

GX

GeneXus by Globant

GeneXus[™]
by **Globant**

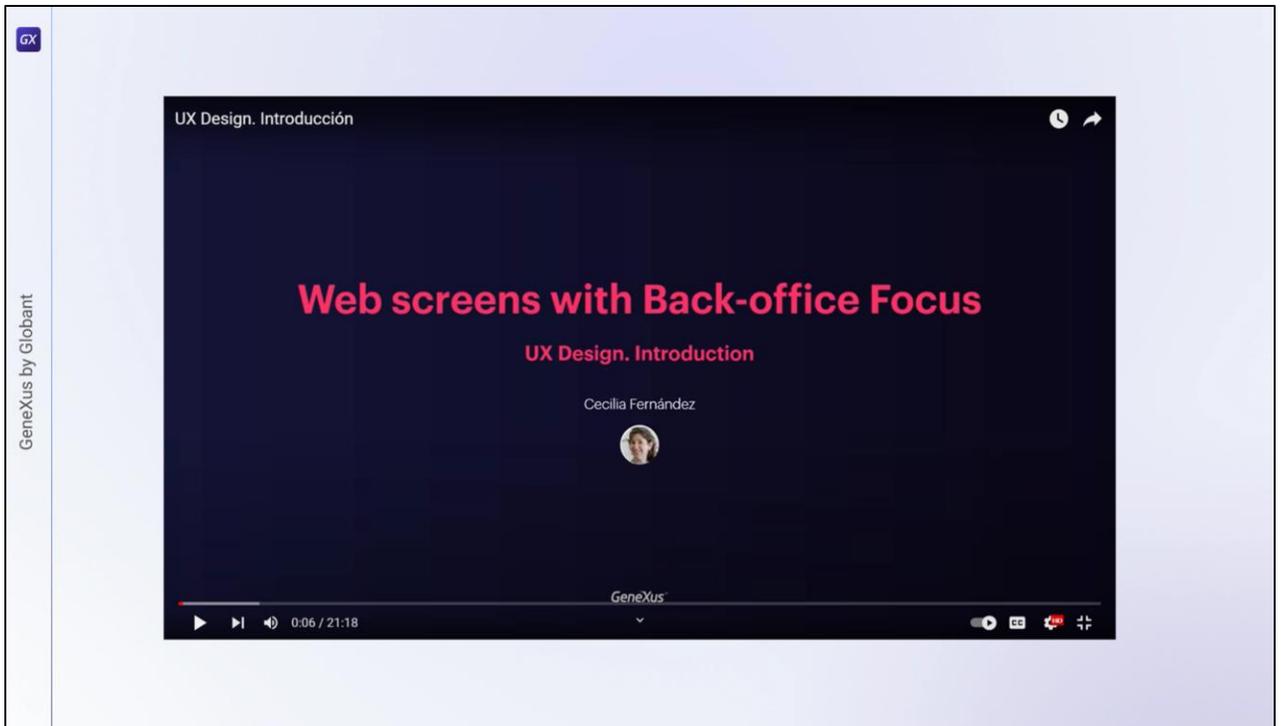
training.genexus.com

Personalização de UI

Introdução



Cecilia Fernández



Neste outro vídeo começamos a personalizar o Design System default que era criado com a KB e que utilizava Unanimo Web.

Aqui vamos apagar o que havíamos feito, para realizar personalizações mais interessantes.

GeneXus by Globant

Attractions

Travel Agency

UNANIMO

Attractions

INSERT ATTRACTIONS PDF

Search Name

Category Name	Photo	Trips	UPDATE	DELETE
Monument		24	UPDATE	DELETE
Tourist site		0	UPDATE	DELETE
Monument		24	UPDATE	DELETE
Tourist site		0	UPDATE	DELETE
Tourist site		0	UPDATE	DELETE
Monument		1	UPDATE	DELETE
Tourist site		0	UPDATE	DELETE
Museum		1	UPDATE	DELETE
Museum		13	UPDATE	DELETE

Patterns usable in this object (underlined means pattern is applied)

Work With for Web Work With

Apply this pattern on save

Work With Pattern Instance

- Transaction (Attraction)
 - Level (Attraction)
 - DescriptionAttribute (AttractionName)
 - Selection (Attractions)
 - modes (Insert, Update, Delete)
 - Attributes
 - AttractionId
 - AttractionName
 - CountryName
 - CityName
 - CategoryName
 - AttractionPhoto
 - trips
 - Orders
 - Order (Name)
 - AttractionName
 - Order (Country)
 - CountryName
 - Filter
 - Attributes
 - AttractionName
 - CountryName
 - Conditions
 - AttractionName like &AttractionName when not &AttractionName.IsEmpty()
 - CountryName like &CountryName when not &CountryName.IsEmpty()
 - Actions
 - Action (AttractionList)

Conditions

```
AttractionName like &AttractionName when not &AttractionName.IsEmpty();
CountryName like &CountryName when not &CountryName.IsEmpty();
```

Começaremos observando as diferenças entre o panel Work With criado por GeneXus e nossa primeira imitação desse panel, aquele que criamos do zero.

O WorkWith filtrava as atrações com este campo, usando o operador like. Não implementava o filtro nas conditions do grid, mas nas gerais do Web panel.

GeneXus Travel Agency

UNANIMO

Attractions

Countries

Country Id

Attraction Name From

Attraction Name To

Grid1's Conditions

```
CountryId = &CountryId
when not &CountryId.IsEmpty();

AttractionName >= &AttractionNameFrom
when not &AttractionNameFrom.IsEmpty();

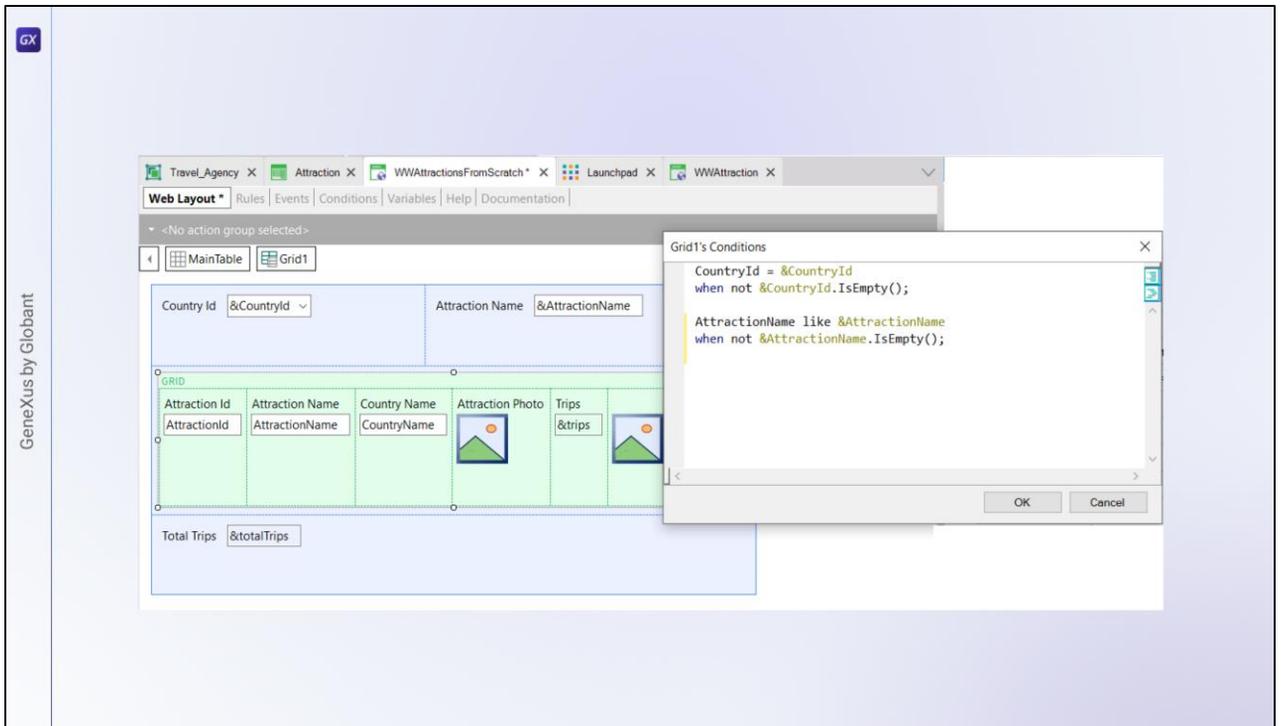
AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();
```

Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips
AttractionId	AttractionName	CountryName		&trips

Total Trips

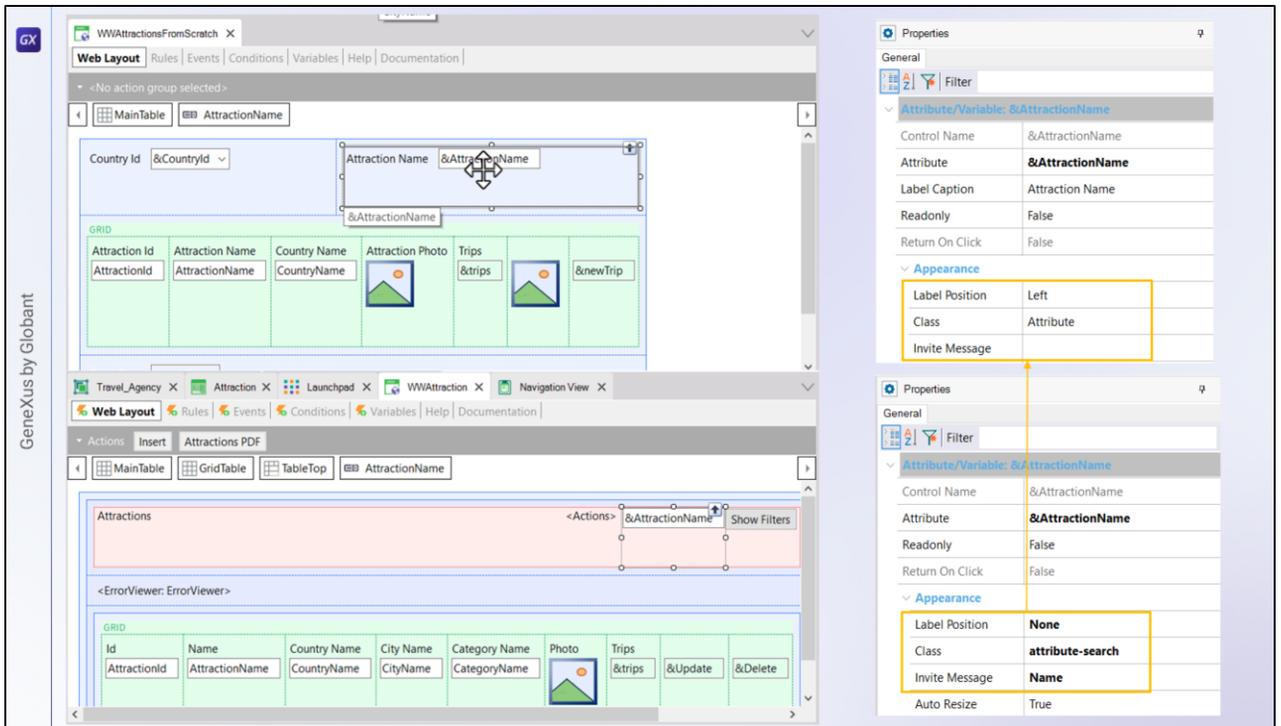
109 Matisse Museum France 13 New Trip

Nós, por outro lado, filtrávamos de acordo com um intervalo e por isso tínhamos estas duas variáveis.



Vamos criar uma variável AttractionName para substituir uma daquelas do intervalo: esta, a trocamos pela nova; eliminamos a outra; movemos esta para a direita do filtro de país; eliminamos esta tabela e agora modificamos as condições do filtro do grid para utilizar o mesmo operador like do WorkWith. Vamos testar.

Bom, mas mesmo assim vemos que nosso campo tem uma estética menos atrativa que a do WorkWith. O que está fazendo a diferença?

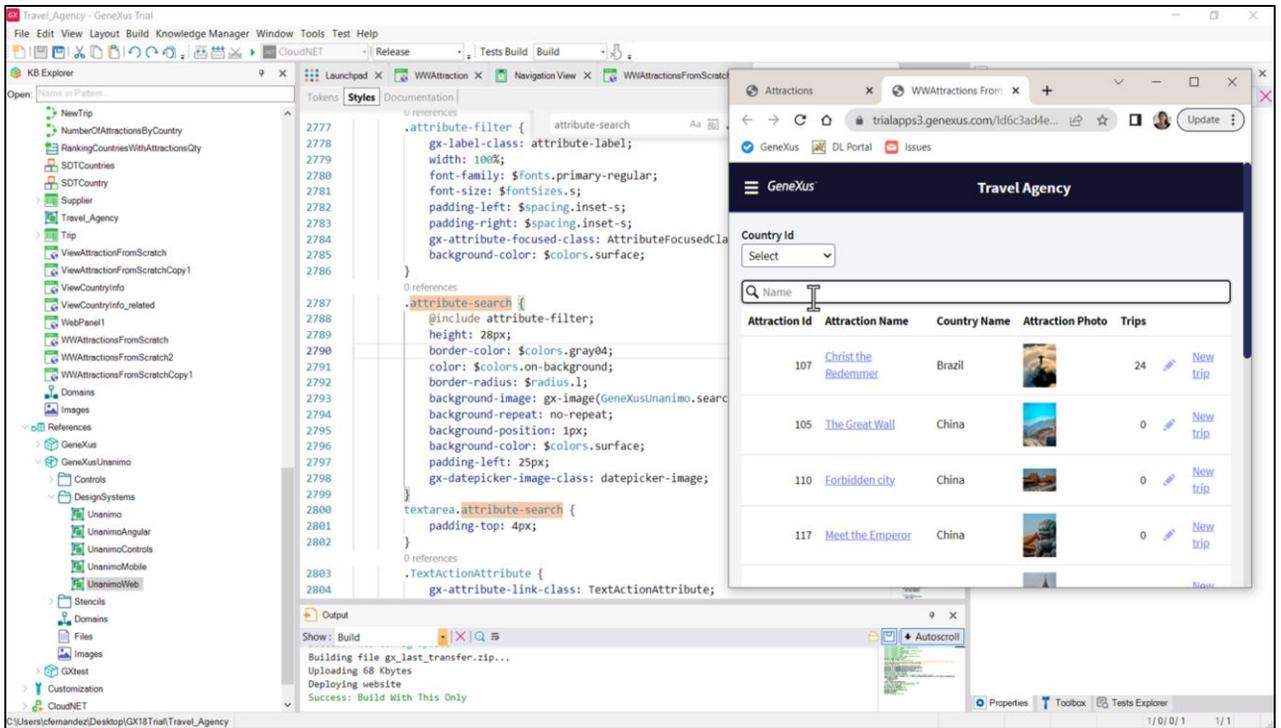


Vamos comparar as propriedades de um campo e do outro...

Aqui vemos as do nosso e aqui as do WorkWith.

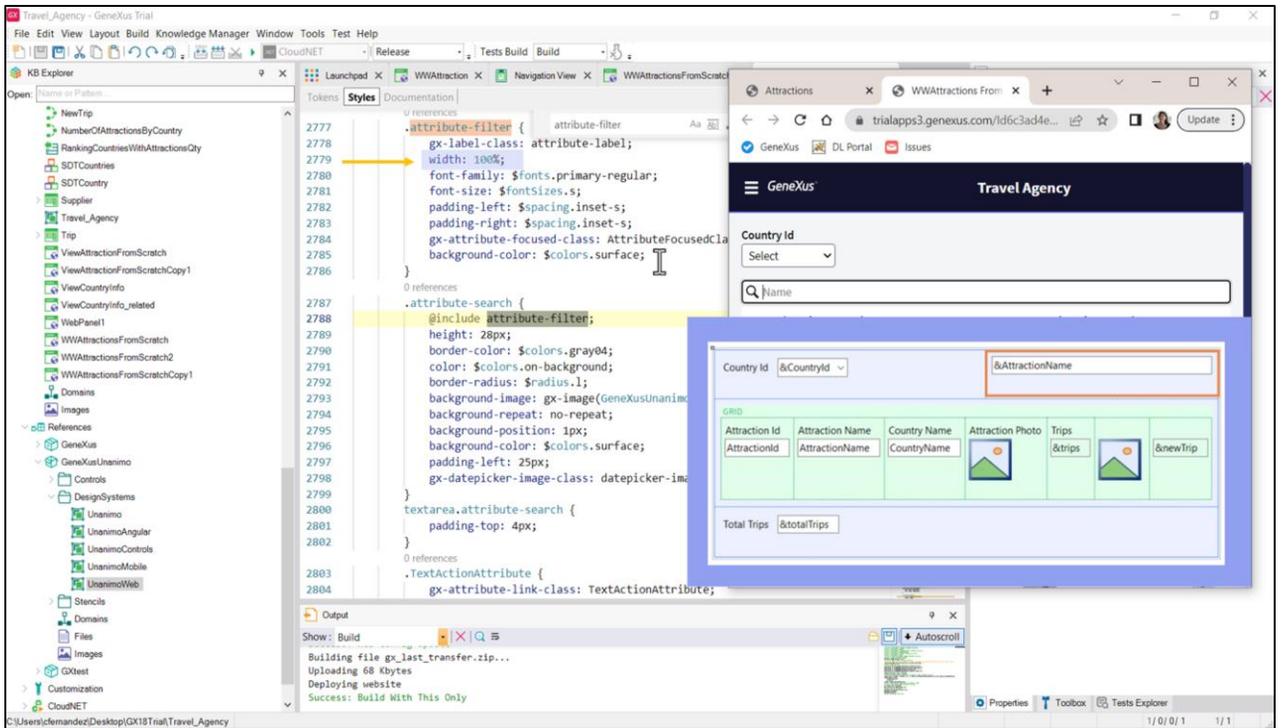
Vamos copiá-las para o nosso... este é o texto que aparece em cinza dentro do campo antes que o usuário digite qualquer coisa. Já não precisamos do rótulo. E aqui a mais importante, a classe. Vamos associar a ela esta classe bem específica, que podemos supor é a que controla sua estética.

Vamos testar. Vamos ver as propriedades da classe?



O Design System Object que comanda o design era esse, que está importando o UnanimoWeb... e ali procuramos o nome da classe. Aqui está. Tem todas estas propriedades.

Por exemplo, a que controla a cor da borda, os cantos arredondados da borda, a imagem de search, a cor de fundo do campo... mas também observamos que inclui outra classe, ou seja, suas propriedades.



E vemos que, entre outras, estão as relacionadas à fonte (sua família e tamanho), e por exemplo, a largura do campo, que é 100% em relação a quem o contém, que neste caso é a célula da tabela em que está localizado nosso campo.

Attractions WWAttractions From Scratch

trialapps3.genexus.com/d6c3ad4e5d6b207af3d2c6ac7a402adbe/wwattractionsfromscratch.aspx

GeneXus DL Portal Issues Update

GeneXus Travel Agency

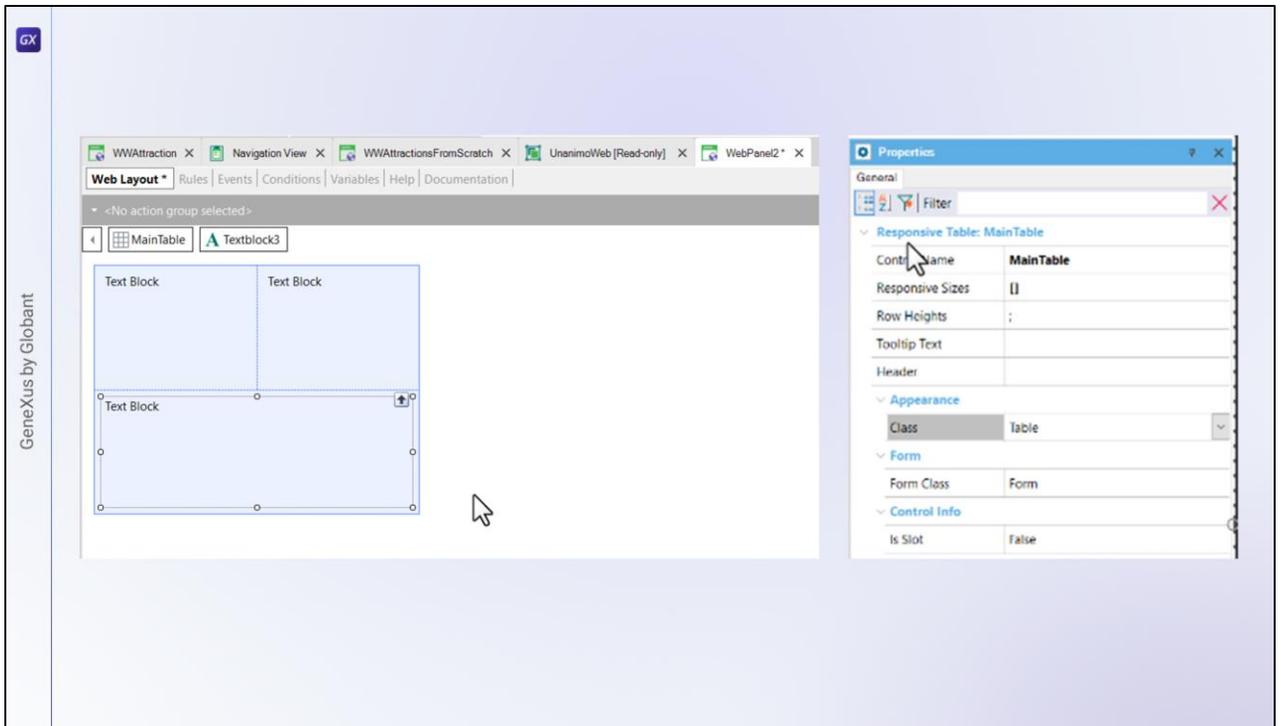
Country Id

Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips
107	Christ the Redeemer	Brazil		24 New trip
105	The Great Wall	China		0 New trip
110	Forbidden city	China		0 New trip
117	Meet the Emperor	China		0 New trip
113	London Towers	England		1 New trip
104	Louvre Museum	France		1 New trip
106	Eiffel Tower	France		24 New trip
109	Matisse Museum	France		13 New trip
111	Cinque Terre	Italy		0 New trip

Mas, por que aqui aparece abaixo deste outro controle e, em vez disso, quando a tela é ampliada, aparece à direita; e ao aparecer, ocupando 50% da tela?

The screenshot displays the GeneXus IDE interface. On the left, the 'KB Explorer' shows a project tree with various views and domains. The main workspace is titled 'Web Layout' and shows a design for a table. At the top, there is a header with 'Country Id' and '&CountryId' on the left, and '&AttractionName' on the right. Below this is a table with columns: 'Attraction Id', 'Attraction Name', 'Country Name', 'Attraction Photo', 'Trips', and '&newTrip'. The table is highlighted with a red border. Below the table is a row for 'Total Trips' with the value '&totalTrips'. On the right side, the 'Properties' window is open, showing the 'Attribute/Variable: &AttractionName' configuration. The 'Control Name' is '&AttractionName', the 'Attribute' is '&AttractionName', and the 'Class' is 'attribute-search'. The 'Output' window at the bottom shows the build process: 'Building file gx_last_transfer.zip...', 'Uploading 68 Kbytes', 'Deploying website', and 'Success: Build With This Only'.

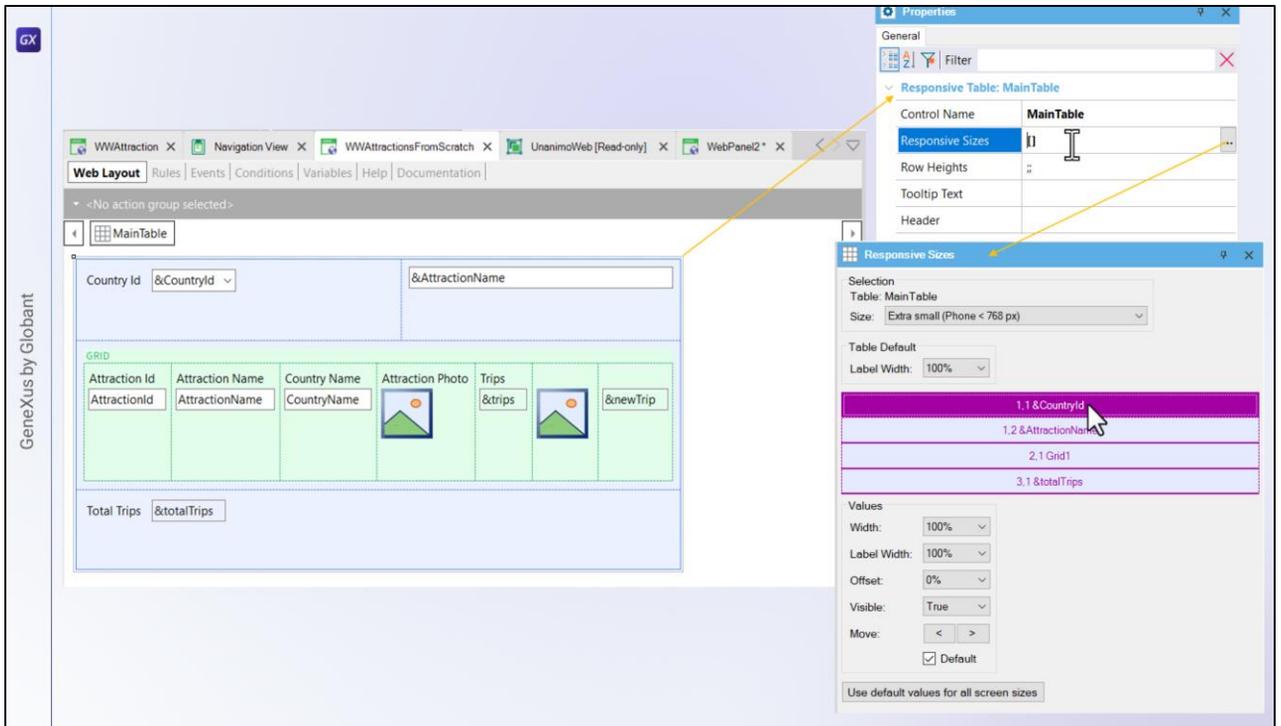
O controle se encontra na célula direita desta linha, desta tabela.



Quando é criado um novo Web panel, seu layout é sempre inicializado com uma tabela, a MainTable.

Qualquer controle que seja inserido estará dentro dessa tabela e qualquer outro controle que seja inserido criará automaticamente outra célula na tabela. Aqui a tabela ficou com duas células, e se agora inserirmos outro controle, teremos outra célula.

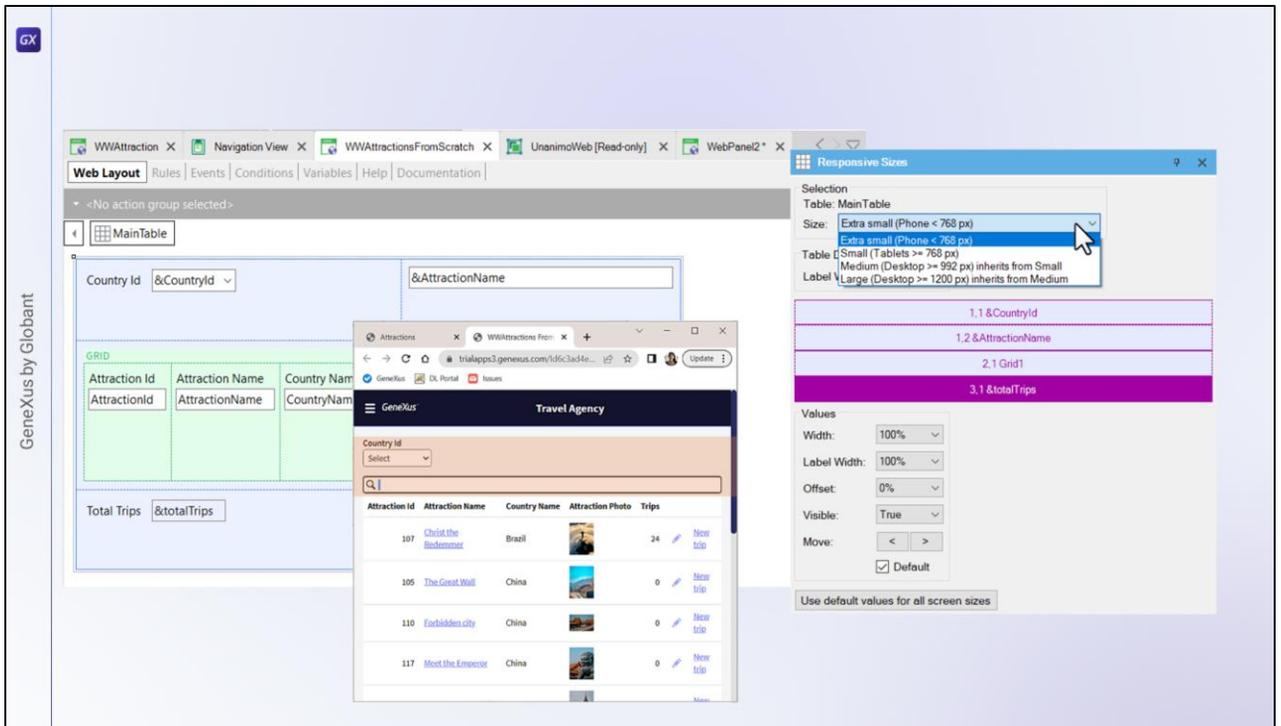
A tabela é uma tabela especial chamada responsiva.



Aqui vemos que a tabela onde estão todos os nossos controles é responsiva e que possui esta propriedade, Responsive Sizes.

O que está indicando? As quatro células da tabela, com os controles que vão em cada uma. Por exemplo, aqui vemos este controle, aqui este outro, depois o grid e, por último, esta variável. Aparecem ocupando 100% da largura e um em cima do outro, como se fosse uma tabela com uma única coluna.

Mas se observamos este combo, Size, vemos que aqui diz que isso é válido para tamanho de tela Extra small, que corresponde ao de um telefone.



Mas observemos que temos outros 3 tamanhos possíveis de tela. Para o Extra small, as duas células da primeira linha que vemos no layout serão exibidas uma em cima da outra, ocupando cada uma 100% da largura da tabela.

É por isso que vemos assim as variáveis em execução para esse tamanho.

Travel_Agency - GeneXus Trial

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Tests Build Build

KB Explorer

Web Layout Rules Events Conditions Variables Help Documentation

Country Id &CountryId &AttractionName

Responsive Sizes

Selection
Table: Main Table
Size: Medium (Desktop >= 992 px) inherits from Small

Table Default
Label Width: 25%

1,1 &CountryId	1,2 &AttractionName
2,1 Grid1	
3,1 &TotalTrips	

Values
Width: 50%
Label Width: 50%
Offset: 0%
Visible: True
Move: < >
 Default

Use default values for all screen sizes

Properties Toolbox Tests Explorer Responsive Sizes

1 / 0 / 1 1 / 1

GeneXus Travel Agency

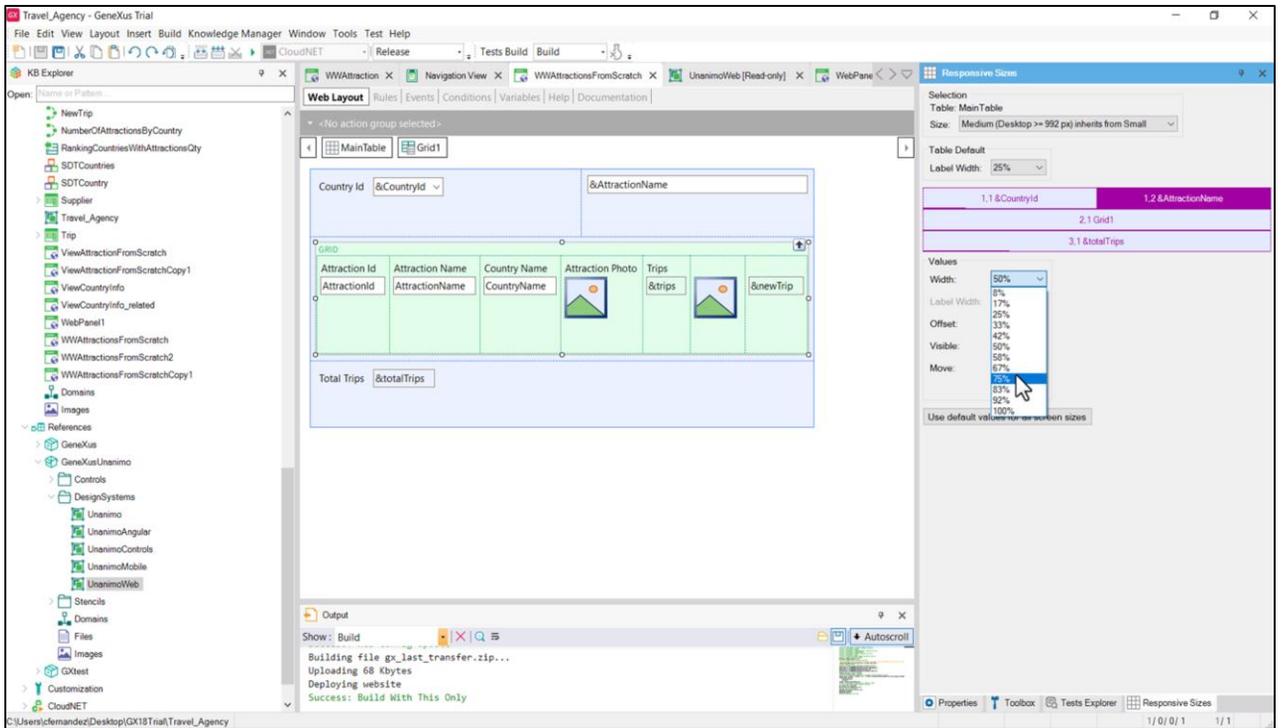
Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips
107	Christ the Redeemer	Brazil		24 New trip
105	The Great Wall	China		0 New trip
110	Forbidden city	China		0 New trip
117	Next the Empress	China		0 New trip
113	London Towers	England		1 New trip

Show: Build

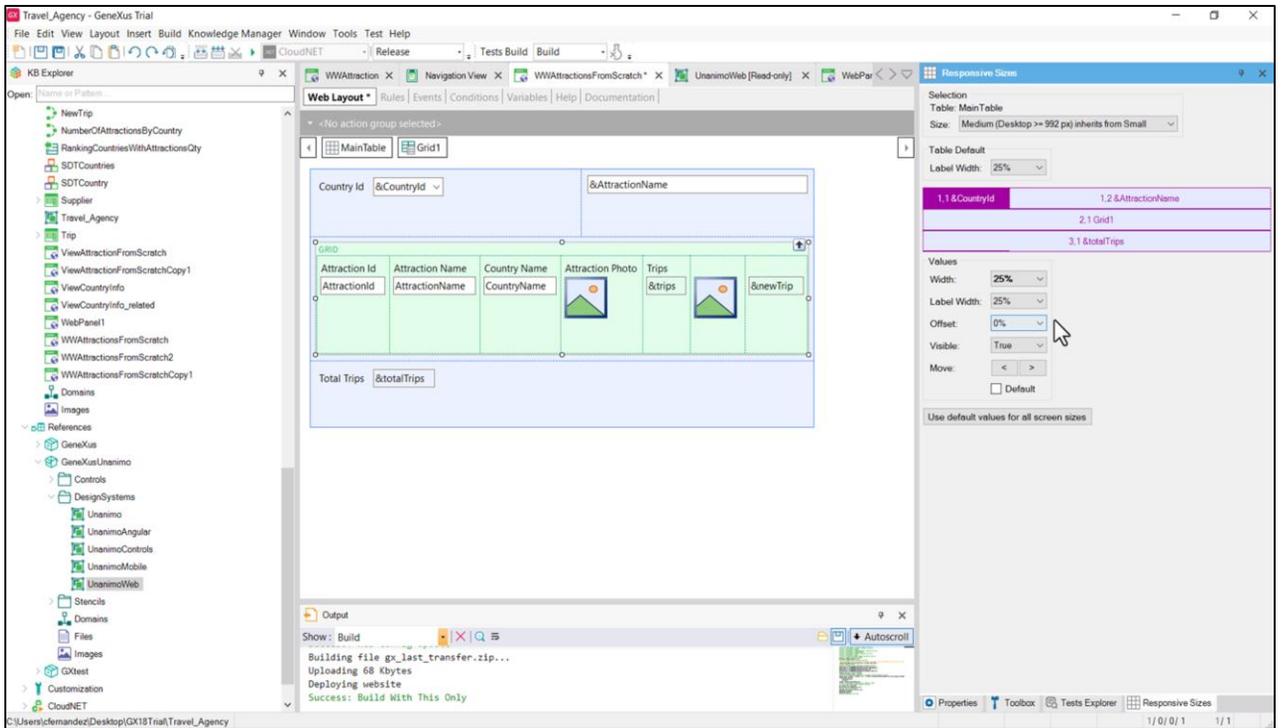
Building file gx_last_transfer.zip...
Uploading 68 Kbytes
Deploying website
Success: Build With This Only

C:\Users\cfemandez\Desktop\GX18Trial\Travel_Agency

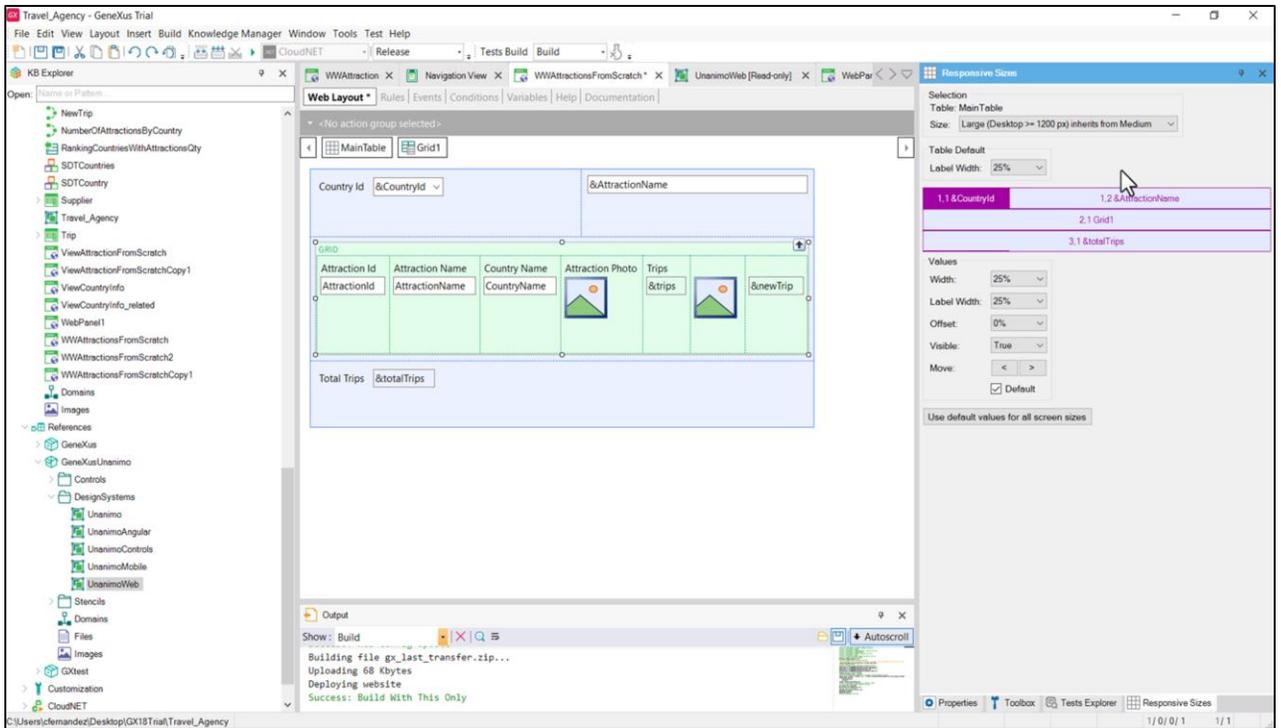
Mas se em vez disso, agora selecionarmos o tamanho Medium que corresponde ao tamanho de desktop, teremos a linha 1 da tabela sendo exibida em uma linha, efetivamente, onde cada célula ocupará 50% da largura da tabela, como observávamos em execução.



Podemos modificar aqui essa largura, com base nestas 12 possibilidades. E assim, por exemplo, indicar que queremos que AttractionName ocupe 75% da largura da tabela.



Mas para que ambos os campos fiquem em uma mesma linha, também teremos que modificar a largura deste outro para que ocupe 25%.



O tamanho Large que segue herda os valores do Medium que acabamos de configurar e, por isso se vê igual.

The screenshot displays the GeneXus IDE interface for a web application titled "Travel Agency". The main window shows a web layout with a search bar and a table of attractions. The table has the following data:

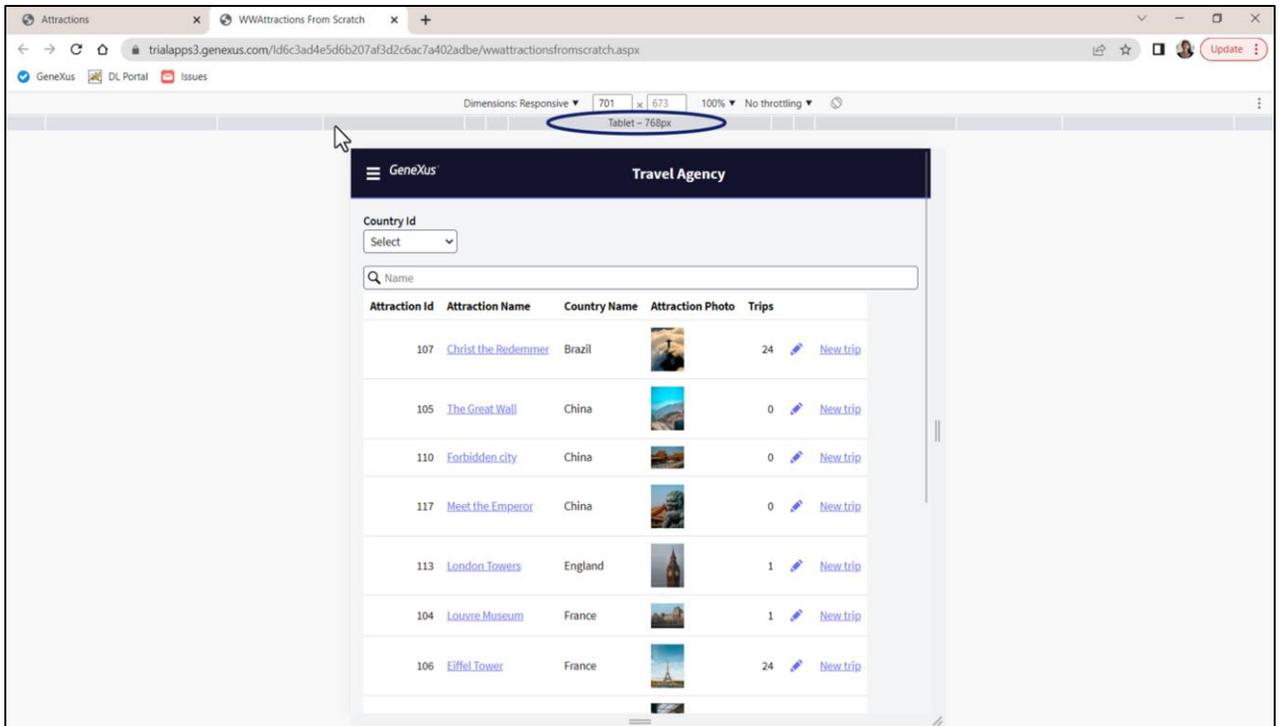
Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips
107	Christ the Redeemer	Brazil		24 New trip
105	The Great Wall	China		0 New trip
110	Forbidden city	China		0 New trip
117	Meet the Emperor	China		0 New trip
113	London Towers	England		1 New trip

The output window at the bottom shows the following logs:

```
Show: Build
Building file gx_last_transfer.zip...
Uploading 52 Kbytes
Deploying website
Success: Build With This Only
```

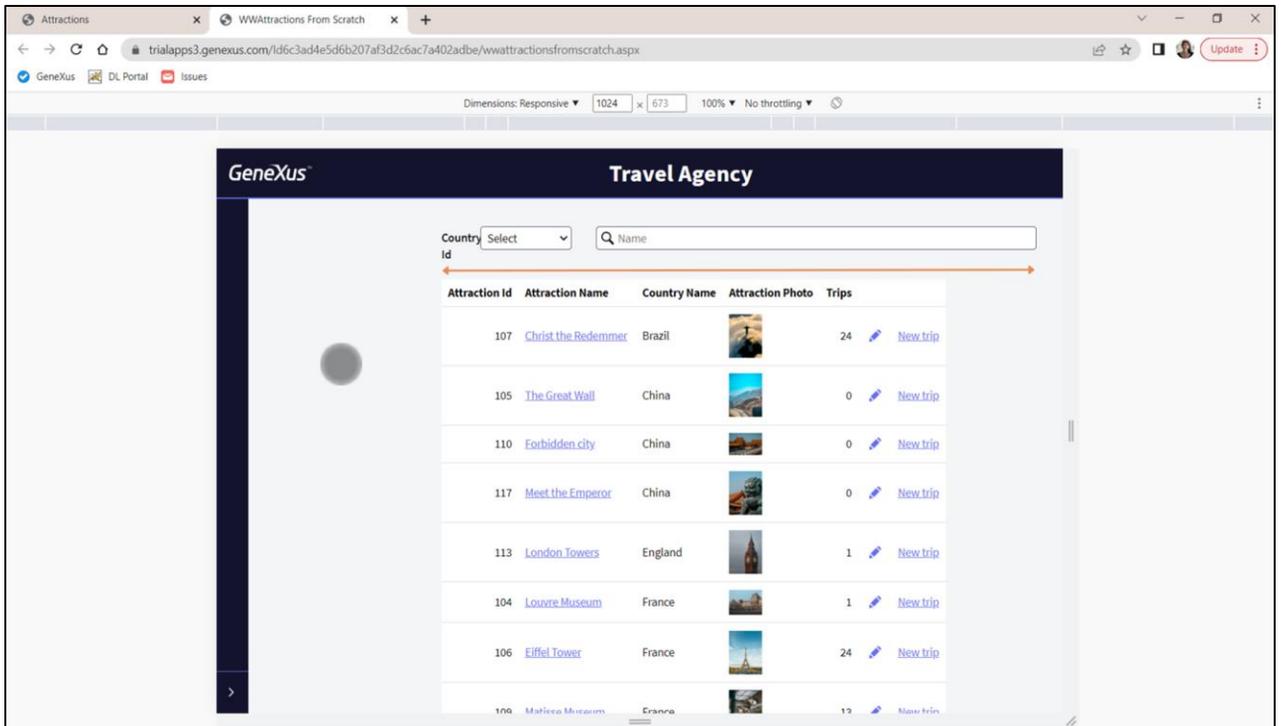
Vamos testar esta mudança em execução. Atualizamos. E aqui vemos essa mudança nas larguras de cada campo: 25% e 75% da largura da tabela.

Continuam sendo respeitadas essas porcentagens à medida que diminuimos a tela, passando pelo tamanho Small que não modificamos, e até chegar ao tamanho de tela Extra small onde as larguras para cada controle da linha 1 eram de 100%.



Outra forma de visualizar as variações por tamanho é pressionar F12 no navegador. Aqui nos permite variar de maneira discreta... e aqui de maneira contínua.

O que vimos até aqui é que podemos variar a largura e a localização dos controles de uma tabela responsiva de acordo com 4 larguras de tela (Extra small, Small, Medium e Large).



Agora vamos prestar atenção em nosso grid e compará-lo com o do WorkWith.

Para começar, nosso grid tem fundo branco, não ocupa 100% da largura da tabela, e vai carregar todas as atrações turísticas...

Attractions

trialapps3.genexus.com/d6c3ad4e5d6b207af3d2c6ac7a402adbe/wwattraction.aspx

GeneXus DL Portal Issues

UNANIMO

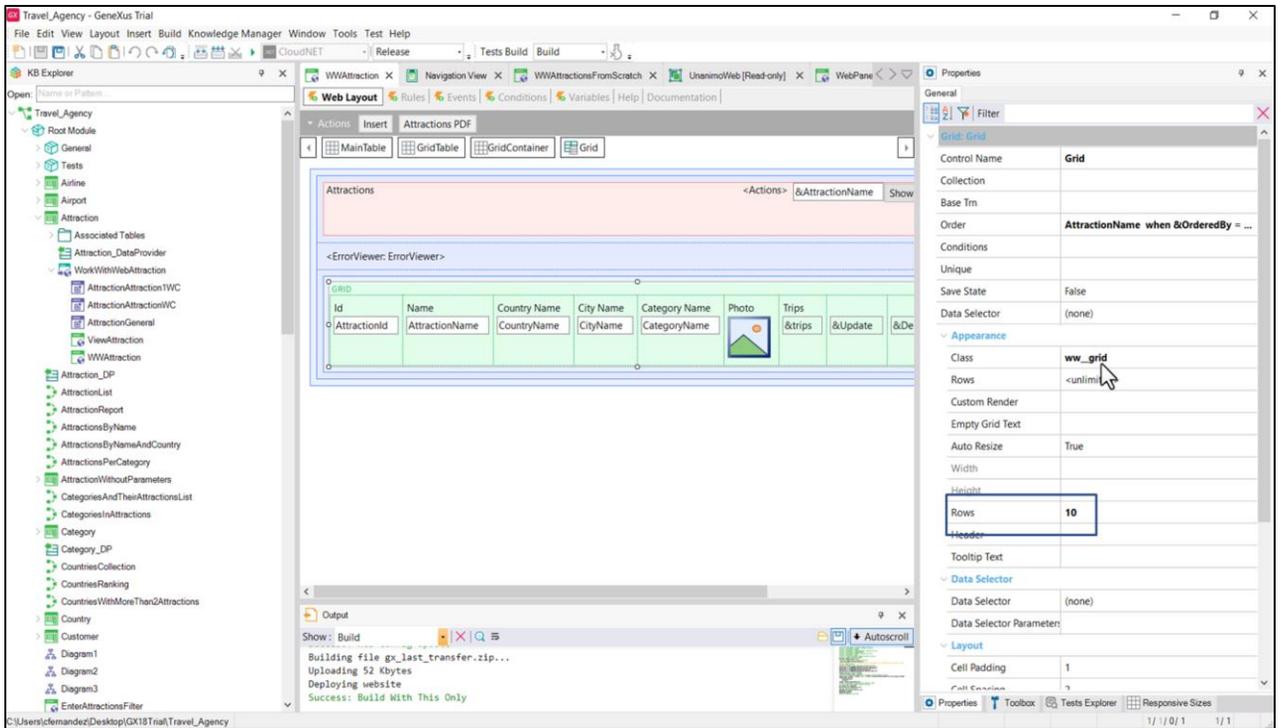
Attractions

Countries

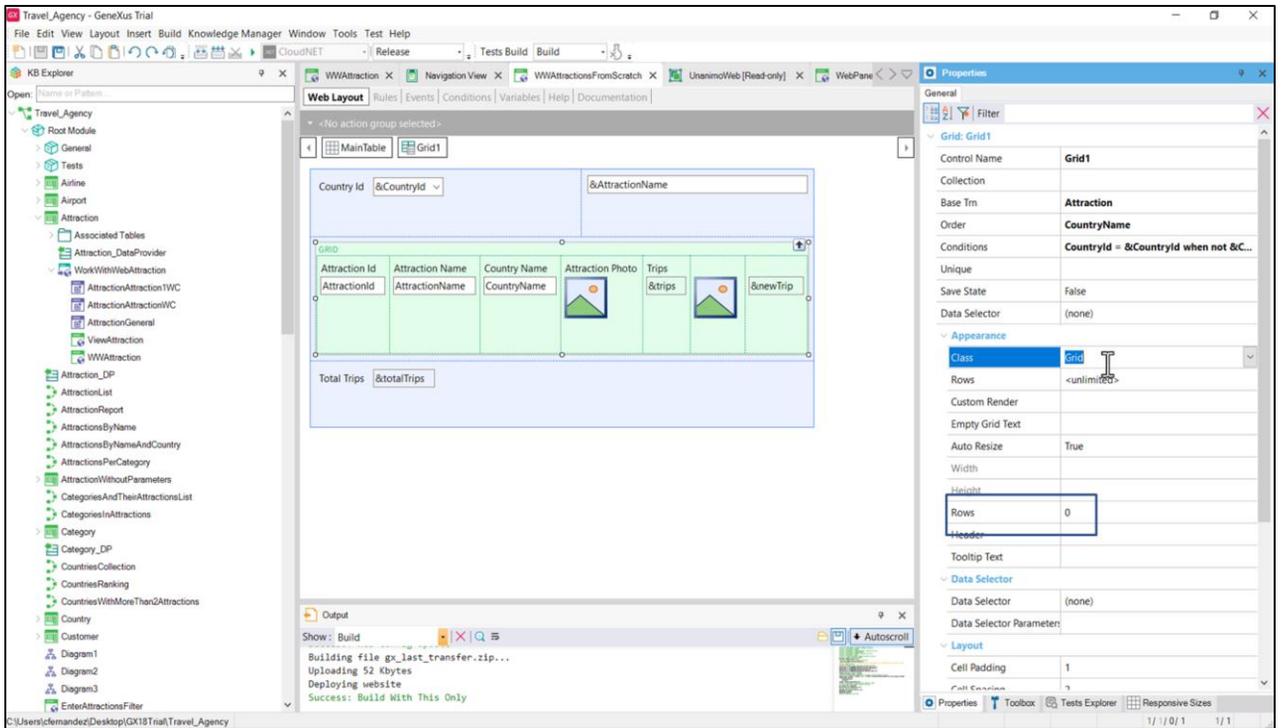
id	name	Country name	City name	Category name	Photo	trips		
107	Christ the Redemmer	Brazil	Rio De Janeiro	Monument		24	UPDATE	DELETE
111	Cinque Terre	Italy	Liguria	Tourist site		0	UPDATE	DELETE
106	Eiffel Tower	France	Paris	Monument		24	UPDATE	DELETE
110	Forbidden city	China	Beijing	Tourist site		0	UPDATE	DELETE
112	Glenfinnan Viaduct	Scotland	Glenfinnan	Tourist site		0	UPDATE	DELETE
113	London Towers	England	London	Monument		1	UPDATE	DELETE
116	Long Bridges	United States	San Francisco	Tourist site		0	UPDATE	DELETE
104	Louvre Museum	France	Paris	Museum		1	UPDATE	DELETE
109	Matisse Museum	France	Nice	Museum		13	UPDATE	DELETE
117	Meet the Emperor	China	Beijing	Monument		0	UPDATE	DELETE

<< < > >>

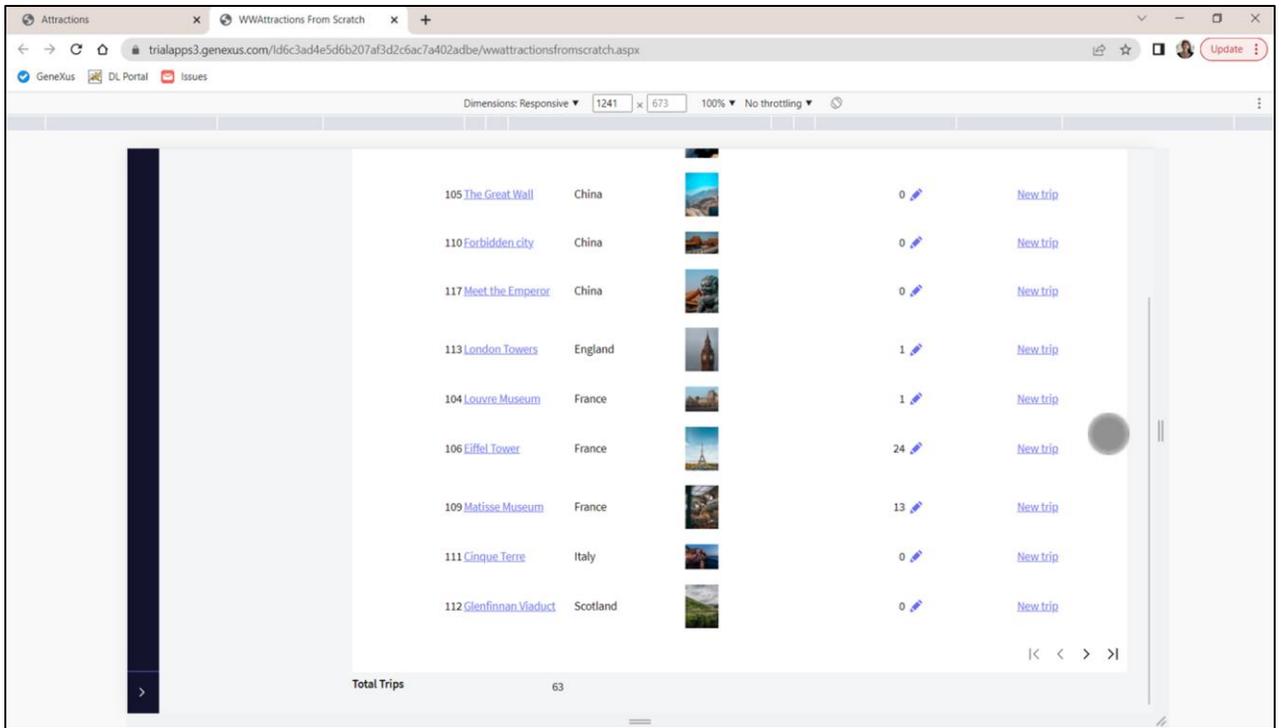
...enquanto o do WorkWith também tem fundo branco, mas ocupa toda a largura, e permite a paginação, carregando 10 de cada vez.



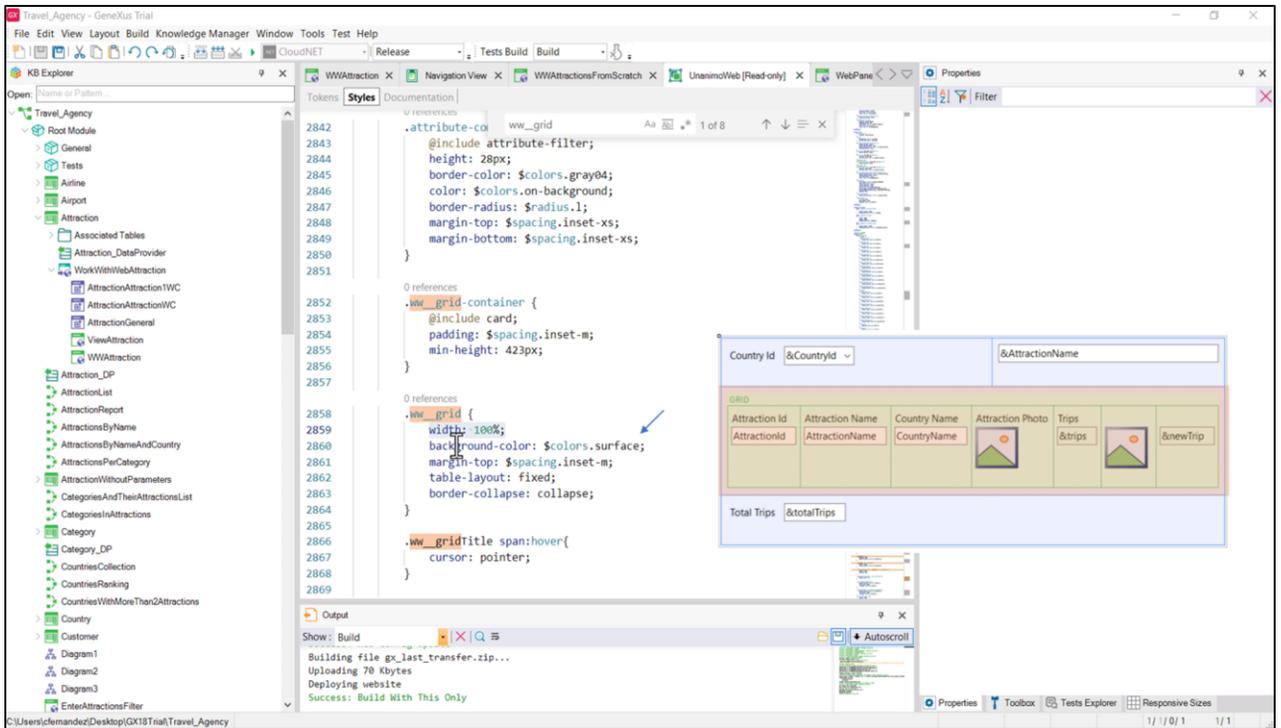
Vemos que o grid do WorkWith tem esta classe associada e tem aqui o número 10 de linhas por página.



No nosso caso, a classe do grid é a padrão. Vamos trocá-la pela do WorkWith e colocaremos 10 para o número de linhas por página.



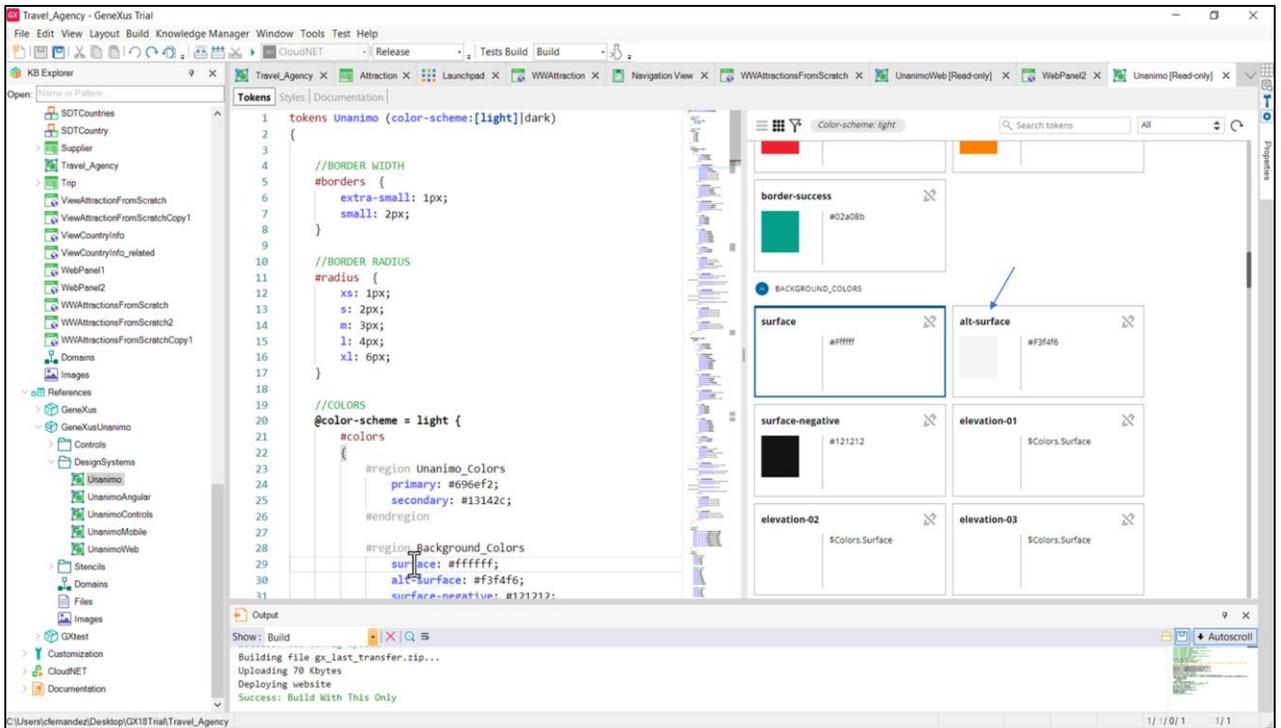
Agora sim está ocupando toda a largura e está paginando.



Vamos inspecionar as propriedades da classe...

Aqui vemos por que agora ocupa 100% da largura do container dentro do qual se encontra (que no nosso caso é esta célula da segunda linha da tabela Main, portanto coincide com a largura da tabela).

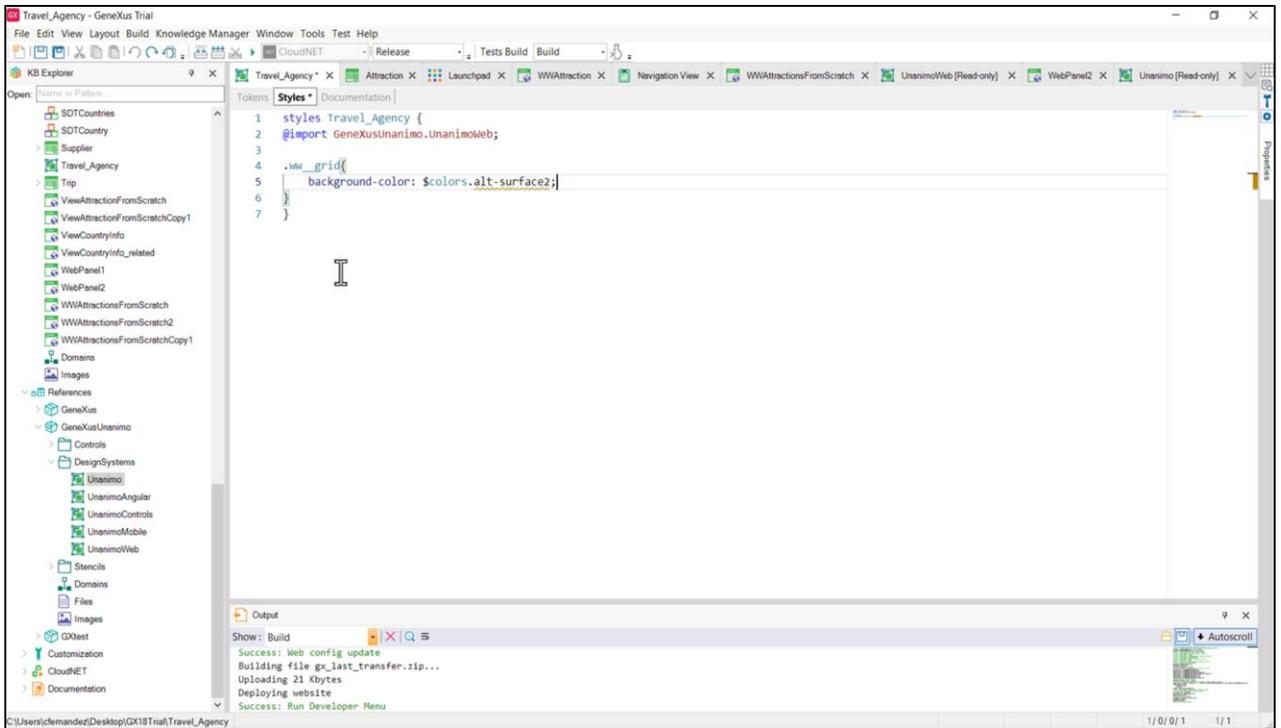
Também podemos ver que foi configurada como cor de fundo a que vem desse token de cor, que, pelo que vimos em execução, é branca.



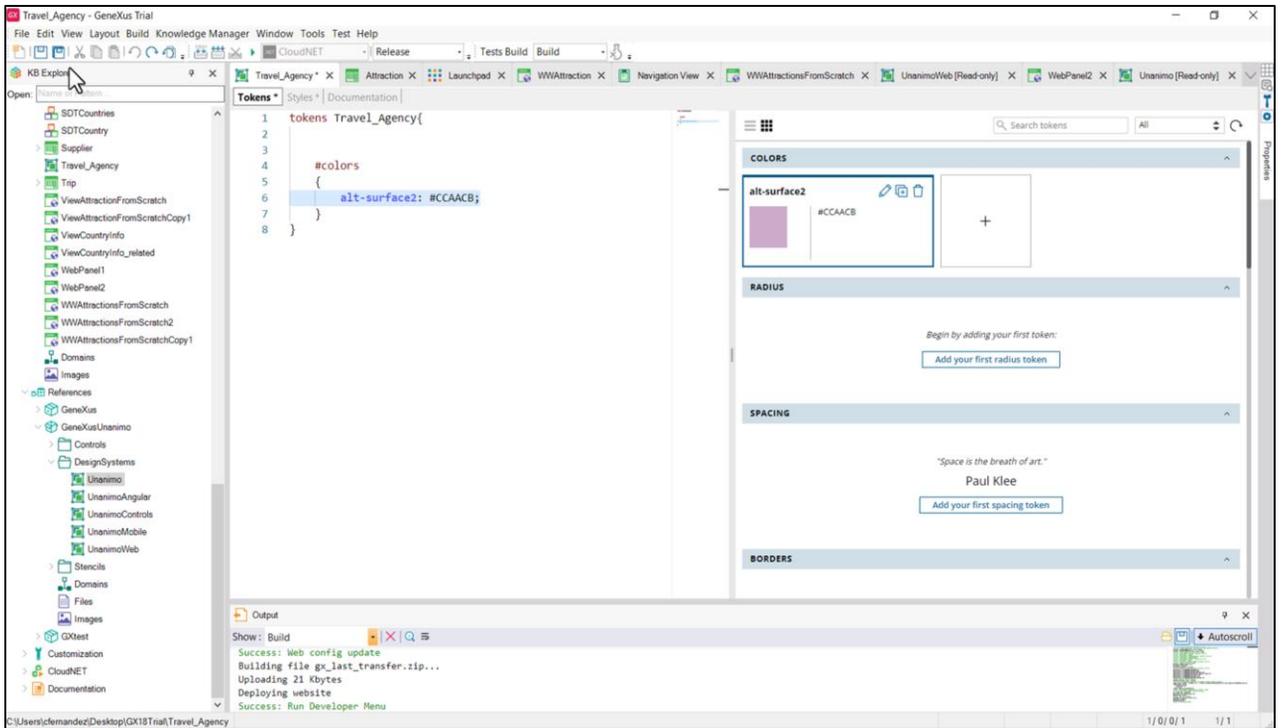
Não o temos definido neste DSO, mas sim neste outro que é importado por UnanimoWeb.

Aqui vemos o branco.

Se para todos os grids que tiverem essa classe definida quisermos mudar a cor de fundo por outra alternativa, por exemplo por esta outra ou por uma nova alternativa que teremos que definir...



Então no DSO de nossa aplicação, aquele que podemos personalizar, selecionamos a classe que queremos personalizar, e sobrescrevemos apenas o valor da propriedade background-color, atribuindo a ela agora como valor o do token alternativo, um novo, que teremos que definir aqui porque não existe na árvore de DSOs importados.



Então, aqui o adicionamos como token de cor. Assim, podemos reutilizar este token de cor como fundo alternativo para todas as classes que quisermos.

Vamos testar.

Attractions x WWAttractions From Scratch x

trialapps3.genexus.com/d6c3ad4e5d6b207af3d2c6ac7a402adbe/wwattractionsfromscratch.aspx

GeneXus DL Portal Issues

Dimensions: Responsive 1241 x 673 100% No throttling

GeneXus™ Travel Agency

Country Id

Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips
107	Christ the Redeemer	Brazil		24  New trip
105	The Great Wall	China		0  New trip
110	Forbidden city	China		0  New trip
117	Meet the Emperor	China		0  New trip
113	London Towers	England		1  New trip
104	Louvre Museum	France		1  New trip
106	Eiffel Tower	France		24  New trip
109	Matisse Museum	France		13  New trip

Pressionamos control F5...

Attractions

WWAttractions From Scratch

trialapps3.genexus.com/ld6c3ad4e5d6b207af3d2c6ac7a402adbe/wwattraction.aspx

GeneXus DL Portal Issues

GeneXus Travel Agency

UNANIMO

Attractions

Countries

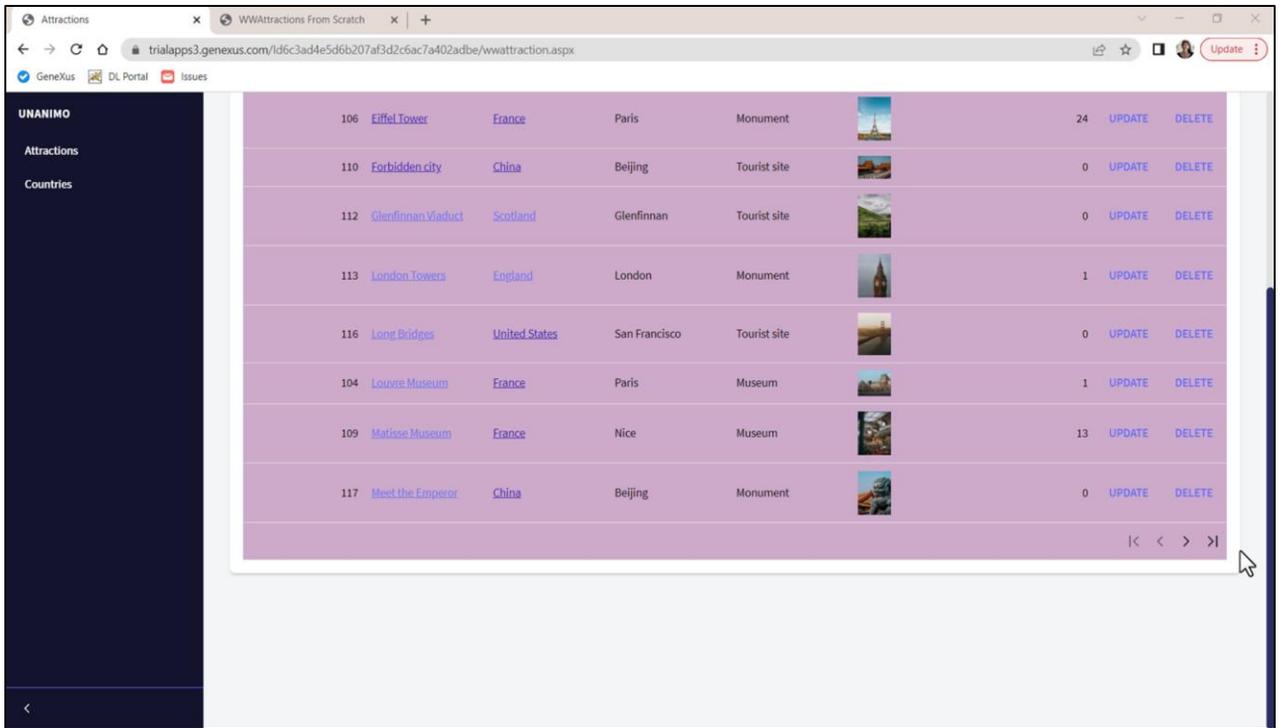
Attractions

INSERT ATTRACTIONS PDF

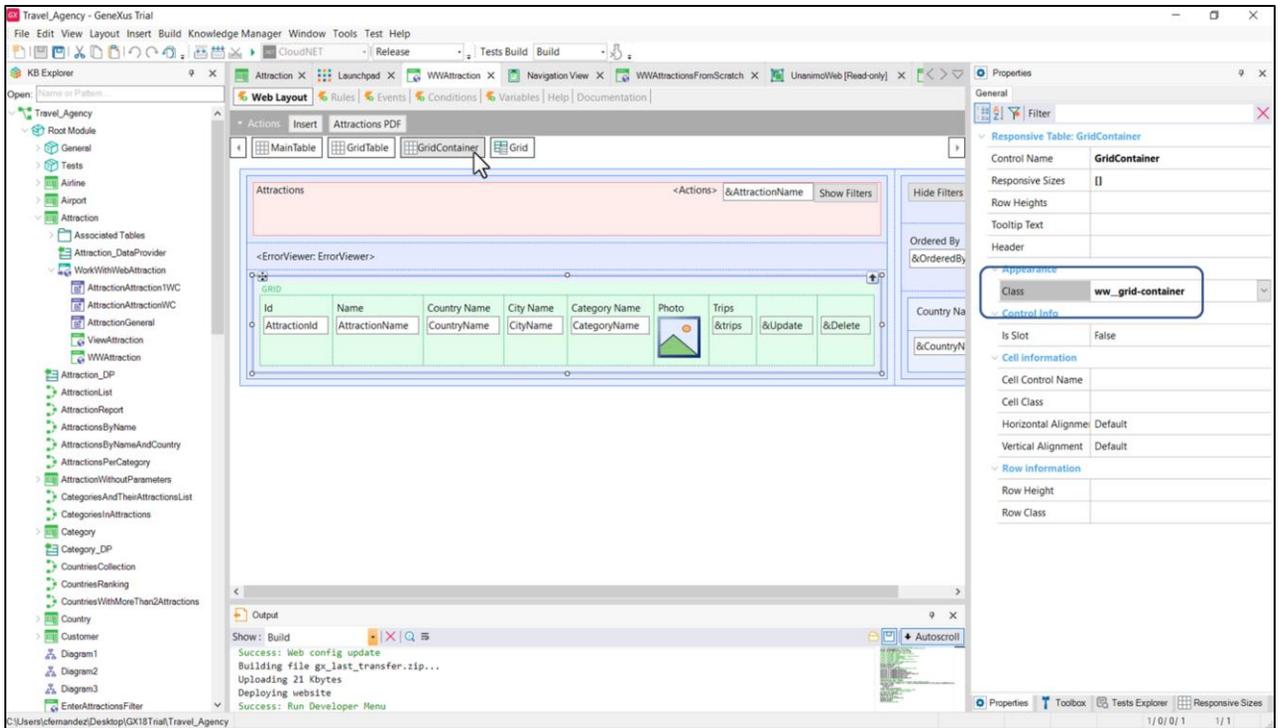
 ▼

Id	Name	Country Name	City Name	Category Name	Photo	Trips
107	Christ the Redeemer	Brazil	Rio De Janeiro	Monument		24 UPDATE DELETE
111	Cinque Terre	Italy	Liguria	Tourist site		0 UPDATE DELETE
106	Eiffel Tower	France	Paris	Monument		24 UPDATE DELETE
110	Forbidden city	China	Beijing	Tourist site		0 UPDATE DELETE
112	Glenfinnan Viaduct	Scotland	Glenfinnan	Tourist site		0 UPDATE DELETE
113	London Towers	England	London	Monument		1 UPDATE DELETE
116	Long Bridges	United States	San Francisco	Tourist site		0 UPDATE DELETE
104	Louvre Museum	France	Paris	Museum		1 UPDATE DELETE
109	Matisse Museum	France	Nice	Museum		13 UPDATE DELETE

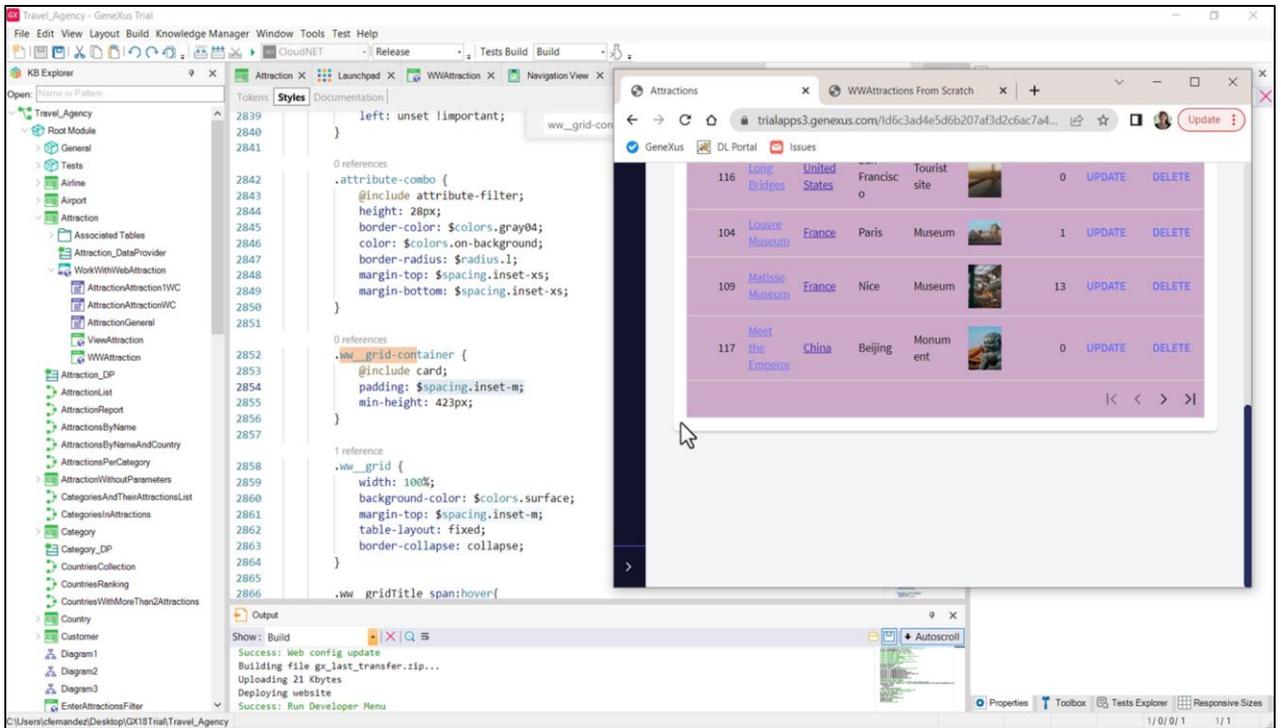
Vemos isso aplicado a ambos os grids.



De onde virá este branco com bordas arredondadas no WorkWith e com este espaçamento em relação ao grid?



Aqui vemos que o grid que tem a classe que acabamos de personalizar está dentro de uma tabela, esta, que tem associada esta outra classe. Aqui deve estar configurada essa estética. Vamos procurá-la no DSO UnanimoWeb...



Vemos aqui a propriedade do espaçamento: padding. Está indicando que o conteúdo do controle que tenha esta classe deverá ser espaçado em relação às bordas do controle nesta magnitude, aquela dada por este token, que estará em pixels.

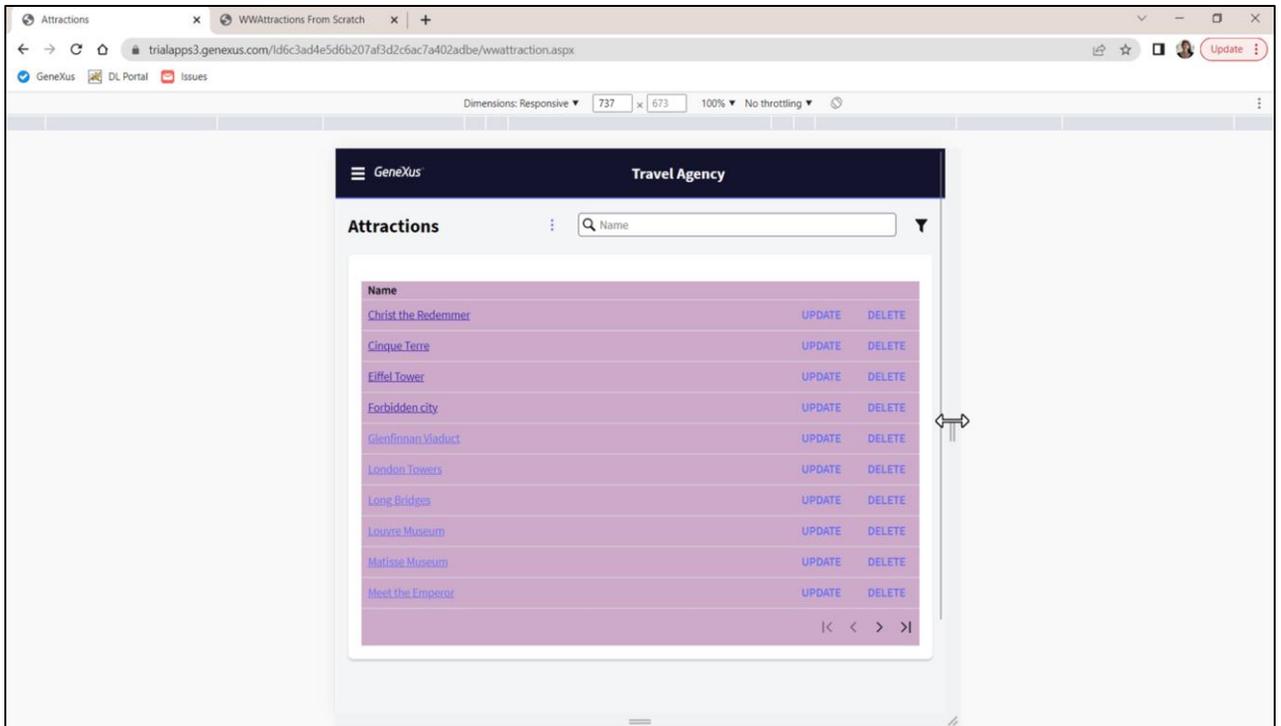
The screenshot displays a development environment with a code editor on the left and a browser preview on the right. The code editor shows the following CSS styles for a card component:

```
1134 .expanded {
1135   margin-left: 57px;
1136   width: calc(100% - 74px);
1137   transition: margin-left .3s, width .2s ease;
1138 }
1139 0 references
1140 .table-login {
1141   width: 360px;
1142   padding: 45px 50px 45px 50px;
1143   background: $colors.surface;
1144   border-radius: $radius.xl;
1145   box-shadow: $shadows.xxs;
1146 }
1147 0 references
1148 .card {
1149   background-color: $colors.surface;
1150   border-radius: $radius.xl;
1151   box-shadow: $shadows.xxs;
1152 }
1153 0 references
1154 .card-body {
1155   padding: $spacing.inset-m;
1156 }
1157 0 references
1158 .card-heading {
1159   box-shadow: 0px 1px 1px $colors.gray01;
1160   padding: $spacing.inset-m;
1161   font-family: $fonts.primary-bold;
1162   font-size: $fontSizes.s;
1163   letter-spacing: $spacing.letter-spacing-s;
1164 }
```

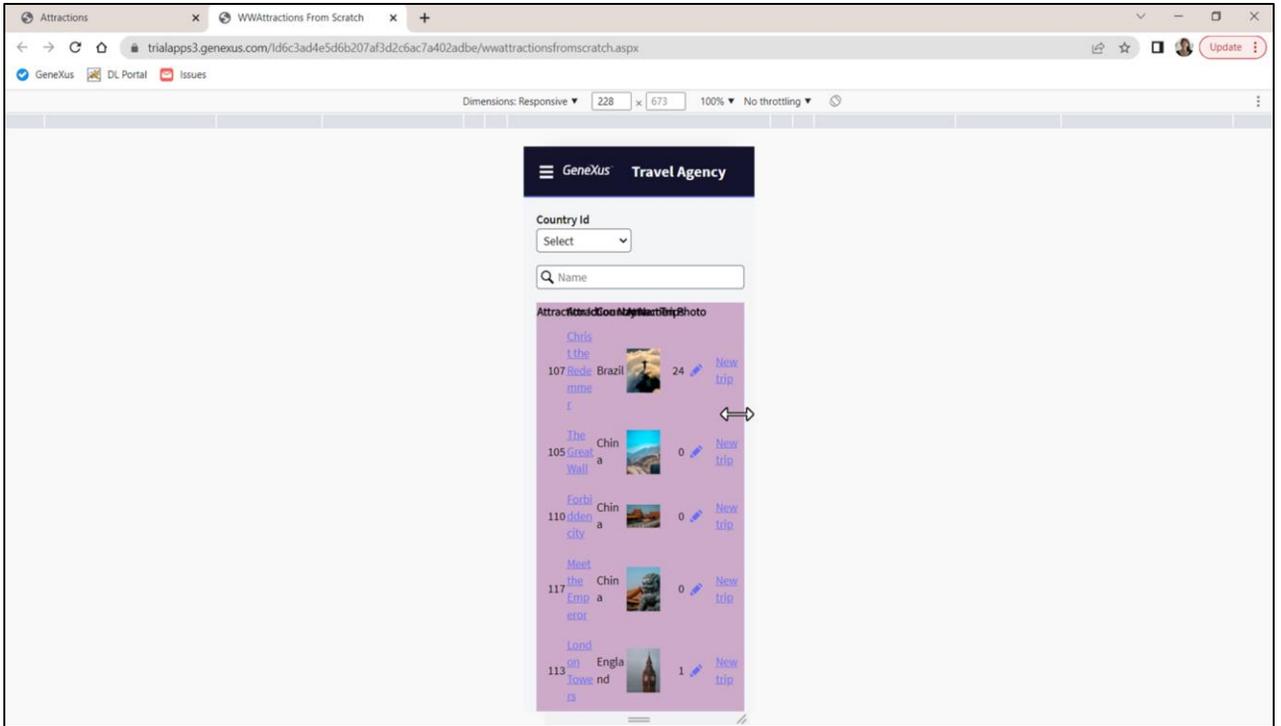
The browser preview shows a table of attractions with the following data:

116	Long Bridges	United States	Francisco	Tourist site		0	UPDATE	DELETE
104	Louvre Museum	France	Paris	Museum		1	UPDATE	DELETE
109	Matisse Museum	France	Nice	Museum		13	UPDATE	DELETE
117	Meet The Emperor	China	Beijing	Monument		0	UPDATE	DELETE

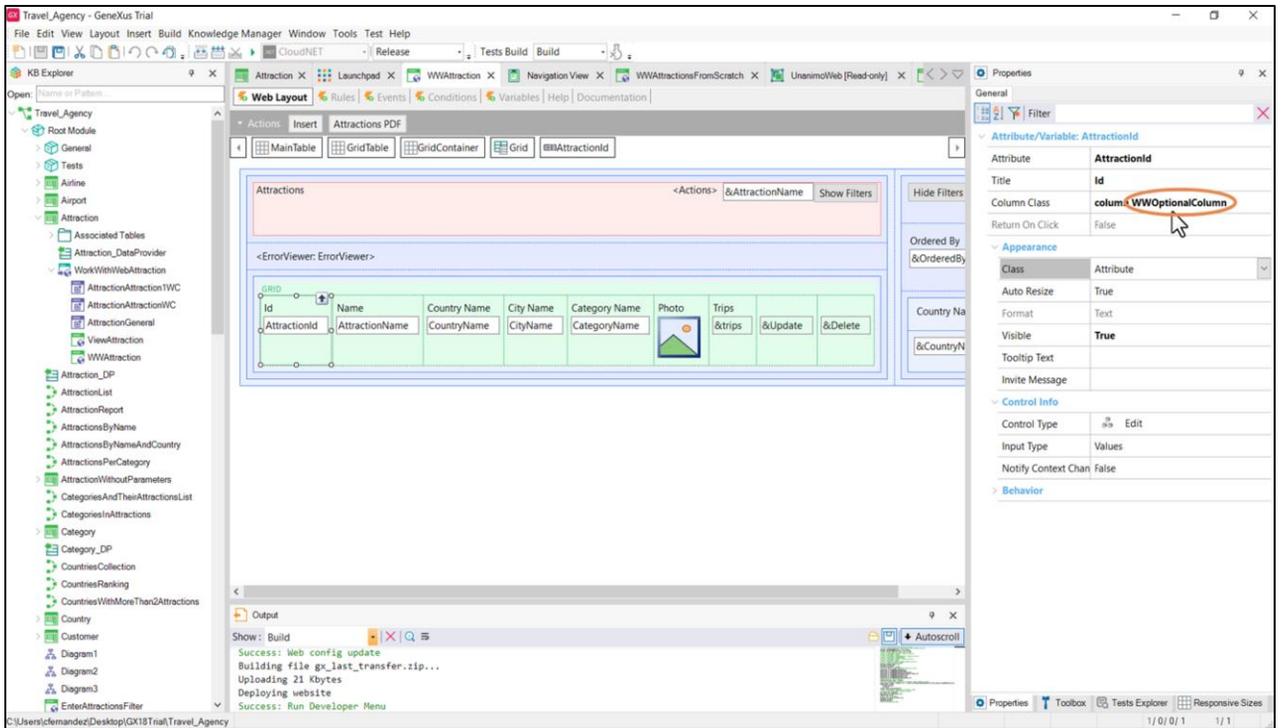
Mas esta classe também está incluindo esta outra, ou seja, suas propriedades. Aqui vemos a cor de fundo (a do token Surface, que era branca) e tanto a propriedade para as bordas arredondadas quanto para esta sombra que é vista.



Agora vamos observar outra questão interessante: enquanto para o grid do WorkWith (vamos pressionar F12) existem colunas que deixam de ser exibidas ao diminuir a largura de tela e apenas permanecem a do nome de atração e as ações, para nosso grid não: todas são exibidas mesmo que não se encaixem bem.

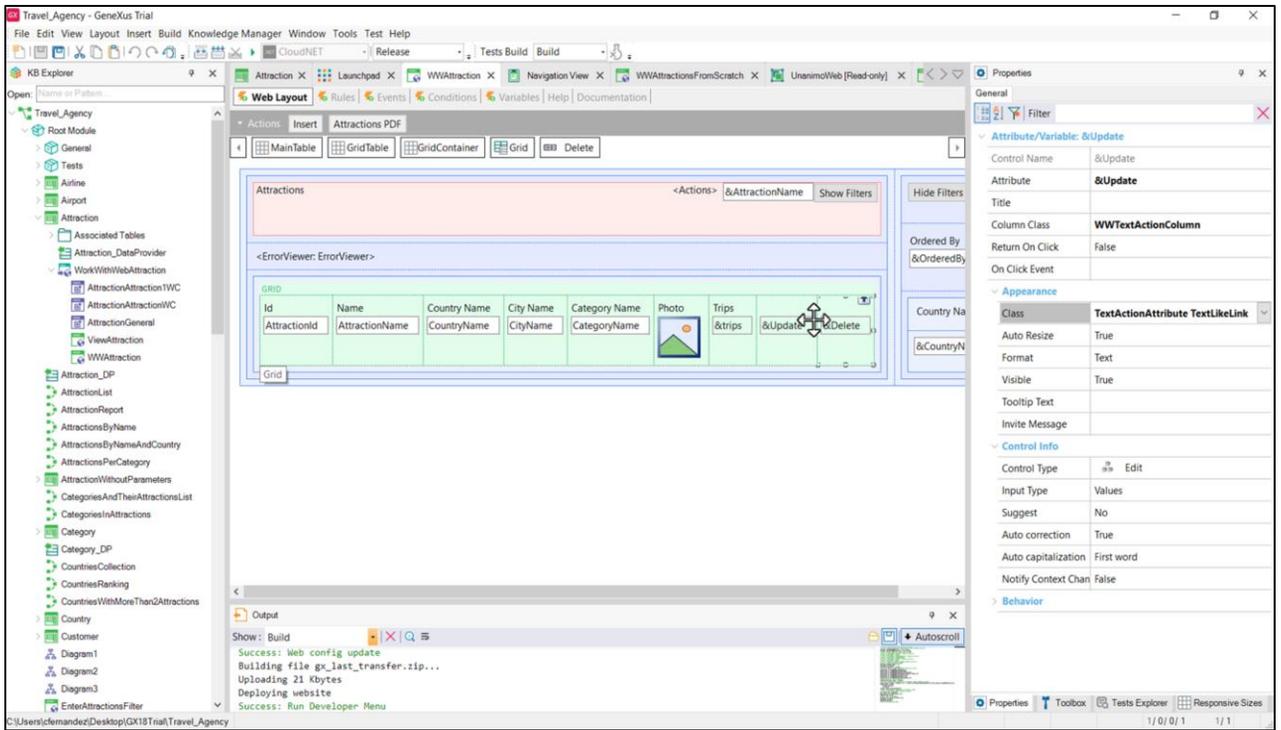


Por quê?

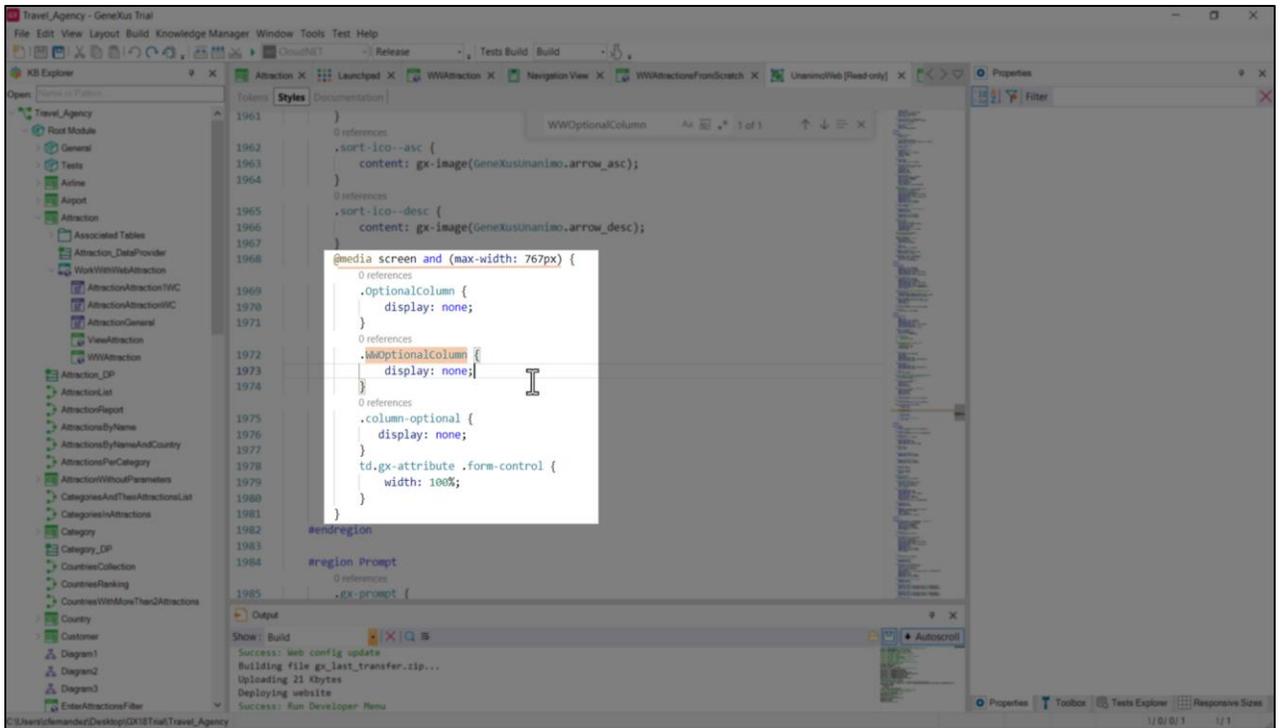


É que essas colunas possuem classes associadas que comandam se são exibidas ou não, de acordo com o tamanho de tela.

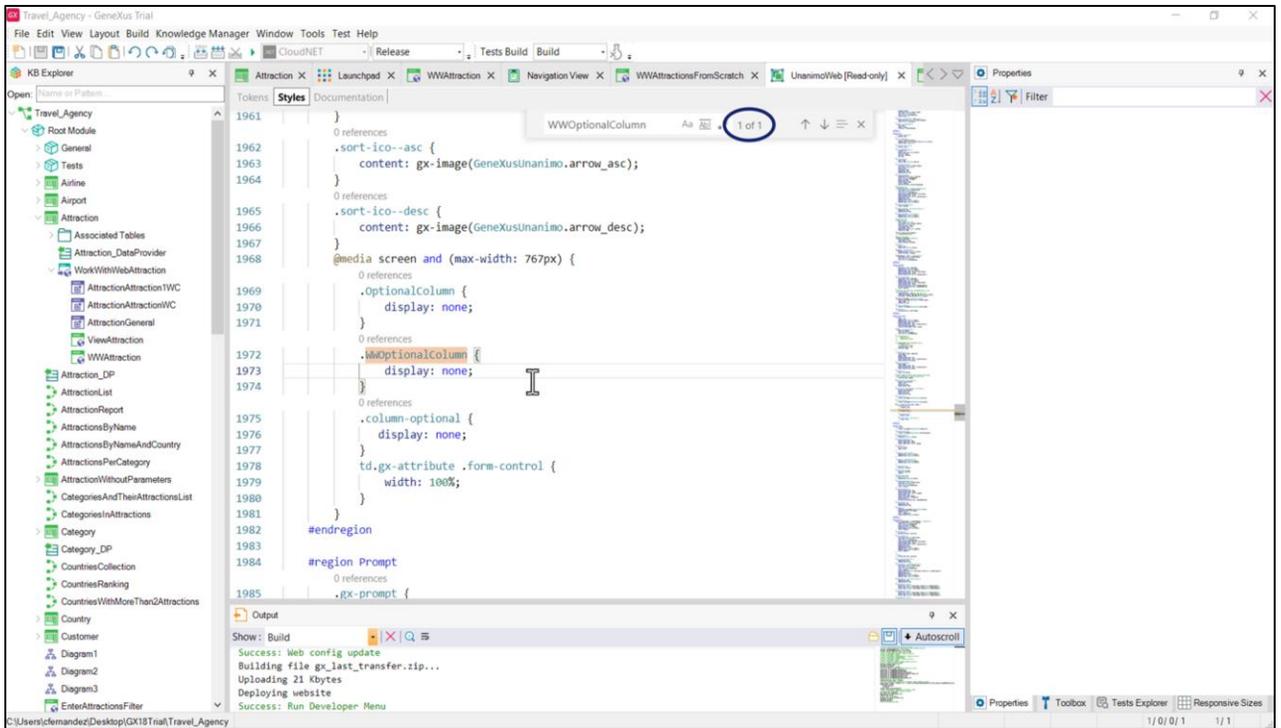
Assim, a coluna do Id tem duas classes (então aplicarão as propriedades de ambas): a chamada "column" que também a tem a coluna do nome de atração, mas também tem esta outra... que se repete para todas as colunas que vimos que, para o tamanho pequeno de tela, se ocultavam.



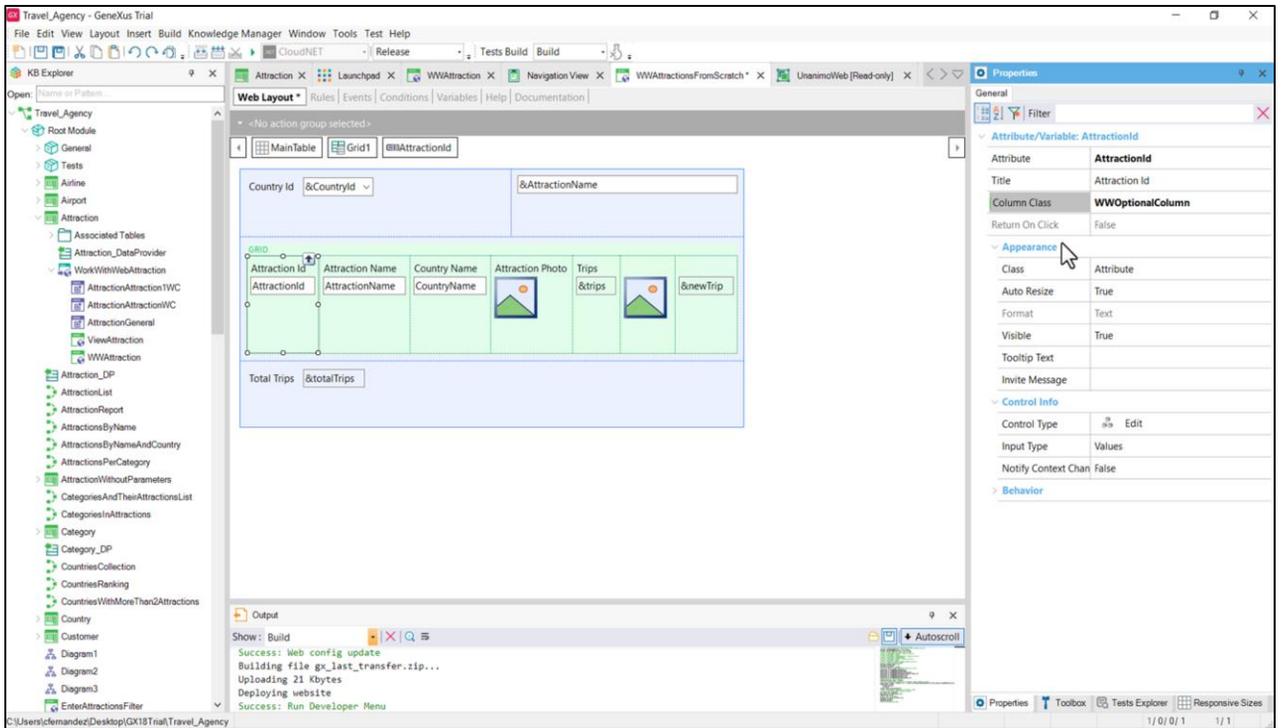
As das ações, não casualmente, não possuem essa classe.



Vamos procurar a classe... Aqui a encontramos, com a propriedade display como none. Mas se observarmos **COM** atenção, esta classe está sendo definida desta forma apenas quando é aplicada esta regra, que é a que permite condicionar a saída a uma tela de largura máxima de 767 pixels. Se o tamanho de tela for maior, não é válida esta definição.

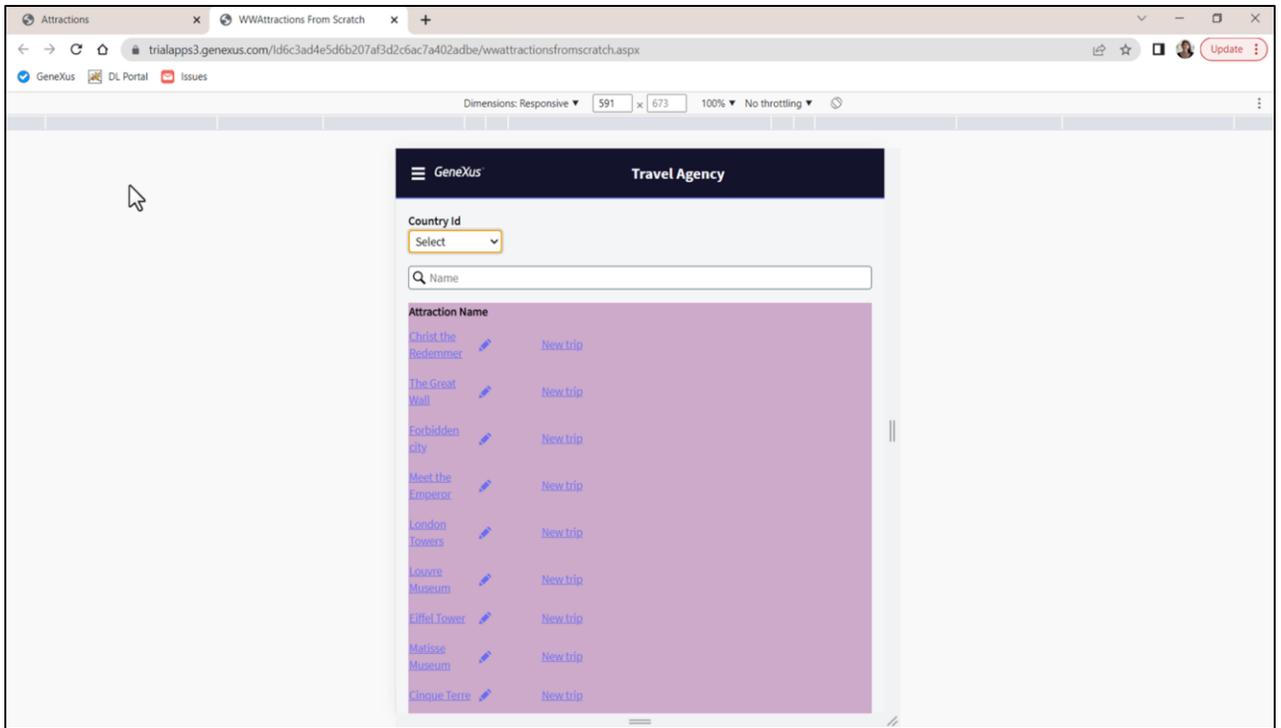


Na verdade, como esta é a única ocorrência desta classe nesta aba de Styles, isso significa que para outro tamanho de tela não existem propriedades para a classe e é por isso que neste caso vemos as colunas serem exibidas em tela.



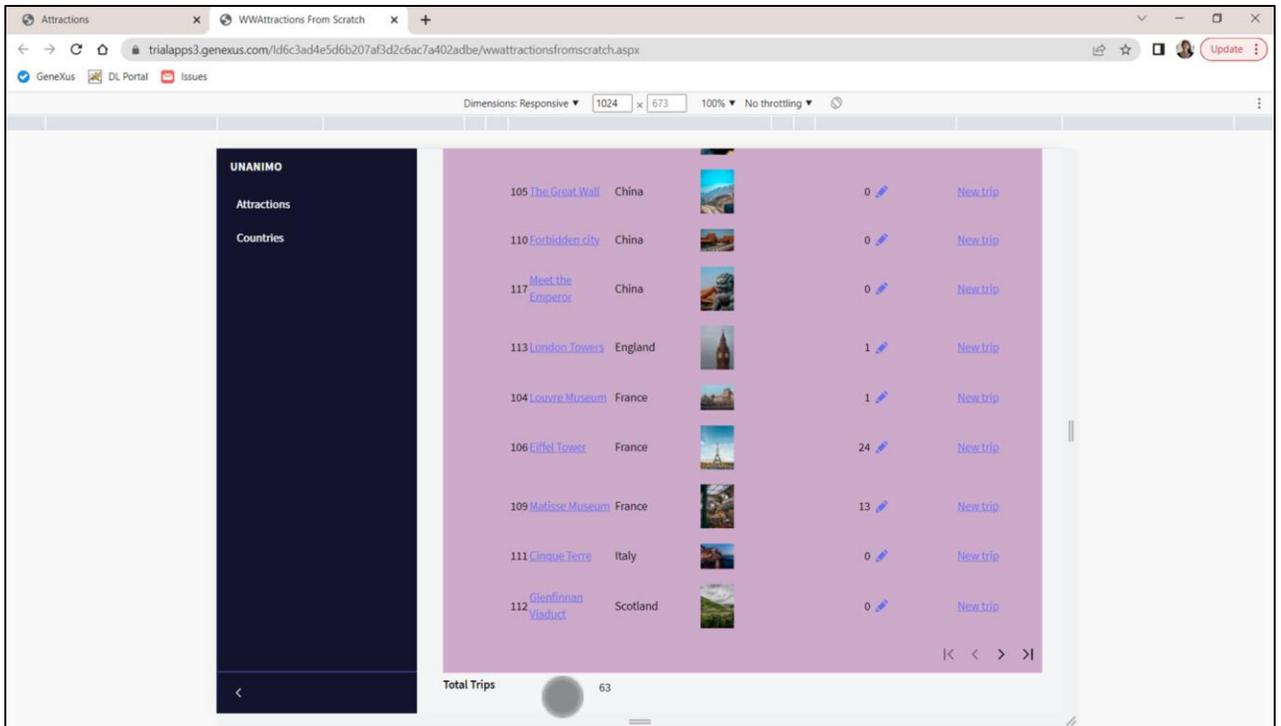
Vamos adicionar em nosso grid esta propriedade para as colunas que queremos que não sejam exibidas em tamanho pequeno.

Geramos...



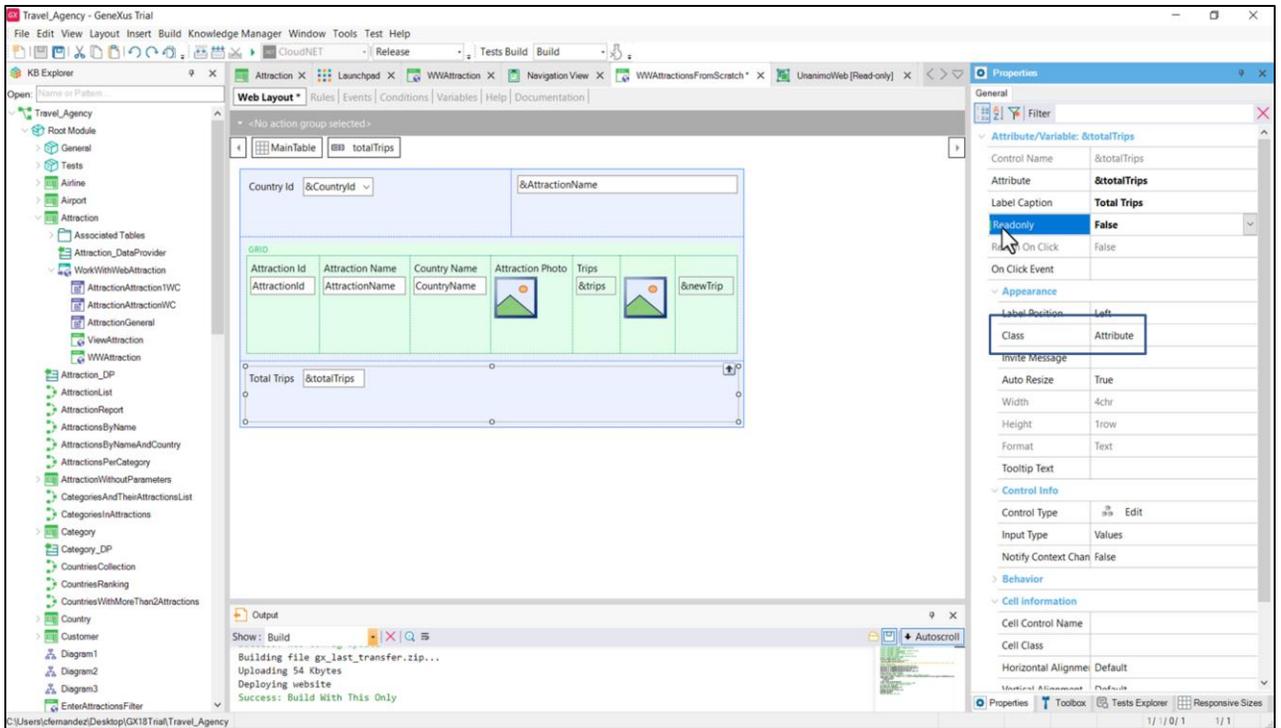
Antes de atualizar, são vistas todas as colunas, e em vez disso, uma vez que atualizamos... são exibidas apenas as que acabamos de manipular, quando a largura de tela é maior que 767 pixels.

Podemos continuar investigando as diferenças entre nosso grid e o do WorkWith para entender como uniformizar o design.



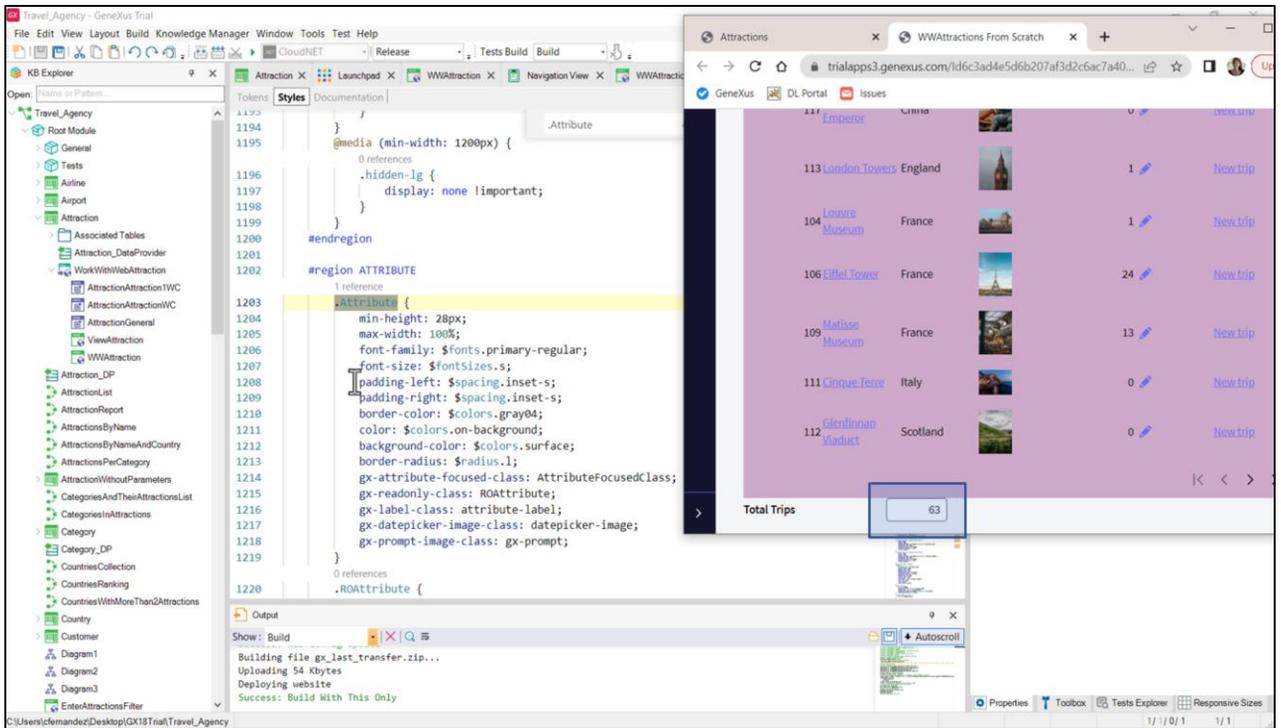
Mas para terminar de entender melhor como estilizar nossos controles, observemos a variável que temos em nosso painel.

Suponhamos que queremos mudar a cor do rótulo, que aqui é um preto destacado, e que o conteúdo da variável também apareça em outra cor e contrastada sobre outra cor de fundo.



As variáveis por padrão são de entrada nos Web panels. Tínhamos alterado para readonly, mas vamos voltar por um momento para colocar o valor default.

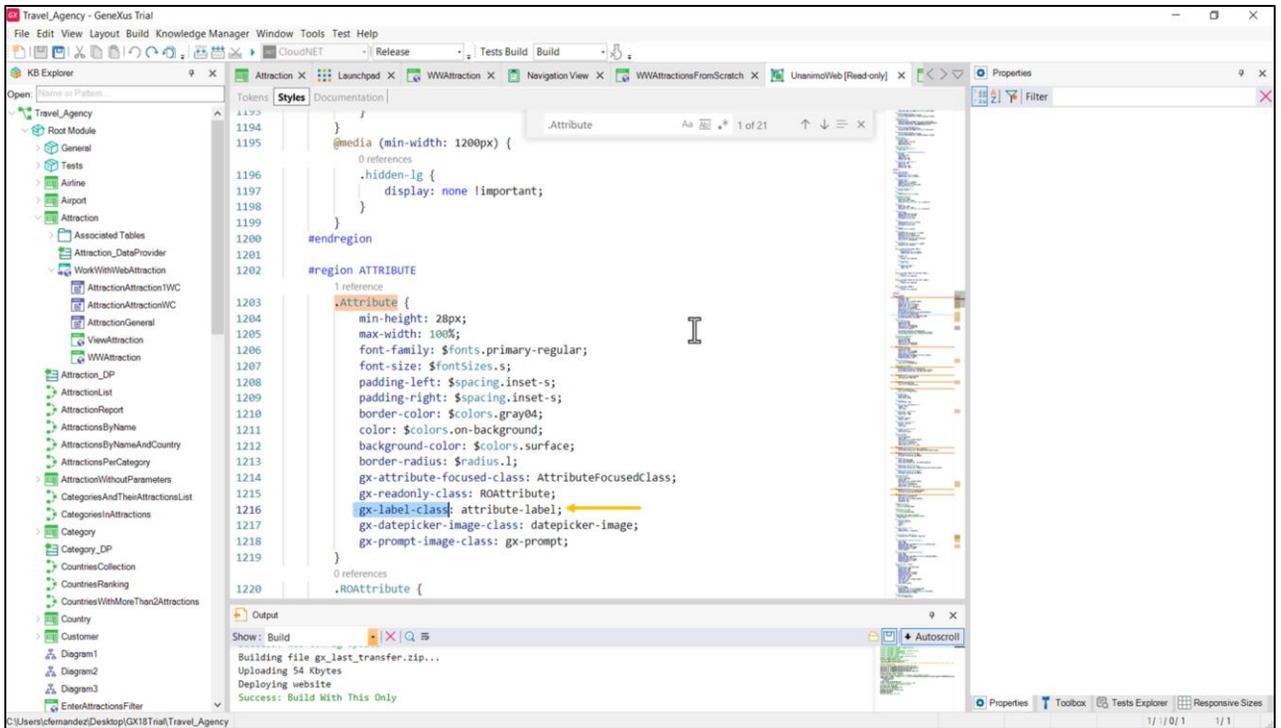
Solicitamos o build e, enquanto isso, vamos ver se tem a classe Attribute associada.



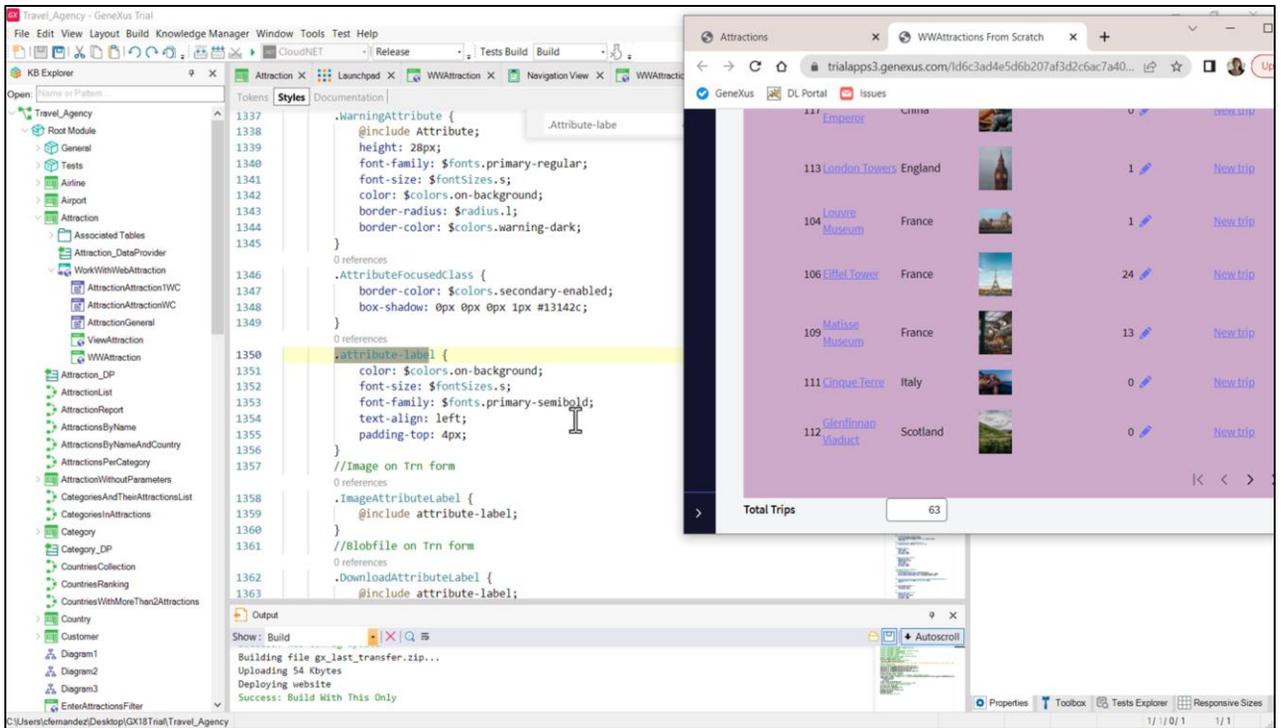
Se a procurarmos, veremos todas estas propiedades configuradas. Executamos... atualizamos...

Font-family, font-size, padding left e right, border-color, color, background-color, border-radius... estão definindo tudo isso, que é o campo da variável.

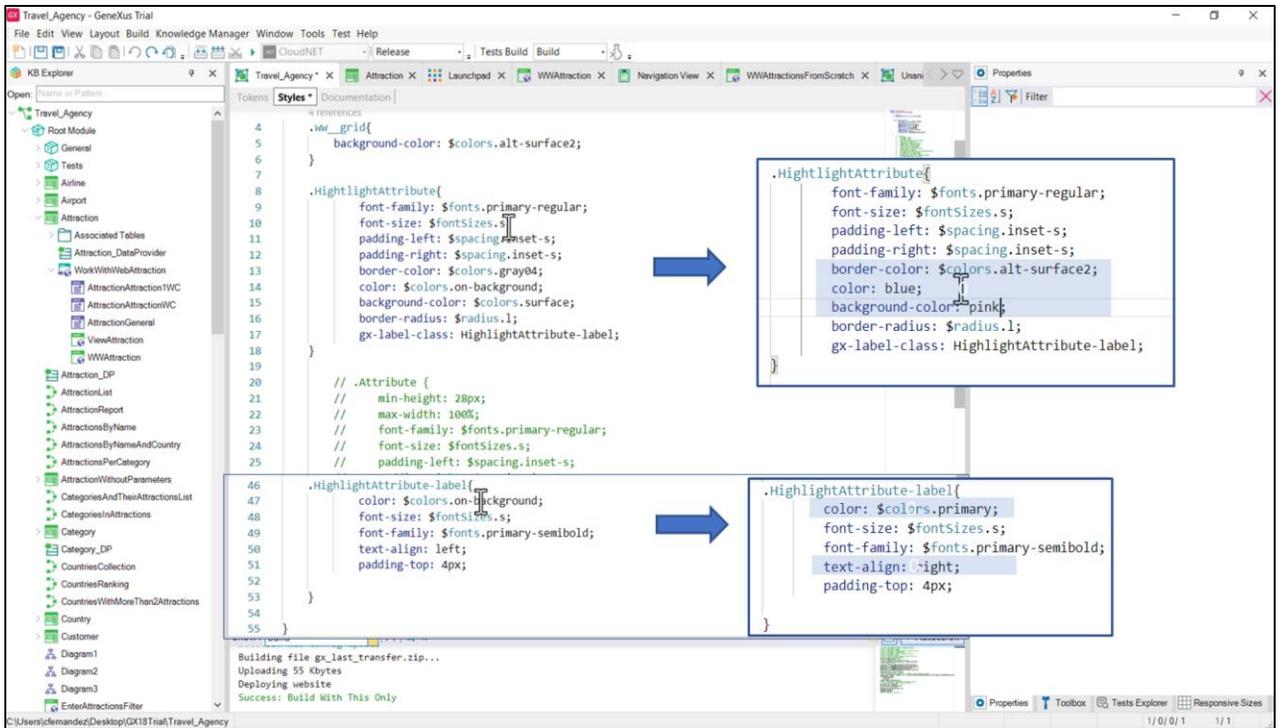
E onde se configura tudo relacionado ao rótulo? Em outra classe, uma que dá estilo ao rótulo. E como dizemos a esta classe que quem comanda o desenho do rótulo é essa outra?



Através desta propriedade do GeneXus... Aqui estamos indicando que a classe chamada attribute-label é a que dá estilo ao rótulo. Vamos buscá-la...



Por exemplo, aqui vemos a fonte em semibold, o texto alinhado à esquerda e com um padding em relação à borda superior da célula onde se encontra o controle, de 4 pixels...

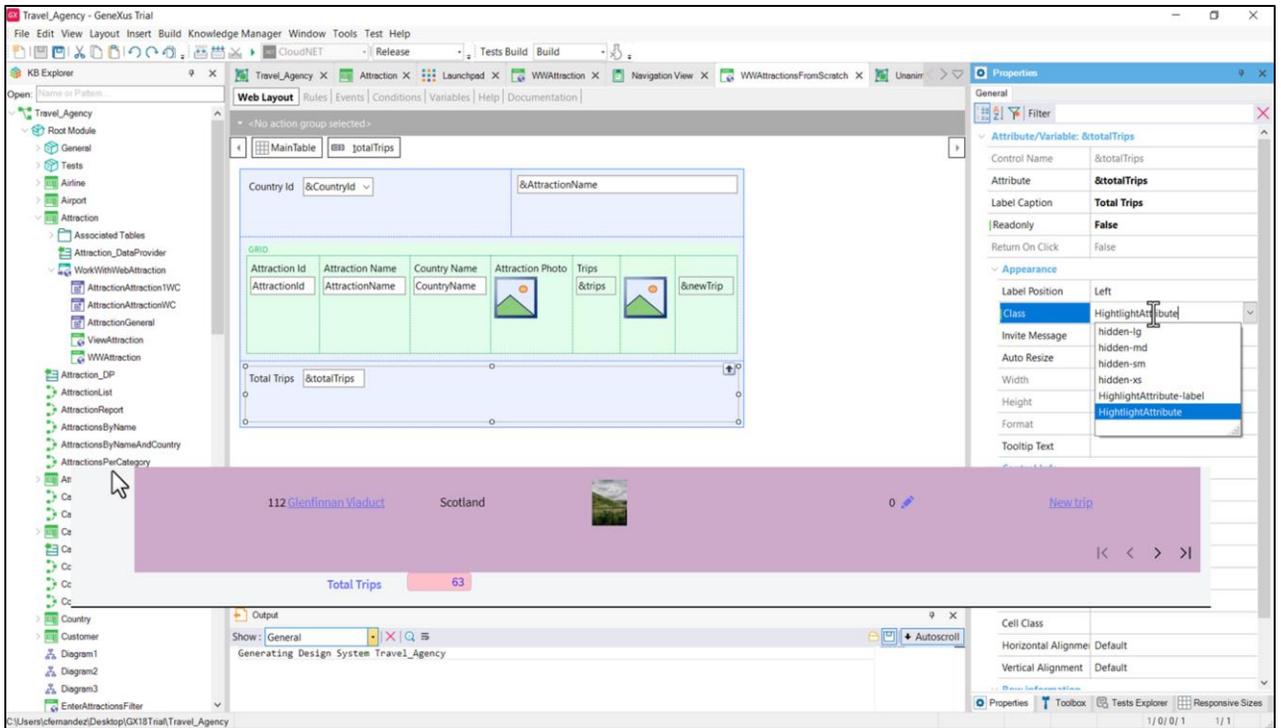


Vamos copiar ambas as classes em nosso DSO e assim vamos fazendo mudanças.

Poderíamos modificar estas aqui, mas isso afetaria todos os controles que as possuem por classe. Melhor criar classes especiais para personalizar nosso caso de uso especial que corresponderá a um atributo ou variável destacado. Então, vamos criar uma classe `HighlightAttribute`.

Vamos copiar todas estas propriedades da `Attribute` e a da classe do rótulo. Nós a chamaremos assim. Vamos deixar comentada a classe `Attribute` que não modificaremos em nada. E vamos selecionar a nova para dar estilo ao rótulo. E vamos copiar as propriedades desta outra.

Agora vamos fazer alterações. Por exemplo para as cores... e vamos alinhar o texto do rótulo à direita. Gravamos o DSO.



E vamos mudar agora a classe da nossa variável para a nova, e vamos executar.

Vemos todas as mudanças.

GeneXus by Globant

<No action group selected>

MainTable totalTrips

Country Id &CountryId &AttractionName

GRID

Attraction Id	Attraction Name	Country Name	Attraction Photo	Trips	&newTrip
AttractionId	AttractionName	CountryName		&trips	
109	Matisse Museum	France		13	New trip
111	Cinque Terre	Italy		0	New trip
112	Glenfinnan Viaduct	Scotland		0	New trip

Total Trips &totalTrips

Properties

General

Filter

Attribute/Variable: &totalTrips

Control Name: &totalTrips

Attribute: &totalTrips

Label Caption: Total Trips

Readonly: True

Return On Click: False

Appearance

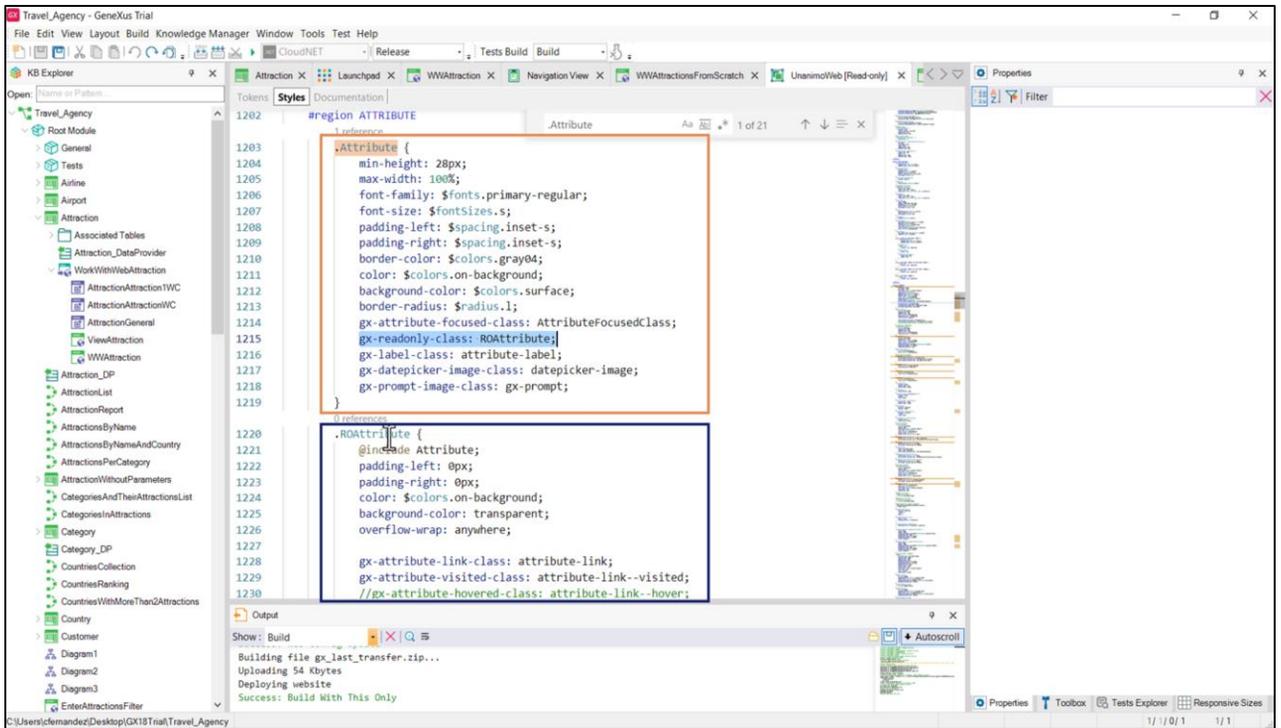
Label Position: Left

Class: HighlightAttribute

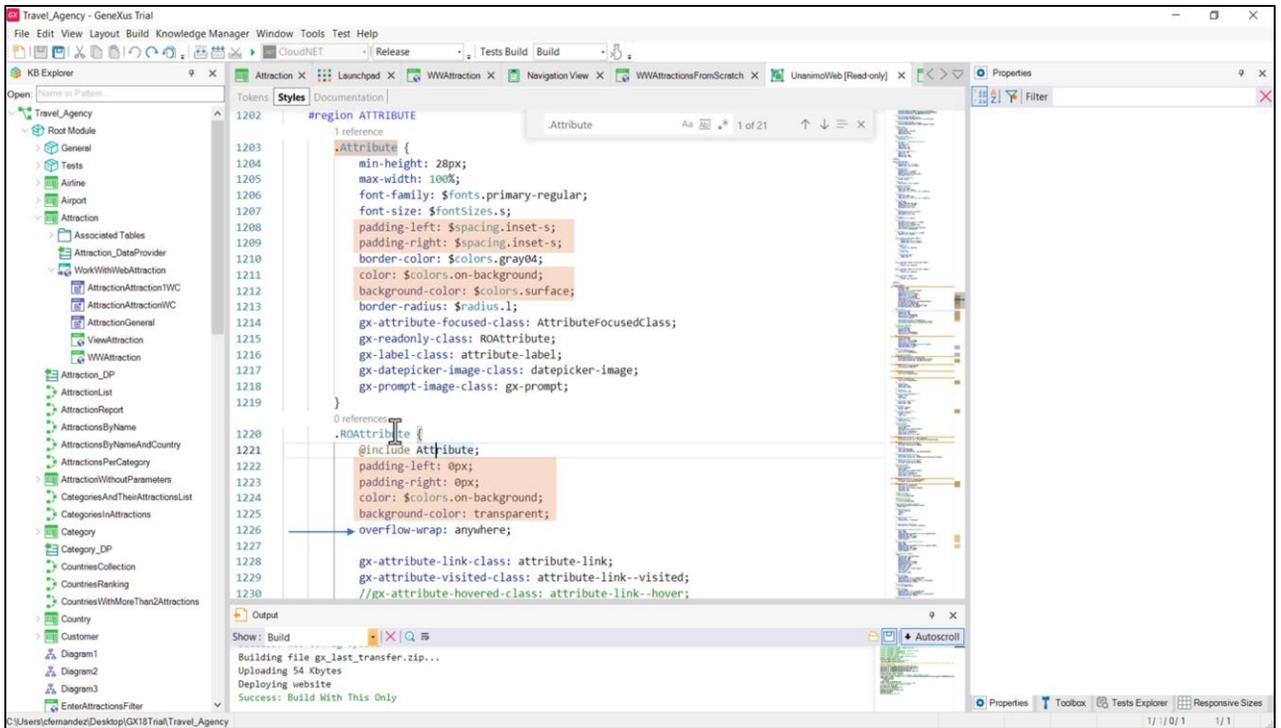
Total Trips 63

Total Trips 63

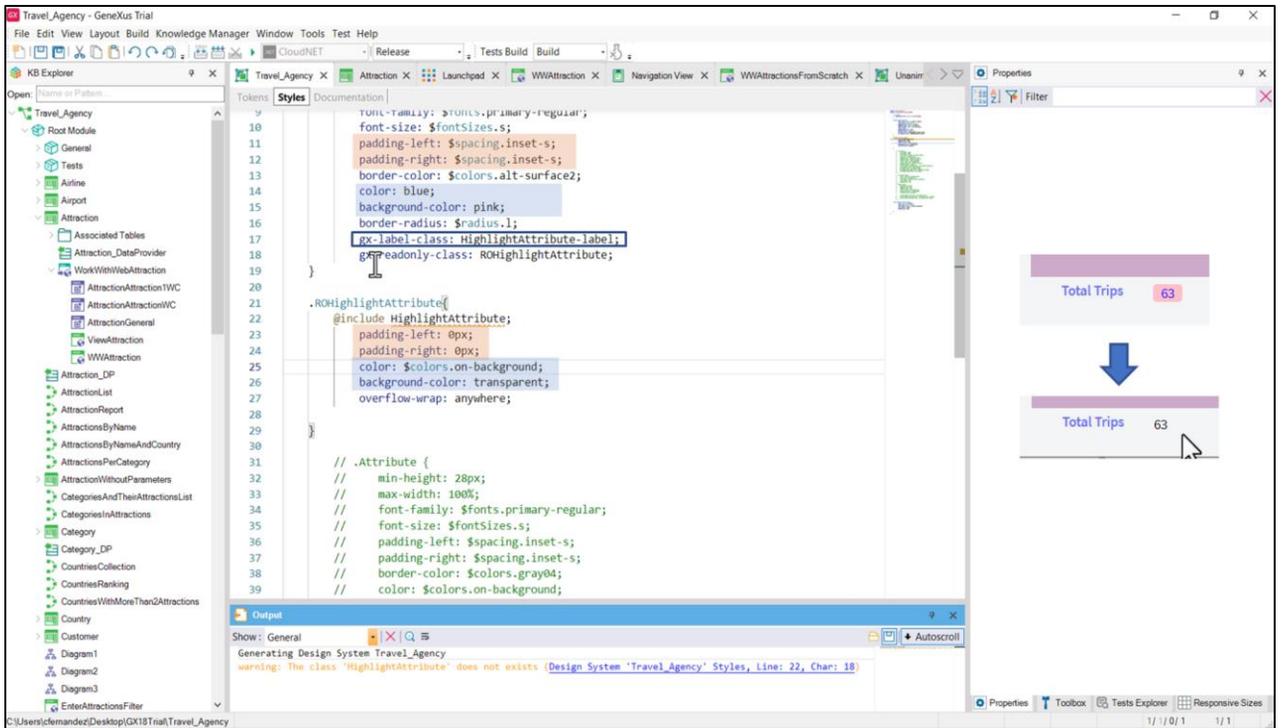
Qual será a aparência da variável se agora a alterarmos para Readonly?



Se formos observar as propriedades da classe predefinida Attribute, veremos que possui esta propriedade: `gx-readonly-class`, que é a que está definindo parte do estilo quando esta classe Attribute está aplicada a um atributo ou variável que é readonly. Ou seja, se não for readonly, então esta propriedade não terá efeito algum sobre o controle, mas se for, então às propriedades da classe Attribute serão adicionadas as desta outra para decidir o estilo, sobrescrevendo aquelas que são repetidas.



Aqui vemos que é incluído tudo o que tem a classe Attribute, mas são sobrescritas estas propriedades e é adicionada esta outra.



Vamos copiar a classe em nosso DSO para tê-la em vista.

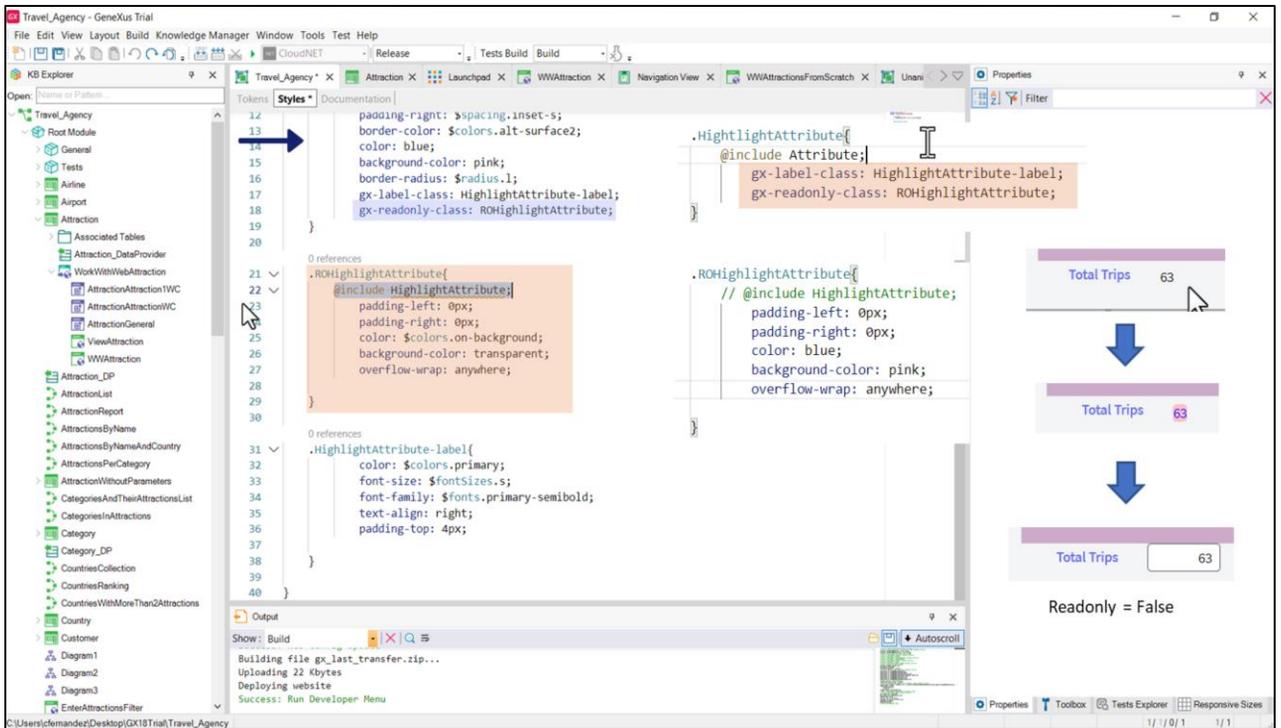
Vamos adicionar à nossa classe HighlightAttribute a propriedade que indicará a classe quando o controle atributo ou variável for readonly. Vamos dar-lhe este nome.

E aqui a definimos. Podemos incluir a classe HighlightAttribute, para garantir as propriedades compartilhadas, e modificar e/ou adicionar as que quisermos. Por exemplo, vamos copiar estas...

É removido o padding left e right, o deixamos em 0, sobrescrevemos a color e background-color.

Vamos testar... Atualizamos... E vemos como agora estão sendo aplicadas estas propriedades.

Deixemos limpo o DSO.

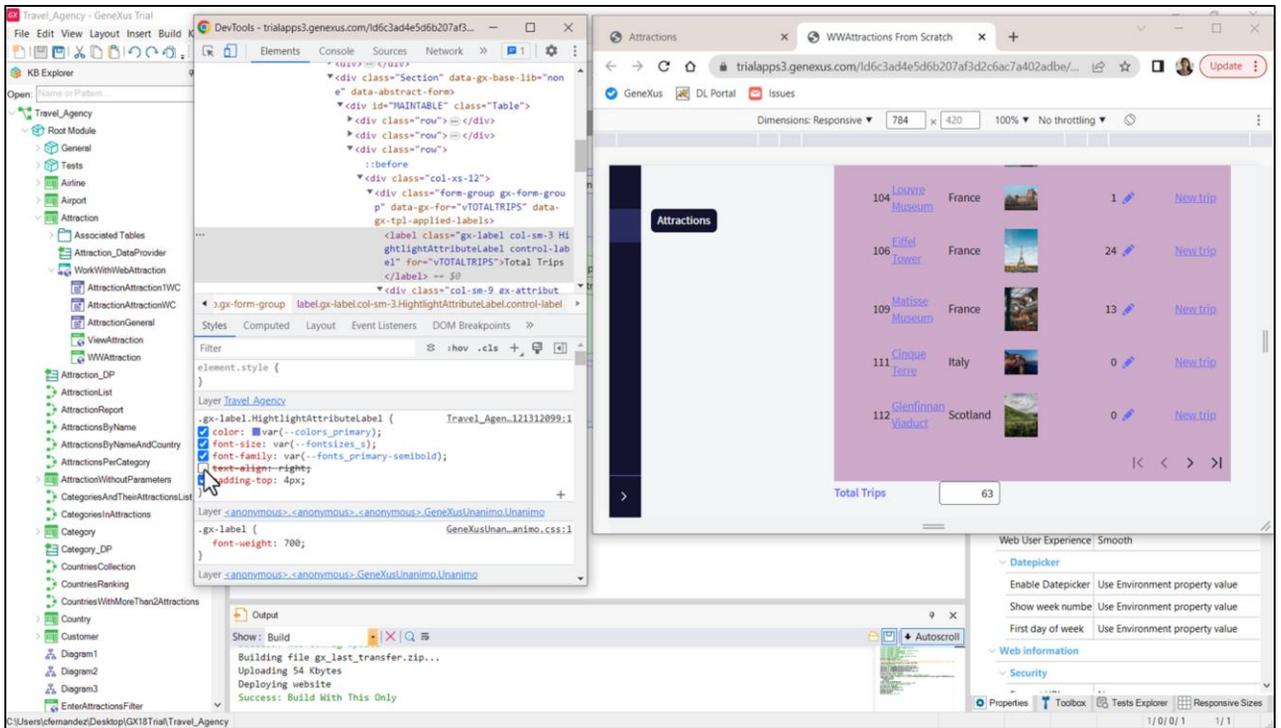


Resumindo: a classe `HighlightAttribute` será aplicada à variável sempre, mas quando é `readonly` aplica também esta outra (pelo que na verdade não era necessário incluir a primeira classe na segunda).

Agora vamos fazer a seguinte alteração: vamos colocar a cor blue e o fundo rosa quando for `readonly` o controle, e em vez disso utilizamos o que vem predefinido na classe `Attribute` em `UnanimoWeb` para a classe em geral, embora sobrescrevendo, sim, estas duas propriedades, porque queremos que sejam aplicadas nossas duas classes nesses casos.

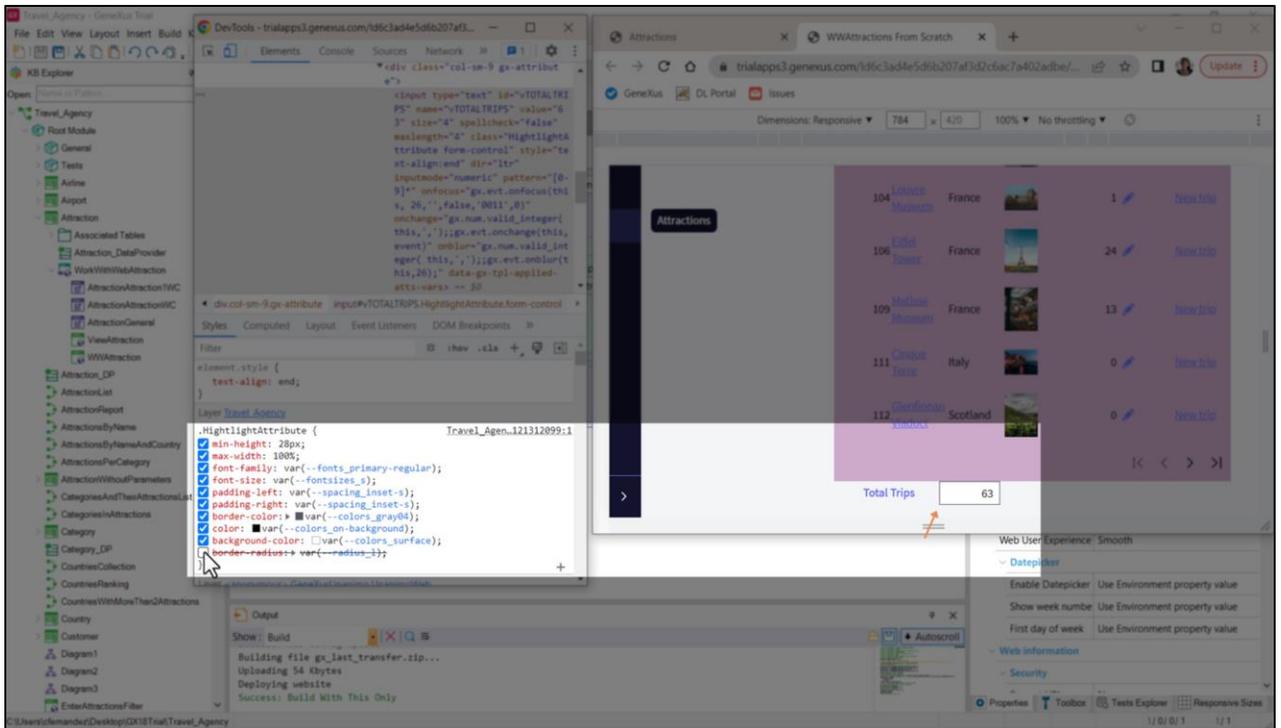
Vamos testar em execução. Nossa variável é `readonly`, então agora a vemos com rosa e azul.

E se agora a fazemos não `readonly`... está pegando o estilo da `Attribute`, mas com o estilo do rótulo de nossa classe.

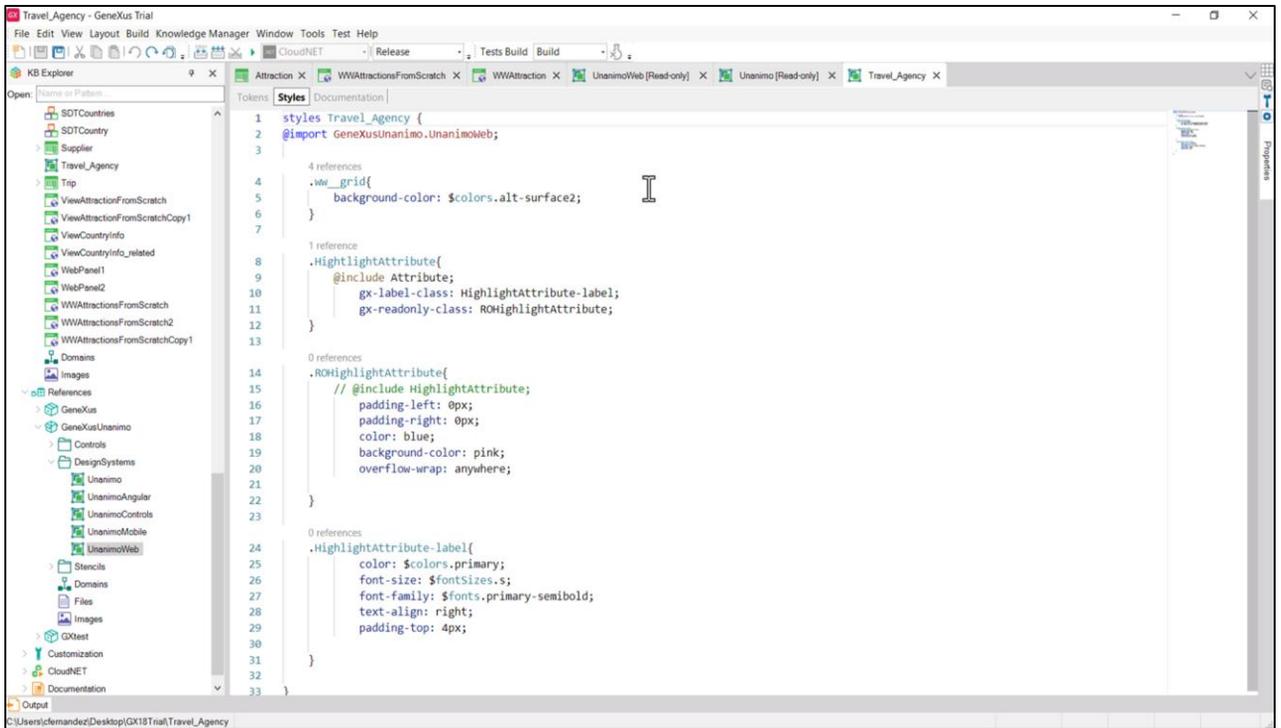


Uma última questão: se quisermos ver quais propriedades estão sendo aplicadas no CSS que comanda a estética deste HTML, como havíamos pressionado F12 podemos inspecionar os elementos do HTML e sua estética.

Aqui temos o elemento correspondente ao rótulo... e vemos que existe esta classe, com as propriedades... Podemos remover esta propriedade e vemos como o rótulo para de alinhar à direita... O mesmo com a `padding-top`. Vemos a mudança entre tê-la e não tê-la. O mesmo com a `color`.



E se agora procuramos o controle de entrada propriamente dito, vemos que possui a classe `HighlightAttribute` com todas estas propriedades. Vamos testar remover o `padding-right...` e o `border-radius...`



Neste vídeo quisemos completar uma pequena amostra dos aspectos técnicos relativos à estilização da User Interface, tomando como exemplo a personalização do Design System de Unanimo para web, um Design System predefinido. Mas é claro que poderíamos criar um Design System completamente do zero.

Como ficará mais ou menos evidente, a maioria das propriedades das classes com as quais trabalhamos são idênticas às propriedades de CSS. Além disso existem algumas que são próprias de GeneXus.

Há todo um mundo a descobrir para o qual você só precisa conhecer os controles de GeneXus e quais propriedades utilizar no Design System.

Convidamos você a continuar explorando.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com