

# Demo: First steps



Cecilia Fernández

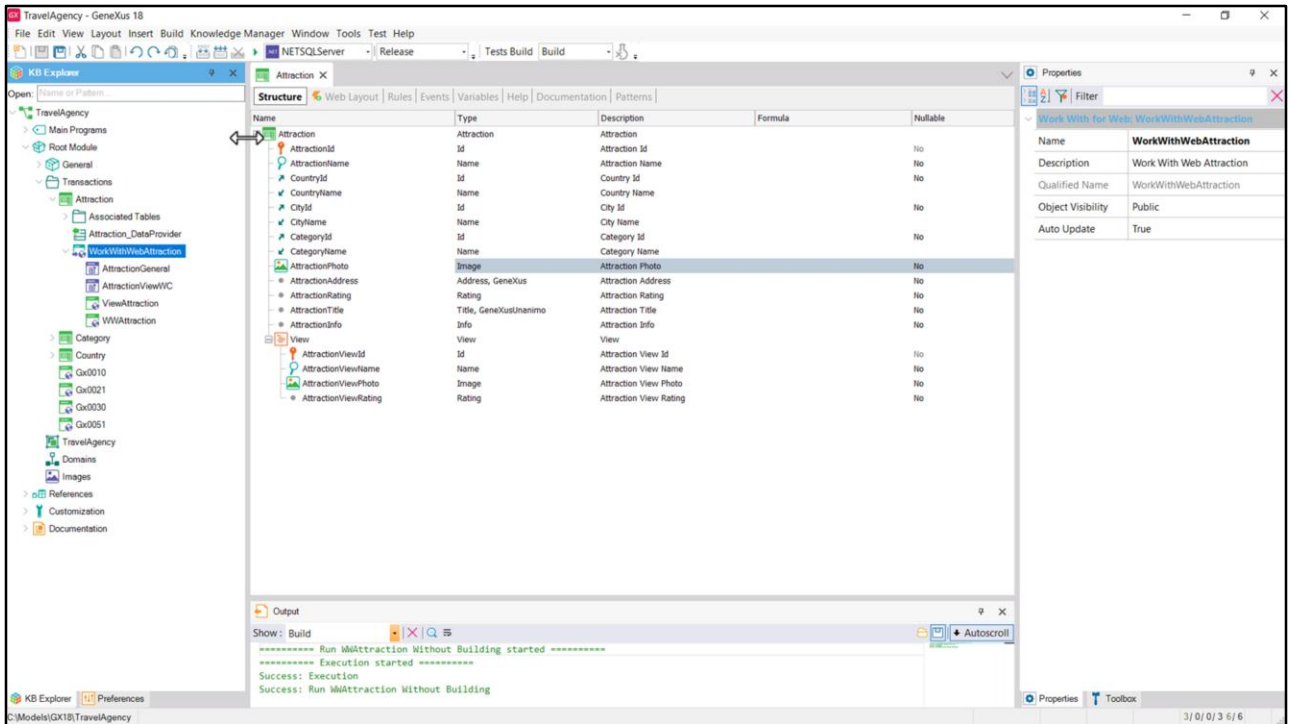
Olá, como estão?



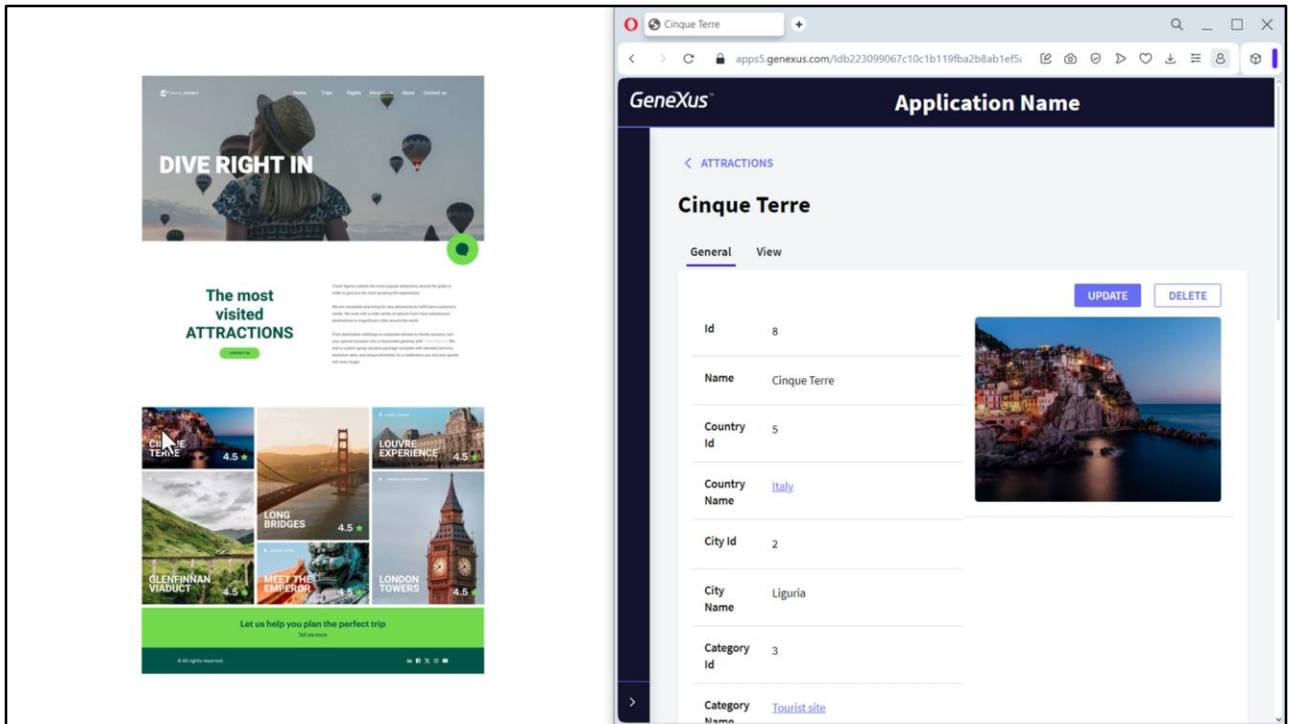
Nesta aula começaremos a trabalhar na KB em GeneXus para poder obter, algumas aulas mais adiante, a aplicação web que implementa essas 4 telas de que falávamos, para tamanho desktop.

Depois disso veremos como variar a aplicação de acordo com o tamanho de tela, para que possa ser executada e ser exibida perfeitamente em dispositivos do tamanho de um telefone ou do tamanho de um Tablet, sempre falando da aplicação web. E então, mais adiante ainda, no final do curso veremos como fazer com que a própria aplicação seja executada de forma nativa nesses dispositivos.

Bem, então vamos começar indo para GeneXus...

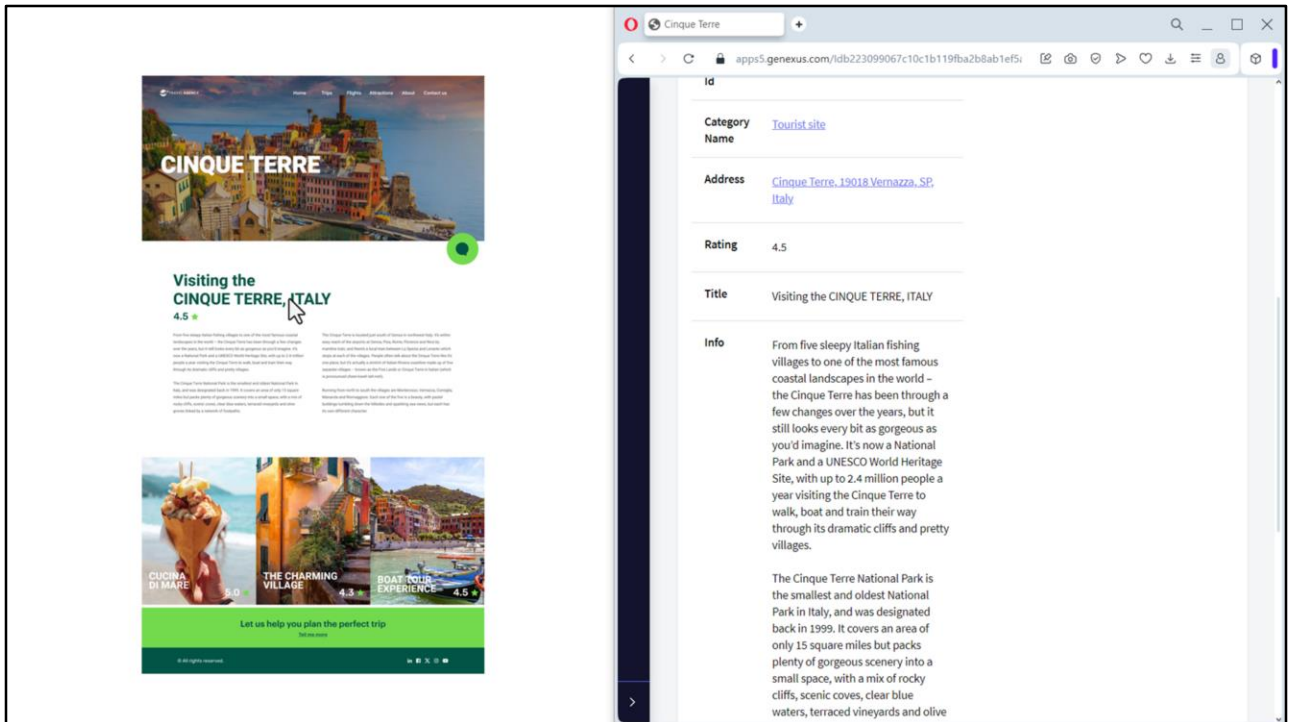


....e observar que criei uma KB chamada TravelAgency, na qual dentro de uma pasta, Transactions, criei as três transações que de forma simplificada registrarão os dados no Backoffice. Preenchi cada uma delas com dados a partir de um data provider e apliquei o padrão Work With Web.

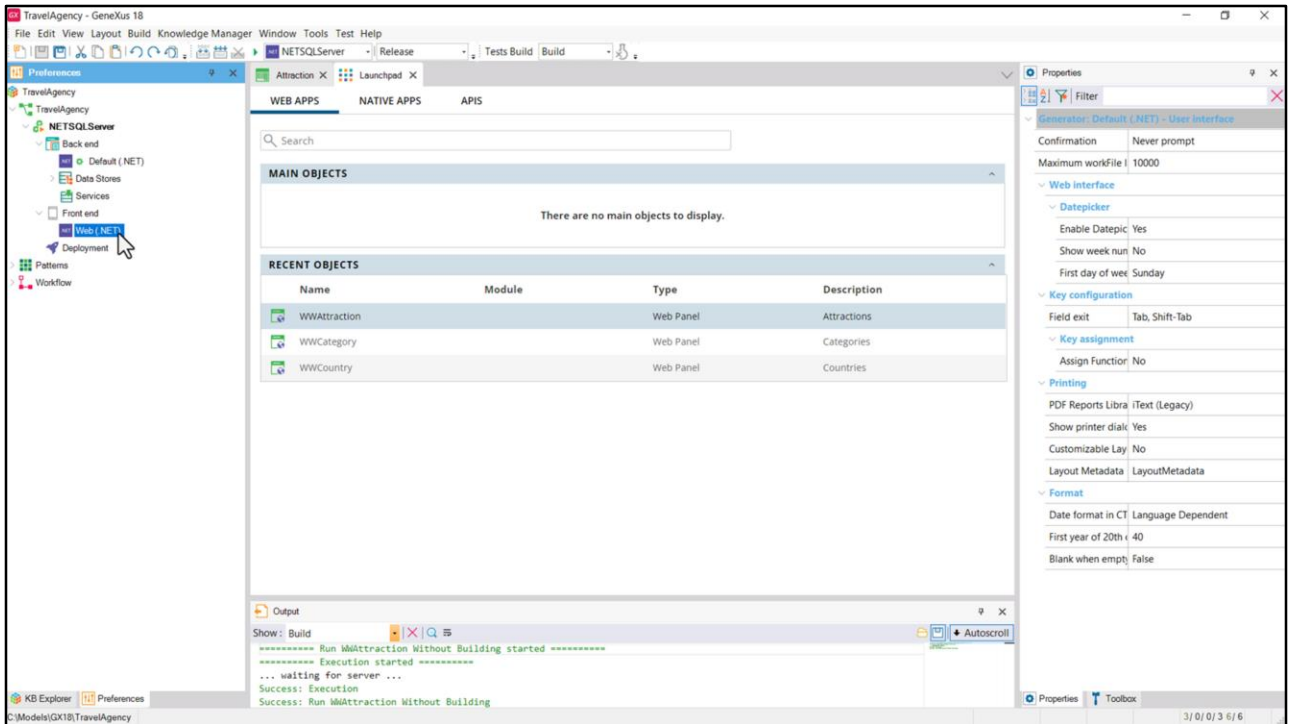


Assim, se eu executar... vemos o Work with das atrações turísticas.

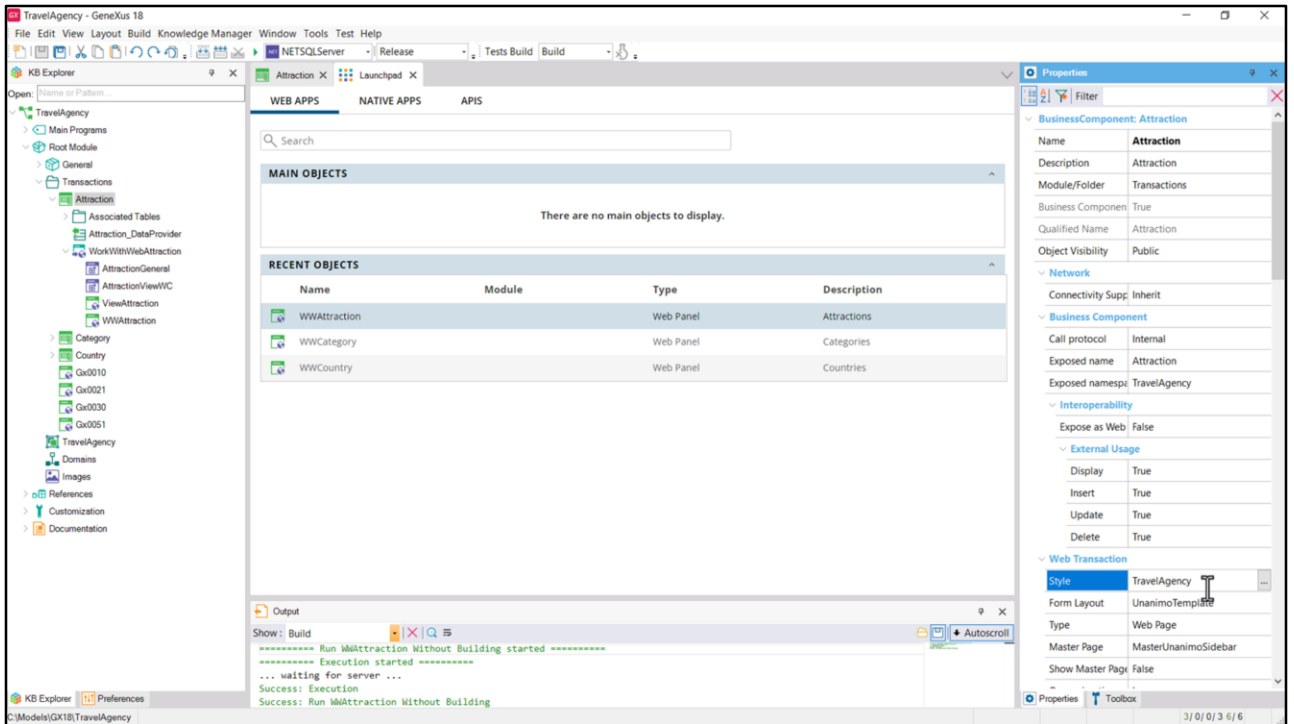
E se eu escolher esta, vemos aqui a foto que será utilizada para a aplicação Customer-facing (aquela que estamos querendo implementar), o nome da atração, o rating...



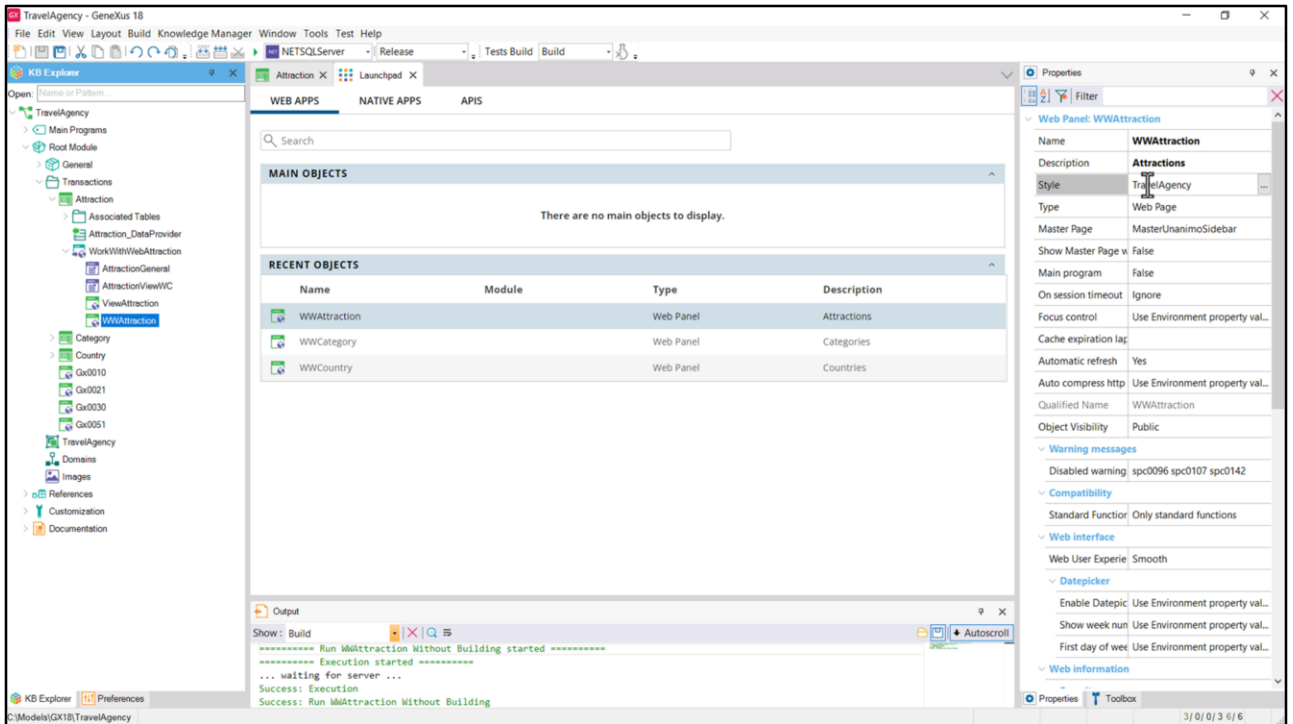
...o título que será visto na página da atração, a informação que será listada nessa página, e se formos na aba View, aqui vemos essa informação relativa à atração turística que será utilizada para exibir em um carrossel, nesta página, que depois poderá ser rolada.



Isto que estamos vendo em execução é o Backoffice que sabemos que por padrão utiliza para seu Front end o gerador Web (.NET).

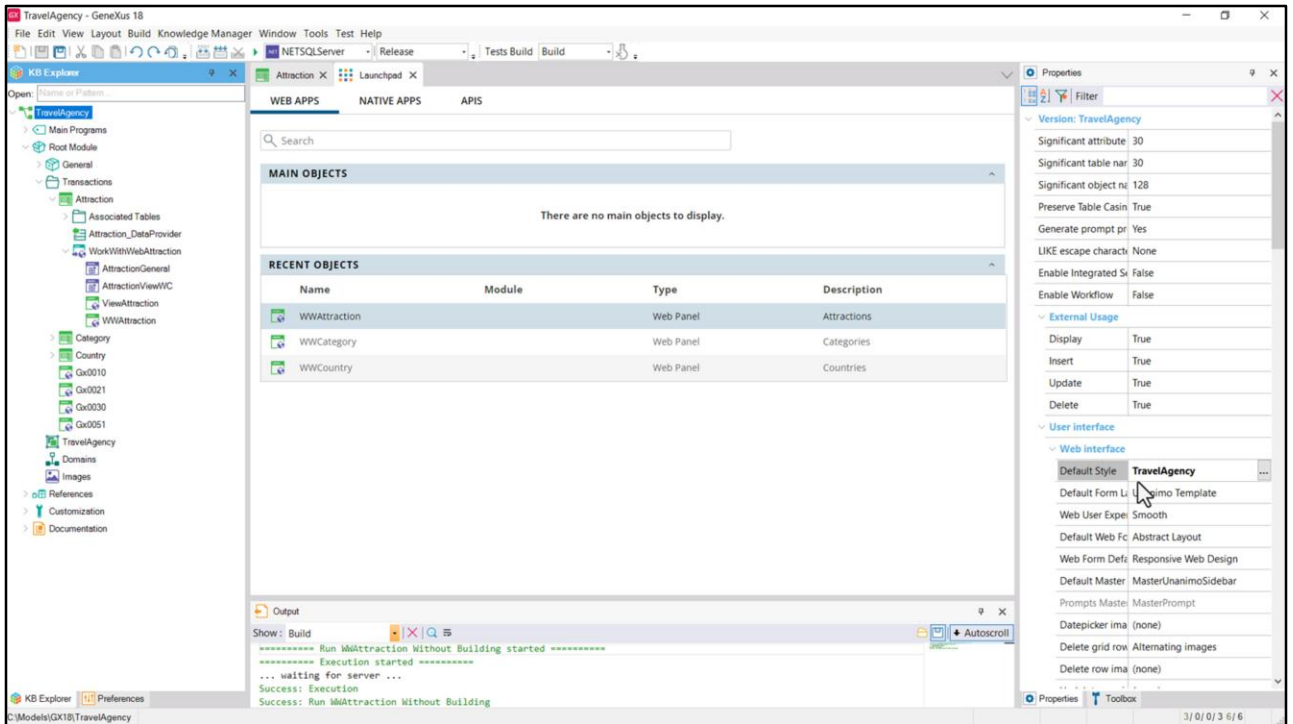


E qual é o objeto Design System que comanda o estilo da User Interface deste Backoffice? Se observarmos as propriedades da transação Attraction, vemos que aquela chamada Style indica o objeto com nome igual à KB.

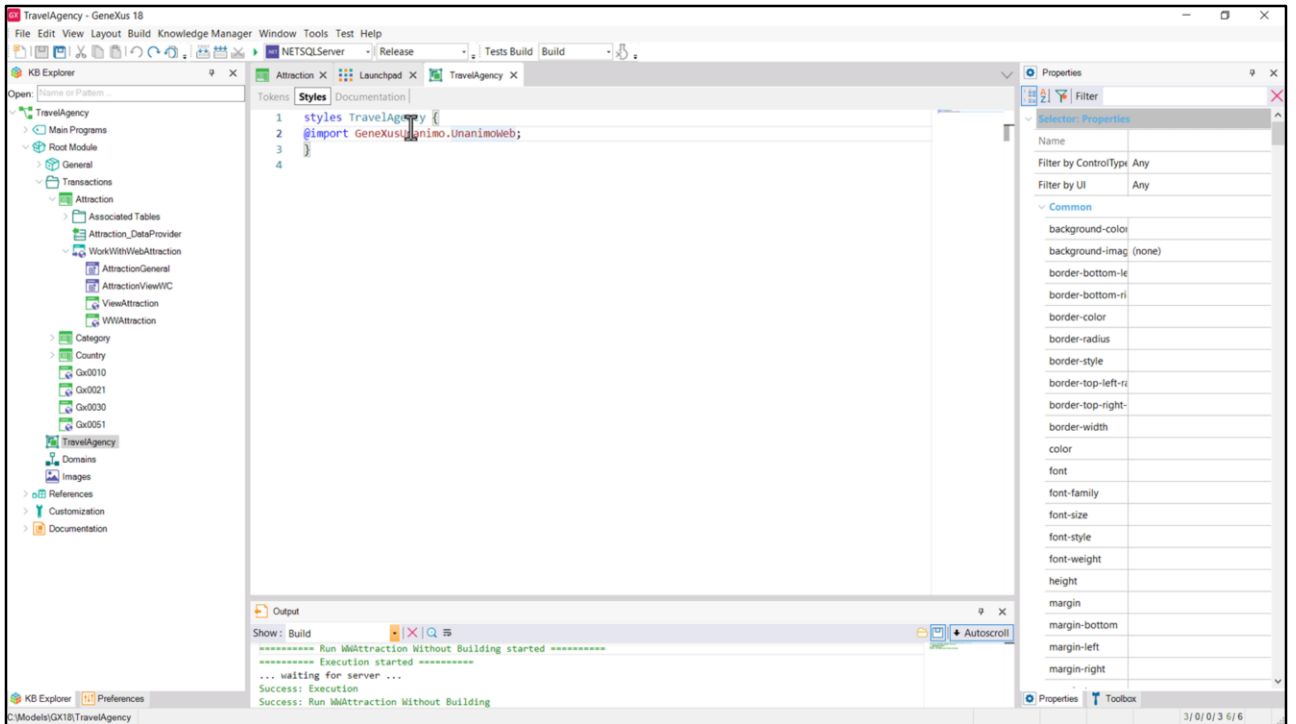


Se formos às propriedades do Web Panel WorkWithAttraction vemos para Style o mesmo objeto Travel Agency. O mesmo para o View, e o mesmo será para as demais transações e Web Panels e objetos com interface que são criados para o Backoffice.

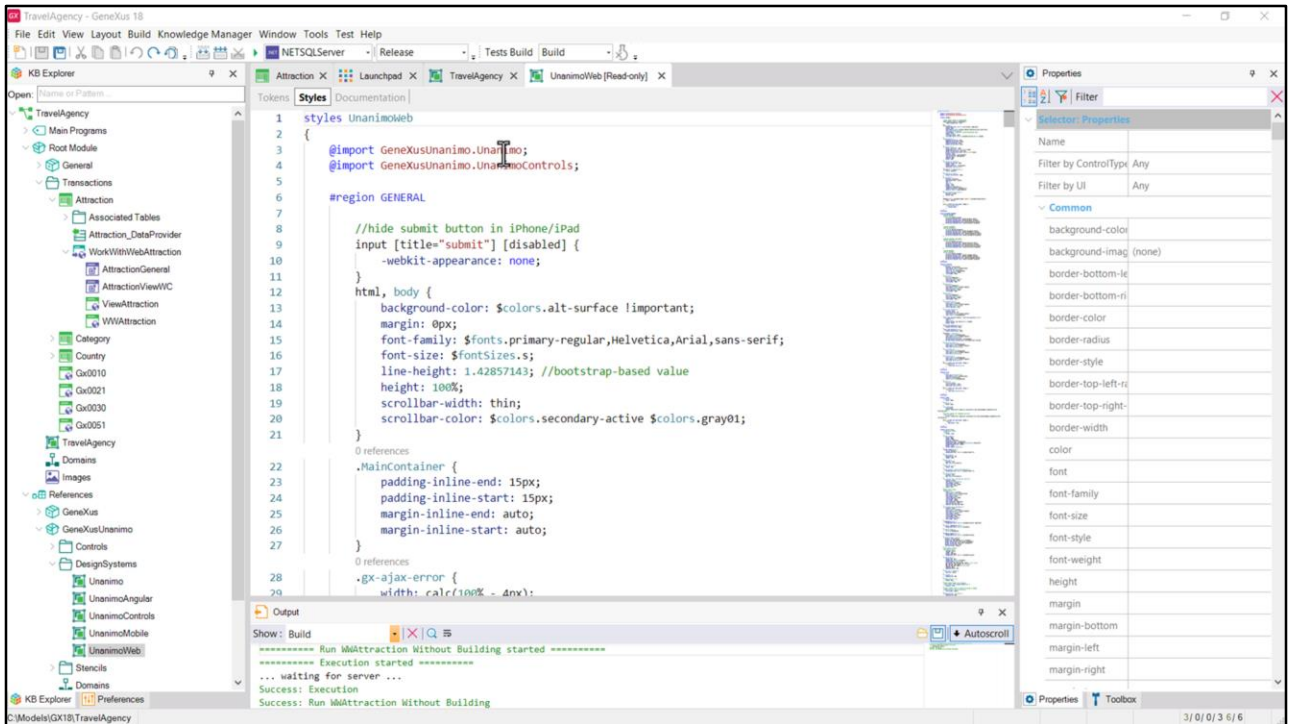




Vemos isso na propriedade Default Style da versão da KB.



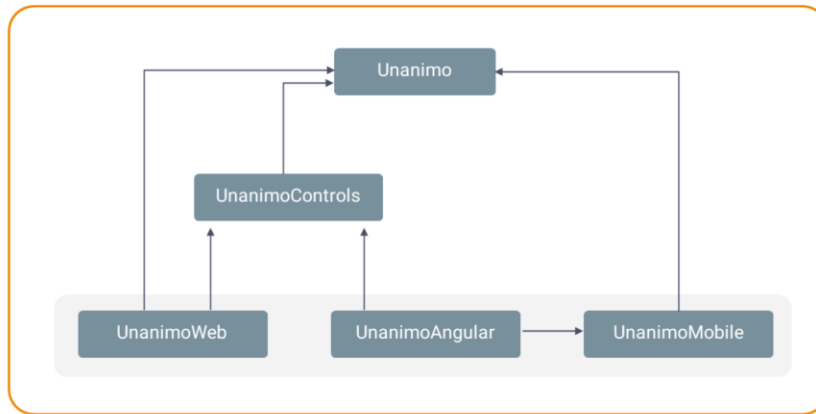
Este objeto DSO, lembre-se, é o único que é criado por default em toda KB, no momento da criação e que assume o mesmo nome.  
Se o abrirmos, vemos que tem vazia a aba de Tokens, mas na aba de Styles tem esta regra, que o que está dizendo é que importe todos os tokens e styles deste outro Design System Object, que se encontra no módulo GeneXusUnanimio.



É onde estava esse módulo GeneXusUnanimoweb? Ele veio instalado por padrão em toda KB, acessível sob o nó References.

Se o abrirmos vemos que é read-only, mas, além de suas definições específicas de tokens e estilos, está importando outros dois DSOs, estes, que estão no mesmo módulo.

Portanto, é construído sobre estes outros dois.



E por outro lado, se abríssemos esses outros que vemos aqui, o que veremos é essa árvore, onde o UnanimousMobile importa apenas o Unanimous, e o UnanimousAngular importa o mobile e também o UnanimousControls.

The screenshot displays the GeneXus 18 IDE interface for a project named 'TravelAgency'. The main window shows a design system diagram with the following components and relationships:

- Unanimo** (top level)
- UnanimoControls** (middle level)
- UnanimoWeb**, **UnanimoAngular**, and **UnanimoMobile** (bottom level, grouped in a light blue box)

Arrows indicate dependencies: UnanimoWeb, UnanimoAngular, and UnanimoMobile all depend on UnanimoControls. UnanimoControls and UnanimoMobile both depend on Unanimo. A blue arrow points from the 'TravelAgency' folder in the KB Explorer to the diagram.

The code editor shows the following tokens:

```
1 styles TravelAgency {  
2 @import GeneXusUnanimo.UnanimoWeb;  
3  
4 }
```

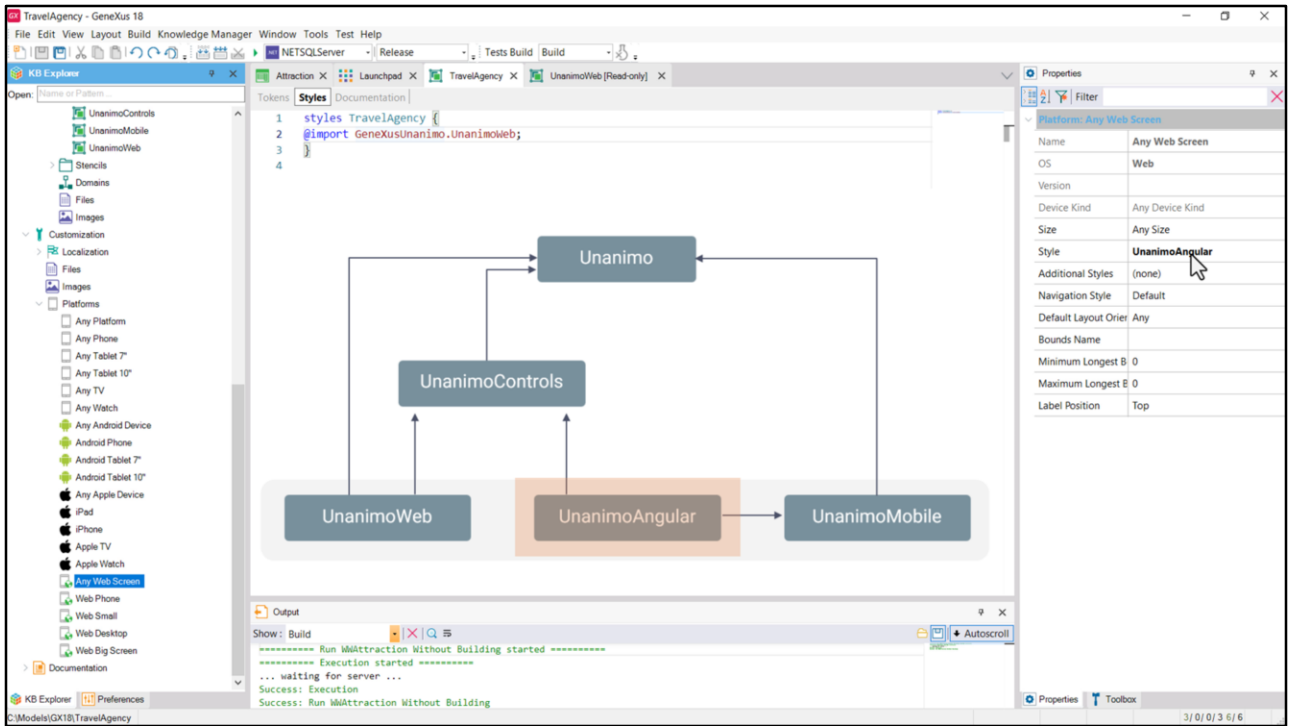
The Properties window on the right shows the following details for the Design System 'TravelAgency':

Name	TravelAgency
Description	Travel Agency
Module/Folder	Root Module
Base CSS	None
Qualified Name	TravelAgency
Object Visibility	Public

The Output window at the bottom shows the following build log:

```
Show: Build  
----- Run MAttraction Without Building started -----  
----- Execution started -----  
... waiting for server ...  
Success: Execution  
Success: Run MAttraction Without Building
```

Se o UnanimoWeb for o que será utilizado por Default no Frontend Web pois, como vimos, utiliza um DSO com o mesmo nome da KB que o importe...



... claramente o UnanimoAngular será o default para o Frontend também Web, mas Angular...

The screenshot shows the Genexus 18 IDE interface for a project named 'TravelAgency'. The main window is divided into several panes:

- KB Explorer (Left):** A tree view showing the project structure with folders for 'UnanimoControls', 'UnanimoMobile', and 'UnanimoWeb'. Under 'Platforms', 'Any Android Device' is selected.
- Code Editor (Top Center):** Displays a 'Styles' definition for 'TravelAgency' with the following code:

```
1 styles TravelAgency {  
2   @import GeneXusUnanimo.UnanimoWeb;  
3 }  
4
```
- Diagram (Center):** A dependency diagram showing 'Unanimo' at the top, which depends on 'UnanimoControls'. 'UnanimoControls' depends on 'UnanimoWeb' and 'UnanimoAngular'. 'UnanimoAngular' depends on 'UnanimoMobile'. 'UnanimoMobile' is highlighted with an orange background.
- Properties Panel (Right):** Shows the properties for the selected 'Any Android Device' platform. The 'Style' property is set to 'UnanimoMobile'.

Platform: Any Android Device	
Name	Any Android Device
OS	Android
Version	
Device Kind	Any Device Kind
Size	Any Size
Style	UnanimoMobile
Additional Styles	(none)
Navigation Style	Default
Default Layout Orier	Any
Bounds Name	
Minimum Shortest E	0
Maximum Shortest E	0
Label Position	Top
- Output Panel (Bottom):** Shows the build output for 'Build', indicating successful execution of 'Run MAttraction Without Building'.

...e o UnanimoMobile será o default para Frontend Mobile (para qualquer plataforma, tanto Android como Apple).

The screenshot displays the Genexus IDE interface. On the left, a project tree shows the structure of the 'TravelAgency' project, including 'NETSQLServer', 'Back end', 'Data Stores', 'Services', 'Front end', 'Web (.NET)', 'Web (Angular)', 'Deployment', 'Patterns', and 'Workflow'. The central editor shows a 'Styles' document with the following code:

```
1 styles TravelAgency {  
2   @import GeneXusUnanimo.Unanimoweb;  
3 }  
4
```

Below the code is a diagram illustrating the architecture. It features a central box labeled 'Unanimo'. Below it are three boxes: 'UnanimoWeb', 'UnanimoAngular', and 'UnanimoMobile'. Arrows indicate dependencies: 'UnanimoWeb' and 'UnanimoAngular' both depend on 'Unanimo'. 'UnanimoAngular' also depends on 'UnanimoMobile'. 'UnanimoMobile' has a dependency on 'Unanimo'. The 'UnanimoWeb', 'UnanimoAngular', and 'UnanimoMobile' boxes are grouped together in a light blue rounded rectangle.

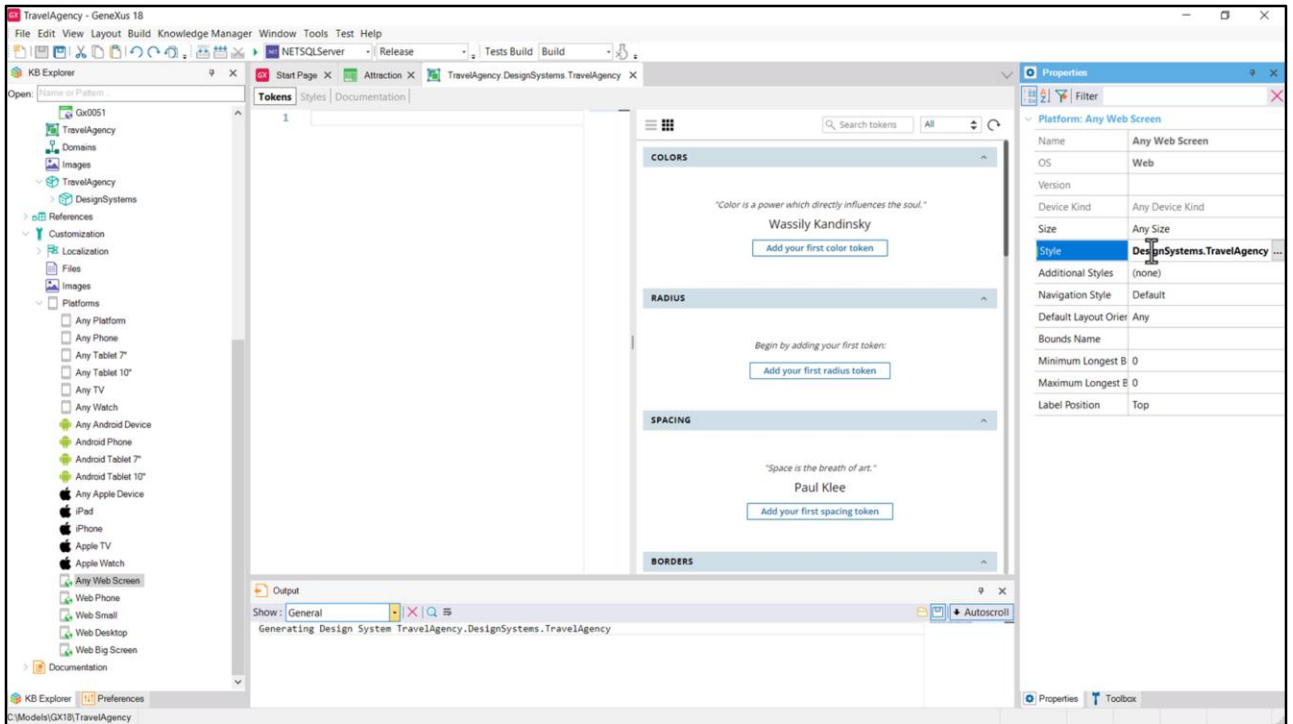
On the right, the 'Properties' panel is open, showing the configuration for the 'Generator: Frontend (Front end)'. The 'Main Platform' is set to 'Angular'. The 'Generate Angular' option is checked (True). Other settings include 'Name: Frontend', 'Services URL: https://apps5.genexus.com/ldb2...', and 'Setup Command: npm update'.

At the bottom, the 'Output' window shows the following log:

```
Show: Build  
***** Run MAttraction Without Building started *****  
***** Execution started *****  
... waiting for server ...  
Success: Execution  
Success: Run MAttraction Without Building
```

Como ainda não temos nenhum Front end para estas plataformas, não podemos ver em ação esses DSOs. De qualquer maneira nós não queremos utilizá-los. Acontece que vamos querer construir a aplicação Customer-facing que terá outro sistema de design, aquele que Chechu desenhou para nós, aquele que estávamos analisando, então vamos criar o DSO para Angular (que é a plataforma em que vamos nos concentrar boa parte do curso) a partir do zero. E então, no final, veremos o caso nativo.



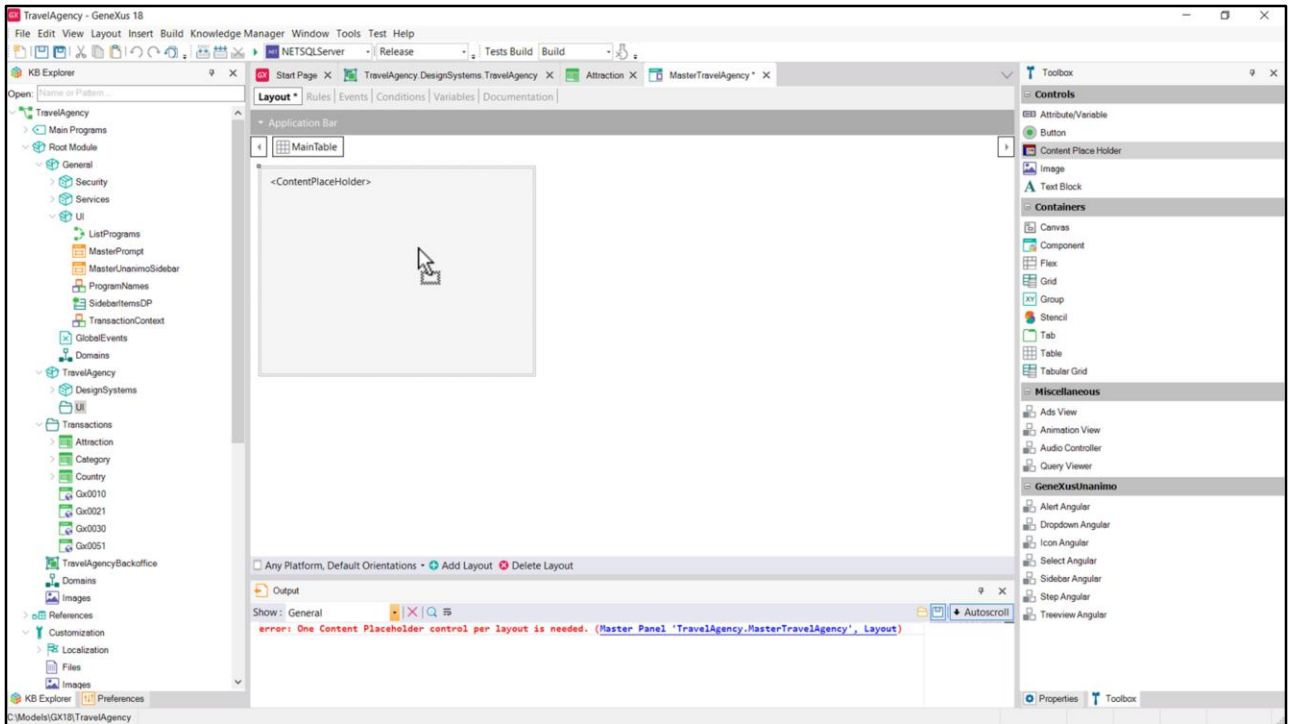


Para deixar bem separados os objetos que serão próprios de nossa aplicação Customer-facing de todos os objetos que vêm por default na KB, e daqueles do Backoffice, vamos criar um módulo que por enquanto chamarei de TravelAgency.

E dentro vou criar outro módulo, DesignSystems, para colocar os objetos DSOs que iremos utilizar. Então veremos porque serão vários e não apenas um. Já podemos criar aquele que comandará o estilo da aplicação Angular que iremos desenvolver inicialmente para tamanho Desktop. Vamos chamá-lo de TravelAgency e definir que seja utilizado este para essa plataforma.

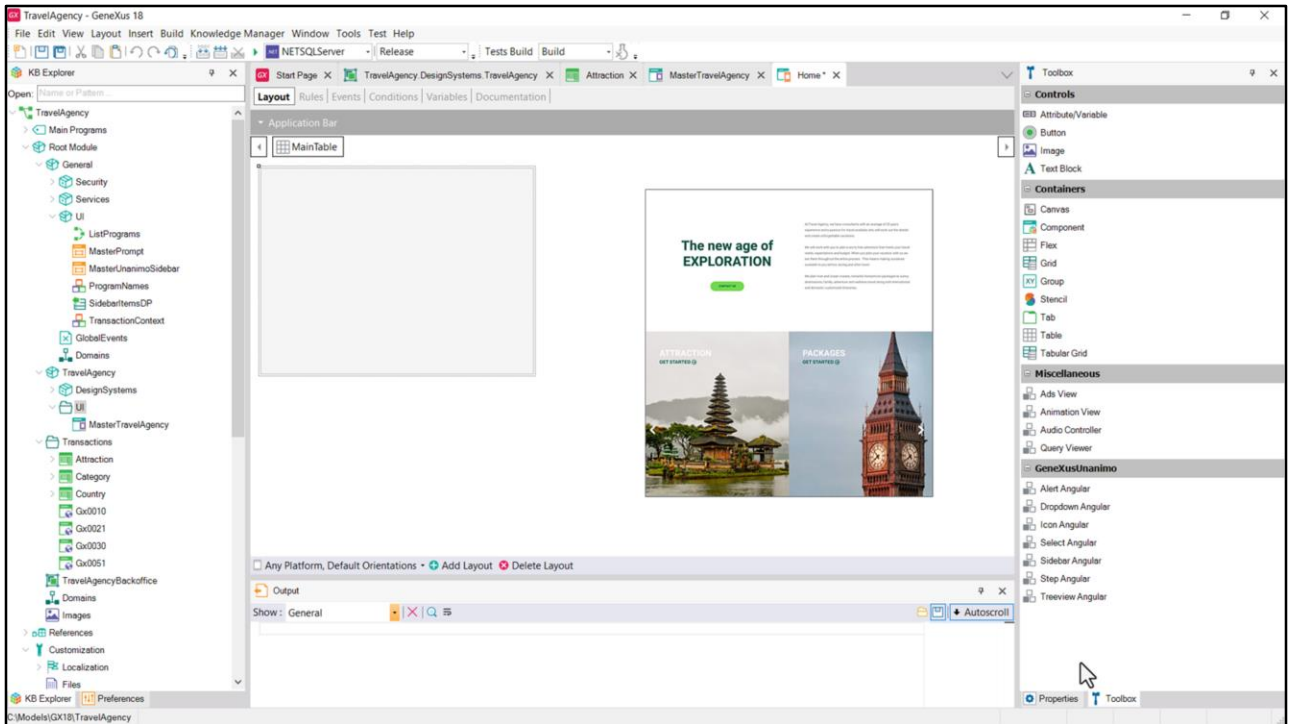
Agora temos dois DSOs TravelAgency, mas escolhemos aquele do módulo DesignSystems que acabamos de criar. E vemos que todas as plataformas mais específicas da web Angular herdarão esse DSO.

Conseguimos chamar da mesma forma este outro DSO porque o colocamos dentro de um módulo. Mas vamos mudar o nome para este outro para que não nos confundam.

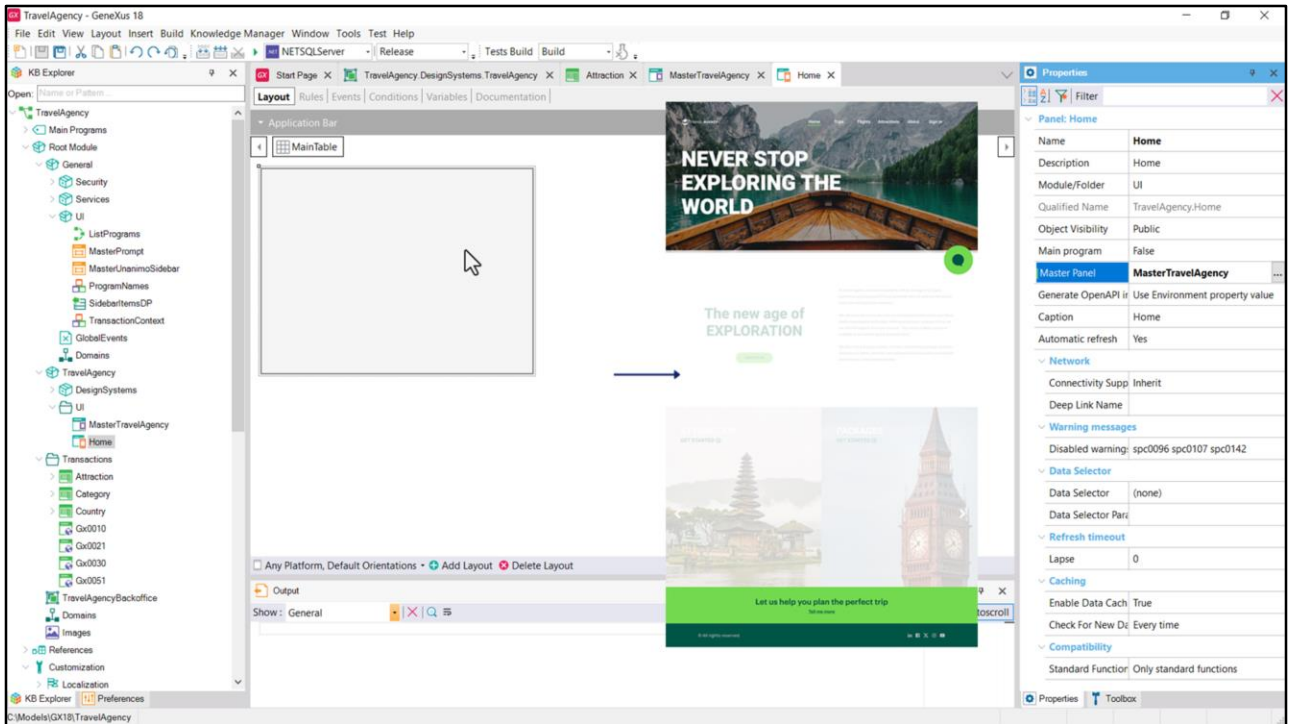


Agora vamos criar uma pasta dentro do módulo para organizar ali dentro todos os objetos com interface de usuário.

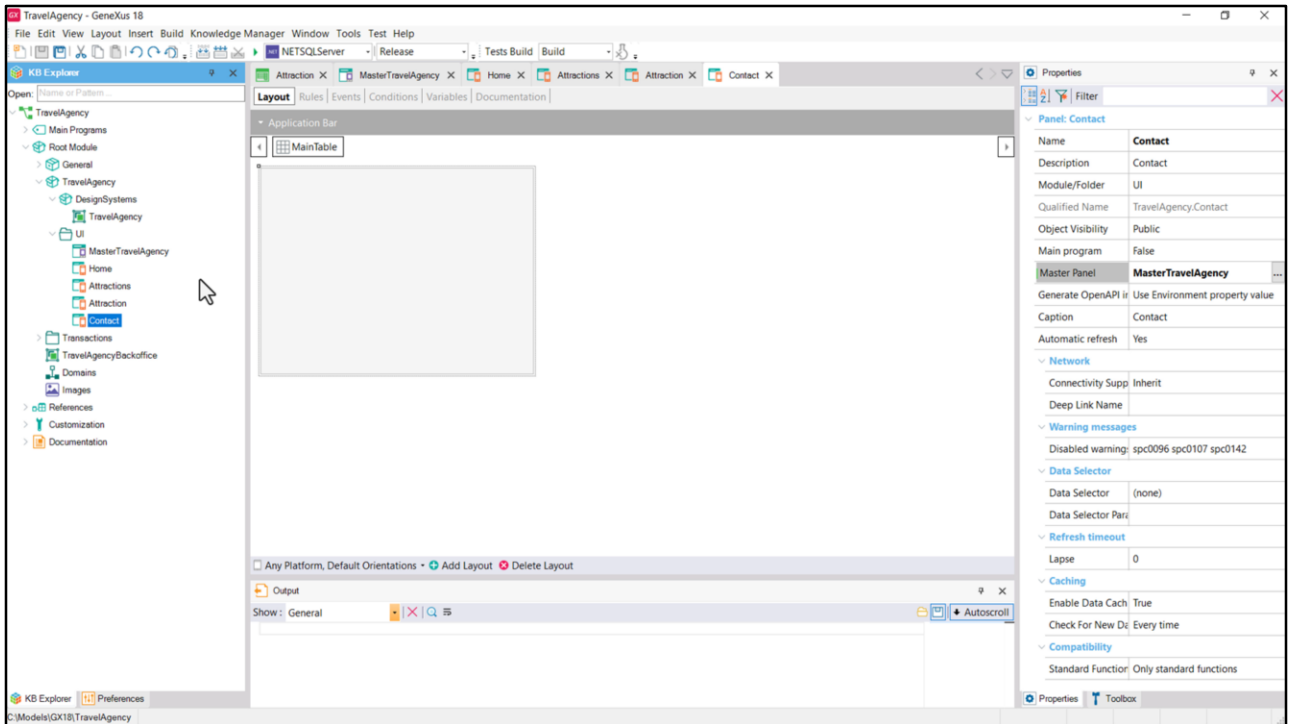
Então vamos criar antes de mais nada o Master Panel, que chamarei assim. Se eu tentar salvá-lo vazio, ocorrerá esse erro. Acontece que é obrigatório colocar o controle ContentPlaceholder, que é onde serão carregados os painéis que o tenham como seu Master Panel. Agora sim teremos permissão para gravar o objeto.



Vamos então criar o primeiro panel, Home, que será o que implementará esta parte da tela que vimos no Figma.

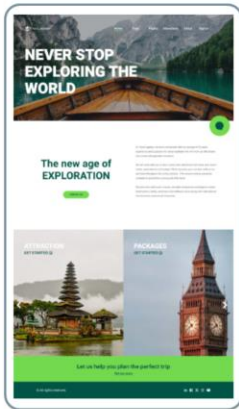


E temos que indicar que seja carregado dentro do Master Panel que acabamos de criar. Ou seja, será carregado aqui dentro...

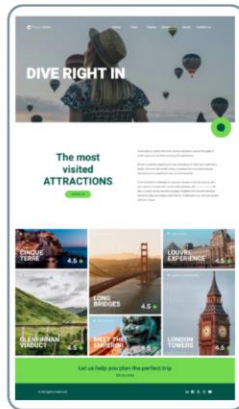


E temos que indicar que seja carregado dentro do Master Panel que acabamos de criar. Ou seja, será carregado aqui dentro...

Pois bem, da mesma forma criaremos os outros três painéis, vazios por enquanto. E com o mesmo Master Panel.



Home



Attractions



Attraction



Contact

Aqui vemos o que temos até o momento.

Pois bem, ao chegar a este ponto já temos a KB inicializada para poder começar a trabalhar em cada um dos objetos e implementar cada uma das partes do layout. Então esperamos por você na próxima aula.

GX

GeneXus by Globant

**GeneXus**<sup>™</sup>  
by Globant

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