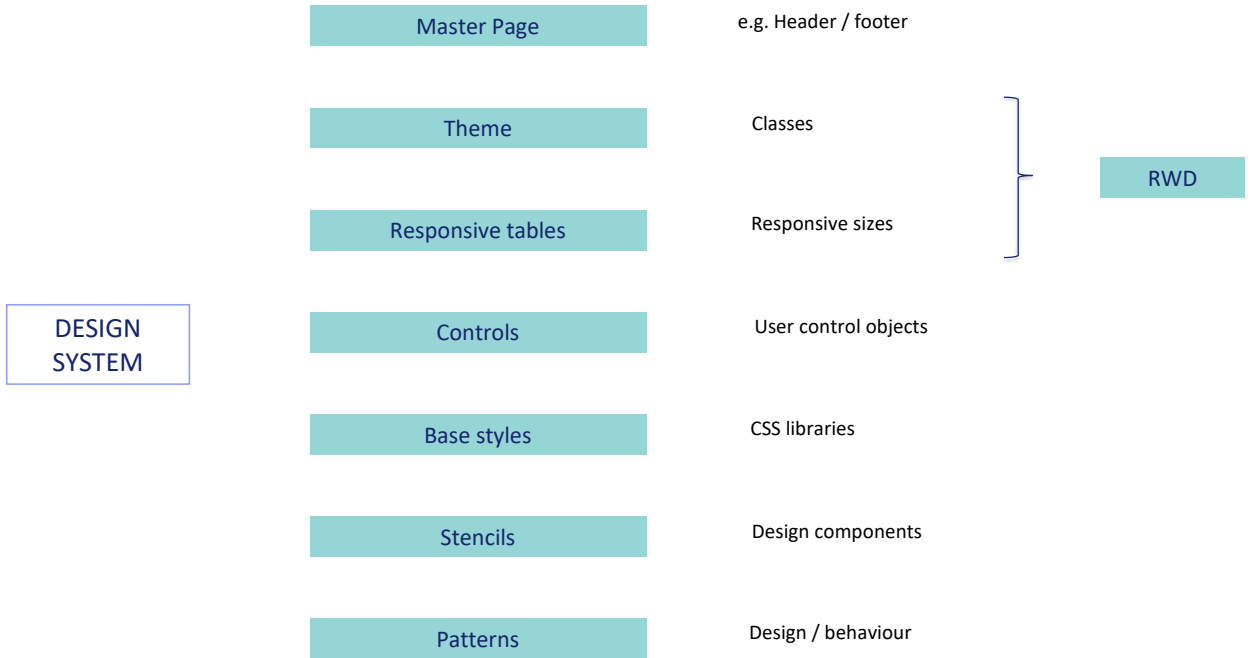


Telas Web

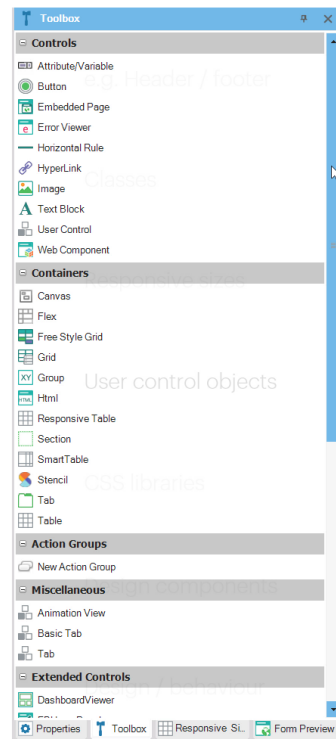
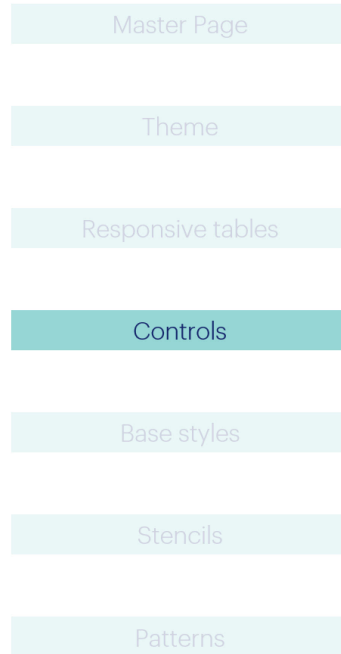
Objeto User Control

GeneXus™

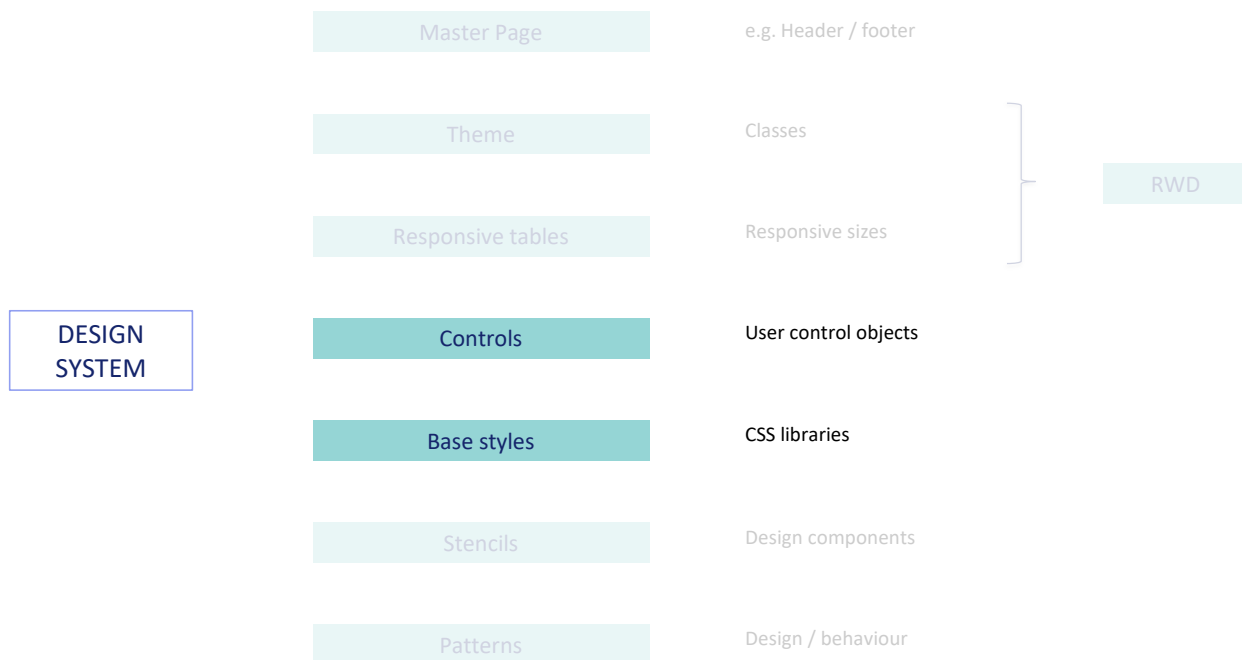


No curso inicial de GeneXus tínhamos abordado de forma superficial cada um dos elementos de GeneXus que participavam na implementação de um Design System para a aplicação.

DESIGN
SYSTEM



Entre eles, tínhamos dito que podíamos não só utilizar em nossos forms os controles que vêm na toolbox de GeneXus em forma predeterminada...



...mas também podíamos definir controles de usuário, que podem ser copiados de plataformas que oferecem esses controles junto com bibliotecas CSS (que são, como nossos themes, as que especificam seu estilo -o estilo desses controles-).

Card | Semantic UI

semantic-ui.com/views/card.html


UI Docs

- Getting Started
- New in 2.4
- Introduction
 - Integrations
 - Build Tools
 - Recipes
 - Glossary
- Usage
 - Theming
 - Layouts
- Globals
 - Reset
 - Site
- Elements
 - Button
 - Container
 - Divider
 - Flag
 - Header
 - Icon
 - Image
 - Input
 - Label
 - List
 - Loader
 - Placeholder

Card

A single card.

To ensure cards are equal height use the plural, `cards`. Card groups automatically uses `flex` styles to match height between cards in the same row.



Kristy
Joined in 2013
Kristy is an art director living in New York.
22 Friends

Card

- Types
- Card
- Cards
- Content
- Variations

Por exemplo, esta, Semantic UI.

Podíamos querer utilizar um controle Card como este, por exemplo...



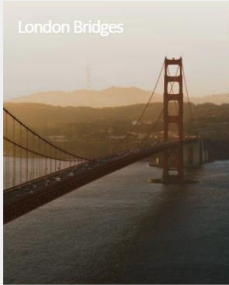



Recents WWAttractions From...

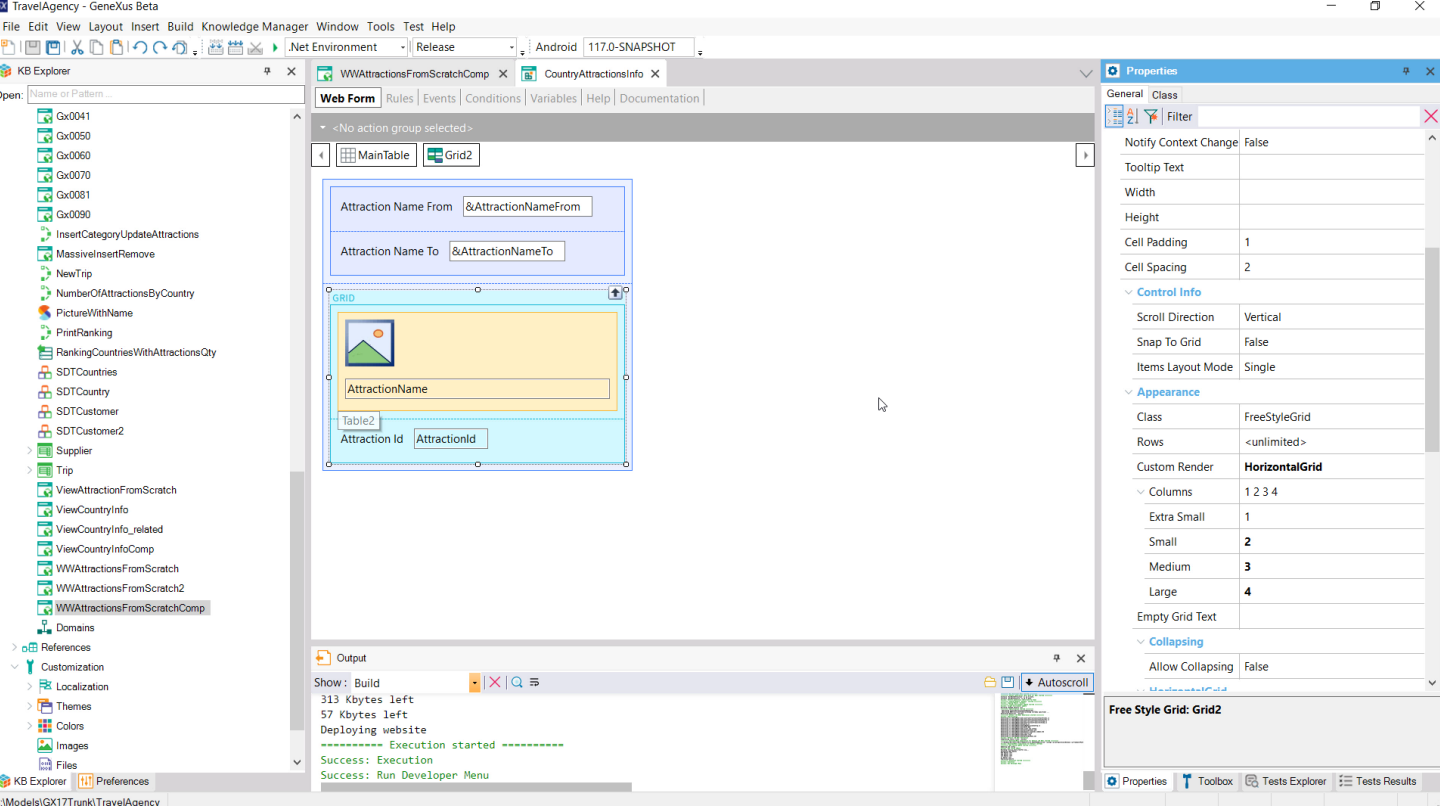
Country Id (None) v

Attraction Name From

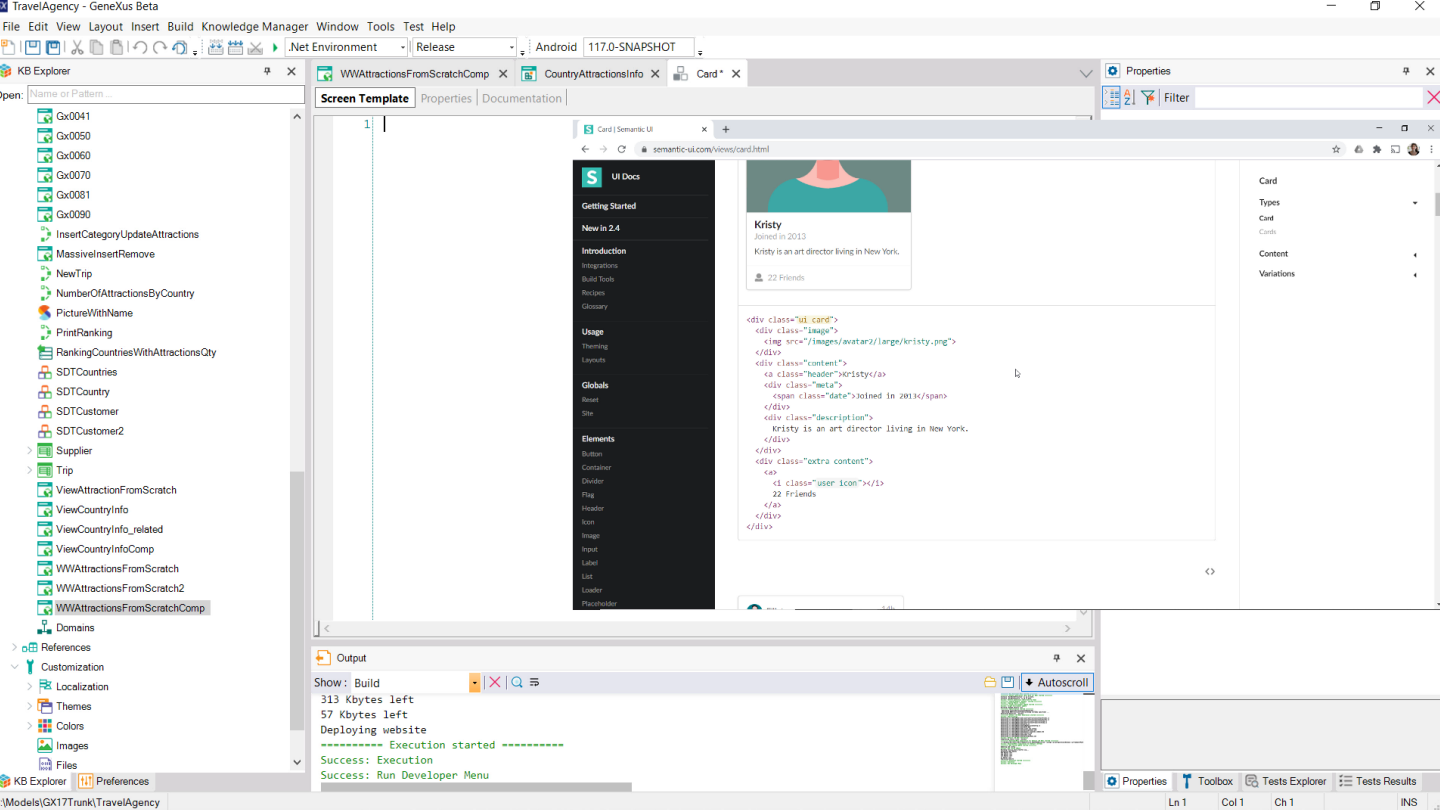
Attraction Name To



para mostrar as atrações turísticas, em vez da tabela canvas que tínhamos colocado no grid horizontal, na qual apenas tínhamos começado a colocar desenho.



Ali vemos o grid e aqui vemos a tabela canvas.



Para isso, conseguirá criando um objeto User control (podemos chamá-lo igualmente)...

...copiar e colar o html que nos oferece Semantic UI...


```
1 <div class="ui card">
2   <div class="image">
3     
6     <a class="header">{{Name}}</a>
7     <div class="meta">
8       <span class="date">{{Date}}</span>
9     </div>
10    <div class="description">
11      {{Description}}
12    </div>
13  </div>
14  <div class="extra content">
15    <a>
16      <i class="user icon"></i>
17      {{LinkToMore}}
18    </a>
19  </div>
20 </div>
```

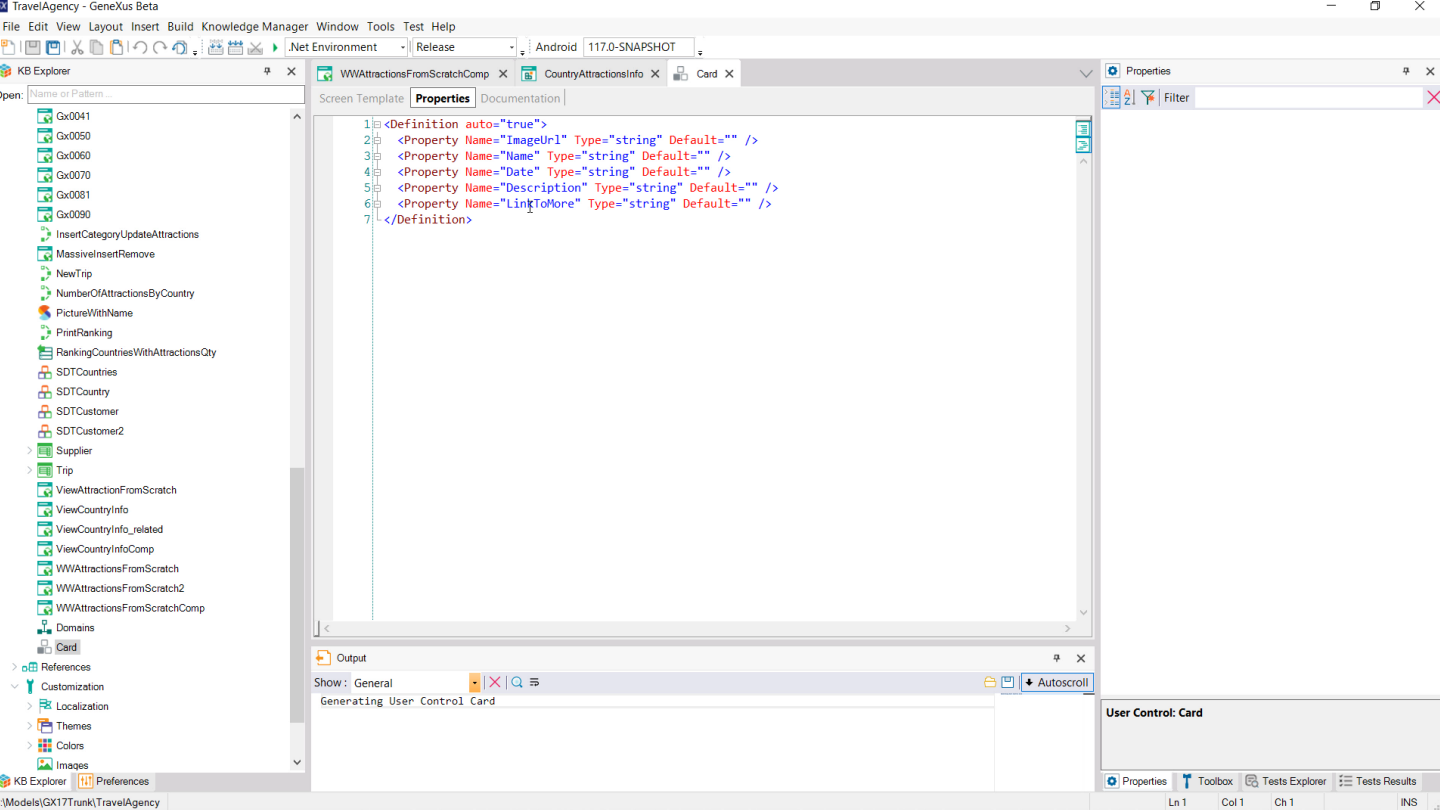
Name	Card
Description	Card
Module/Folder	Root Module
Is Control Type	False
References	
Base Control Type	None
Base Style	Default
Qualified Name	Card
Object Visibility	Public

Output
Show: General
Generating User Control Card

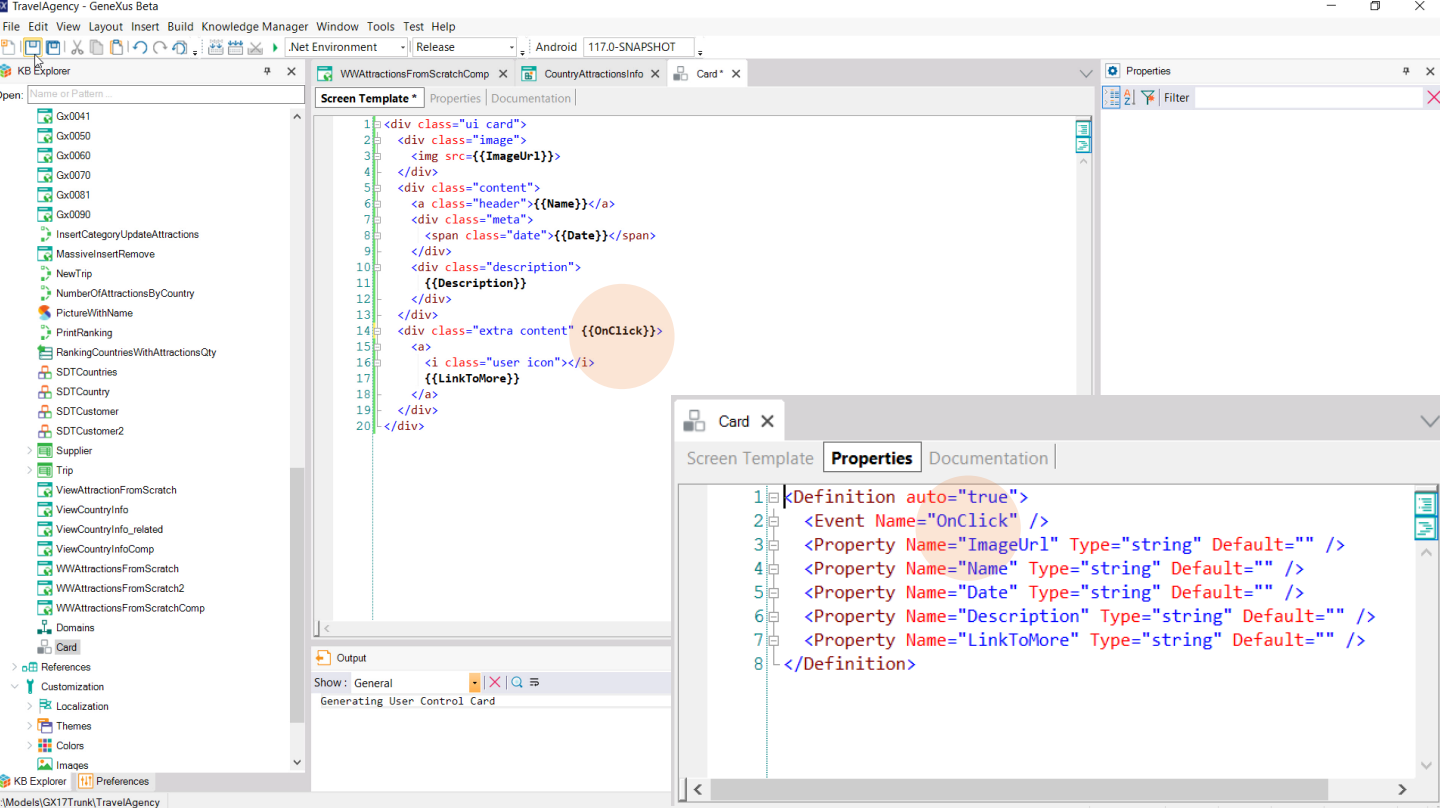
User Control: Card

...alterar estes dados fixos para algo assim, como nomes de elementos de um SDT (para poder utilizá-los de maneira dinâmica, ou seja, carregar dinamicamente esse controle com dados que vamos poder especificar, por exemplo, da base de dados).

Com esta sintaxe, que é conhecida como sintaxe de bigode, estamos então, modificando os dados de fixos para dados variáveis.



Se agora vamos ver as Properties, vemos que ali estão aparecendo.



Vamos também adicionar um evento, que é o evento que ocorre ao clicar sobre este componente do HTML.

E vemos que está aparecendo aqui.

Desta forma, vamos poder programar o clique sobre esse elemento em execução.

Card
A card displays site content in a manner similar to a playing card

4 Themes [Download](#)

Semantic UI
[Download ZIP](#) [Getting Started](#)

Package Managers

NPM
`npm install semantic-ui`

Git
`https://github.com/Semantic-Org/Semantic-UI.git`

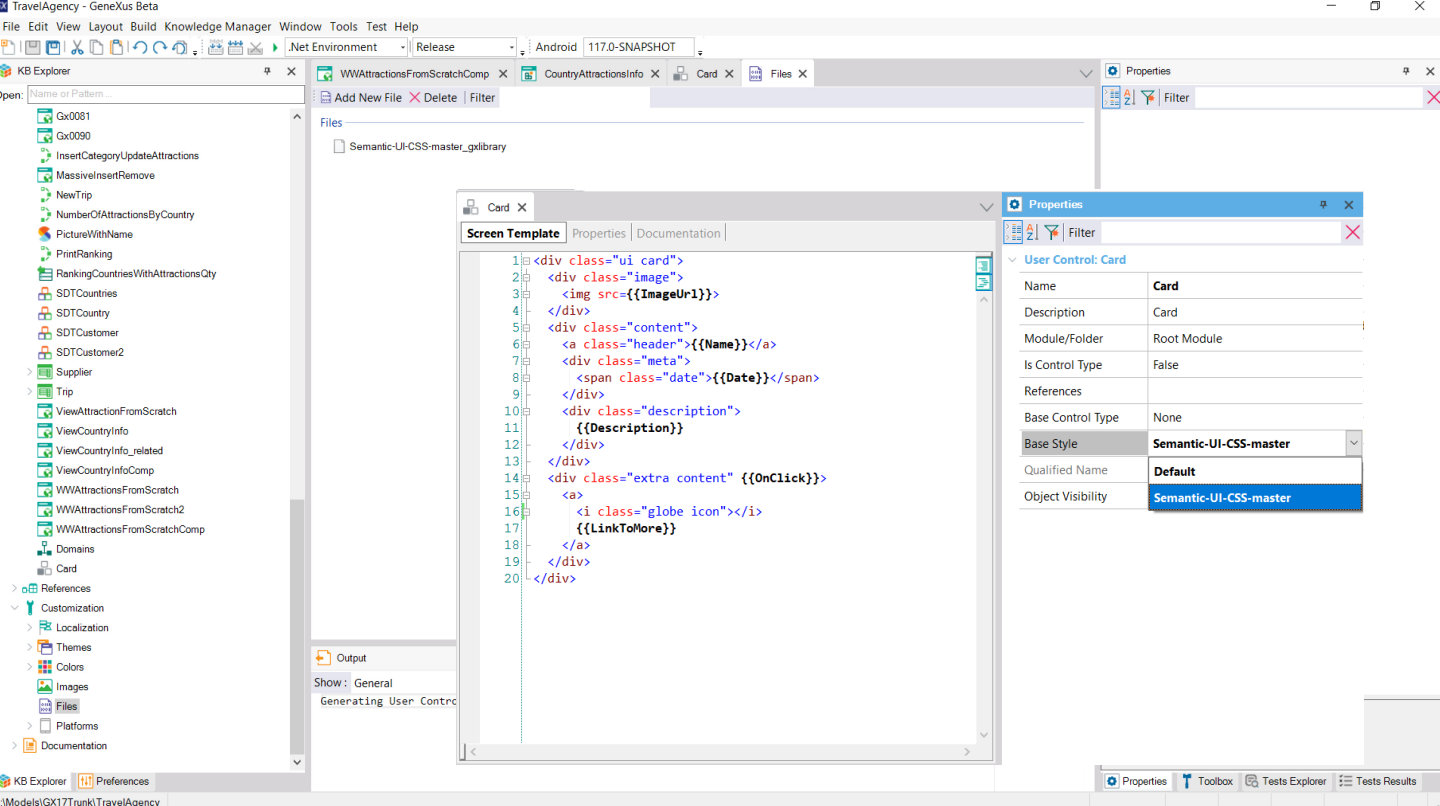
To ensure cards are equal height use the plural, `cards`. Card groups automatically uses `flex` styles to match height between cards in the same row.

Example

Card
Types
Card
Cards
Content
Variations

Agora o que vamos fazer é baixar o CSS, que é o que verdadeiramente implementa o desenho de todas estas tags do HTML.

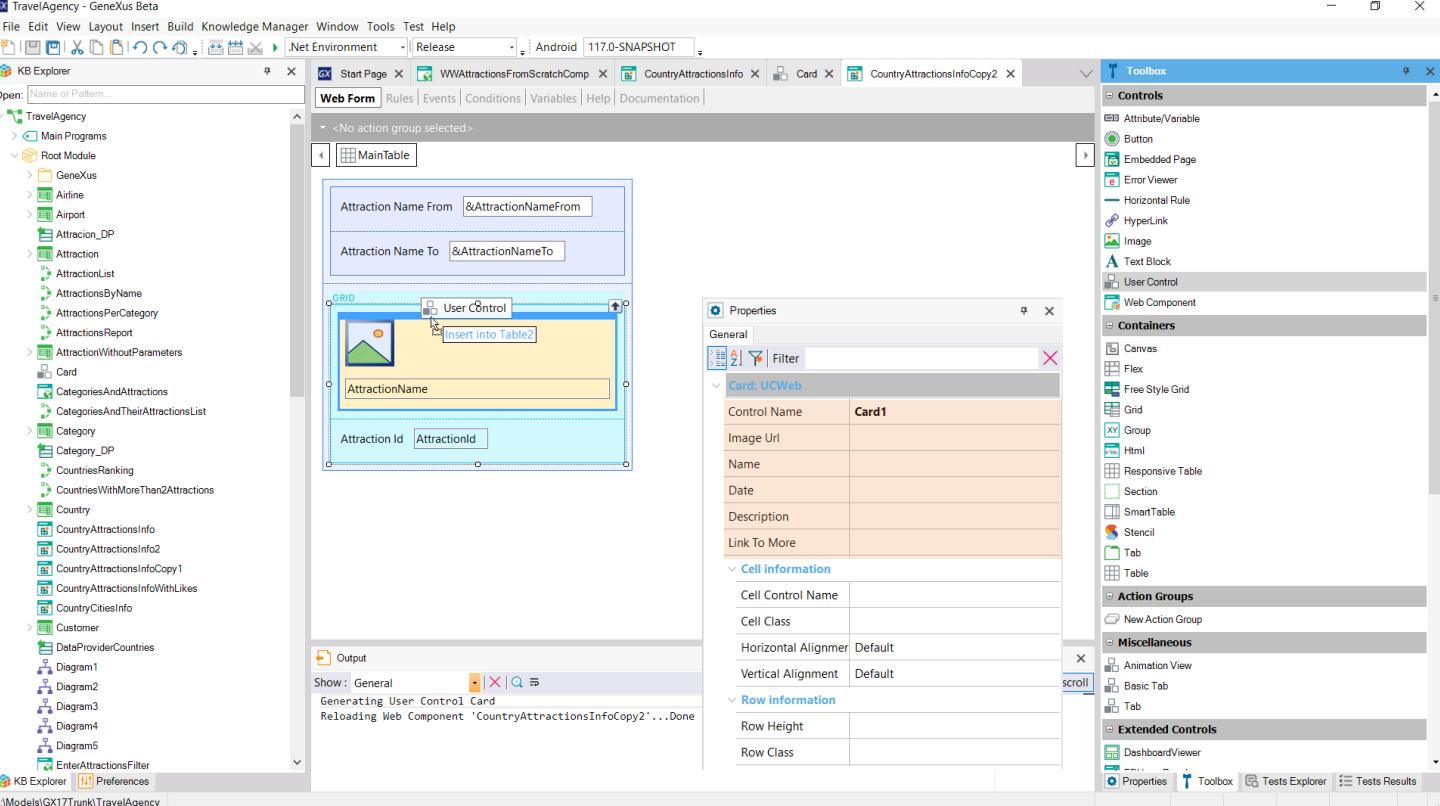
Aqui o temos. O baixamos como um zip. Vamos mudar a extensão para gxlbrary, que é a extensão que é necessária em GeneXus para interpretar este CSS.



Agora o que vamos fazer é inserir este CSS como um arquivo.

E, finalmente, vamos dizer ao user control Card que sua Base Style pegue de lá.

Bem, muito simples.



Já temos o User Control para ser utilizado em nossa KB. Agora o que vamos fazer é gravar este componente com outro nome e vamos inserir dentro do grid este User Control, Card.

Se vemos suas propriedades, vemos que estão aparecendo todas aquelas que modificamos, e aparecem tanto de forma estática, como para lhes atribuir um valor, como - como vamos ver – de forma dinâmica.

Attraction Name From	&AttractionNameFrom
Attraction Name To	&AttractionNameTo
GRID	
<Card: Card1>	
Country Id	CountryId

```

Card X CountryAttractionsInfoCopy2* X
Web Form | Rules | Events * | Conditions | Variables | Help | Documentation |
Events
1 | Event Start
2 |     CountryId.Visible = False
3 | Endevent
4 |
5 | Event Grid2.Load
6 |     Card1.ImageUrl = AttractionPhoto.ImageURI
7 |     Card1.Name = AttractionName
8 |     Card1.Description = AttractionDescription
9 |     Card1.LinkToMore = CountryName
10 | Endevent
11 |
12 | Event Card1.OnClick
13 |     ViewCountryInfo(CountryId)
14 | Endevent
15 |

```

Vamos remover tudo o que já não nos interessa. Vamos adicionar o atributo CountryId, porque vamos querer passá-lo como parâmetro. AttractionId já não nos interessa. E vamos colocar CountryId como invisível.

Bem, então agora o que vamos ter que fazer é carregar no evento Load esse User Control com os valores de cada atração. Então programaremos o evento Load do Grid. E vemos que, ao escrever o nome do controle, Card1, já nos aparecem todas as propriedades que vimos antes no User Control.

À propriedade Name vamos atribuir o valor do atributo AttractionName. À propriedade ImageUrl temos que dizer a url onde se encontrará a imagem que queremos que seja carregada nesse Card. E é a do atributo AttractionPhoto. Para a descrição, vemos que adicionamos à transação Attraction o atributo AttractionDescription, que contém então a descrição da atração turística. Carregamos, então, esse elemento do User Control com o valor do atributo.

E, por último, este vai ser o link. Queremos que seja carregado o valor de CountryName, ou seja, que diga o nome do país, e o que vamos fazer depois é estabelecer um link ali, ou seja, que CountryName seja um link para o painel que tínhamos definido para listar a informação do país, e suas atrações e cidades. Portanto, o evento que tínhamos antes, que era o que se executava ao clicar sobre AttractionName, já não precisamos mais dele, mas agora temos que programar o evento OnClick do user control. E o que faremos será invocar o painel que tínhamos, passando-lhe como

parâmetro o CountryId.

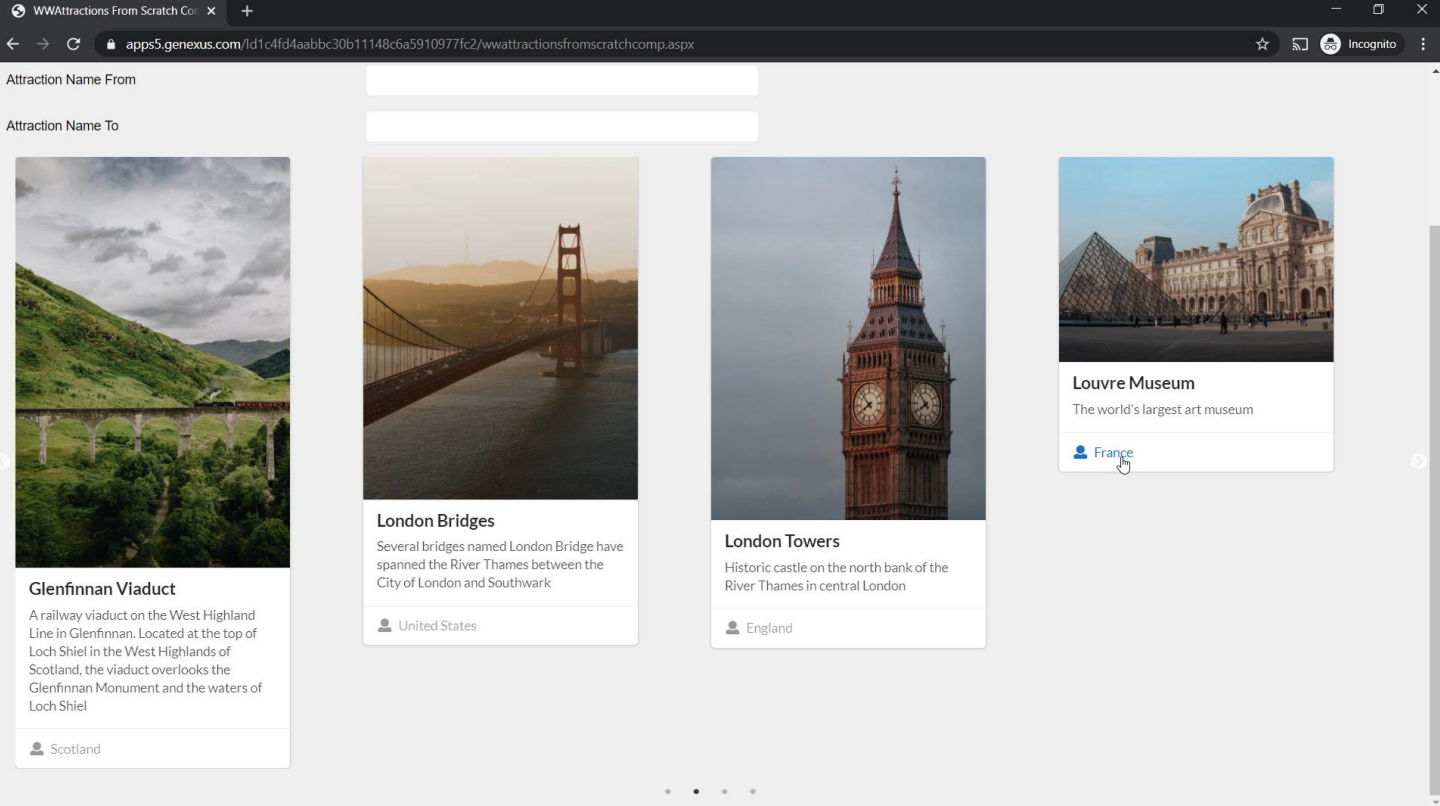
Country Id &CountryId

<Component: CountryAttractionsInfoCopy2>

Event &CountryId.ControlValueChanged
 Component1.Object = CountryAttractionsInfoCopy2.Create(&CountryId)
 Endevent

Web Component: Component1	
Control Name	Component1
Object	CountryAttractionsInfoCopy2
Parameters	&CountryId
Cell information	
Cell Control Name	
Cell Class	
Horizontal Alignmer	Default
Vertical Alignment	Default
Row information	
Row Height	
Row Class	

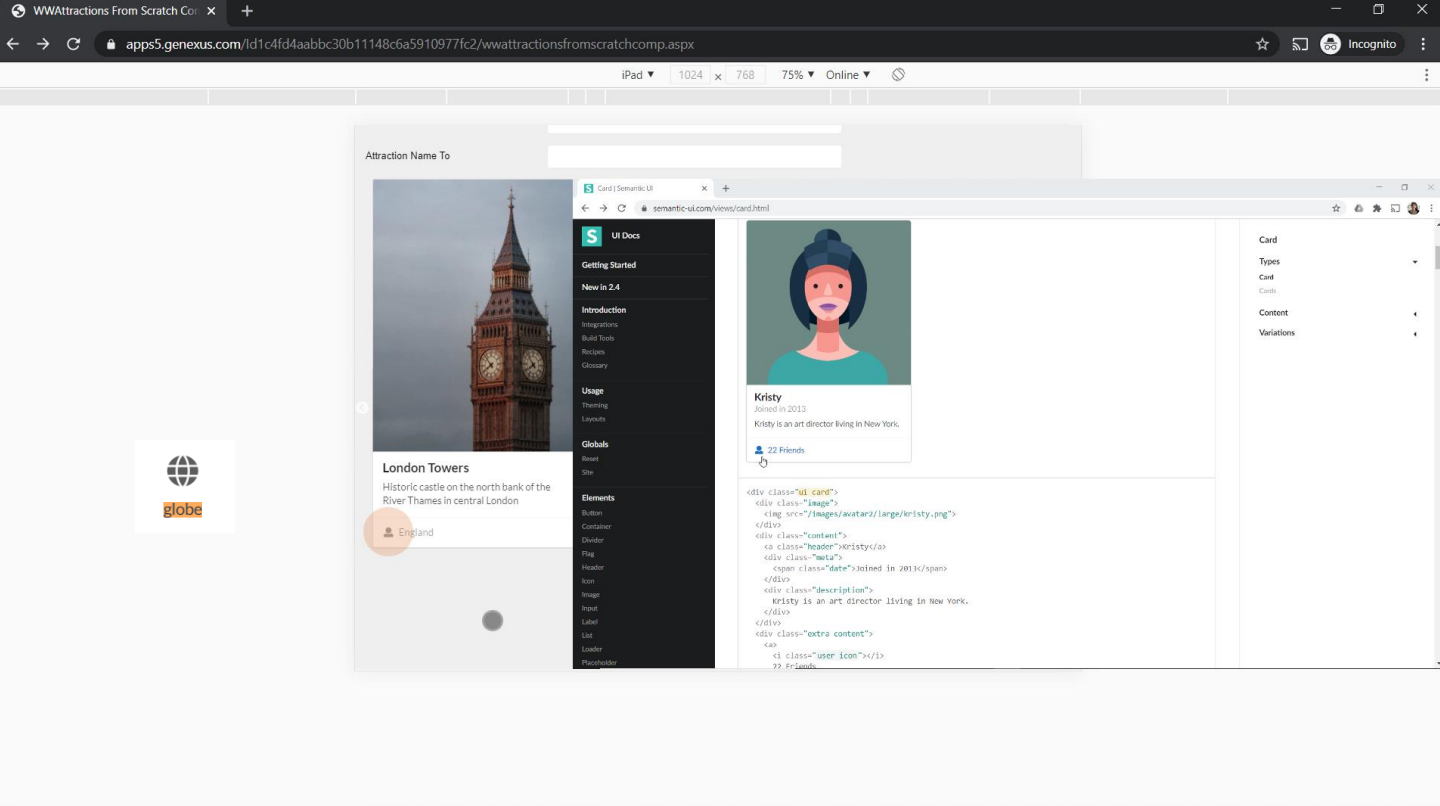
Por último, vamos modificar aquí. Queremos que o componente seja carregado com o web component que acabamos de criar. Agora, executemos para testar.



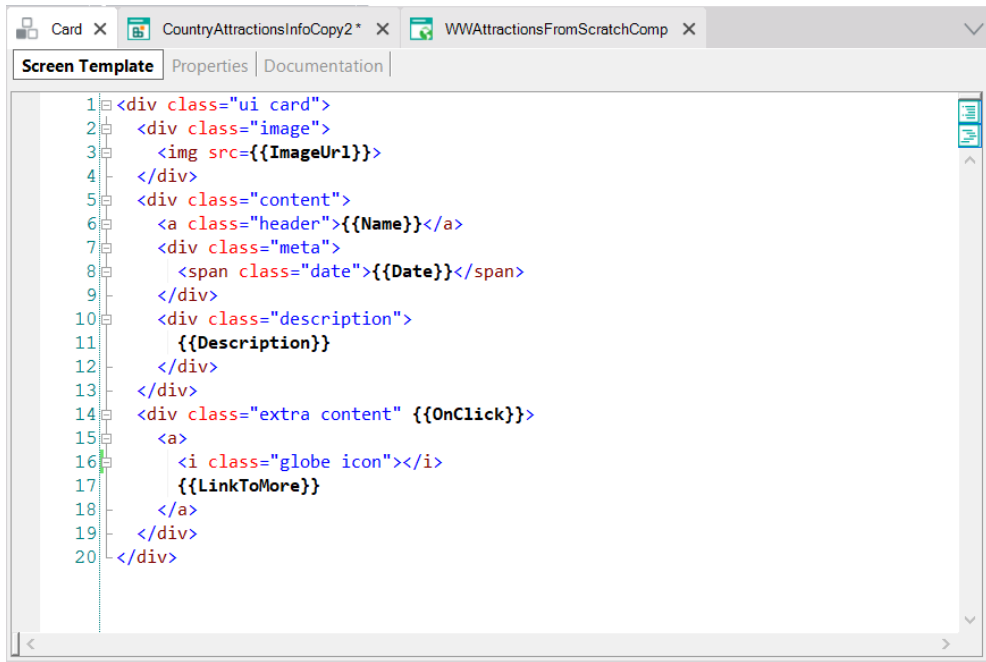
E vemos que estão saindo os cartões.

E aqui vemos o view do país.

Aqui estamos vendo 4 cartões por página porque tínhamos especificado que para esse tamanho, para o tamanho Large, no grid horizontal se mostrariam 4 colunas, enquanto que se o tamanho era Small se mostrariam 2, e 3 quando era Medium.



Vemos que o ícone que está sendo mostrado é um ícone de usuário que não é o que nos interessa. Nós queremos um ícone que reflita a realidade de um país. Ou seja, vamos ficar com este, com o ícone de Globo.



```
1 <div class="ui card">
2   <div class="image">
3     
6     <a class="header">{{Name}}</a>
7     <div class="meta">
8       <span class="date">{{Date}}</span>
9     </div>
10    <div class="description">
11      {{Description}}
12    </div>
13  </div>
14  <div class="extra content" {{OnClick}}>
15    <a>
16      <i class="globe icon"></i>
17      {{LinkToMore}}
18    </a>
19  </div>
20 </div>
```

Para isso, a única coisa que temos que fazer é trocar aqui a classe por Globe icon, porque está especificada dessa maneira em Semantic UI.



Recents WWAttractions From...

Country Id (None) v

Attraction Name From

Attraction Name To

Christ the Redemmer
An Art Deco statue of Jesus Christ in Rio de Janeiro

Cinque Terre
A string of centuries-old seaside villages on the rugged Italian Riviera coastline

Italy

Eiffel Tower
Wrought-iron lattice tower on the Champ

Forbidden city
The former Chinese imperial palace and state residence of the Emperor of China from the Ming dynasty

China

Testemos e vemos como o ícone está aparecendo.

DESIGN SYSTEMS

- Web Master Panel object
- Themes
- Stencil object
- User Control object**
 - Related topics to work with it
 - Properties
 - Events
 - Scripts definition and usage
 - Advanced usages
 - KB Sample
 - Patterns
 - See Also

<User Control object

This documentation is valid for:

- [GeneXus 16 Help](#)
- [GeneXus 17 Help](#)

Defines a specific UI control (in addition to the predefined in the GeneXus toolbox).

Most important benefits

- It provides the possibility to define User Controls in a built-in way into the [Knowledge Base](#).
- Allows you to base its appearance on a style provided by a designer, CSS Framework, etc.
- Generates server-side HTML code, which means power, since it is easier to scale a server than the clients.

User control object sections

A User Control object contains three tabs:

Com isto, temos uma amostra de como é simples utilizar um User Control externo. Na verdade, poderíamos criar os nossos próprios User Controls, escrevendo desta maneira o html e fornecendo a biblioteca css correspondente.

Para saber mais sobre User Controls, recomendamos que acesse nossa wiki.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications