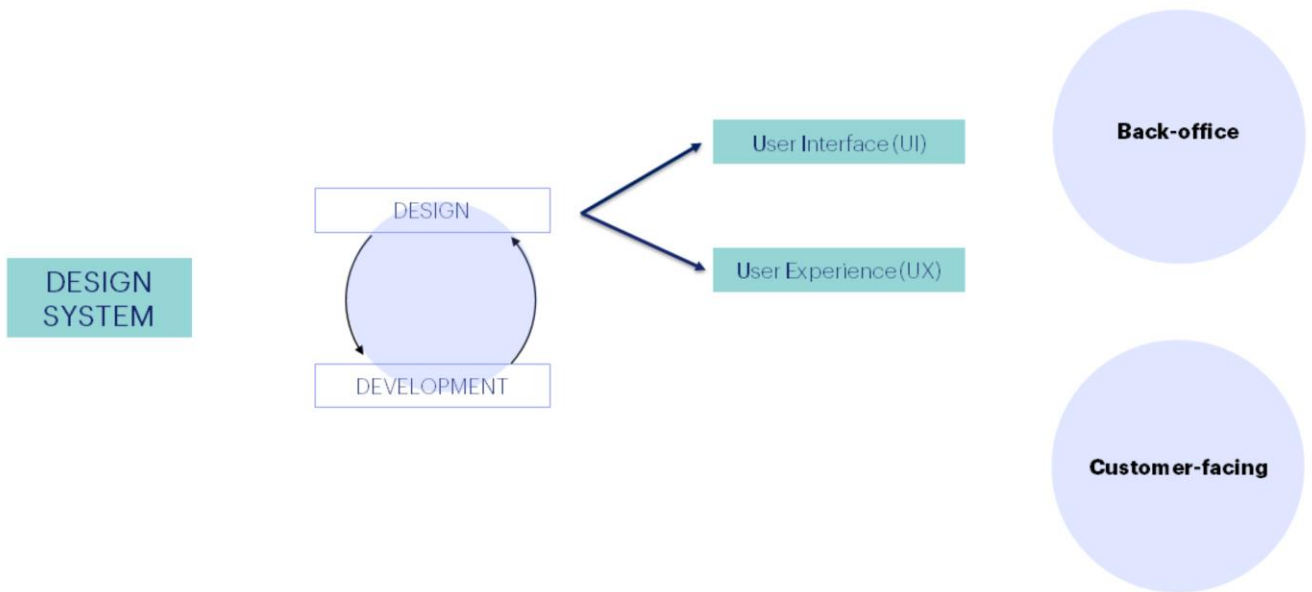


Design System em GeneXus

Personalizado

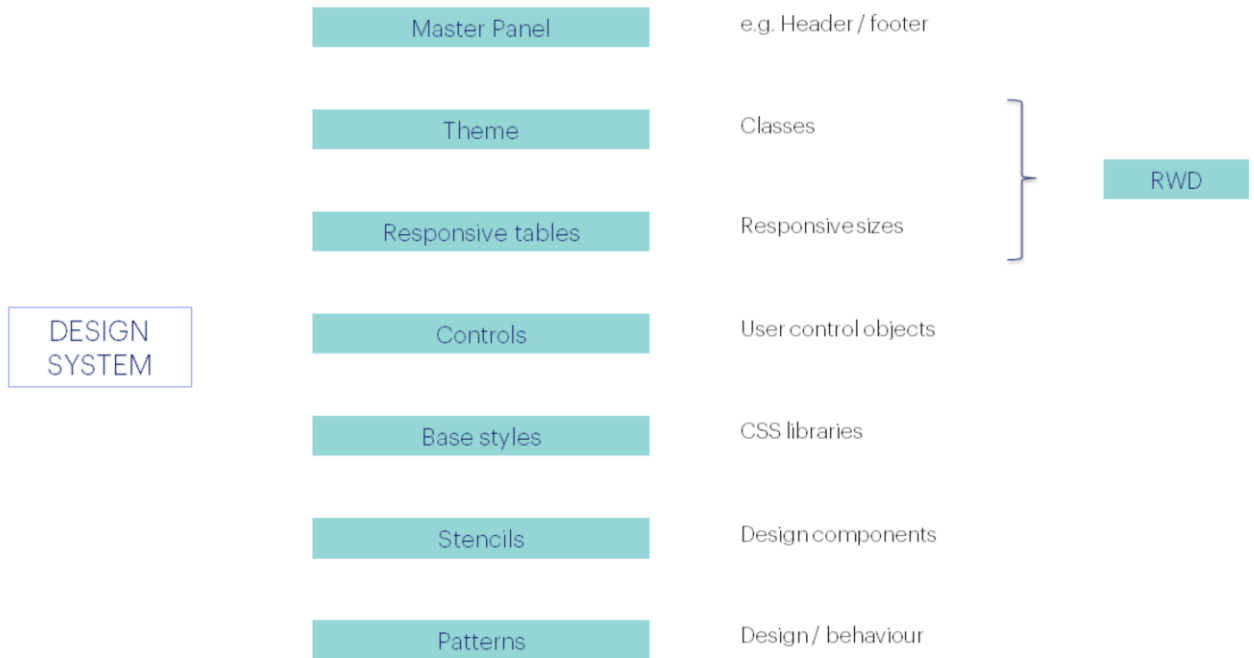
GeneXus[™]



Nos vídeos anteriores explicamos que tanto as aplicações de Back-office como as Customer-Facing têm requisitos de User Interface e User Experience cada vez mais exigentes, que conduziram ao aparecimento de um novo conceito, o Design System.

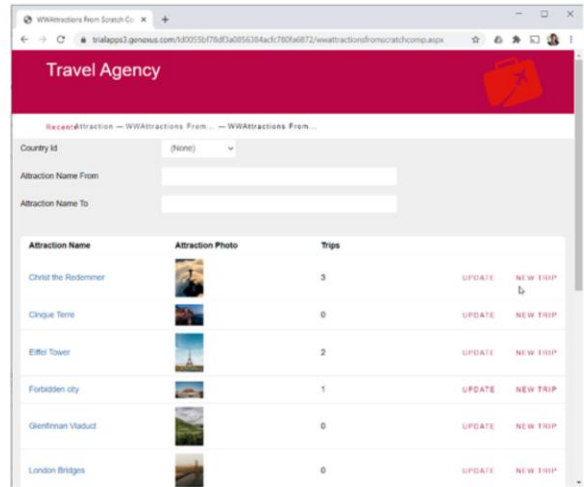
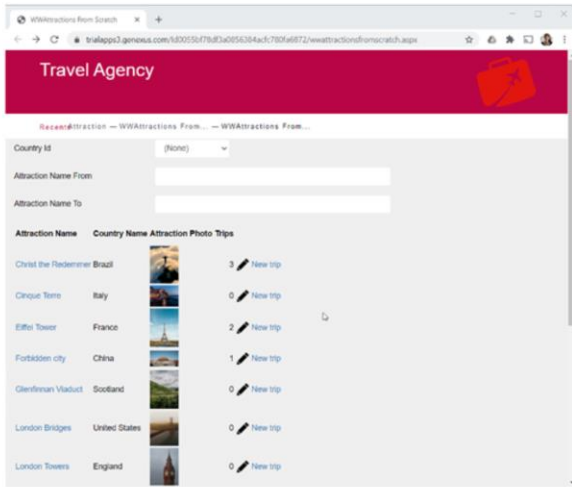
Vimos então, que o desenvolvimento de uma aplicação requer um trabalho conjunto com designers gráficos, que são os que mais intervêm na definição desse design system e de sua aplicação em nossas telas.

Então, nos focamos na aplicação de Back-office, para mostrar como já contávamos com um Design System predeterminado.



Então, vimos os diferentes atores envolvidos em sua implementação, começando pelo panel mestre default, seguido pelo tema default e tudo o que o padrão Work With já utilizava de tudo isso, através das classes que definia nesse theme.

A partir daí, vimos como podia começar a se personalizar tudo aquilo, criando classes novas e aplicando-as aos controles de nossos forms, ou modificando as classes existentes para conservar a uniformidade com o desenho das telas do padrão, cuidando da responsividade, etc.



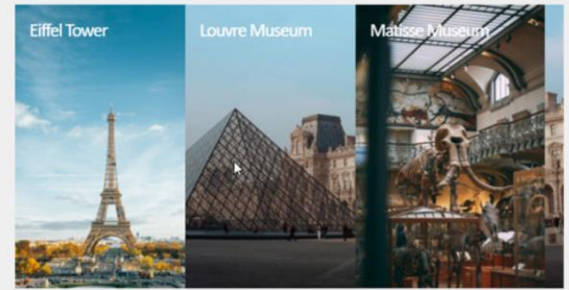
Mas quando tivemos que atender ao desenho de nossas próprias telas, as que criamos do zero, não tínhamos em que nos apoiar, a não ser tentar imitar o que fez o Work With. Na vida real, para estes casos é que contaremos com os designers gráficos.



Country Name France

Attraction Name From

Attraction Name To



City Name

City Id City Name Attractions

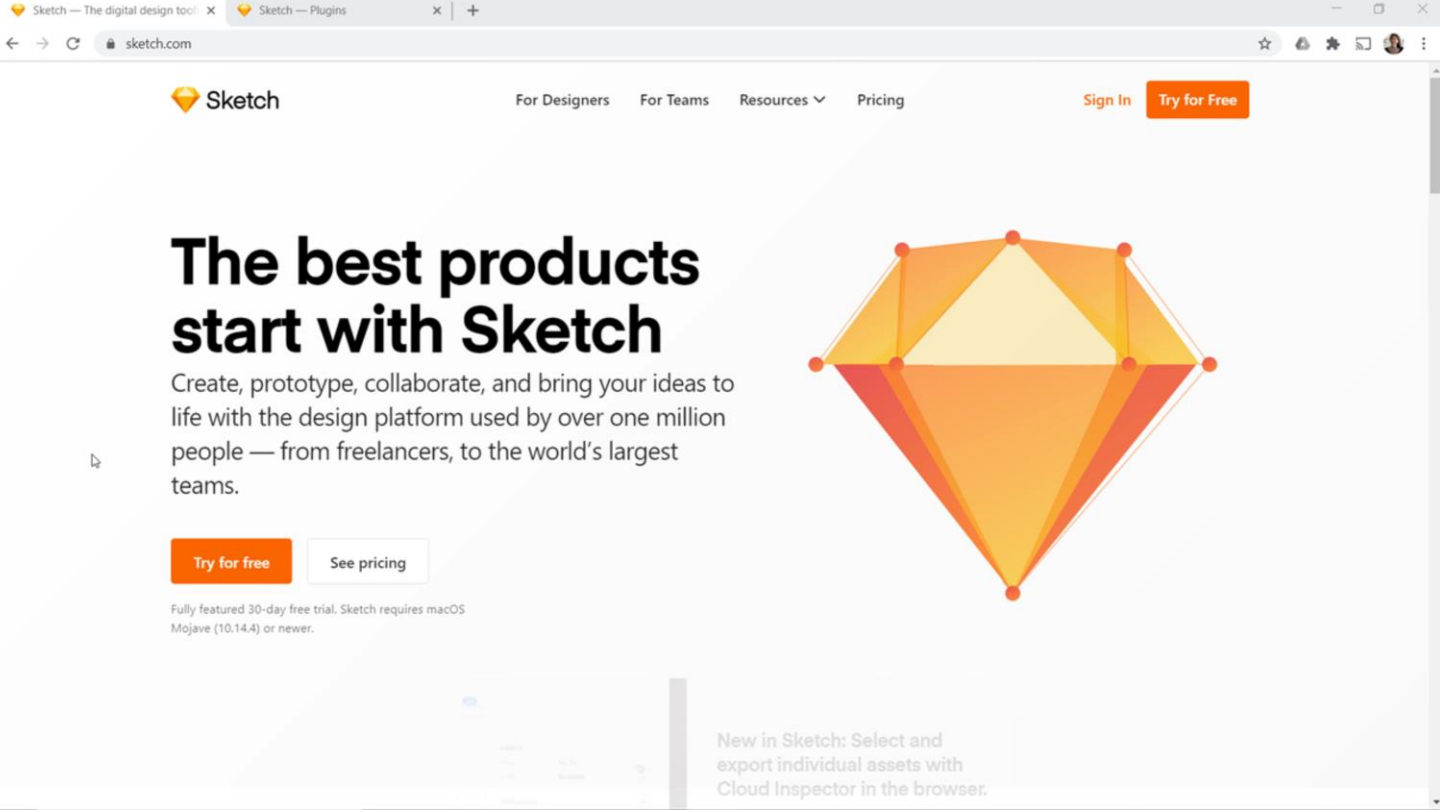
1	Paris	2
2	Nice	1
Total Attractions		3

Suponhamos que tenhamos pedido ao nosso designer gráfico para desenhar uma tela para o nosso Back-office, para mostrar as informações de um país com suas atrações turísticas (como a que implementamos manualmente aqui, mas sem as cidades).

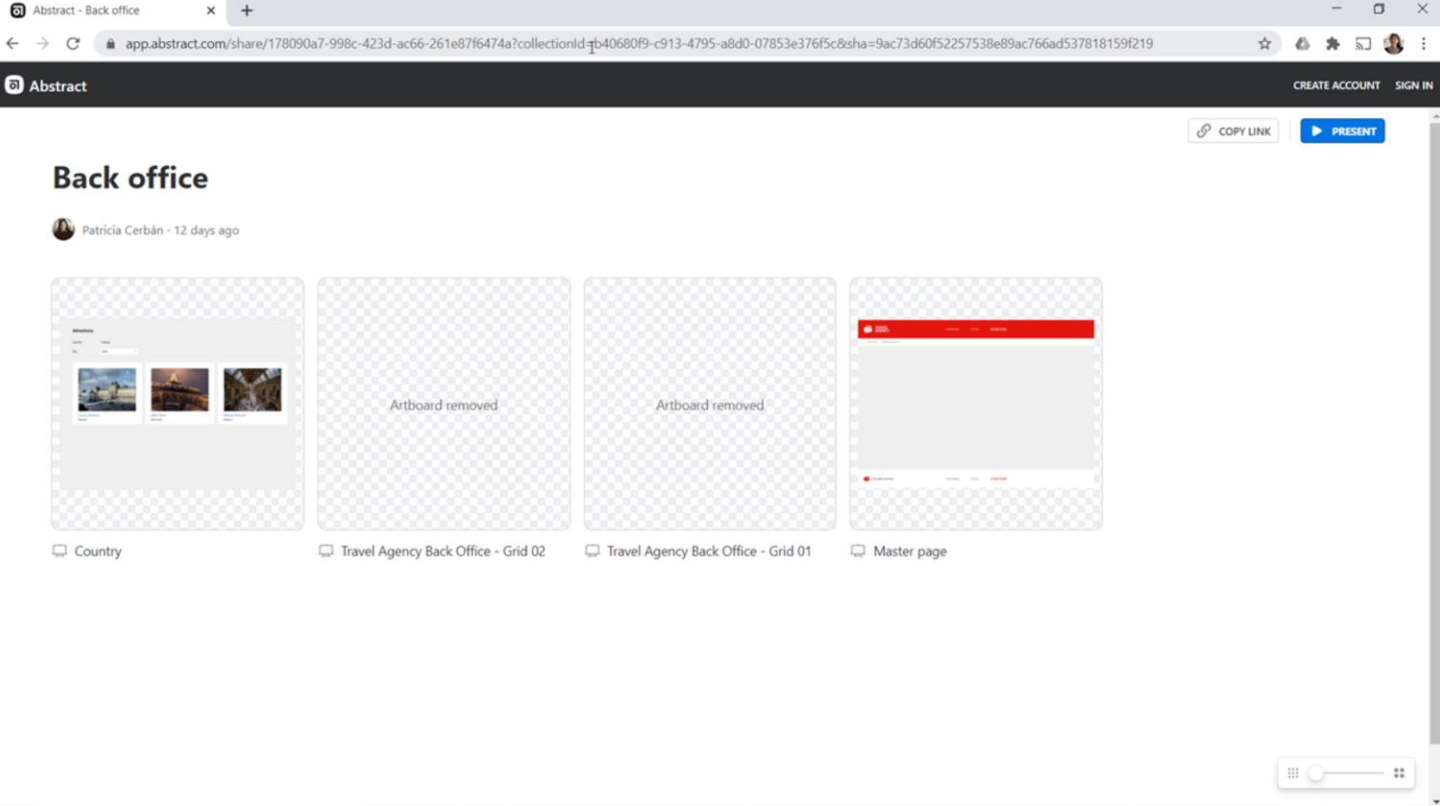
Na verdade, na vida real, nunca a teríamos implementado do zero se tivéssemos um designer gráfico. Vejamos os passos que, se soubéssemos, teríamos seguido.



Para simplificar, lhe pedimos que desenhe, a princípio, apenas o panel mestre para o Back-office e essa tela em particular. Mas, na verdade, também adequará nossas telas do Work With, possivelmente alterando cores, fontes e assim por diante. E, claro, desenhará a página Home do Back-office.

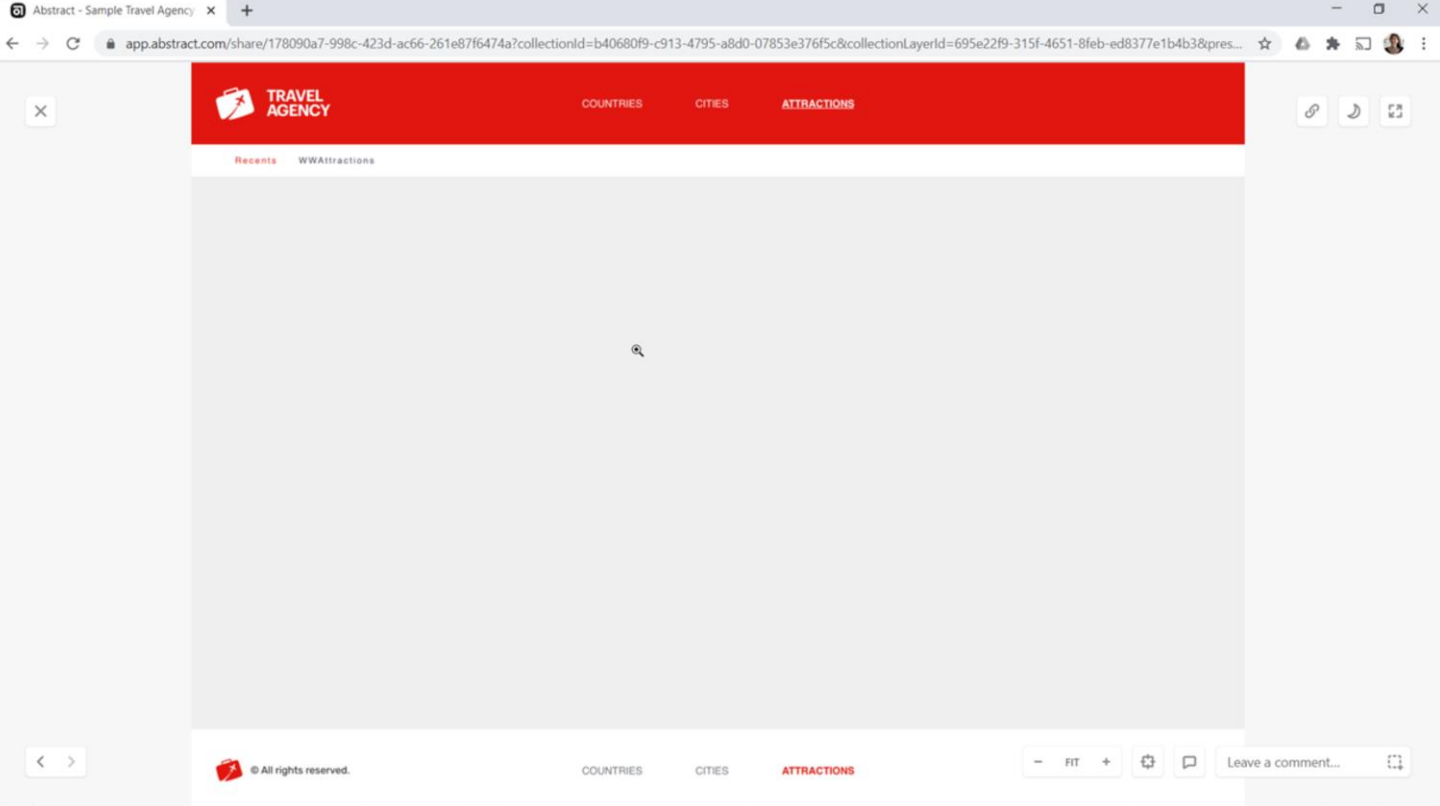


O designer utiliza a ferramenta Sketch para realizar seu trabalho, porque é uma ferramenta de desenho à qual pode-se adicionar um plugin de GeneXus que facilitará toda a magia que veremos a seguir, integrando automaticamente o desenho à nossa KB. E até criando objetos.



Então, o designer realiza o desenho da página mestra e do panel que lhe pedimos em sketch. Para isso, deve seguir algumas orientações para que a integração posterior com GeneXus seja simples e eficiente.

Uma vez terminado o desenho, nos mostra para que o validemos. Neste caso, nos envia este link para que ali vejamos essas telas, nós que não temos usuário do Sketch.



Assim ficará a página mestra.


Abstrack - Sample Travel Agency x +

app.abstrack.com/share/178090a7-998c-423d-ac66-261e87f6474a?collectionId=b40680f9-c913-4795-a8d0-07853e376f5c&collectionLayerId=f444627f-6329-4c7d-940e-efc6d3c3839c&pres...


Attractions

Country: France


City: Paris



Louvre Museum
Museum



Eiffel Tower
Monument



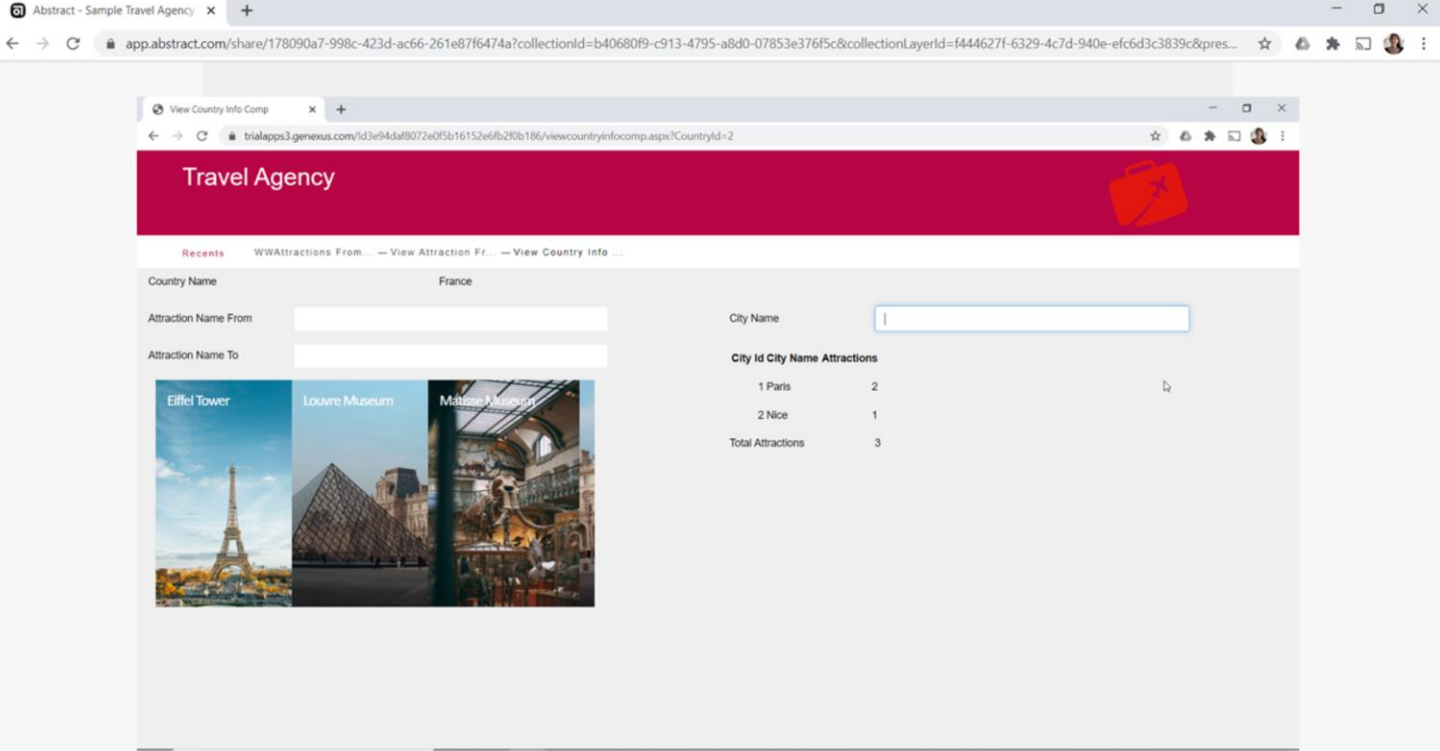
Matisse Museum
Museum

< >

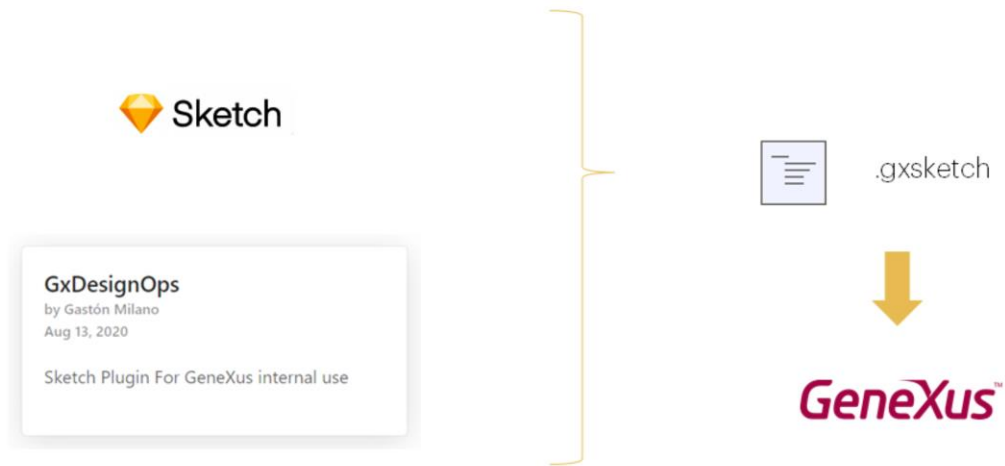
- FIT +

Leave a comment...

E assim o panel de atrações do país...

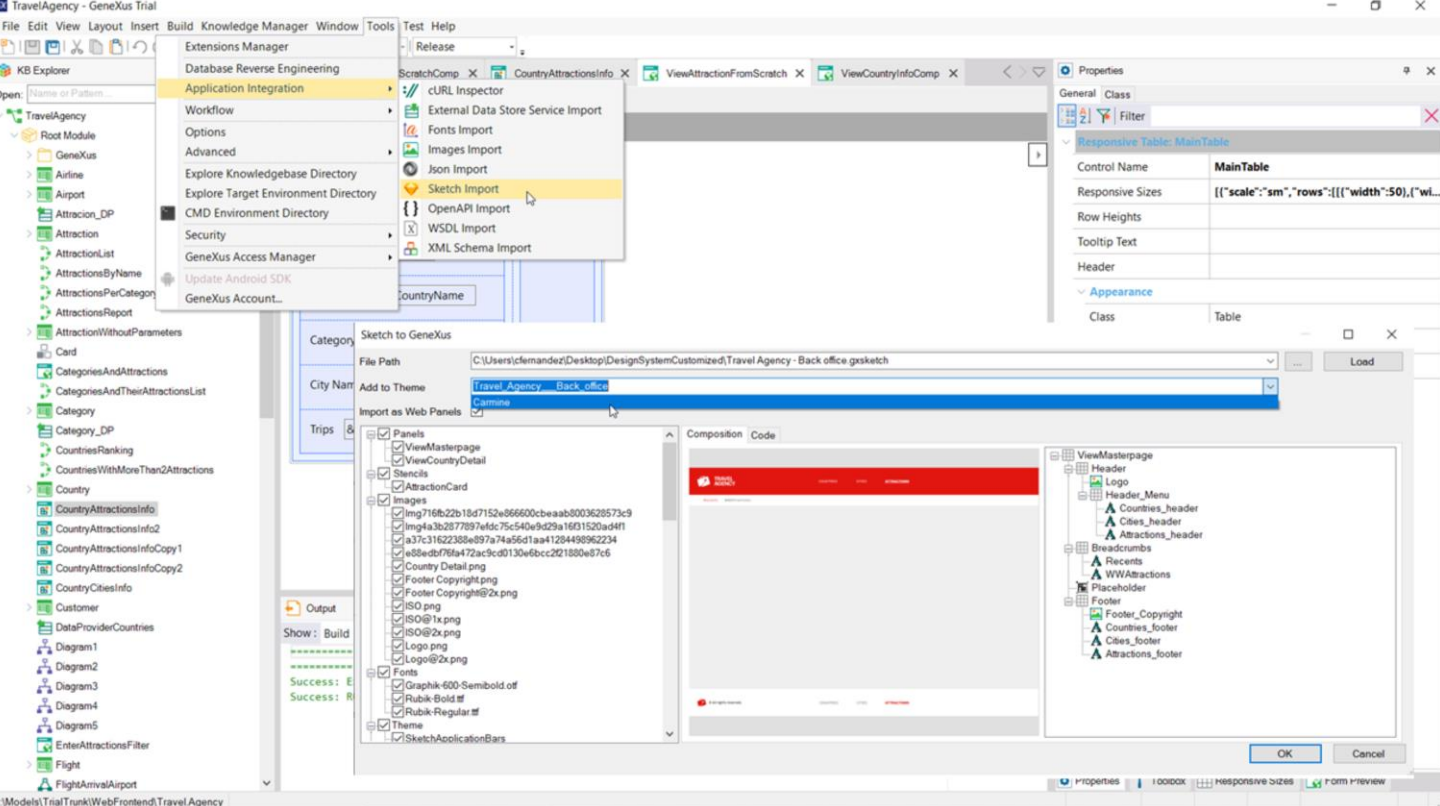


...que substituirá este outro.



Quando lhe damos o ok, utilizando esse plugin do qual falávamos, cria um arquivo de extensão gxsketch, que é o que nos envia.

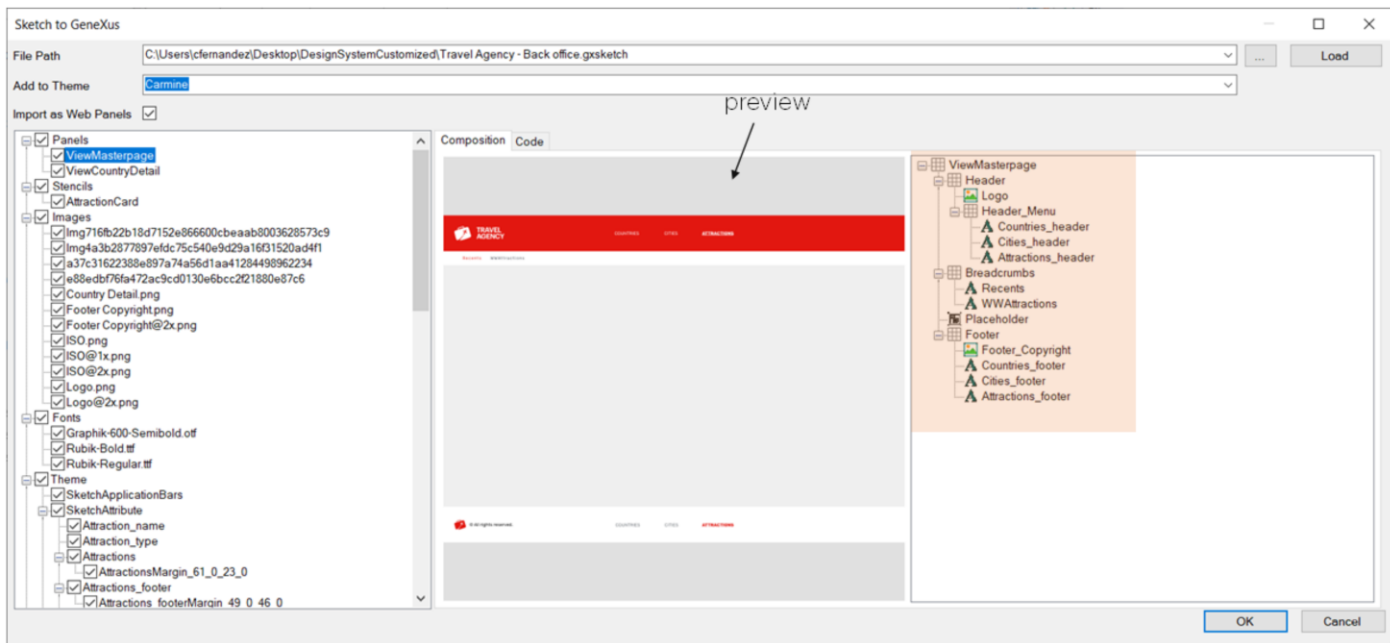
O que fazemos a seguir é importá-lo em nossa KB. Como?



Selecionando em nossa KB **Sketch Import**.

Ali devemos indicar onde se encontra o arquivo gxsketch que o designer nos enviou, e marcar que queremos importar Web Panels.

Além disso, devemos dizer-lhe se todas as classes que vai criar para estabelecer o desenho e comportamento dos controles vai importar como um objeto theme novo, ou se queremos que as incorpore ao nosso Theme Carmine. Queremos este último, para que tudo se integre em nosso Back-office.



À esquerda, nos indica tudo o que vai importar. Por exemplo, vemos dois web panels.

O primeiro vai corresponder ao panel mestre. Aqui podemos ver o preview, e aqui nos indica os controles que conterà.

Vemos que terá uma tabela Header, que conterà, por sua vez, um controle image, e outra tabela para especificar o menu com as três opções, apresentadas como text blocks. Depois, outra tabela para os recent links. Uma para o contentplaceholder. E uma última para o footer.

Sketch to GeneXus

File Path: C:\Users\cfernandez\Desktop\DesignSystemCustomized\Travel Agency - Back office.gxsketch

Add to Theme: Camins

Import as Web Panels:

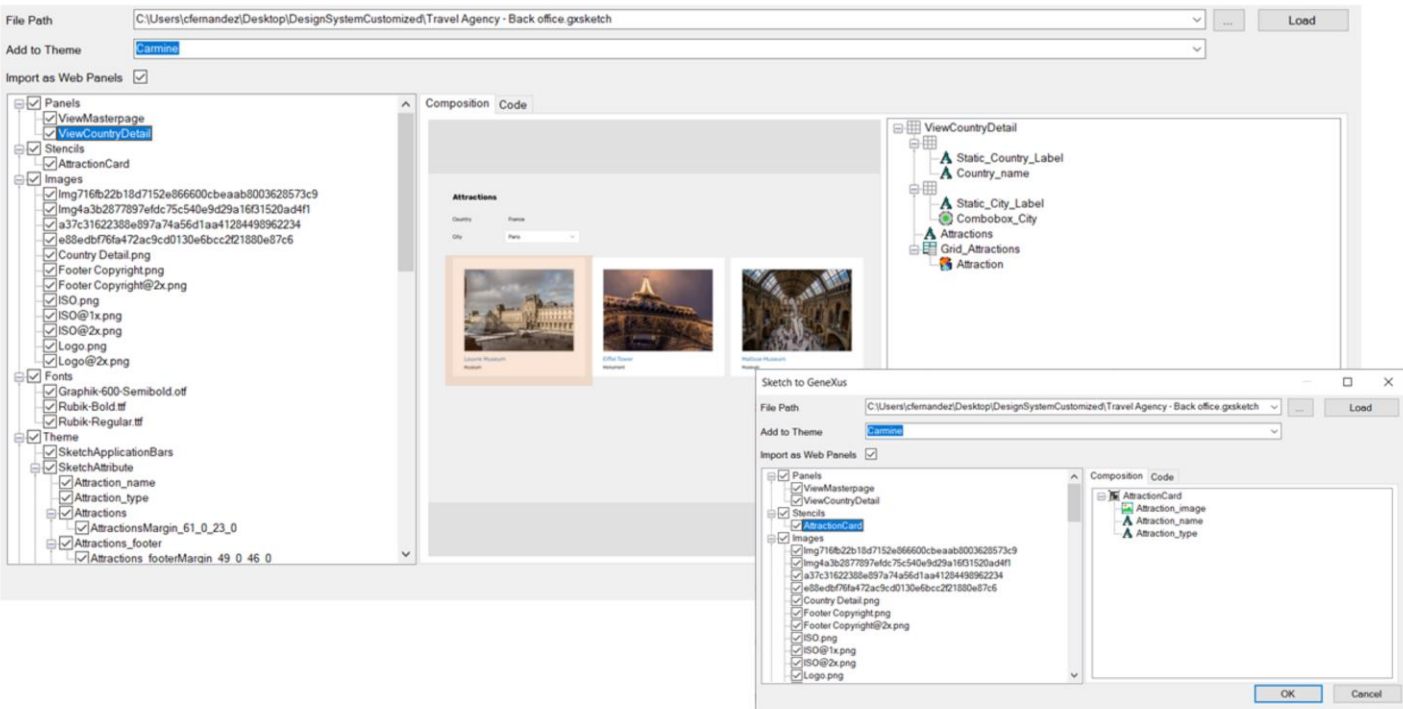
Composition Code

```
<Root Id="2ACD9572-97D0-45A6-89CD-3AF255EB757B" importFor="web">
  <Controls>
    <table tableType="SmartTable" controlName="ViewMasterpage" width="100%" height="100%"
      columnsStyle="100%" rowsStyle="112px;43px;100%;111px" class="Master_page" expandBounds="None"
      formClass="SketchForm">
      <row rowHeight="112px">
        <cell hAlign="Center" vAlign="Top">
          <table tableType="SmartTable" controlName="Header" width="100%" height="112px"
            columnsStyle="362px;1077px" rowsStyle="112px" class="Background_header" expandBounds="None"
            formClass="SketchForm">
            <row rowHeight="112px">
              <cell hAlign="Left" vAlign="Middle">
                <table tableType="SmartTable" controlName="wrapLogo" width="100%" height="100%"
                  columnsStyle="100%" rowsStyle="50%;44px;50%" class="SketchTable" expandBounds="None" formClass="SketchForm">
                  <row rowHeight="50%">
                    <cell />
                  </row>
                  <row rowHeight="44px">
                    <cell hAlign="Left" vAlign="Middle">
                      <data attribute="&Logo" class="LogoMargin_0_34_0_0" labelPosition="None"
                        readonly="True" type="BITMAP" controlNameForStencil="&Logo" initialValue="Image:Logo.Link()"
                        PATTERN_ELEMENT_CUSTOM_PROPERTIES="&Properties&";&Property&";&Name&;AutoGrow&;/Name&;&;
                    </cell />
                  </row>
                </table>
              </cell />
            </table>
        </cell />
      </table>
    </table>
  </Controls>
</Root>
```

OK Cancel

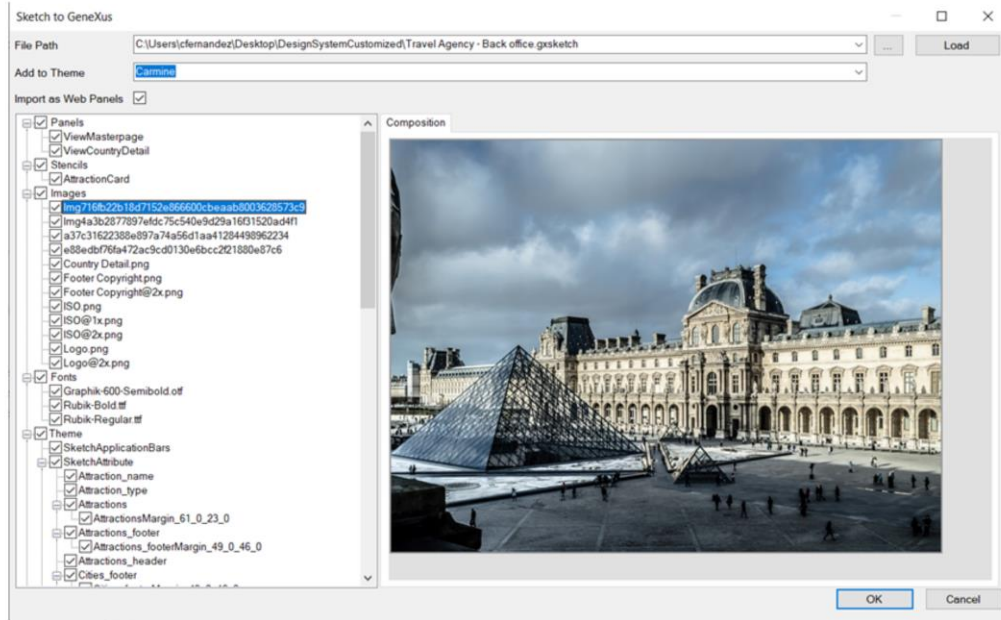
También podemos ver o código html correspondente.

Sketch to GeneXus

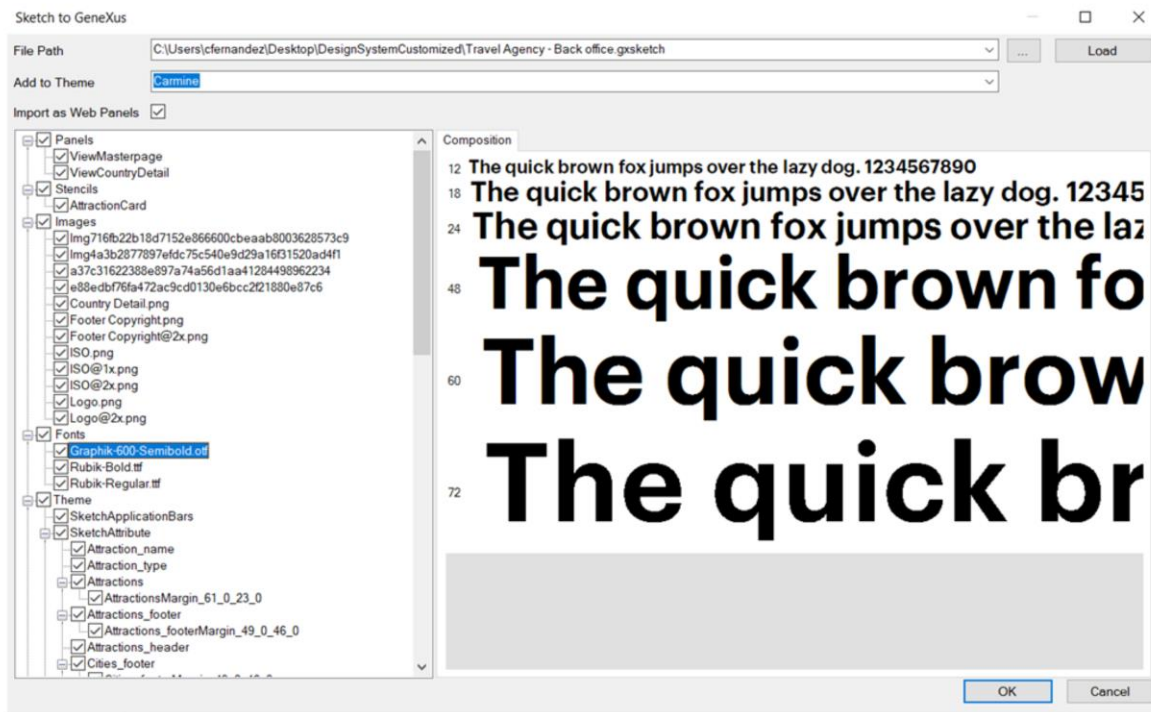


Depois, se vemos o próximo panel, a mesma coisa.

Vejamos que criará um grid que, como veremos, será horizontal, e um controle stencil, justamente para não repetir este pedaço de desenho. Aqui o vemos.



Também importará as imagens de que necessita. Por exemplo, estas são para poder executar o panel com dados fixos e mostrar atrações carregadas, que depois os desenvolvedores deverão substituir pelas da base de dados de nossa aplicação.

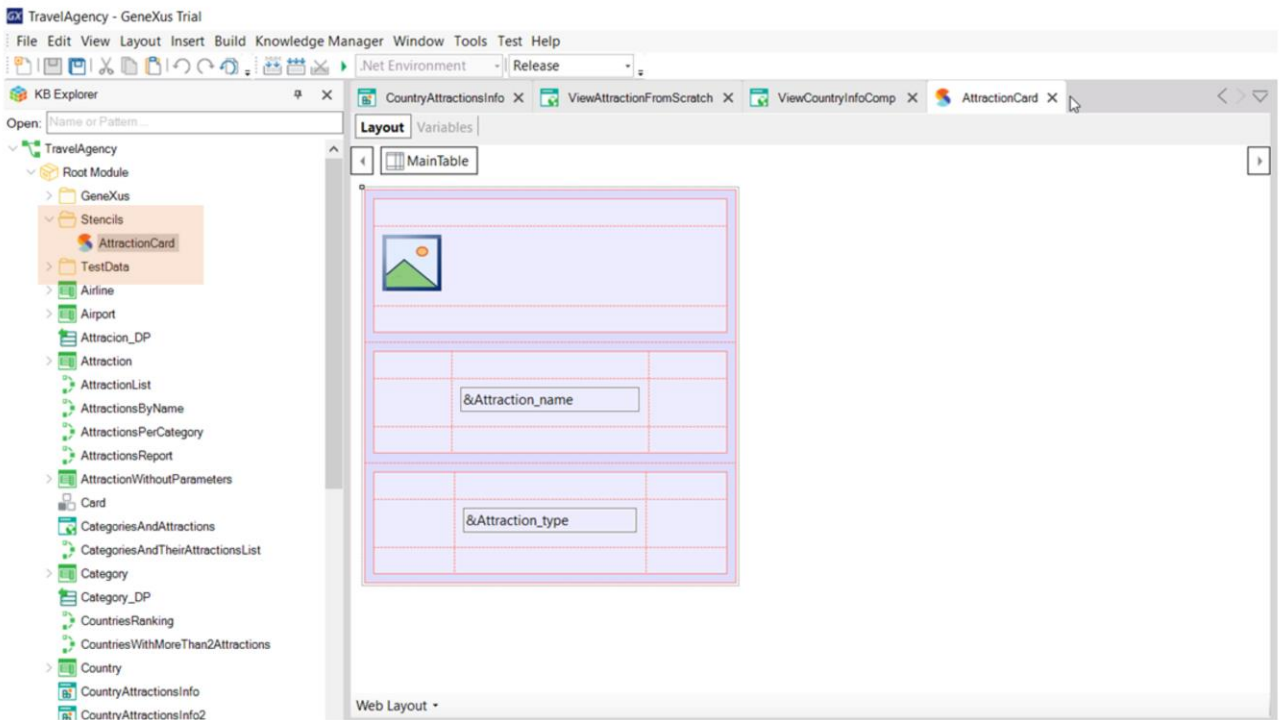


Aqui vemos as fontes que serão utilizadas neste desenho.

E por último, todas as classes que serão adicionadas ao nosso theme.

Se tivessem sido definidas novas cores também seriam mostradas aqui.

Como estamos de acordo com importar tudo isto, pressionamos Ok.



Observemos que foram criados estes dois folders.

Aqui aparece o stencil que tínhamos identificado para o cartão que mostra foto, nome e categoria da atração turística.

```
1 ViewCountryDetail_Grid_Attractions_DP
2 {
3   ViewCountryDetail_Grid_Attractions_SDT
4   {
5     Layout = "Attraction"
6     Attraction_image = Image:Img716fb22b18d7152e866600cbeaab8003628573c9.Link()
7     Attraction_name = "Louvre Museum"
8     Attraction_type = "Museum"
9   }
10  ViewCountryDetail_Grid_Attractions_SDT
11  {
12    Layout = "Attraction"
13    Attraction_image = Image:Img4a3b2877897efdc75c540e9d29a16f31520ad4f1.Link()
14    Attraction_name = "Eiffel Tower"
15    Attraction_type = "Monument"
16  }
17  ViewCountryDetail_Grid_Attractions_SDT
18  {
19    Layout = "Attraction"
20    Attraction_image = Image:a37c31622388e897a74a56d1aa41284498962234.Link()
21    Attraction_name = "Matisse Museum"
22    Attraction_type = "Museum"
23  }
24 }
25
```

Data Provider: ViewCountryDetail_Grid_Attractions_DP	
Name	ViewCountryDetail_Grid_Attractions_DP
Description	View Country Detail_Grid_Attractions_DP
Expose as Web Service	False
Main program	False
Call protocol	Internal
Module/Folder	TestData
Qualified Name	ViewCountryDetail_Grid_Attractions_DP
Object Visibility	Public
Output	
Infer Structure	No
Output	ViewCountryDetail_Grid_Attractions_SDT
Collection	True
Collection Name	ViewCountryDetail_Grid_Attractions_DP
Network	
Connectivity Support	Inherit
Miscellaneous	
Generate Object	True

E aqui um Data Provider que é o que carregará com dados fixos as três atrações que vimos no preview.

The screenshot displays the GeneXus IDE interface for designing a web form. The main workspace shows a grid-based layout for a master page. The layout consists of three main sections: a header, a main content area, and a footer. The header section contains a logo on the left and three columns of content labeled '&Countries_header', '&Cities_header', and '&Attractions_header'. The main content area contains two columns of content labeled '&Recents' and '&WWAttractions'. The footer section contains a logo on the left and three columns of content labeled '&Countries_footer', '&Cities_footer', and '&Attractions_footer'. The Properties panel on the right shows the control name 'ViewMasterpage' and its appearance settings, including Columns Style (100%), Rows Style (112px:43px:100%:111px), Width (100%), Height (100%), Class (Master_page), Columns Gap (0px), Rows Gap (0px), and Form Class (SketchForm).

Control Name	ViewMasterpage
Tooltip Text	
Header	
Appearance	
Columns Style	100%
Rows Style	112px:43px:100%:111px
Width	100%
Height	100%
Class	Master_page
Columns Gap	0px
Rows Gap	0px
Form	
Form Class	SketchForm

E aqui abaixo encontramos o web panel que implementava a página mestre. E aqui o que implementava o detalhe do país.

Se abrimos o primeiro, vemos que já está criado todo o seu conteúdo visual.

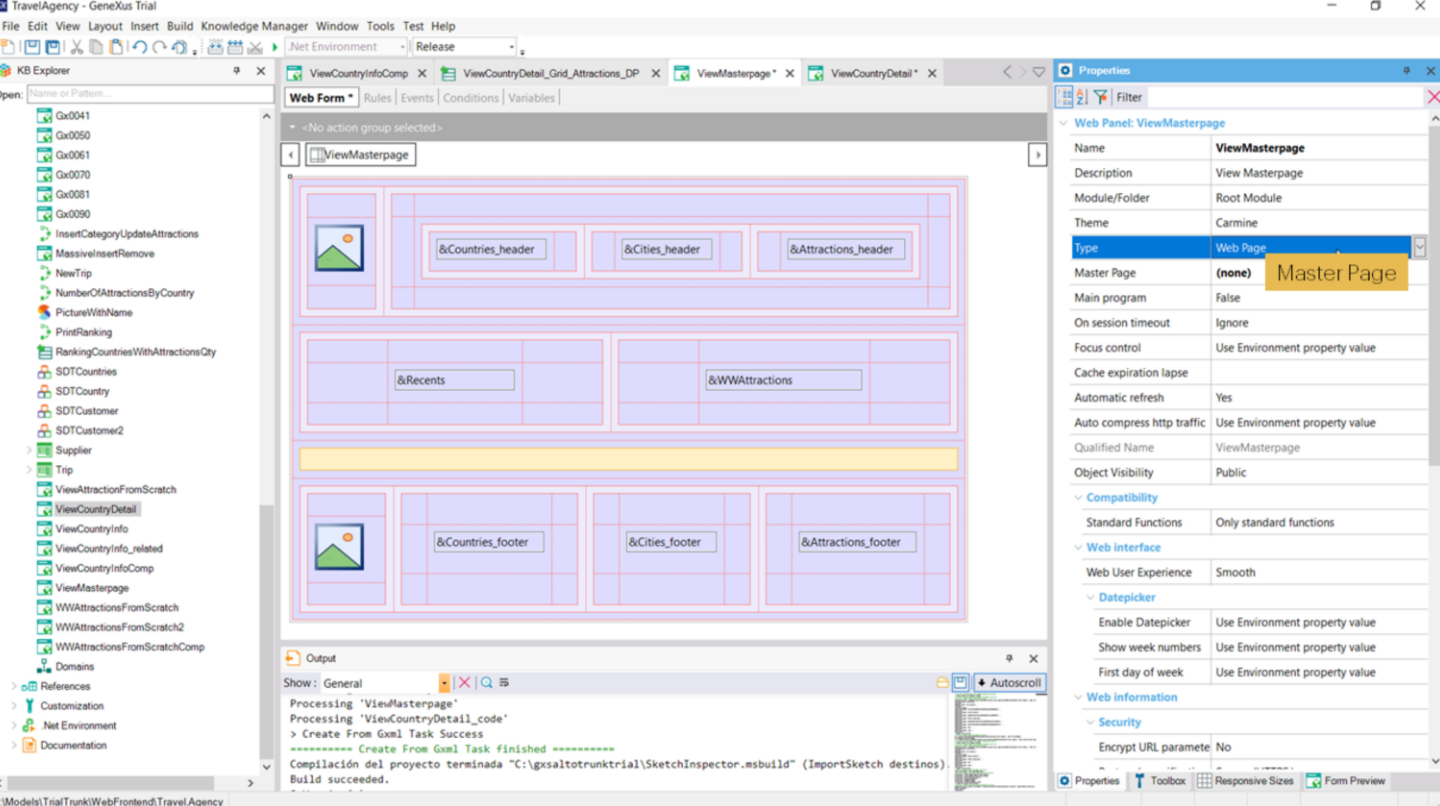
The screenshot displays the GeneXus IDE interface. On the left, a project tree shows a list of views, with 'ViewCountryDetail' selected. The main workspace shows a 'Web Form' design for 'ViewCountryDetail'. The form is structured as follows:

- A top section with a label '&Attractions' and an input field.
- A middle section with a label 'Country' and an input field '&Country_name'.
- A bottom section with a label 'City' and a dropdown menu '&Combobox_City'.

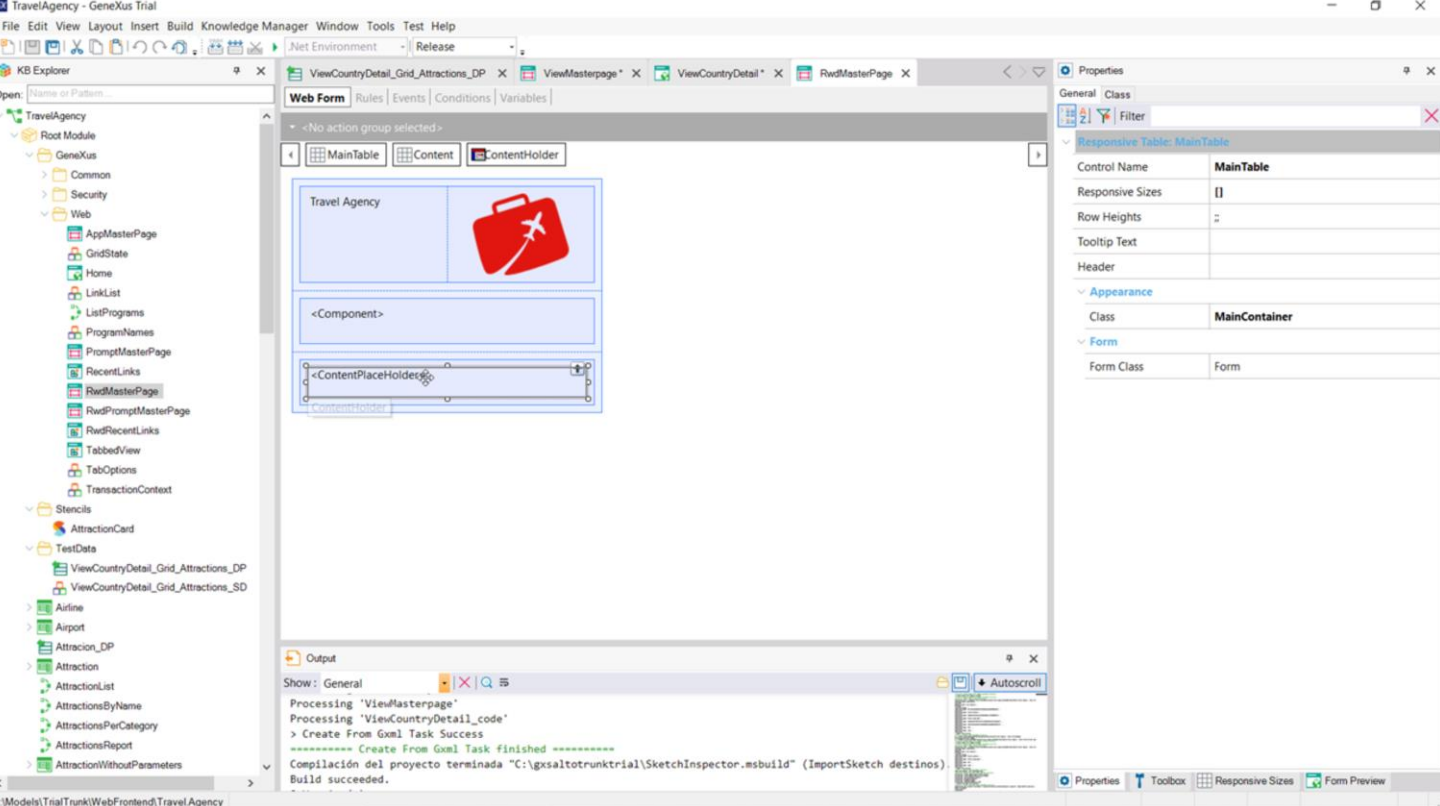
At the bottom of the form, a 'GRID' section is partially visible. On the right, the 'Properties' panel is open, showing the following details for the 'ViewCountryDetail' control:

Property	Value
Control Name	ViewCountryDetail
Tooltip Text	
Header	
Appearance	
Columns Style	100%
Rows Style	112px;5%;8%;86%
Width	100%
Height	100%
Class	SketchTable
Columns Gap	0px
Rows Gap	0px
Form	
Form Class	SketchForm

E o mesmo se abrimos o segundo.

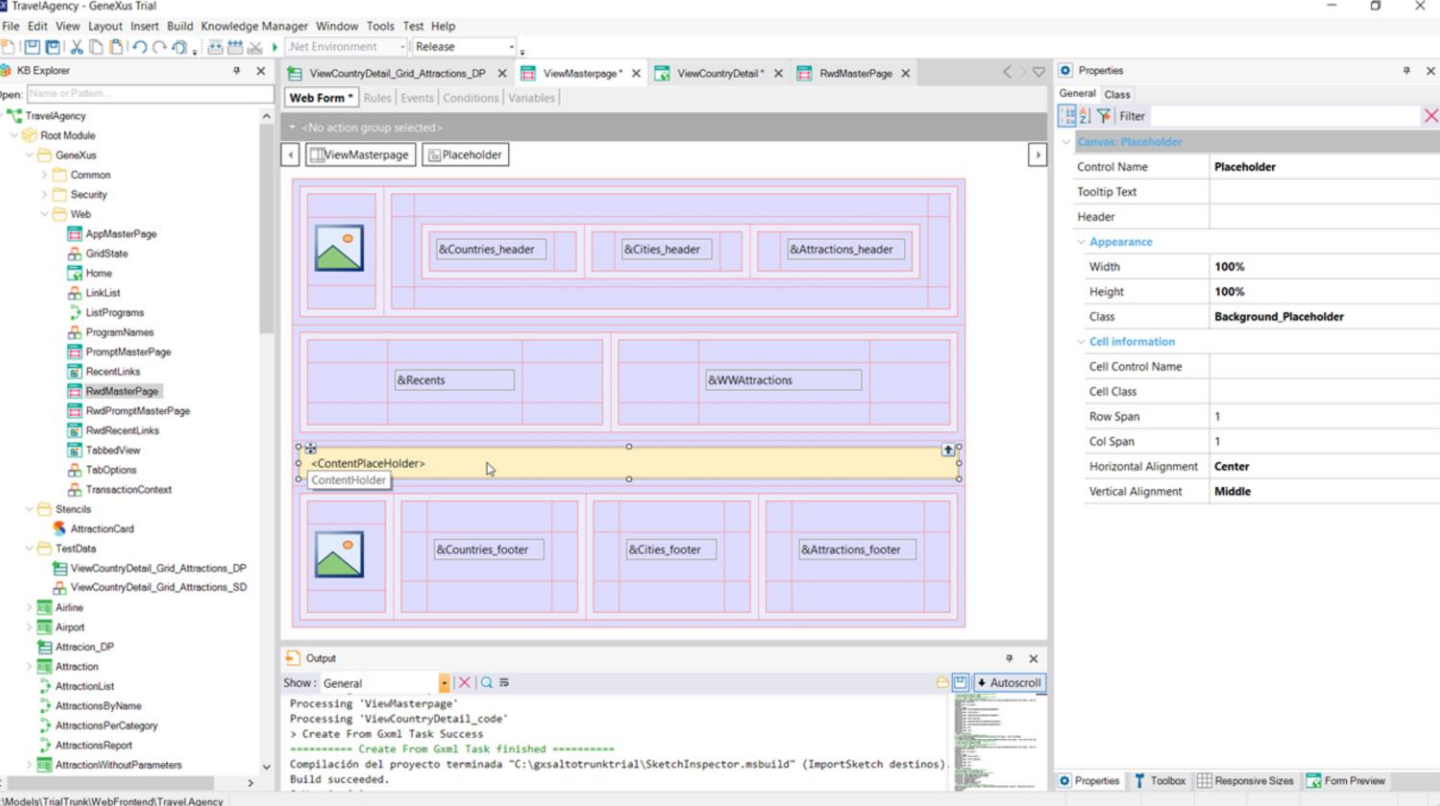


No entanto, o que ainda não está fazendo o programa de importação de Sketch (porque ainda se encontra em processo de desenvolvimento), mas o fará no futuro, é importar como página mestre este panel. Observemos que o importou como Web Page comum. Então deveremos modificar seu tipo.

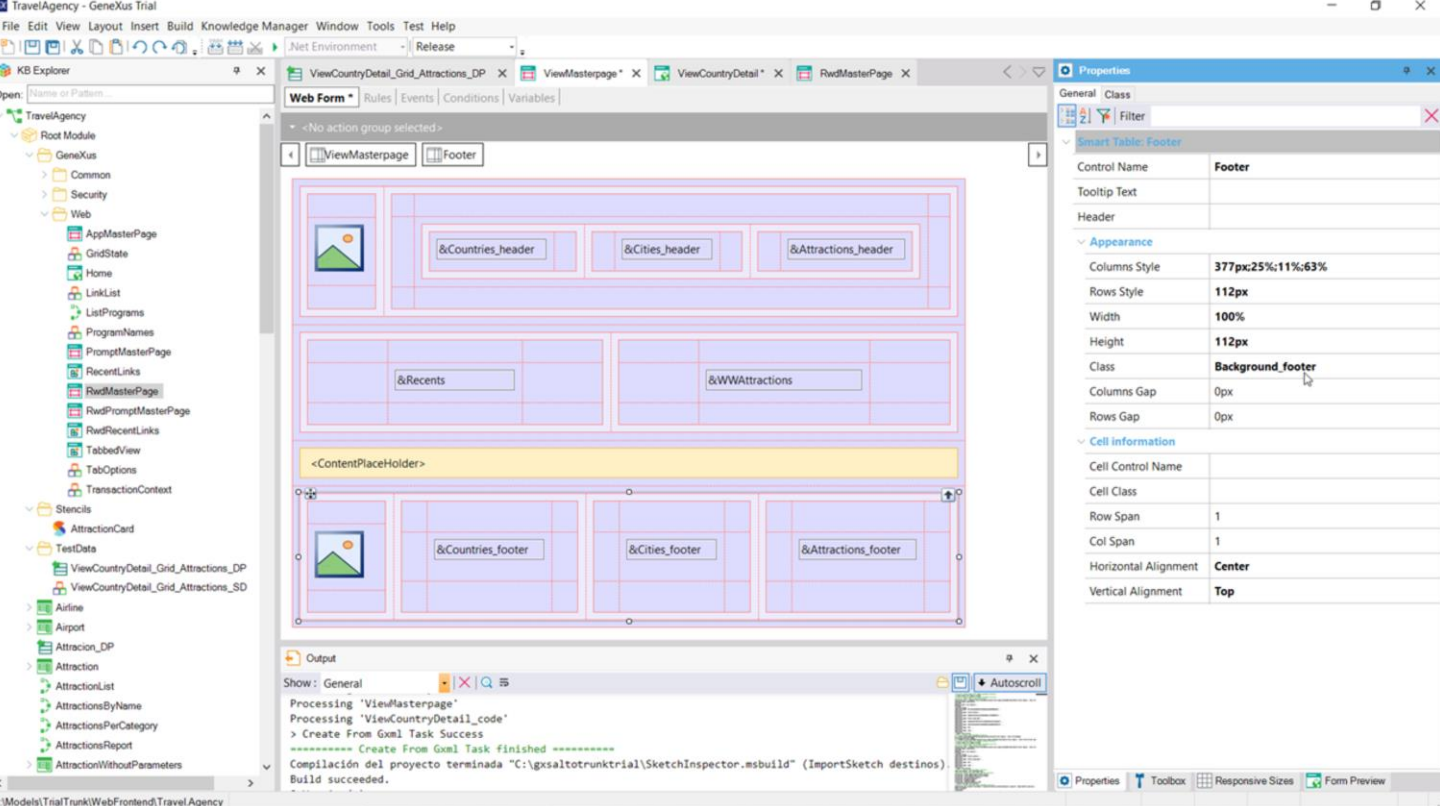


Recordemos que em nossa KB os objetos web estavam utilizando a página mestra default, que tínhamos personalizado um pouco.

Aproveitemos que a temos aberta para copiar o content place holder para a nova, que acabamos de especificar como mestra, porque exigirá ter este controle em seu form (caso contrário, não nos permitirá gravar o objeto).

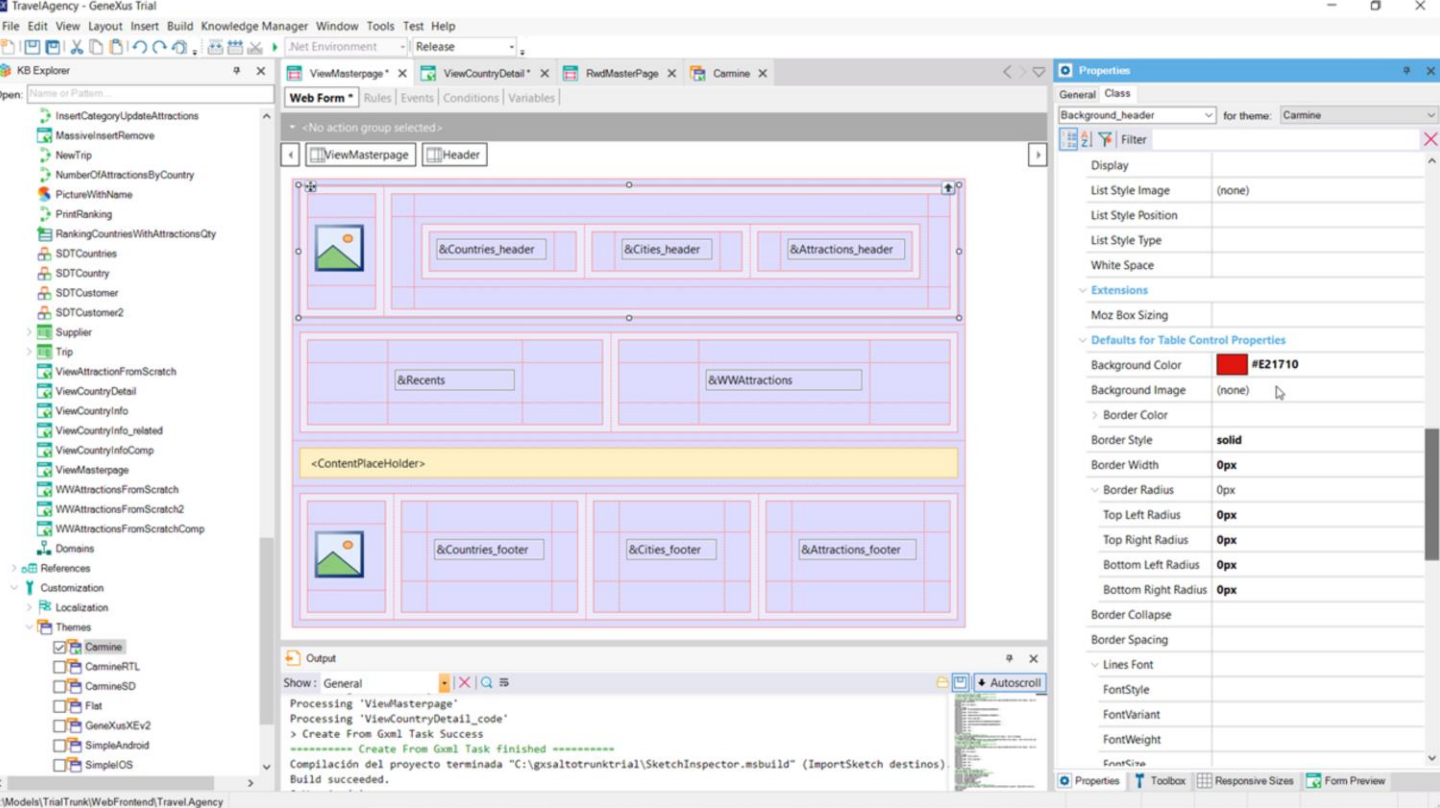


O copiamos no lugar que o designer nos deixou para ele.

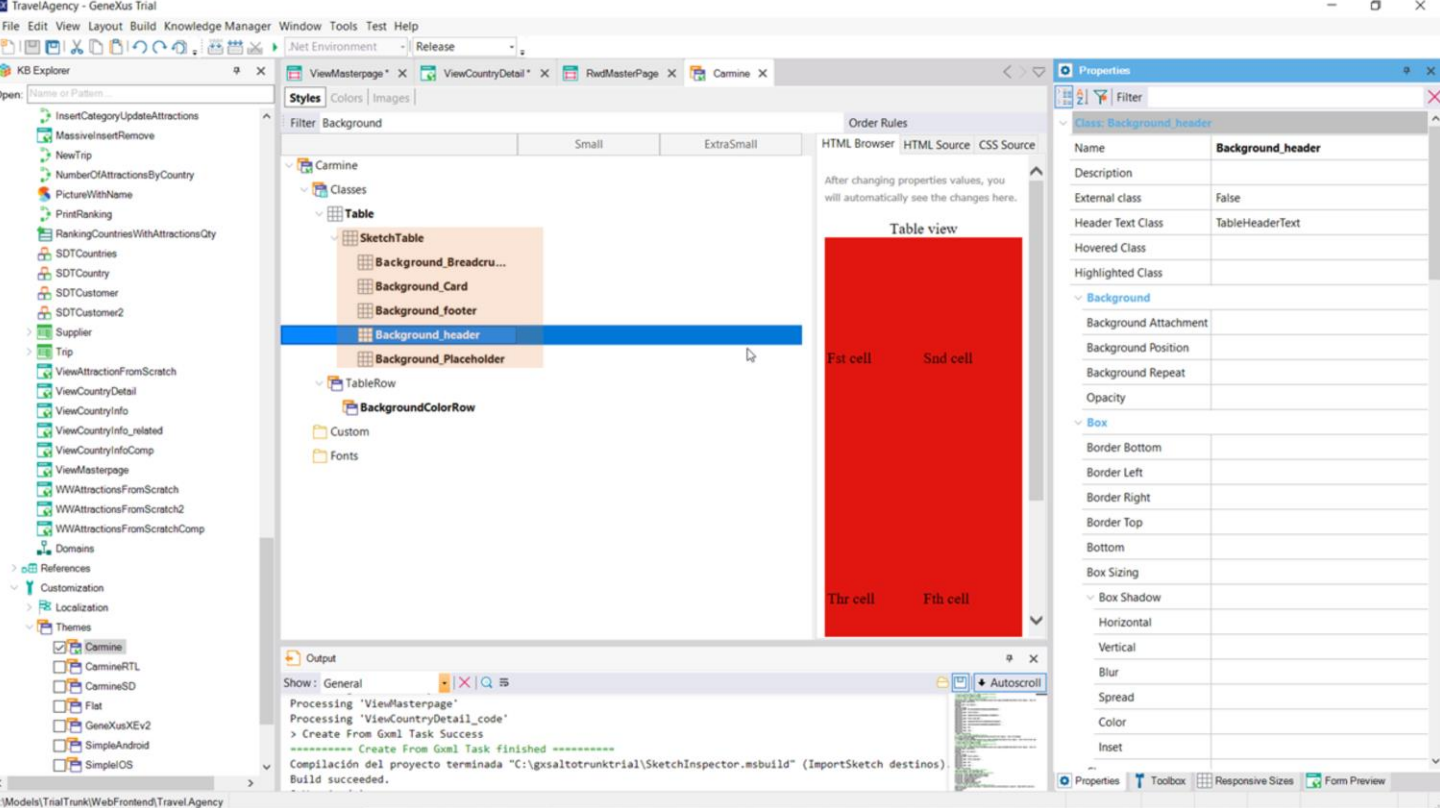


Aqui vemos a tabela Header, que tínhamos visto antes de importar. Aqui a dos links recentes.

E aqui a do footer. Vemos que lhe foi atribuída esta classe, que foi importada dentro do nosso theme Carmine...

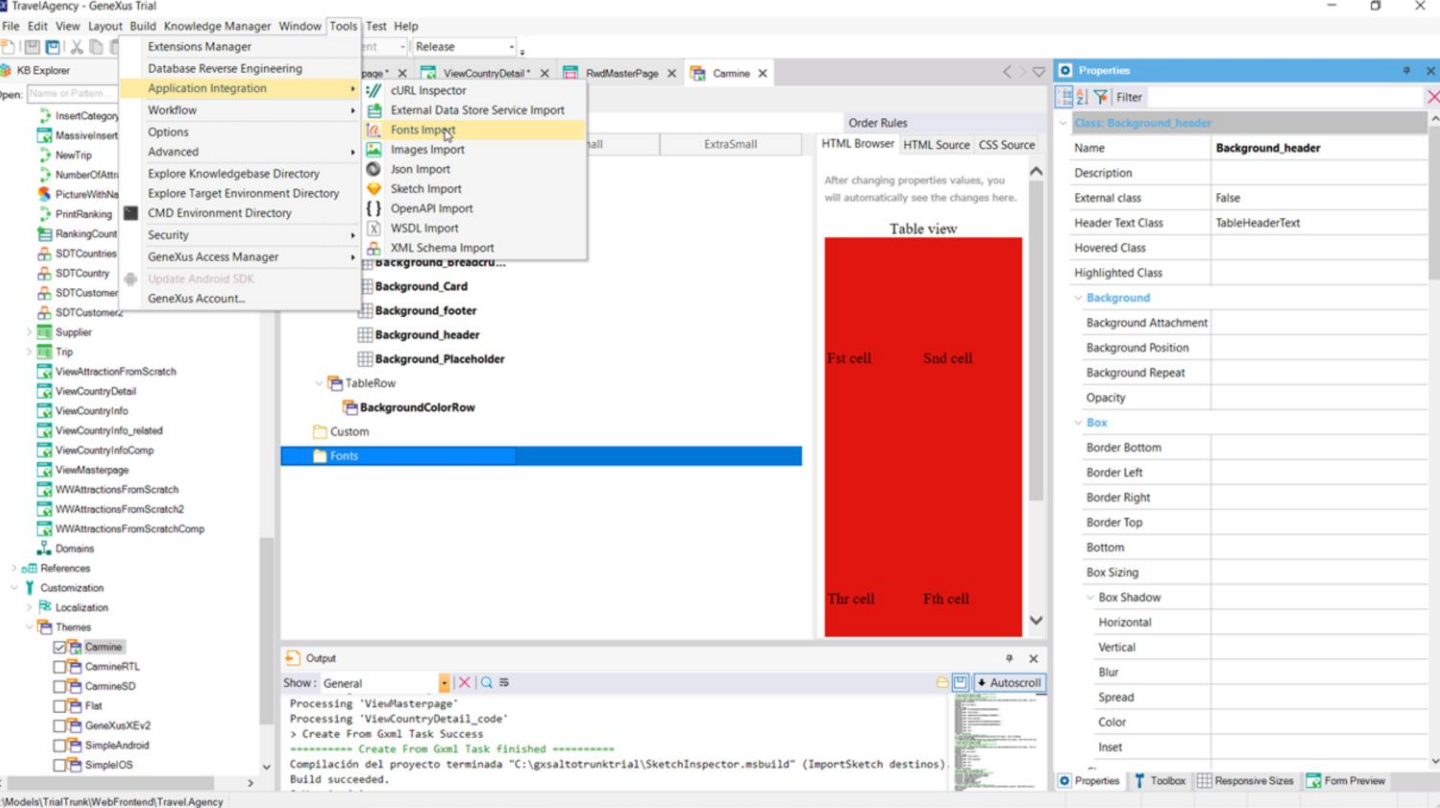


Se o abrimos... e buscamos a classe para a tabela Header que tem especificada esta cor de fundo...

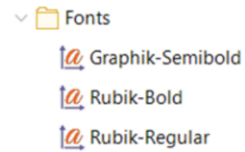
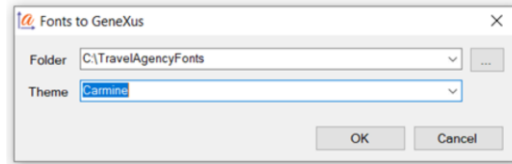
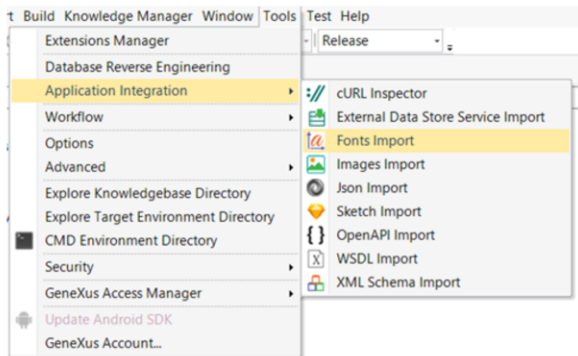


...aqui a vemos.

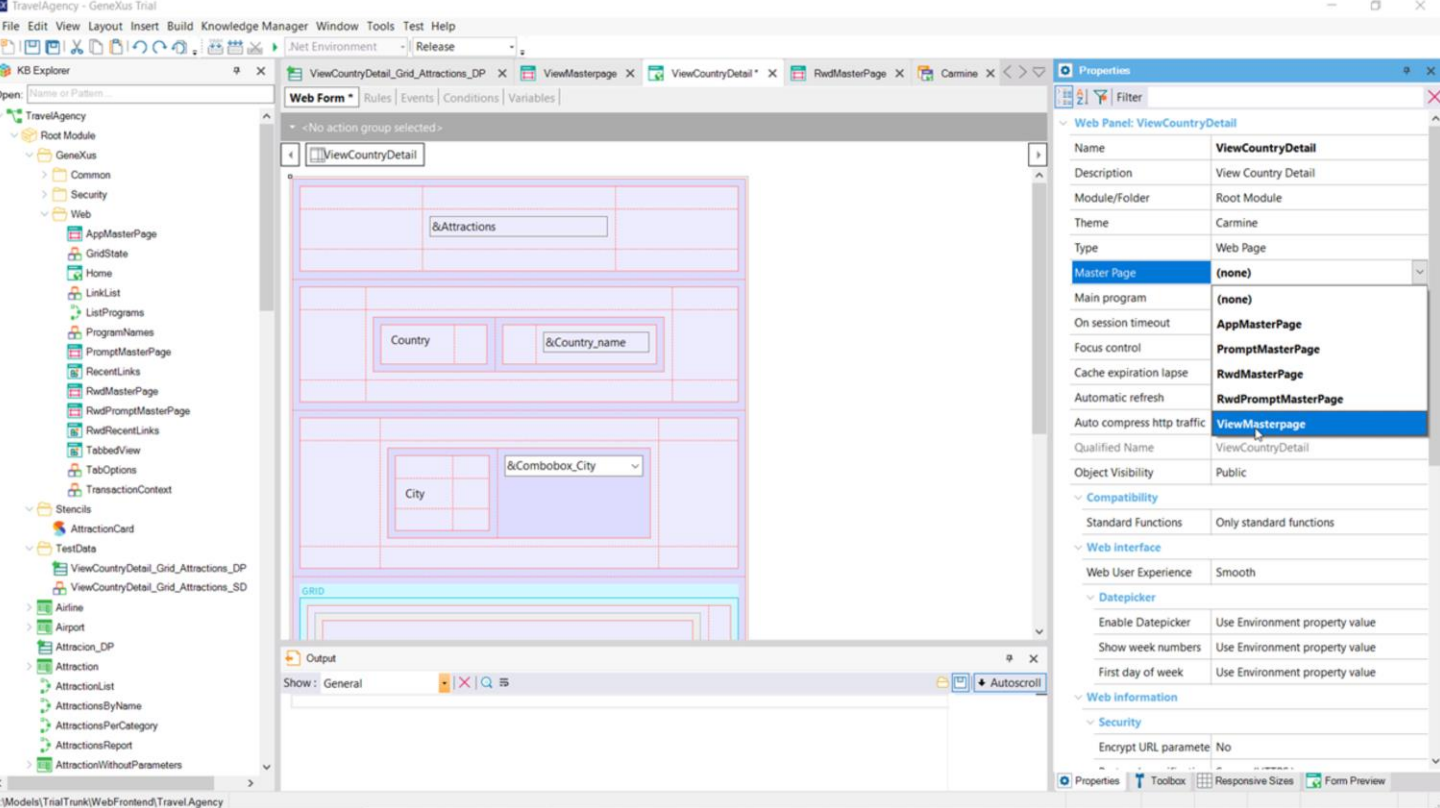
Com a importação foram criadas todas estas classes de tabelas que não existiam.



Também deveriam ter sido criados aqui abaixo as fontes que vinham especificadas no Sketch. Em próximas versões será feito. Por enquanto, teremos que importá-las explicitamente.

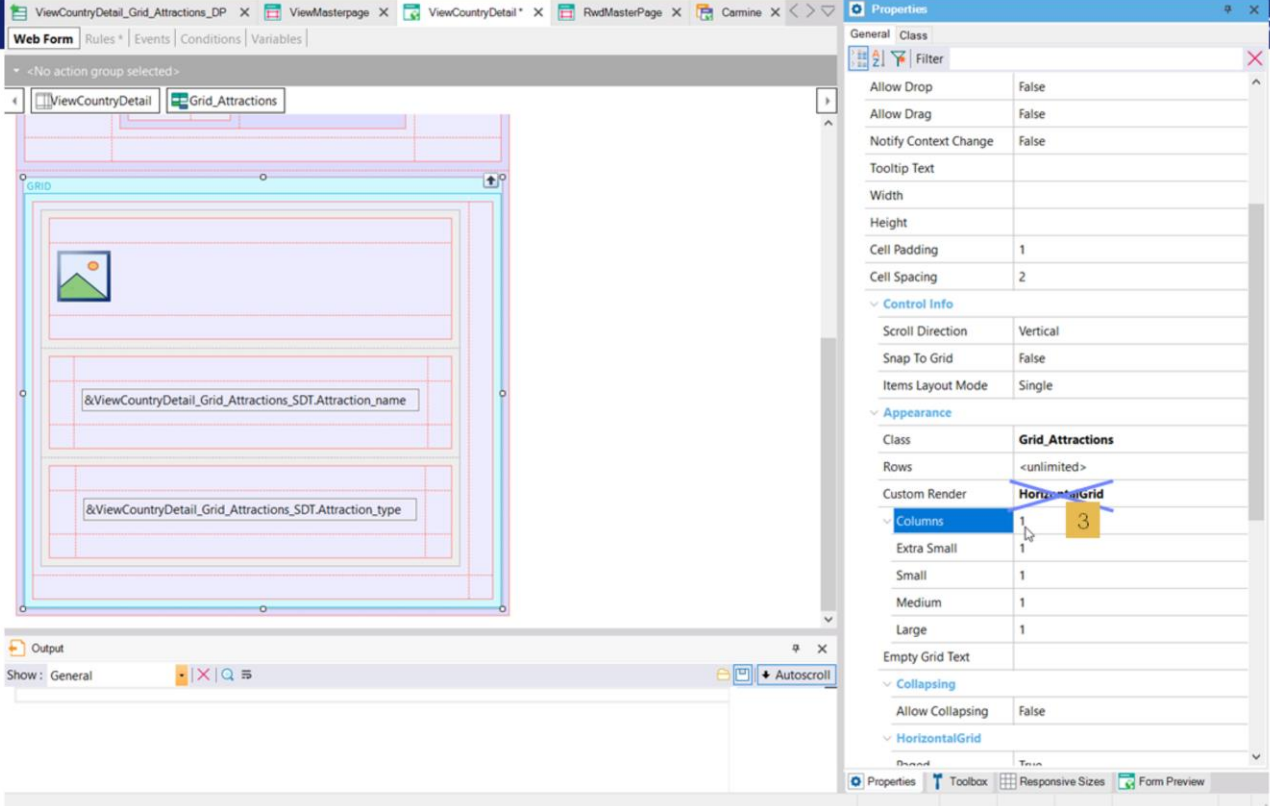


Indicamos em que folder se encontram... e que queremos que se incorporem ao theme Carmine. Ali estão.



Gravemos o novo panel mestre.

E agora devemos indicar ao panel de detalhe de um país que se carregue dentro dele.



Antes de executar, observemos que foi inserido um grid horizontal, embora não tenham sido especificadas corretamente (ainda) as 3 colunas por página horizontal. Isso ocorre porque ainda a importação do Sketch para web panels não está totalmente madura.

Então, o modificamos.

Também poderíamos personalizá-lo para testar, antes de mais nada como pareceria se o grid não fosse horizontal, mas padrão, embora com 3 colunas.

Executemos para testar.

Attractions

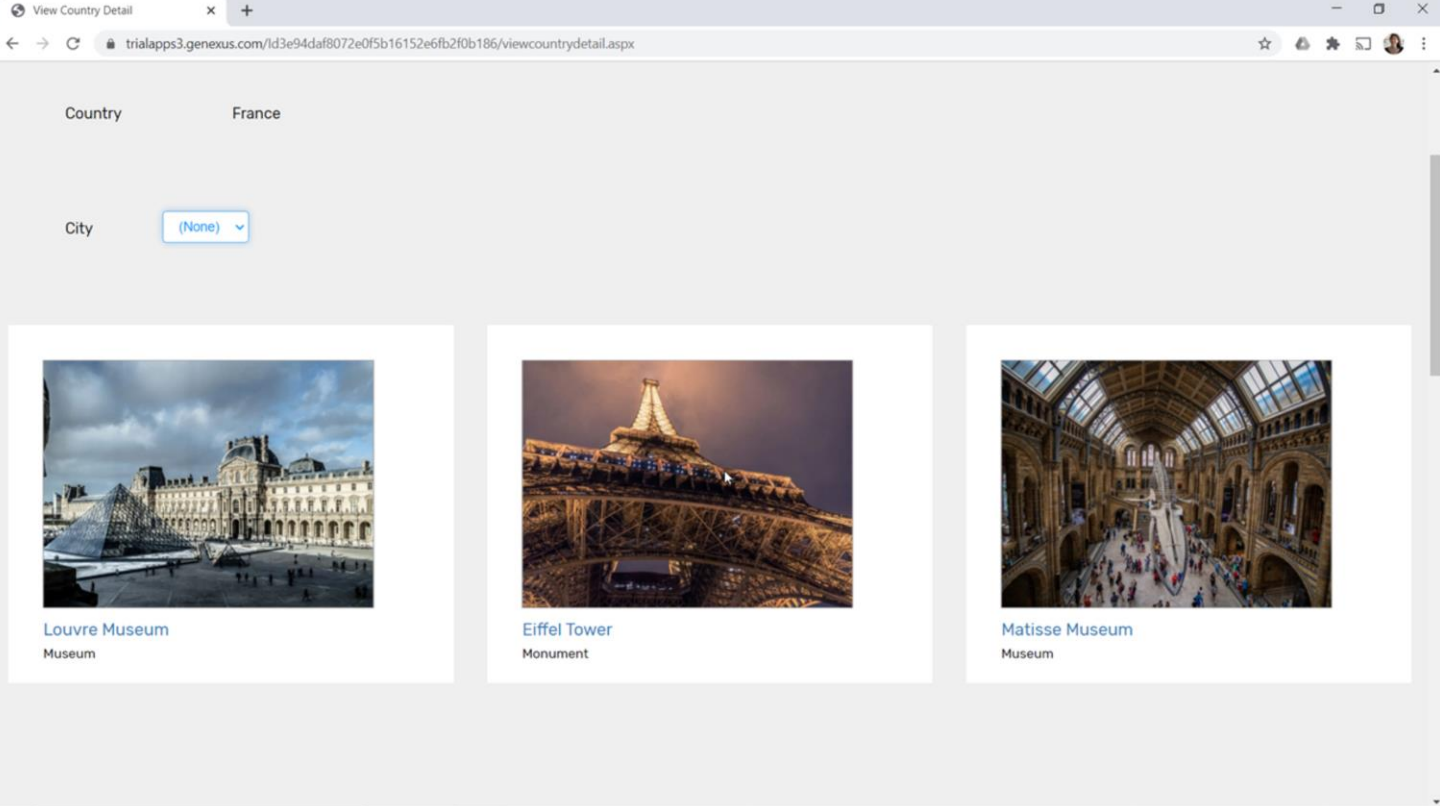
Country France

City

- (None)
- (None)**
- Paris

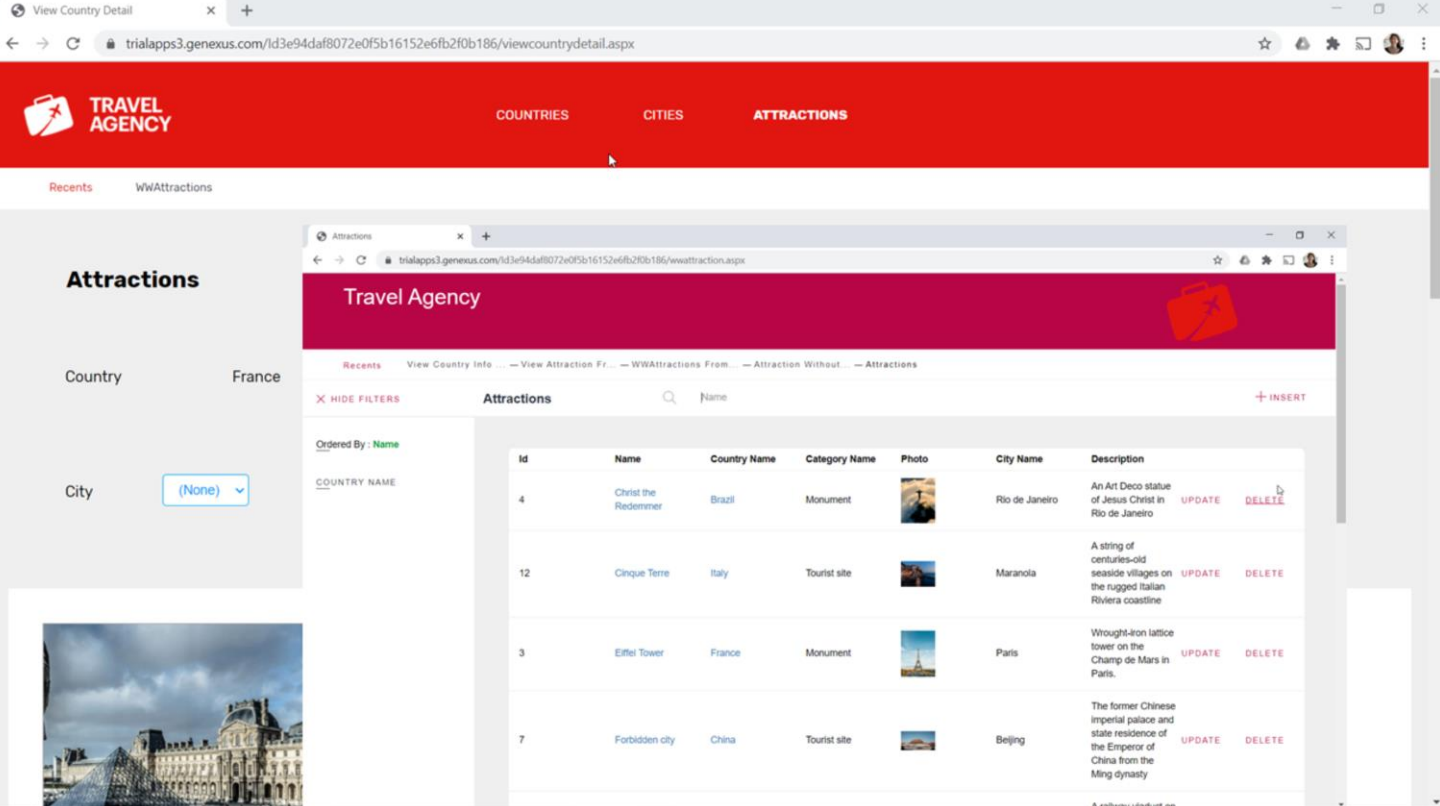


Vemos que este panel está sendo carregado com dados fixos, que são os que se importaram com o Sketch, para que se possa visualizar imediatamente como ficaria. Por isso, vemos o nome de país France e um combo que permite escolher entre None e Paris.



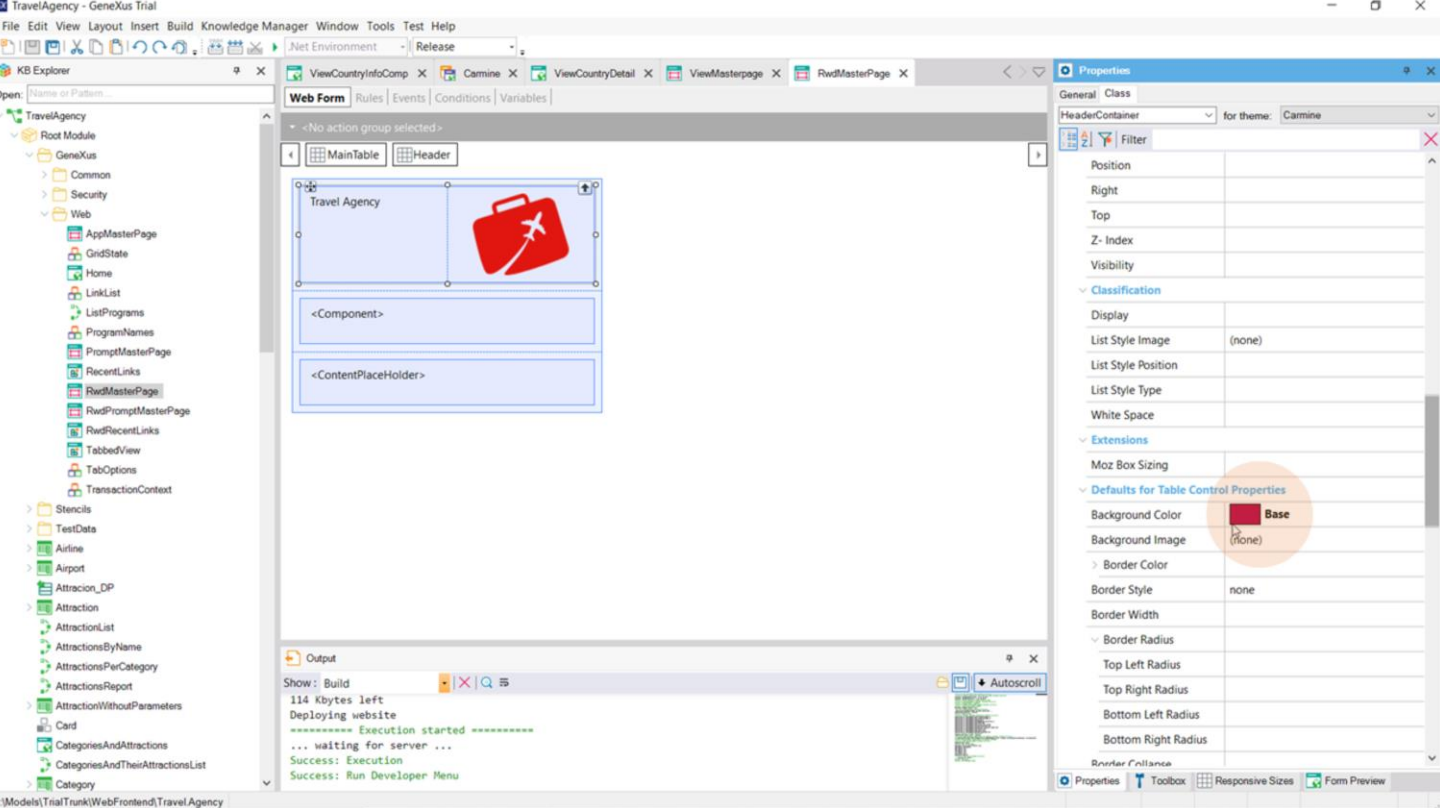
Assim como vemos três atrações turísticas que estão *hardcodeadas*, ou seja, especificadas manualmente. Evidentemente teremos que alterar tudo isto para pegar os dados da base de dados.

E talvez modificar algumas coisas, como o espaçamento entre controles, dado que esta importação não está sendo pixel-perfect, como será em pouco tempo uma vez que amadureça. Mas, para começar, a importação direta do desenho nos tem poupado muito trabalho. Pensemos na diferença entre isto e implementar tudo do zero.

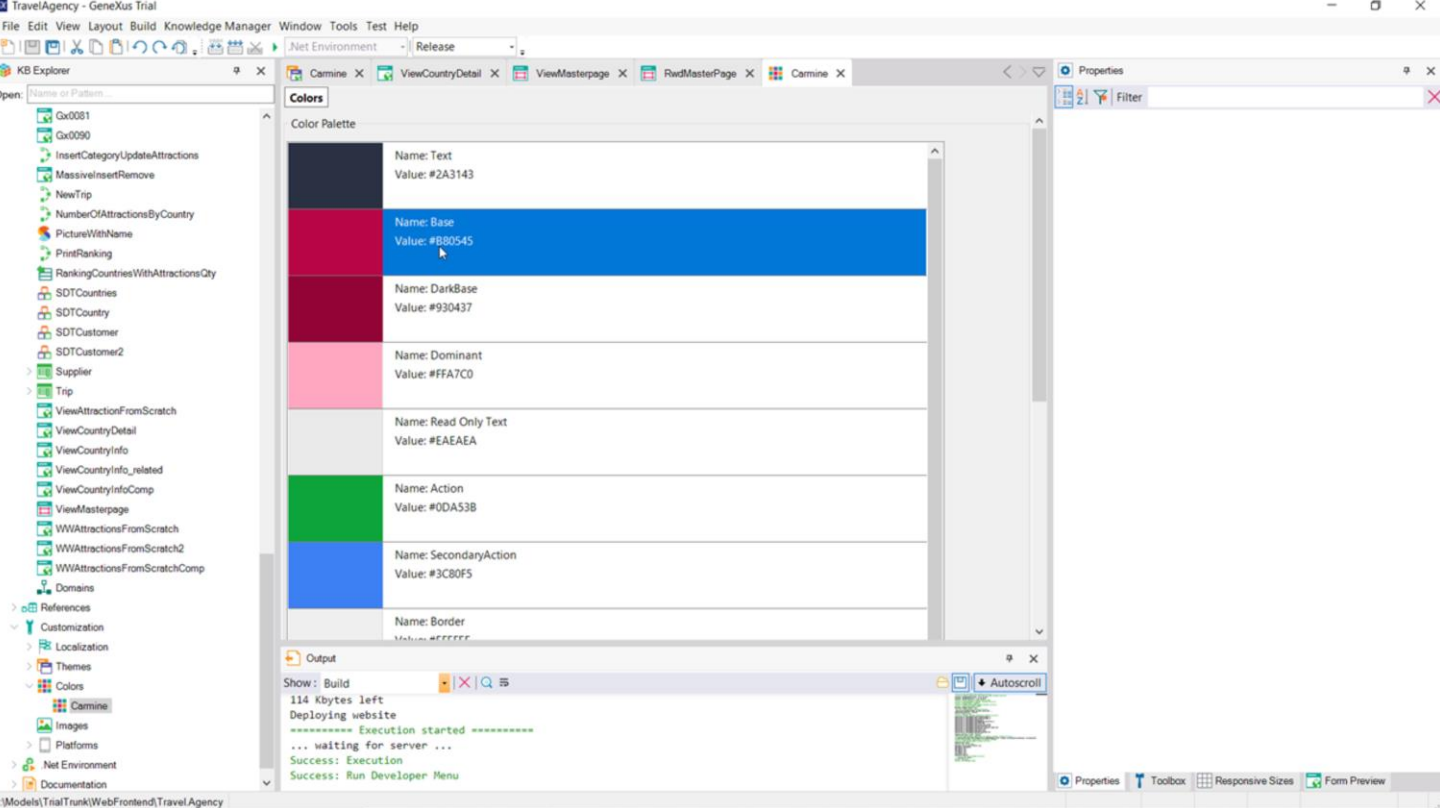


Evidentemente, se nossa intenção é que a página mestra deste painel se transforme na página mestra de todo o back-office, a cor de base do resto da aplicação deverá ser esta e não a anterior.

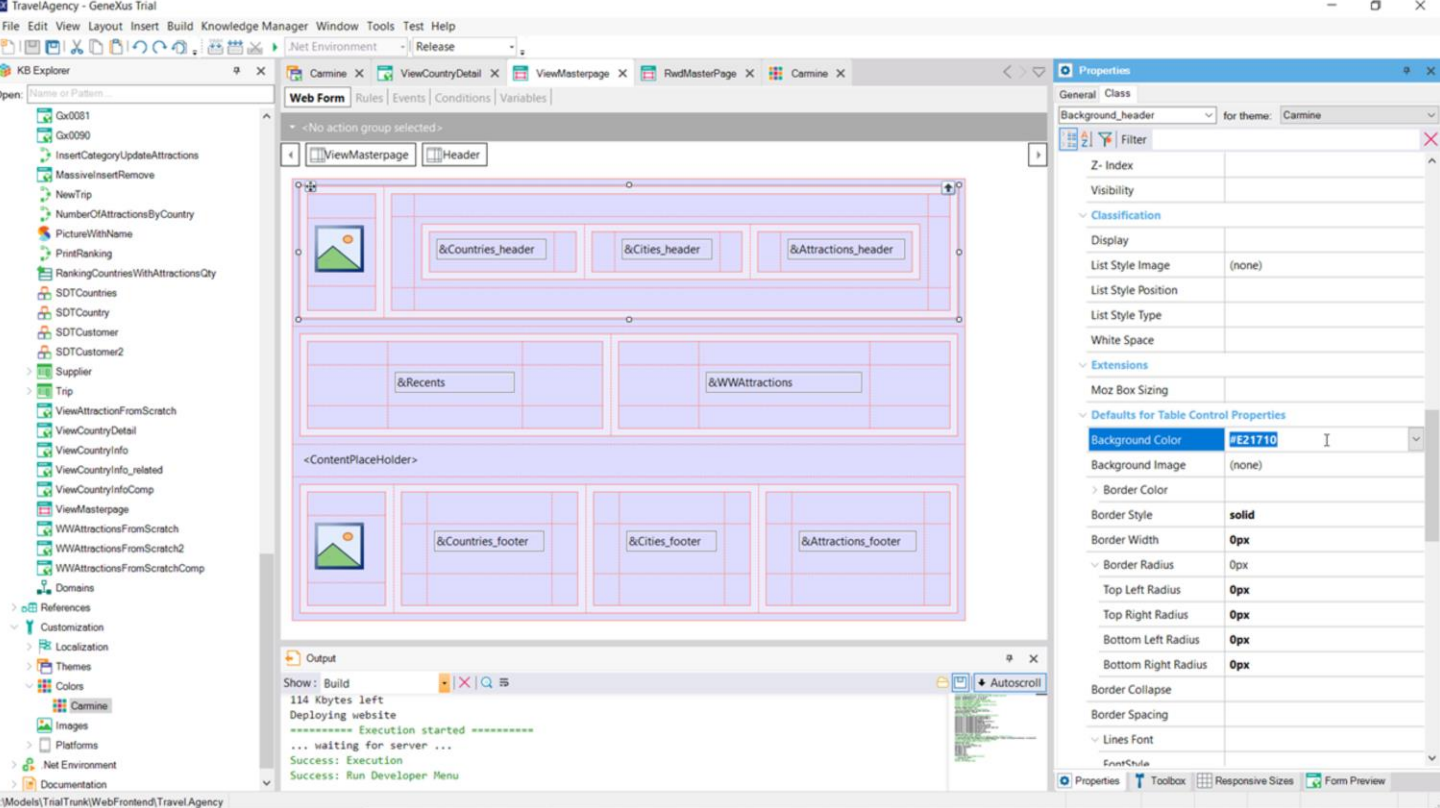
Não só a barra da aplicação, mas todas as ações e botões deverão assumir esta outra cor.



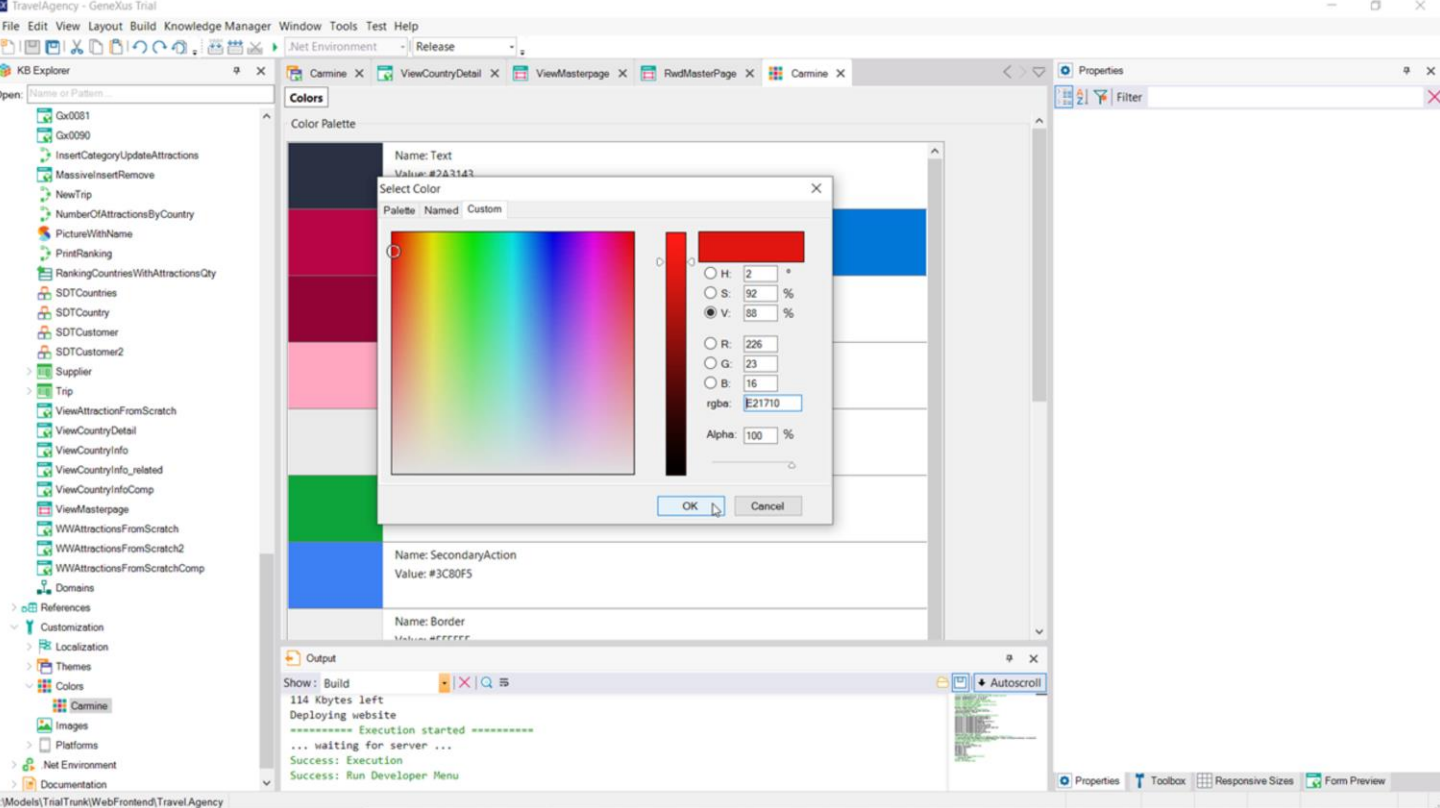
Se observamos a classe da tabela que implementa a barra da aplicação na master page default, vemos que a cor de fundo era esta, Base. Devemos modificá-la. Onde se encontra configurada?



Na paleta de Cores associada ao tema Carmine, que recebe o mesmo nome. Ali, então, modificamos sua cor pela que os designers utilizaram na barra da aplicação da nova página mestra.



Copiamos o código.



Clicamos duas vezes aqui e o colamos neste outro lugar.

Pronto. Gravamos e executamos.





Travel Agency

Recapts WWAttractions From... — Attractions

X HIDE FILTERS Attractions + INSERT

Ordered By: Name

COUNTRY NAME

Id	Name	Country Name	Category Name	Photo	City Name	Description		
4	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	An Art Deco statue of Jesus Christ in Rio de Janeiro	UPDATE	DELETE
12	Cinque Terre	Italy	Tourist site		Maranola	A string of centuries-old seaside villages on the rugged Italian Riviera coastline	UPDATE	DELETE
3	Eiffel Tower	France	Monument		Paris	Wrought-iron lattice tower on the Champ de Mars in Paris.	UPDATE	DELETE
7	Forbidden city	China	Tourist site		Beijing	The former Chinese imperial palace and state residence of the Emperor of China from the Ming dynasty	UPDATE	DELETE
						A railway viaduct on		

Attractions x View Country Info Comp x +

trialapps3.genexus.com/ld3e94daf8072e0f5b16152e6fb2f0b186/viewcountryinfocomp.aspx?CountryId=2

Travel Agency

Recents Attraction — Attractions — WWAttractions From... — View Attraction Fr... — View Country Info ...


Country Name France

Attraction Name From

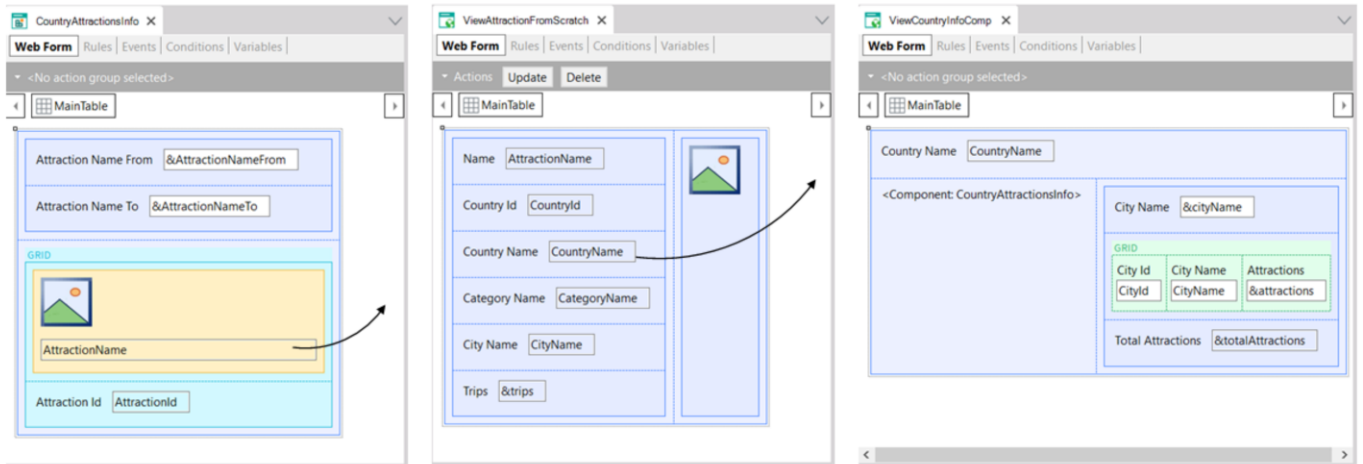
Attraction Name To

City Name

City Id	City Name	Attractions
1	Paris	2
2	Nice	1
Total Attractions		3



E se agora queremos mudar a página mestra para estes web panels?



Master Page: ViewMasterpage

Master Page: ViewMasterpage

Vamos para GeneXus...

Este é o componente que mostra em um grid horizontal as atrações e quando clicamos no nome de uma, nos abre este outro, que é o que mostra a informação da atração, e se clicamos em seu nome de país, nos abre o último, que é o que depois substituiremos pelo novo.

Então, vamos modificar a página mestra para este, e para o anterior.

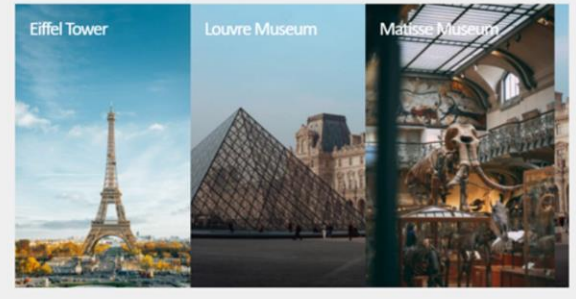
Country Name France

Attraction Name From

City Name

Attraction Name To

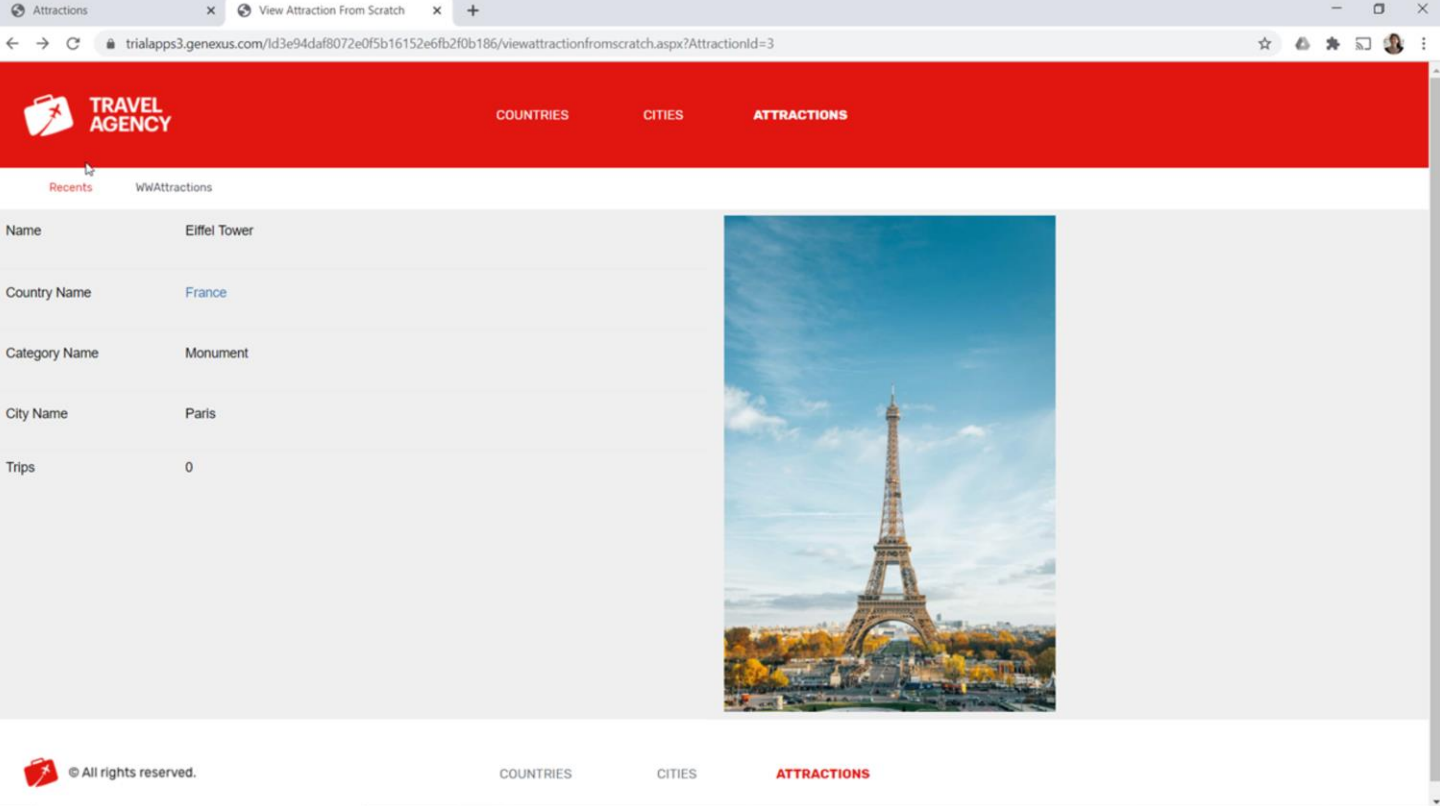
City Id City Name Attractions



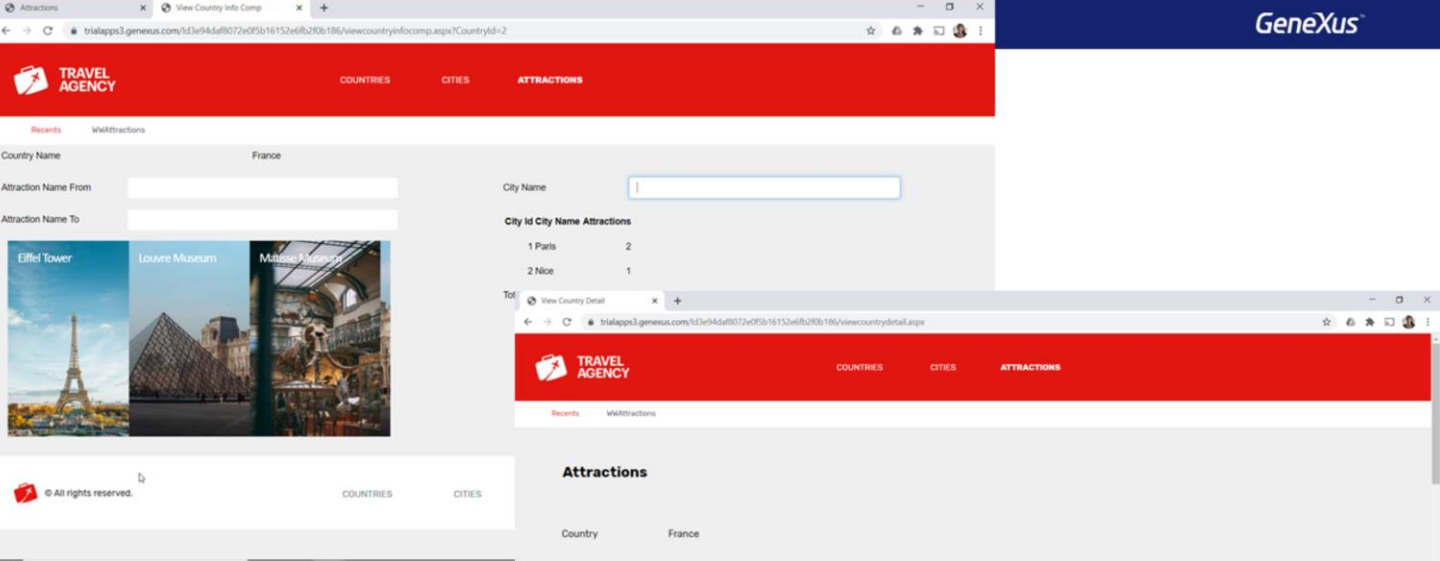
1	Paris	2
2	Nice	1
Total Attractions		3

Testemos.

Atualizamos. E vemos aqui perfeitamente aplicada a master page.



E também para o panel anterior: o que chamou este último. Não é assim para o primeiro porque não o mudamos.



Agora queremos substituir este panel pelo novo, mas carregando todos os seus dados da base de dados.

Web Form Rules Events Conditions Variables

```
1 parm(in: &CountryId);
2
```

CountryAttractionsInfo x ViewAttractionFromScratch x ViewCountryInfoComp x Camine x ViewCountryDetail * x Properties

Web Form Rules * Events Conditions Variables * <No action group selected>

ViewCountryDetail Wrap control0 WrapCountry_name Country_name

&Attractions

Country &Country_name

City &Combobox_City

GRID

Name	Type
& Variables	
Standard Variables	
Attractions	VarChar(40)
Combobox_City	Name
Country_name	Name
CountryId	Attribute:CountryId
ViewCountryDetail_Grid_At...	ViewCountryDetail_Grid_Attractions_SDT

Control Name	&Country_name
Attribute	&Country_name
ReadOnly	True
Return On Click	False
On Click Event	
Appearance	
Label Position	None
Class	Country_name
Invite Message	
Auto Resize	True
Width	40chr
Height	1row
Maximum Number Of Li	0
Format	Text
Tooltip Text	
Control Info	
Control Type	Edit
Input Type	Values
Suggest	No
Auto completion	True

A primeira coisa que devemos fazer é receber o identificador de país como parâmetro. Adicionamos a variável ao objeto, que estará baseada no atributo de mesmo nome.

Então, vemos que a variável Country_name deveria ter como tipo de dados o do atributo CountryName, que é Name, então nós o modificamos.

```

1 /* Generated by GeneXus Sketch Import [Start] */
2
3 Event Load
4   For &ViewCountryDetail_Grid_Attractions_SDT in ViewCountryDetail_Grid_Attractions_DP()
5     Load
6   EndFor
7 EndEvent
8
9 /* Generated by GeneXus Sketch Import [End] */
10
11
12 Event Start
13   For each Country
14     where CountryId = &CountryId
15     &Country_name = CountryName
16   endfor
17 Endevent

```

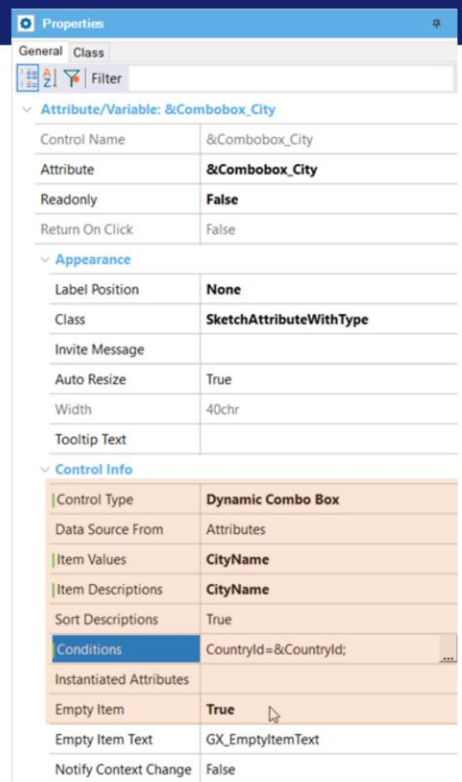
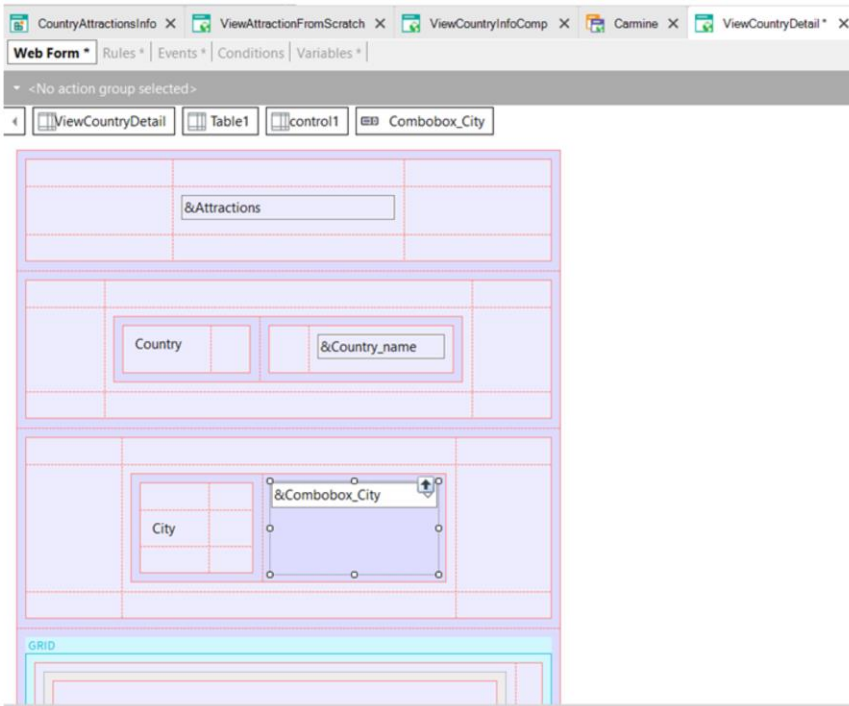
&Country_name = find(CountryName, CountryId = &CountryId, "")

Variable: &Country_name	
Name	Country_name
Description	Country_name
Column title	Name
Class	Attribute
Type Definition	
Based on	Name
Data Type	Character
Length	50
Collection	False
Dimensions	Scalar
Initial value	France
Validation	

Este web panel não tem tabela base, como mostraremos. Por isso, temos que ir procurar na base de dados o nome do país recebido por parâmetro. Para isso, podemos programar um for each no evento Start. Vamos procurar na tabela Country o registro para o qual CountryId é igual à variável do parâmetro e para esse registro atribuímos à variável do form &Country_name o valor do atributo CountryName.

Outra maneira de obter o mesmo teria sido utilizar a fórmula inline find, em vez do for each.

Agora vamos tirar esse valor inicial que estava assumindo porque já não precisamos dele.



Devemos carregar esta variável combo box com as cidades desse país. Portanto, devemos mudar o tipo de controle para que se trate de um combo cujos valores sejam carregados dinamicamente da base de dados, e não de forma estática, como era o caso. Os valores que o combo vai assumir serão os do atributo CityName e as descrições serão iguais, CityName (o que salvará internamente e o que mostrará será o mesmo). Mas também não queremos mostrar todas as cidades, apenas as do país recebido por parâmetro. Deixamos em true a propriedade para adicionar o item vazio (esse valor None que víamos em execução).

The screenshot displays the GeneXus IDE interface. On the left, a web form is being designed, featuring a grid with a placeholder for an attraction image and text fields for attraction name and type. The main window shows the 'Properties' panel for the selected element, 'Attraction_name', with attributes like 'Attribute', 'Field Specifier', 'Readonly', 'Return On Click', and 'On Click Event'. The 'Appearance' section shows the class 'Attraction_name'. The 'Behavior' section shows 'Input History' as True, 'Is Password' as False, and 'Nulls in Forms' as Empty as Null. Below the main window, the 'Structure' window shows the 'ViewCountryDetail_Grid_Attractions_SDT' object with its components: 'Layout', 'Attraction_image', 'Attraction_name', and 'Attraction_type', each with a 'Type' of 'VarChar(40)' and a 'Description'.

Bem, agora precisamos modificar o que aparece no grid e também filtrá-lo.

Como vemos, o grid não tem Transação Base especificada, nem Order, nem Conditions. Nem sequer tem atributos, mas elementos de um SDT que foi criado ao importar de Sketch.

Então este grid carregará uma coleção desse SDT. Para cada item, serão exibidos seus elementos de acordo com o desenho do stencil que vimos.

Aqui está definido o SDT, com um elemento que armazenará uma imagem da atração, outro o nome e outro a categoria.

Esses elementos substituirão no stencil do grid estas variáveis genéricas.

```

ViewCountryDetail_Grid_Attractions_DP X
Source Rules Variables
1 ViewCountryDetail_Grid_Attractions_DP from Attraction
2 where CountryId = &CountryId when not &CountryId.IsEmpty()
3 where CityName = &CityName when not &CityName.IsEmpty()
4
5 {
6     ViewCountryDetail_Grid_Attractions_SDT
7     {
8         Layout = "Attraction"
9         Attraction_image = AttractionPhoto
10        Attraction_name = AttractionName
11        Attraction_type = CategoryName
12    }
13
14    // ViewCountryDetail_Grid_Attractions_SDT
15    // {
16    //     Layout = "Attraction"
17    //     Attraction_image = Image:Img716fb22b18d7152e866600cbeaab8003628573c9.Link()
18    //     Attraction_name = "Louvre Museum"
19    //     Attraction_type = "Museum"
20    // }
21    // ViewCountryDetail_Grid_Attractions_SDT
22    // {
23    //     Layout = "Attraction"
24    //     Attraction_image = Image:Img4a3b2877897efdc75c540e9d29a16f31520ad4f1.Link()
25    //     Attraction_name = "Eiffel Tower"
26    //     Attraction_type = "Monument"
27    // }
28    // ViewCountryDetail_Grid_Attractions_SDT
29    // {
30    //     Layout = "Attraction"
31    //     Attraction_image = Image:a37c31622388e897a74a56d1aa41284498962234.Link()
32    //     Attraction_name = "Matisse Museum"
33    //     Attraction_type = "Museum"
34    // }
35 }

```

```

ViewCountryDetail_Grid_Attractions_DP X
Source Rules Variables
1 param(in: &CountryId, in: &CityName);
2

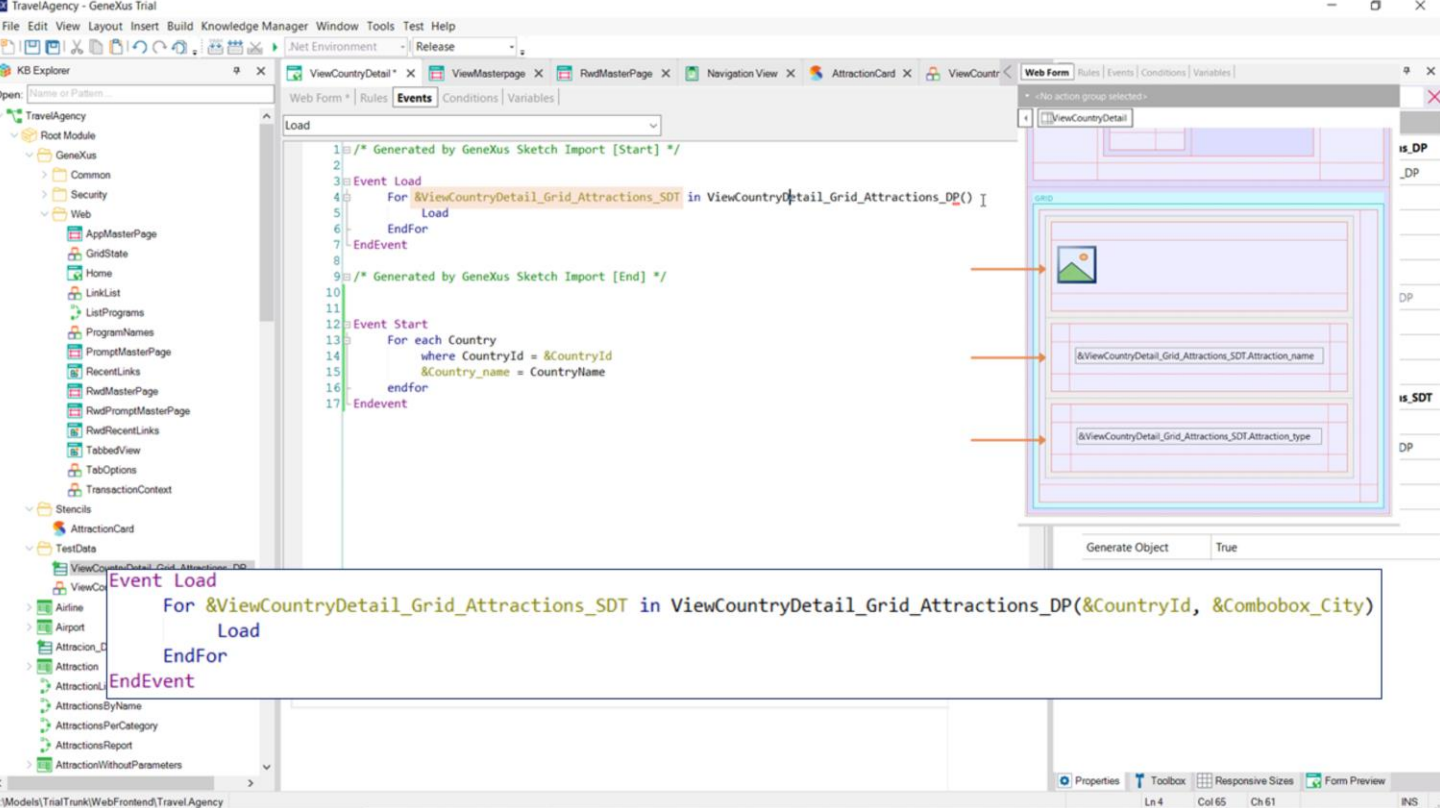
```

Mas quem realmente devolverá uma coleção de dados deste SDT com as atrações carregadas será este DataProvider importado com o Sketch. Tudo o que temos que fazer é modificá-lo para que os itens sejam carregados a partir das atrações da base de dados, e não destes dados fixos.

Por isso, deixamos comentados estes três grupos e acrescentamos um que será variável, especificando que pegaremos os dados da transação base Attraction, filtrando por identificador de país quando não está vazio. E pelo nome da cidade, quando não estiver vazia. Deveremos definir ambas as variáveis na regra parm.

Mas antes, para cada registro encontrado, queremos que se adicione um item na coleção resultante, com o valor da foto da atração, seu nome e o nome de sua categoria.

Agora sim, definamos os dois parâmetros que este data provider deverá receber ao ser invocado.



E agora vamos ver a sua invocação.

Esta é realizada no evento Load, claro!

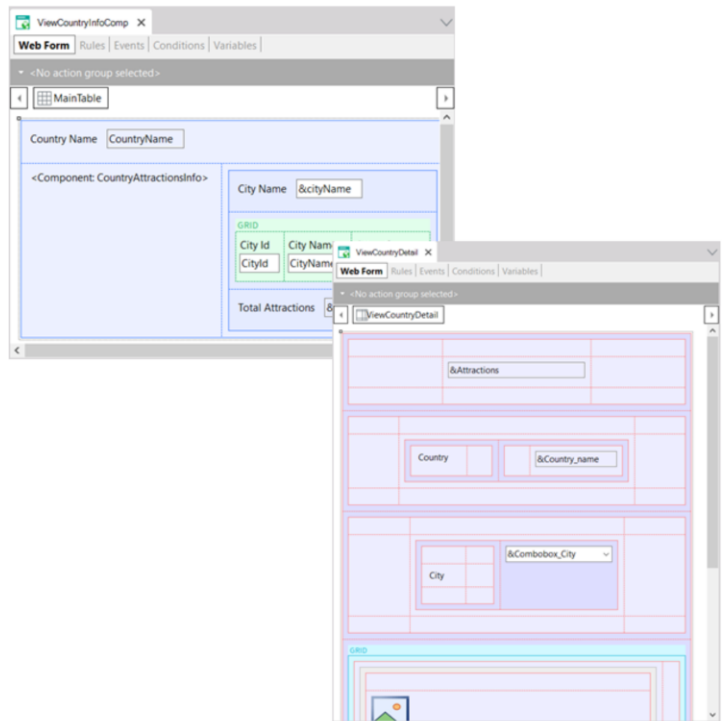
Vemos que temos um comando For in que percorre cada item da coleção devolvida pelo Data Provider, e o carrega no grid.

Aqui é onde deveremos enviar-lhe os parâmetros, para que filtre por país e cidade.

```

ViewAttractionFromScratch * X
Web Form | Rules | Events * | Conditions | Variables |
CountryName.Click
1 Event Load
2   &trips = Count(TripDate)
3 Endevent
4
5
6 Event CountryName.Click
7   //ViewCountryInfo(CountryId)
8   //ViewCountryInfo_related(CountryId)
9   //ViewCountryInfoComp(CountryId)
10  ViewCountryDetail(CountryId)
11 Endevent
12
13
14 Event Start
15   CountryId.Visible = False
16 Endevent
17

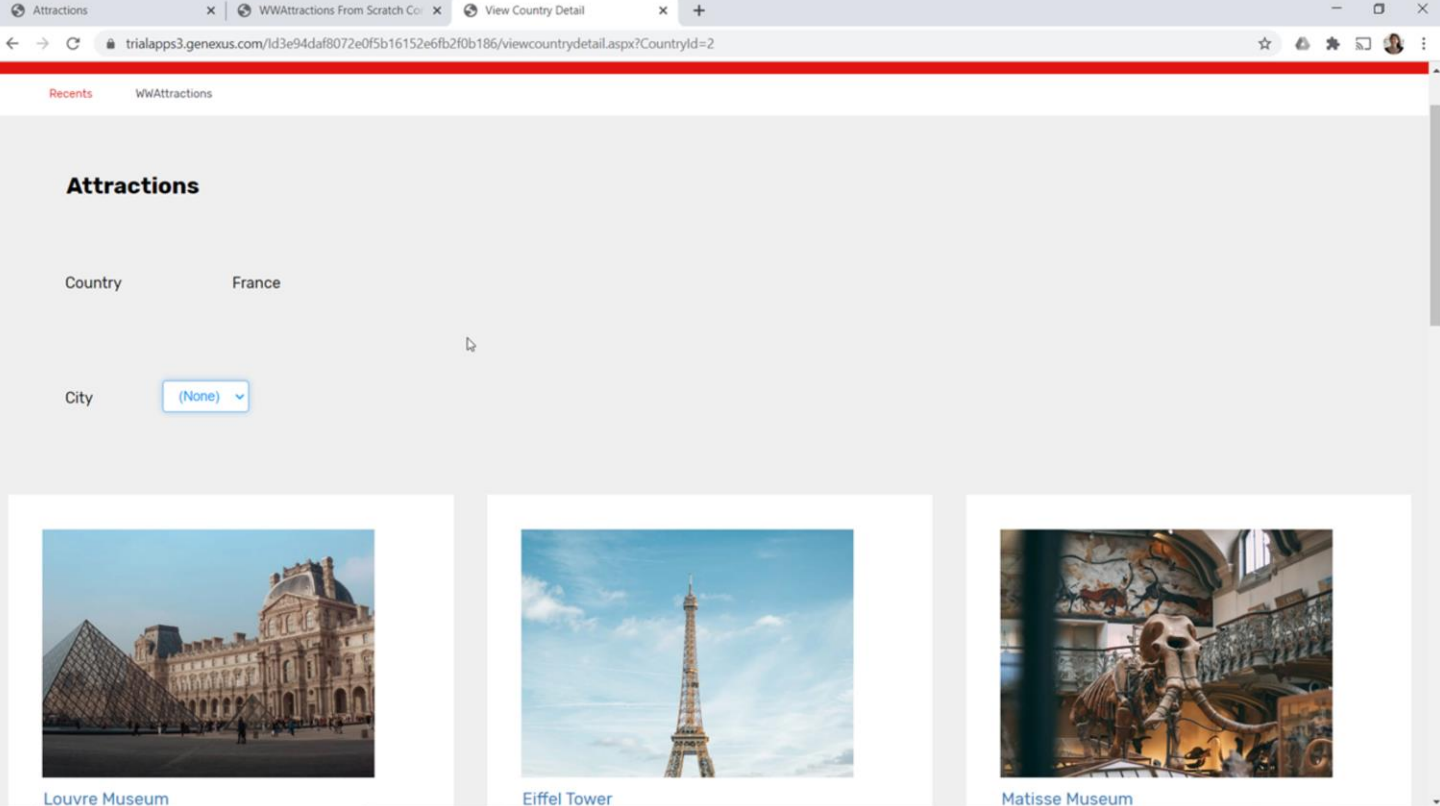
```



Só nos resta invocar este web panel em vez do que tínhamos.

Ou seja, em vez deste, este outro.

Gravamos e executamos.



Vamos escolher esta atração. E ver a informação de seu país, France.

Ali estamos vendo exibidas as três atrações que temos da França na base de dados.

Attractions


trialapps3.genexus.com/ld3e94daf8072e0f5b16152e6fb2f0b186/viewcountrydetail.aspx?CountryId=2

Recents WWAttractions


Attractions

Country France


City (None) (None) Nice Paris



Louvre Museum

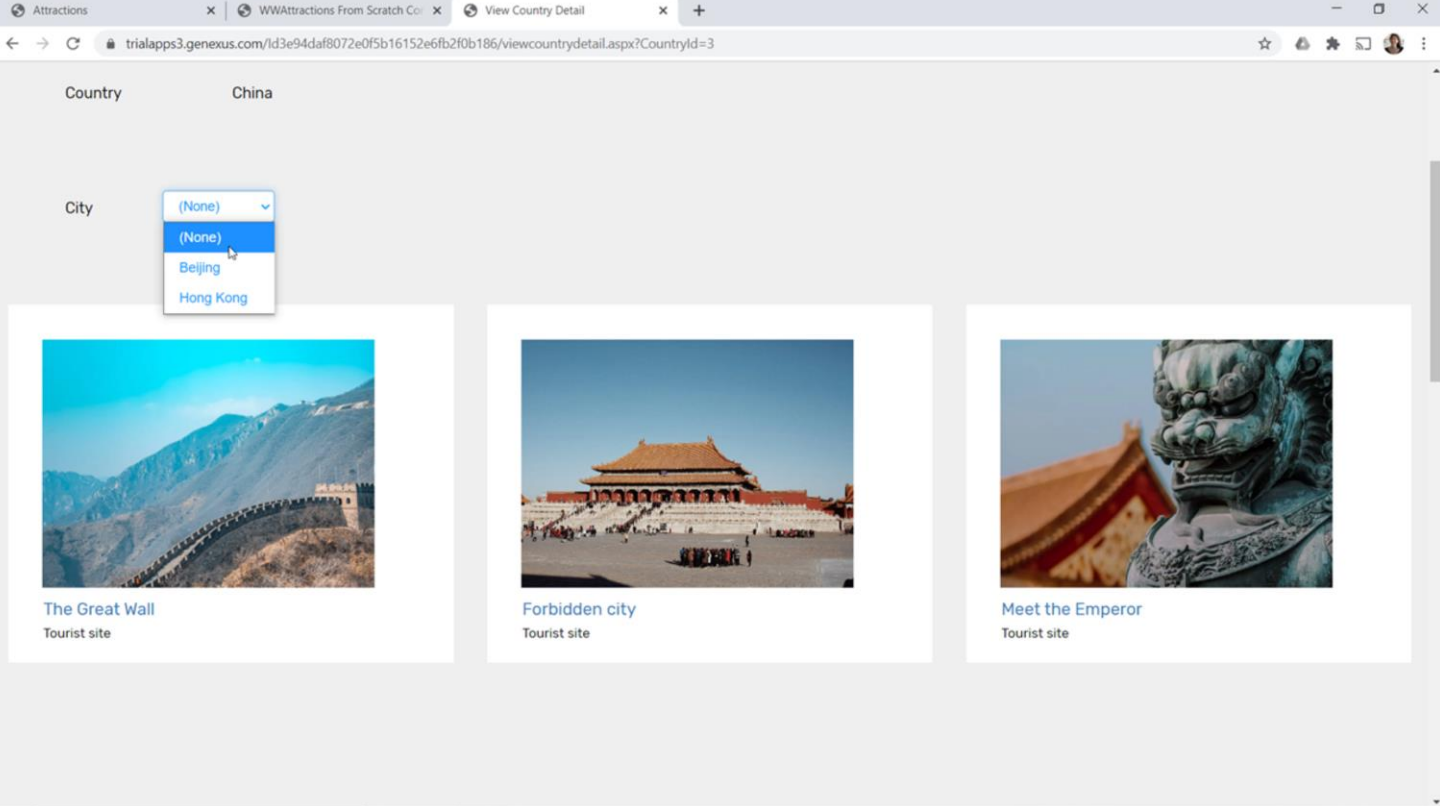


Eiffel Tower

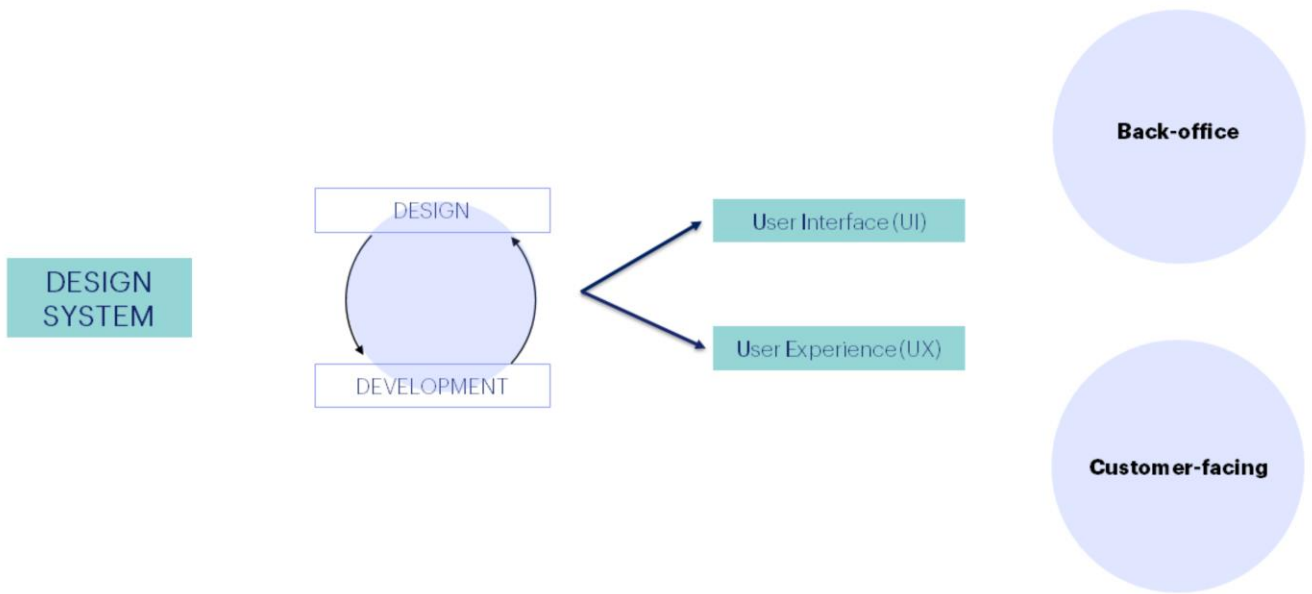


Matisse Museum

E as duas cidades, e filtrando por Paris vemos suas duas atrações e por Nice, sua atração.



Se escolhermos outra atração, da China desta vez, vemos as três atrações que temos da China.



Com este pequeno exemplo já podemos ir tendo uma ideia de como será o processo incremental de desenvolvimento de uma aplicação tanto nos aspectos de sua lógica, como de seu desenho, e em que medida nos convém trabalhar em equipe com pessoas com diferentes habilidades.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications