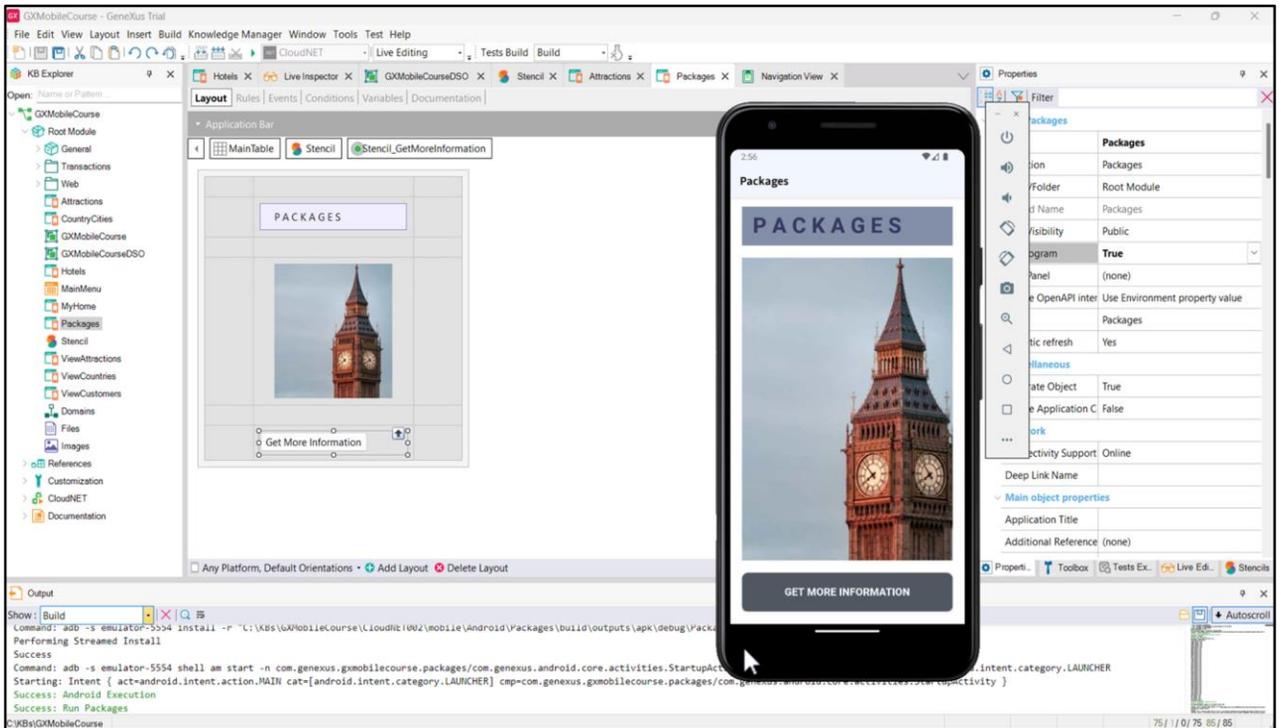
Se for criado um Stencil a partir de um Panel que contém atributos, ao salvar o Stencil os atributos são convertidos em variáveis. Isso ocorre porque um Stencil não tem navegação nem lógica definida.



Vamos criar, agora sim, o Panel Attractions, e vamos usar o Stencil. Vamos arrastá-lo a partir do KB Explorer.

Se clicarmos na tabela principal, vemos que já não temos as propriedades de configuração que tínhamos antes. Isso ocorre porque agora todas as propriedades relacionadas ao seu design são acessíveis somente a partir do objeto Stencil.

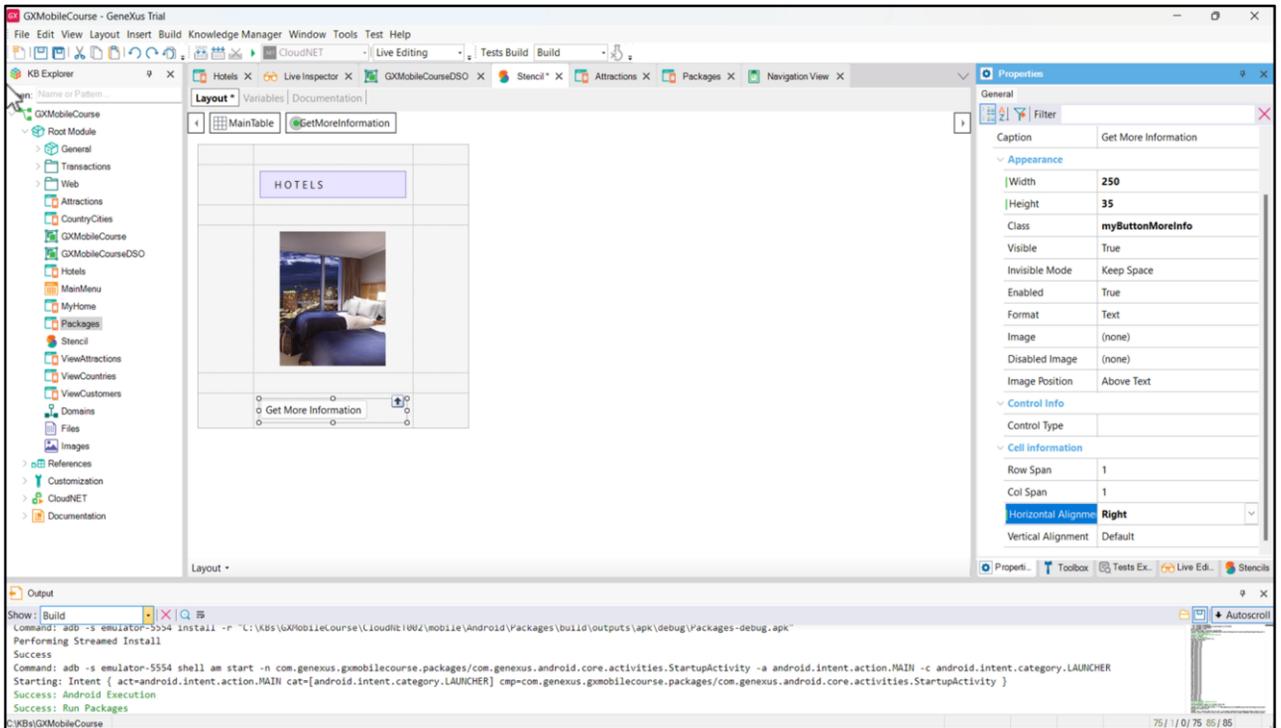
Vamos mudar o rótulo do título para *A T T R A C T I O N S*, e a imagem para a chamada *imgAttractions*. Como o botão deve ter sempre um evento associado, vamos clicar duas vezes nele para que seja criado.



Agora vamos criar o Panel Packages, e adicionar o Stencil a partir da janela Stencils, usando a opção *INSERT*. A opção *OPEN* abre o objeto Stencil em questão, e a opção *SWAP* serve para alterar uma tabela que tenha o mesmo conjunto de controles do Stencil para o Stencil.

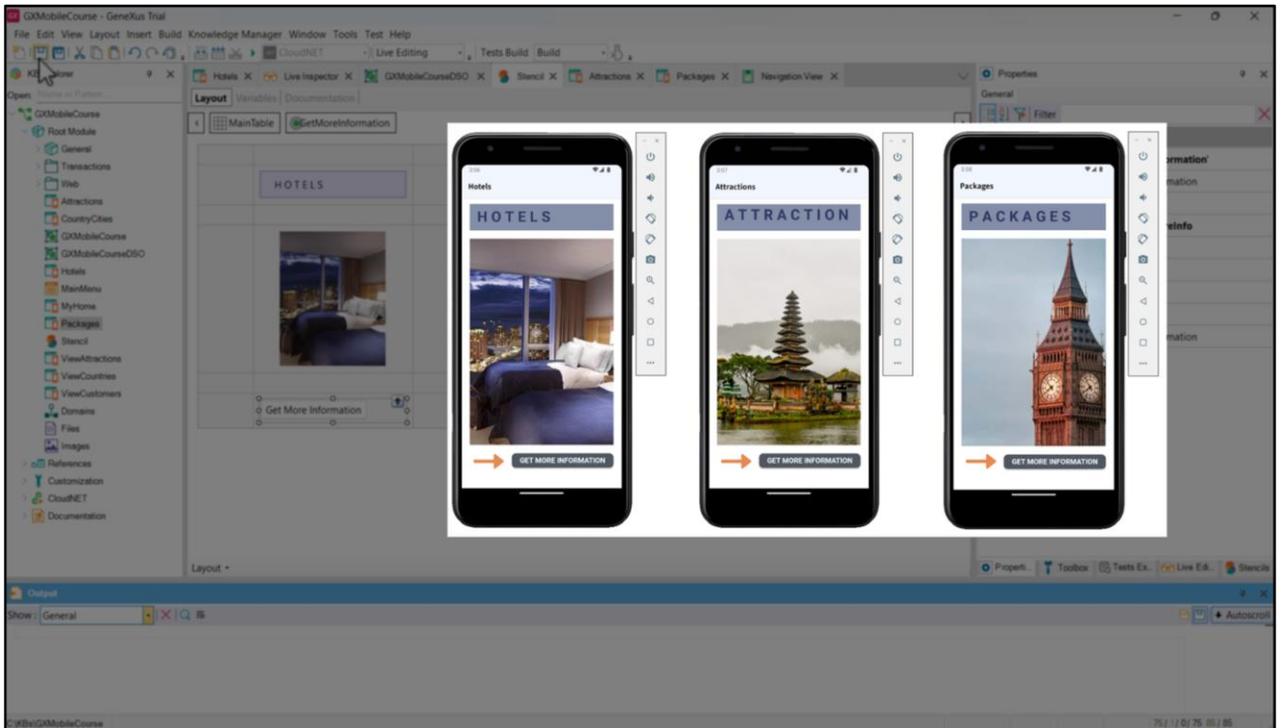
A qualquer momento podemos deixar de utilizá-lo, usando a opção *Unwrap Stencil*.

Mudaremos o rótulo do título para *P A C K A G E S*, e a imagem para a chamada *imgPackages*.



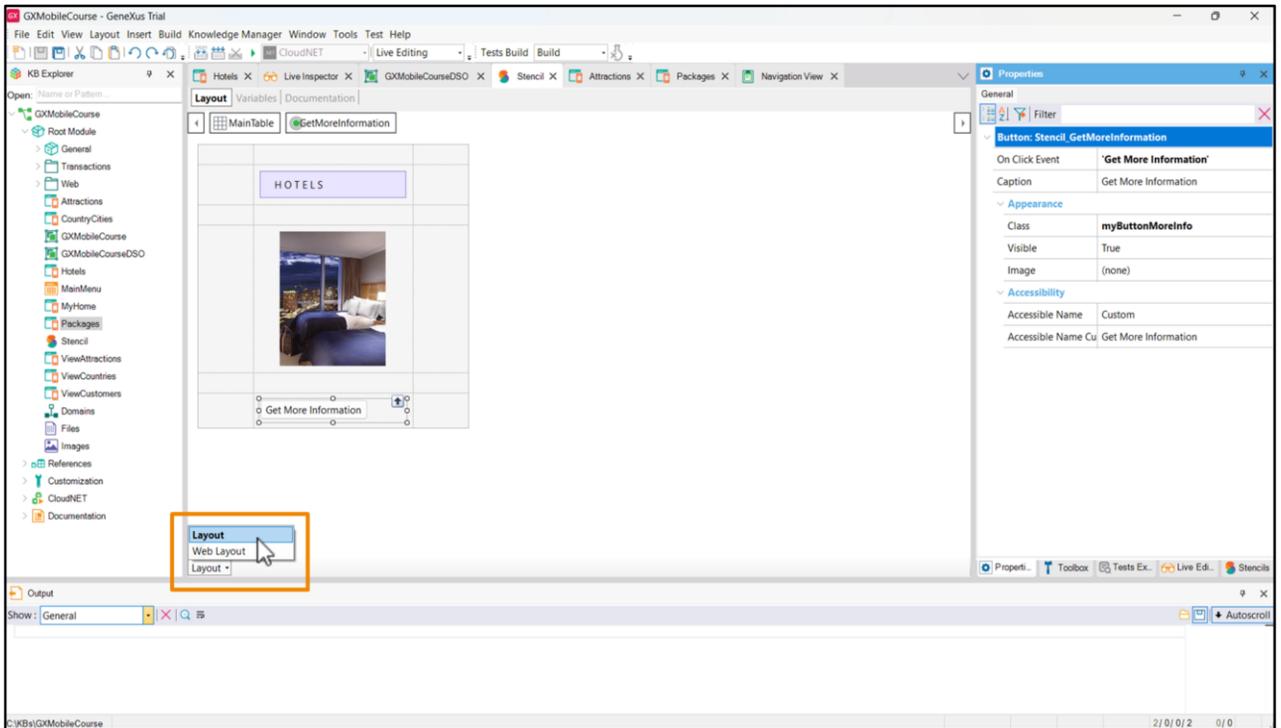
Agora, suponhamos que o cliente não gostou do nosso botão e nos pede para que seja menor e esteja alinhado à direita.

Como já explicamos, as alterações de aparência realizadas em um objeto Stencil são aplicadas aos objetos que o contêm, então voltamos ao Stencil e alteramos a propriedade Width do botão para 250, e a propriedade Height para 35. Além disso, modificamos o alinhamento horizontal para colocá-lo à direita.



Vemos que todos os Panels adotaram a mudança no botão.

Em resumo, se os controles podem ser pensados como as unidades mínimas que compõem a UI, e suas classes permitem centralizar seu design, os Stencils criam unidades de design maiores, compostas por controles, para definir seu design (sem código associado), de forma a ser reutilizado em diferentes objetos.

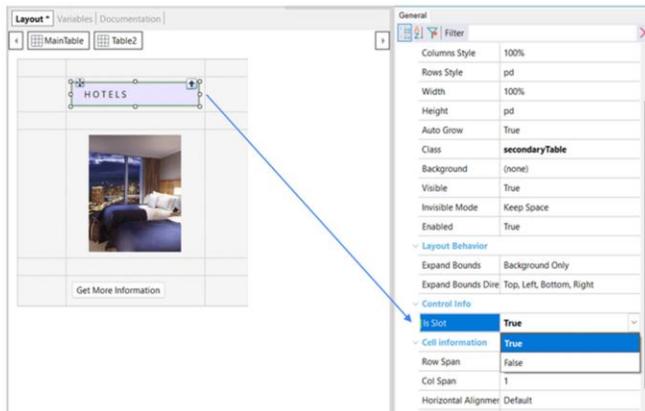


Ao desenhar uma aplicação tanto para web como para dispositivos móveis, existem alguns elementos visuais em ambas as plataformas que são conceitualmente iguais, mesmo que devam ser desenhados de forma diferente.

Para levar isso em consideração, os Stencils contêm dois designs:

- o Web Layout: para ser utilizado em objetos que contenham um Web Layout, como Web Panels, Web Components, etc.
- e o Layout: para ser utilizado em objetos que contenham um Layout, como Panels e Components.

## Slots in Stencils



<https://wiki.genexus.com/commwiki/wiki?51385>

<https://wiki.genexus.com/commwiki/wiki?38418>

A partir do Upgrade 11 de GeneXus 17, é possível configurar um contêiner como Slot.

Isso significa que é possível editar e personalizar essa parte do Stencil em cada objeto onde ele está incluído, configurando sua propriedade Is Slot como True.

Para obter mais informação sobre este tema, consulte o artigo da Wiki de Slots em Stencils:

<https://wiki.genexus.com/commwiki/wiki?51385>

Você também pode ver outros exemplos e cenários de uso de Stencils neste outro artigo:

<https://wiki.genexus.com/commwiki/wiki?38418>

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

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