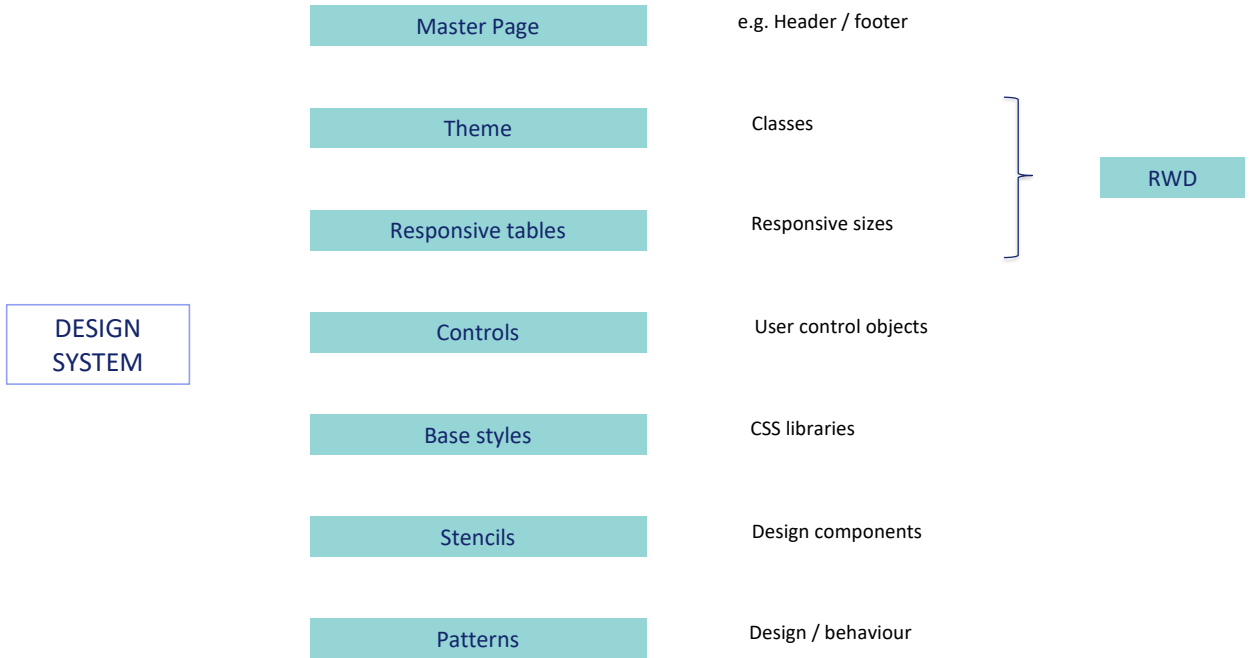


# Pantallas Web

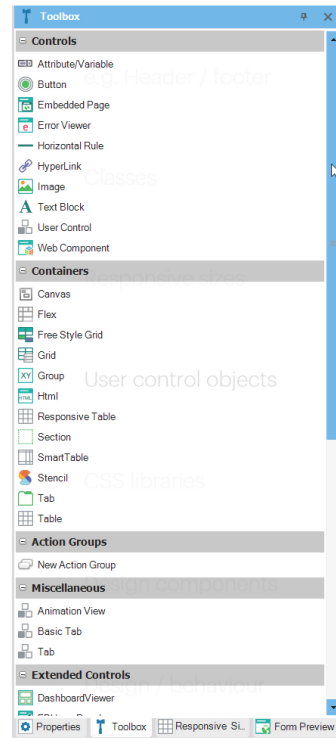
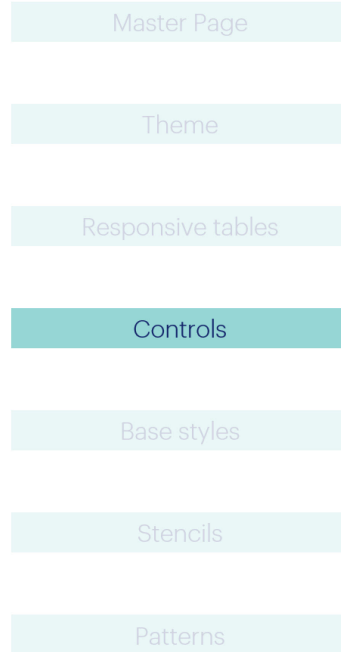
Objecto User Control

*GeneXus™*



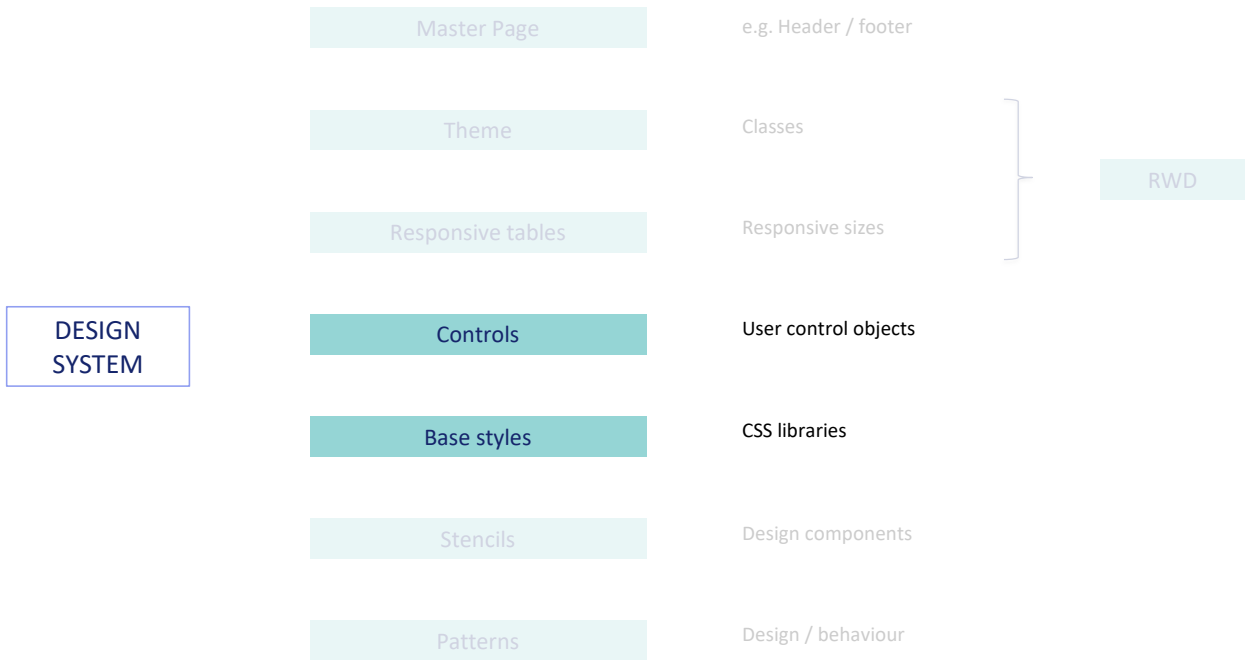
En el curso inicial de GeneXus habíamos abordado de manera superficial cada uno de los elementos de GeneXus que participaban en la implementación de un Design System para la aplicación.

DESIGN  
SYSTEM

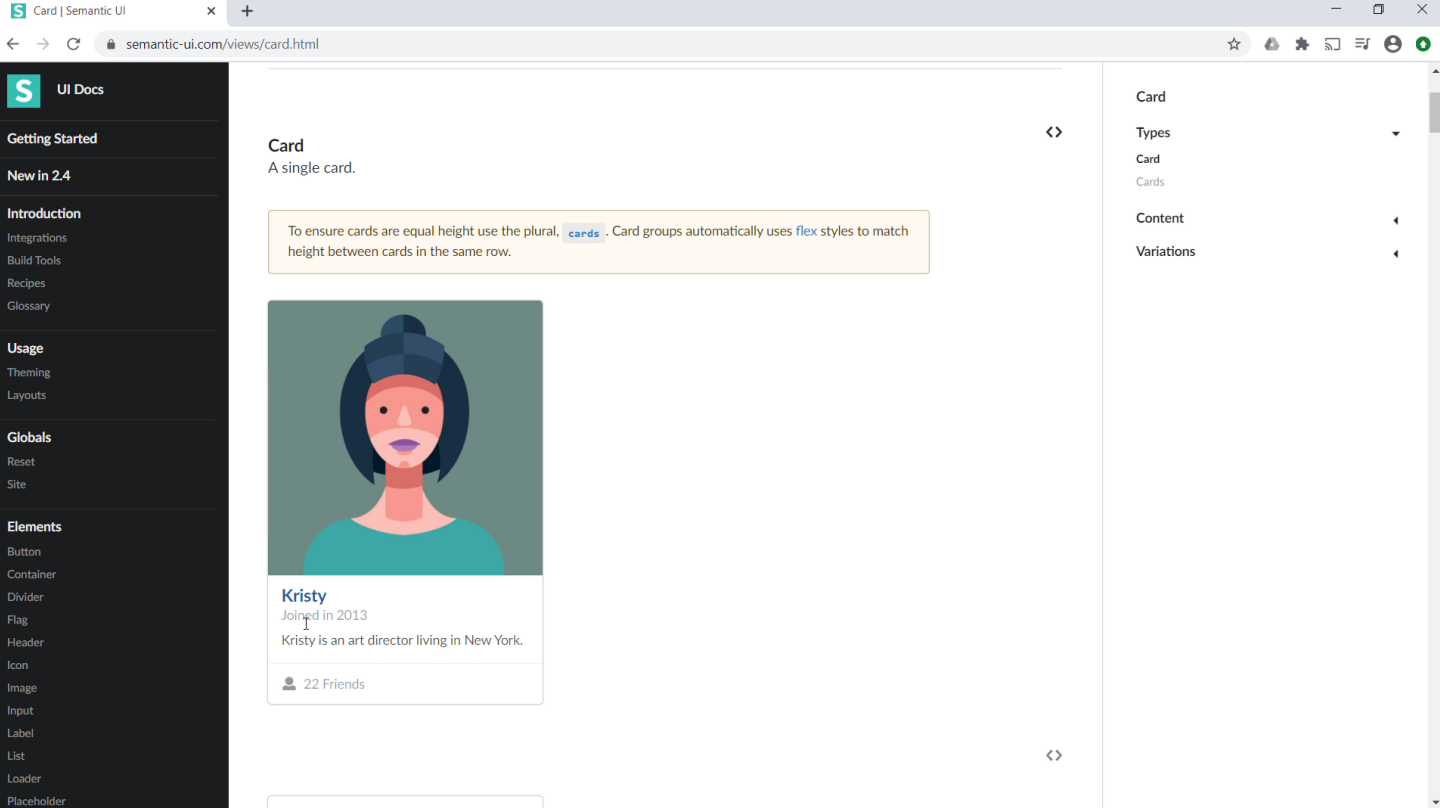


RWD

Entre ellos, habíamos dicho que podíamos no solo utilizar en nuestros forms los controles que vienen en la toolbox de GeneXus en forma predeterminada...



...sino que además podíamos definir controles de usuario, que pueden ser copiados de plataformas que ofrecen esos controles junto con bibliotecas CSS (que son, como nuestros themes, las que especifican su estilo –el estilo de esos controles-).



Por ejemplo, esta, Semantic UI.

Podíamos querer utilizar un control Card como este, por ejemplo...



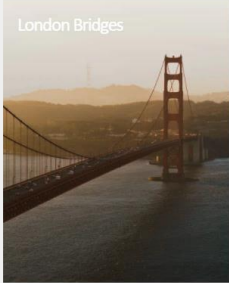



Recents WWAttractions From...

Country Id (None) v

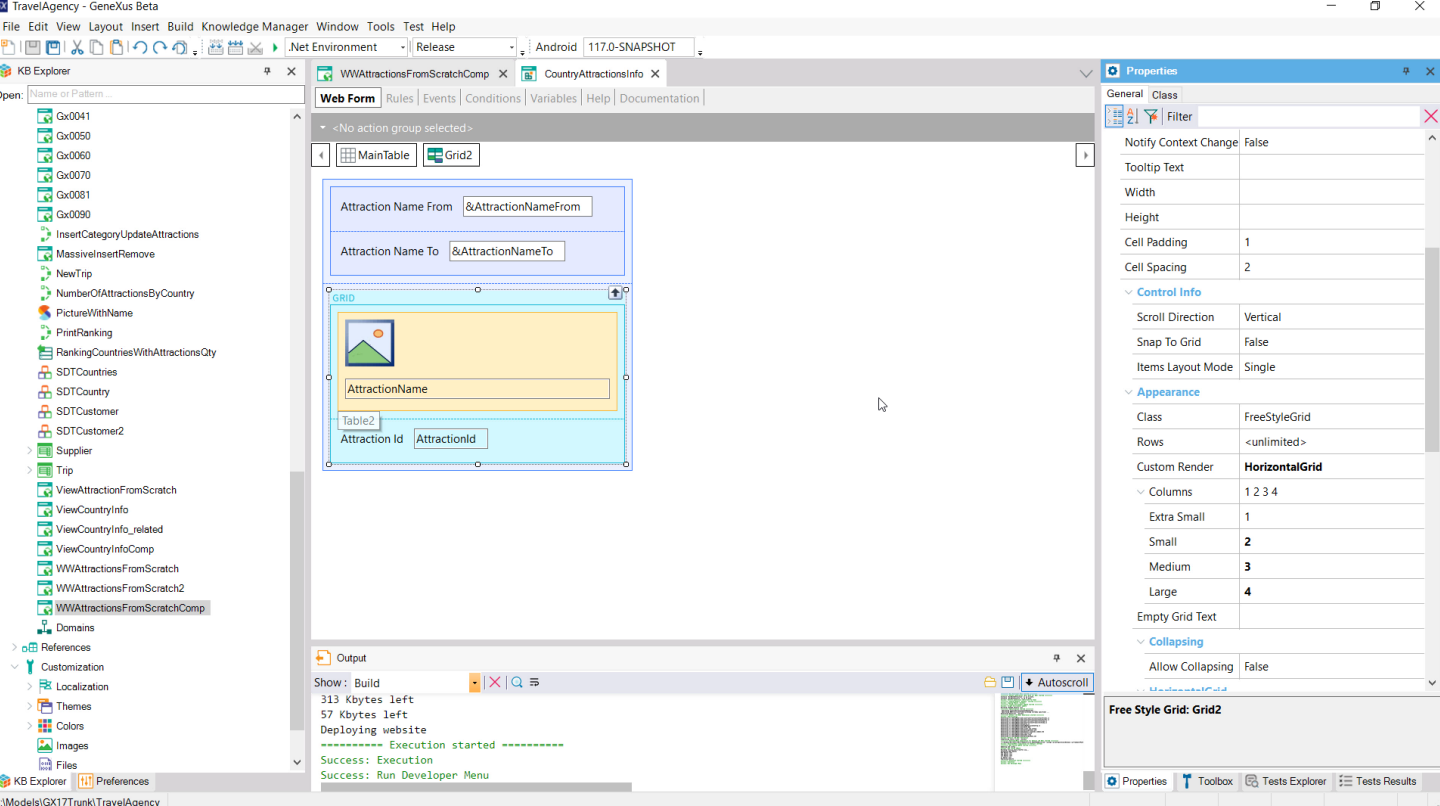
Attraction Name From

Attraction Name To

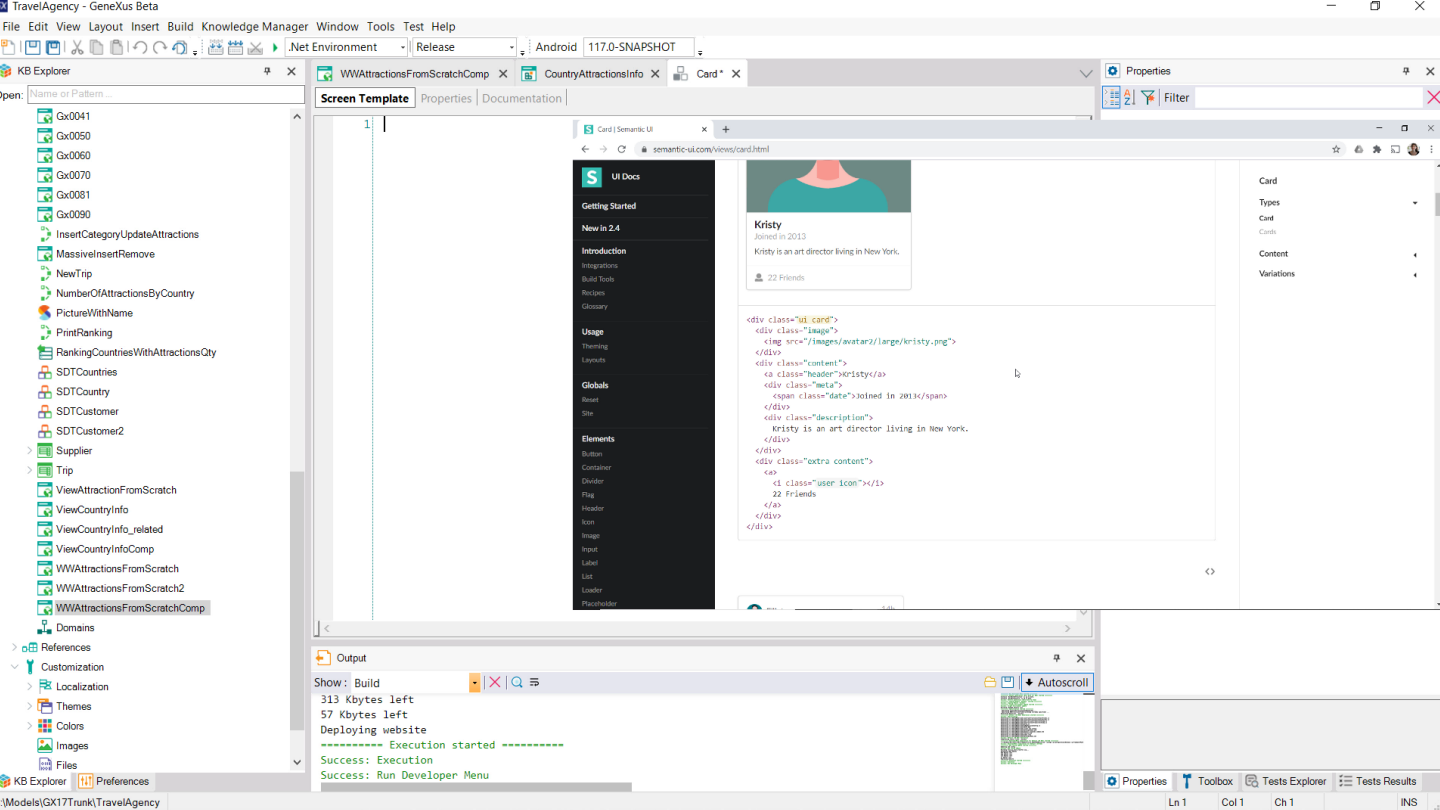


... ..

para mostrar las atracciones turísticas, en lugar de la tabla canvas que habíamos colocado en el grid horizontal, al que apenas habíamos empezado a colocarle diseño.



Allí vemos el grid y aquí vemos la tabla canvas.



Para ello alcanzará con crear un objeto User control (le podemos llamar igual)...

...copiar y pegar el html que nos ofrece Semantic UI...



The screenshot displays the GeneXus IDE interface. The main editor shows a 'Screen Template' with the following HTML code:

```
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>
```

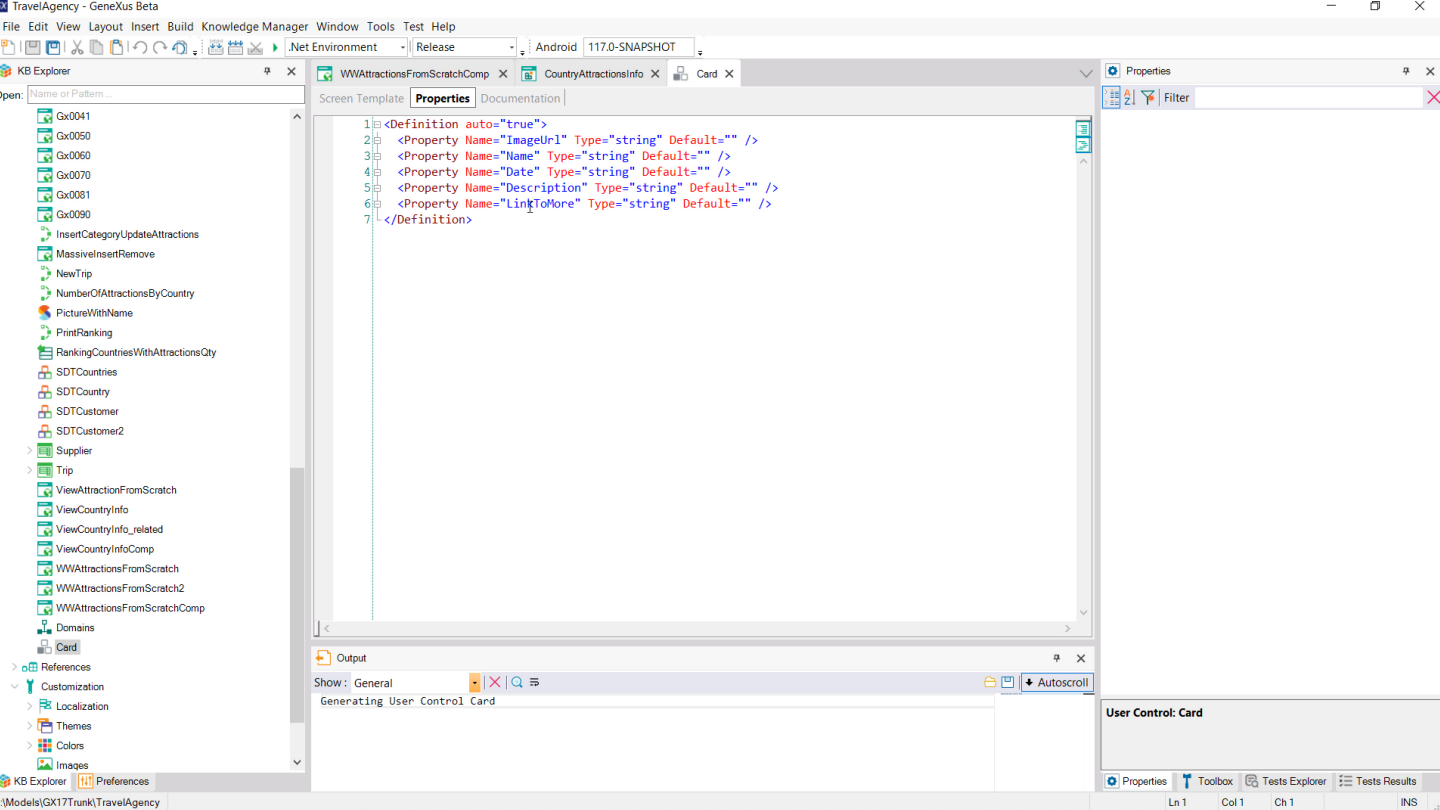
The Properties window on the right shows the configuration for the 'User Control: Card':

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

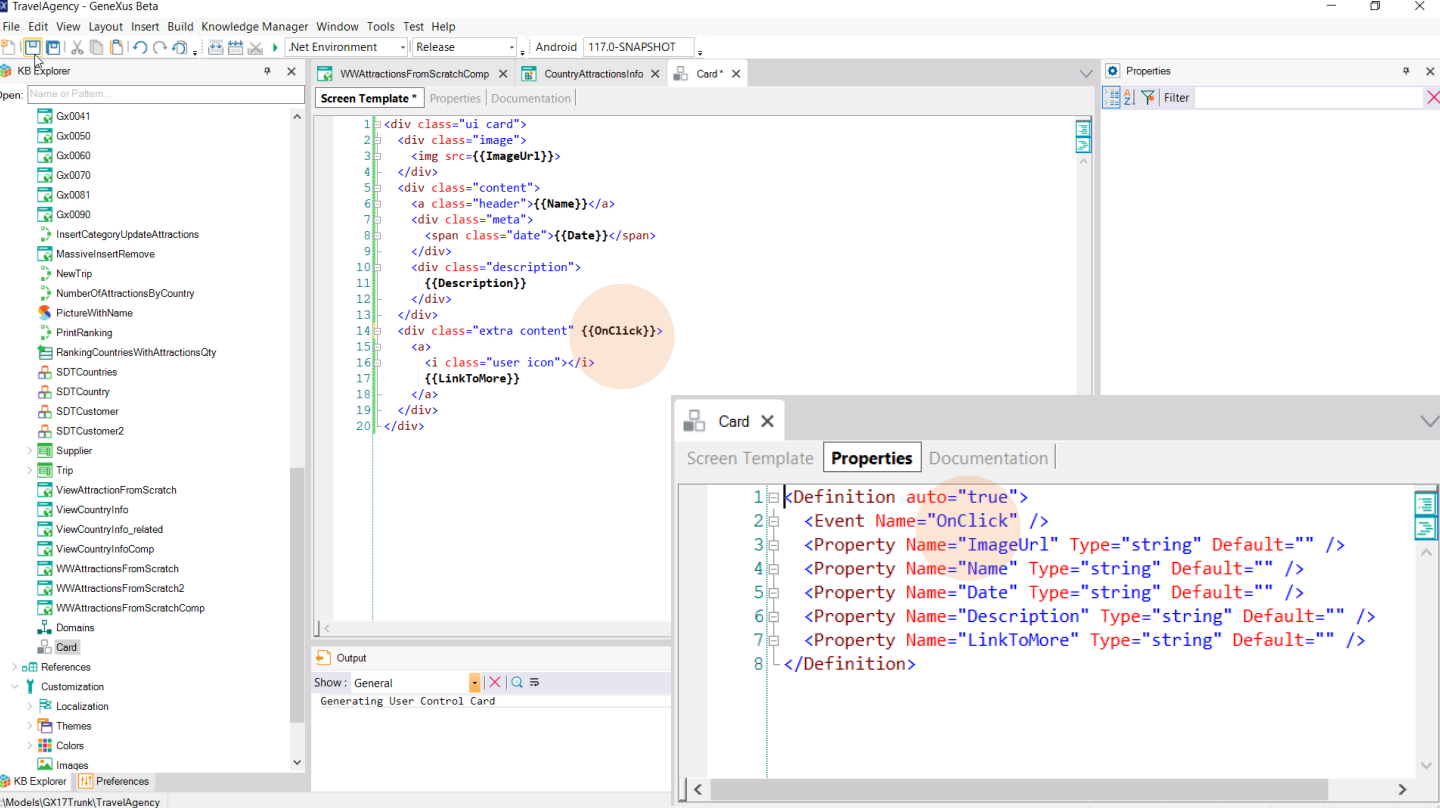
The Output window at the bottom shows the message: 'Generating User Control Card'.

...cambiar estos datos fijos por algo así como nombres de elementos de un SDT (para poder utilizarlos en forma dinámica, es decir, cargar dinámicamente ese control con datos que vamos a poder especificar por ejemplo de la base de datos).

Con esta sintaxis, que se conoce como sintaxis de mostacho, estamos entonces modificando los datos fijos por datos variables.



Si ahora vamos a ver las Properties, vemos que allí están apareciendo.



Vamos también a agregar un evento, que es el evento que se produce al hacer click sobre este componente del HTML.

Y vemos que está apareciendo aquí.

De esta manera vamos a poder programar el clic sobre ese elemento en ejecución.

Card | Semantic UI

semantic-ui.com/views/card.html

# Card

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

4 Themes [Download](#)

Star 48,432 [Tweet](#)

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

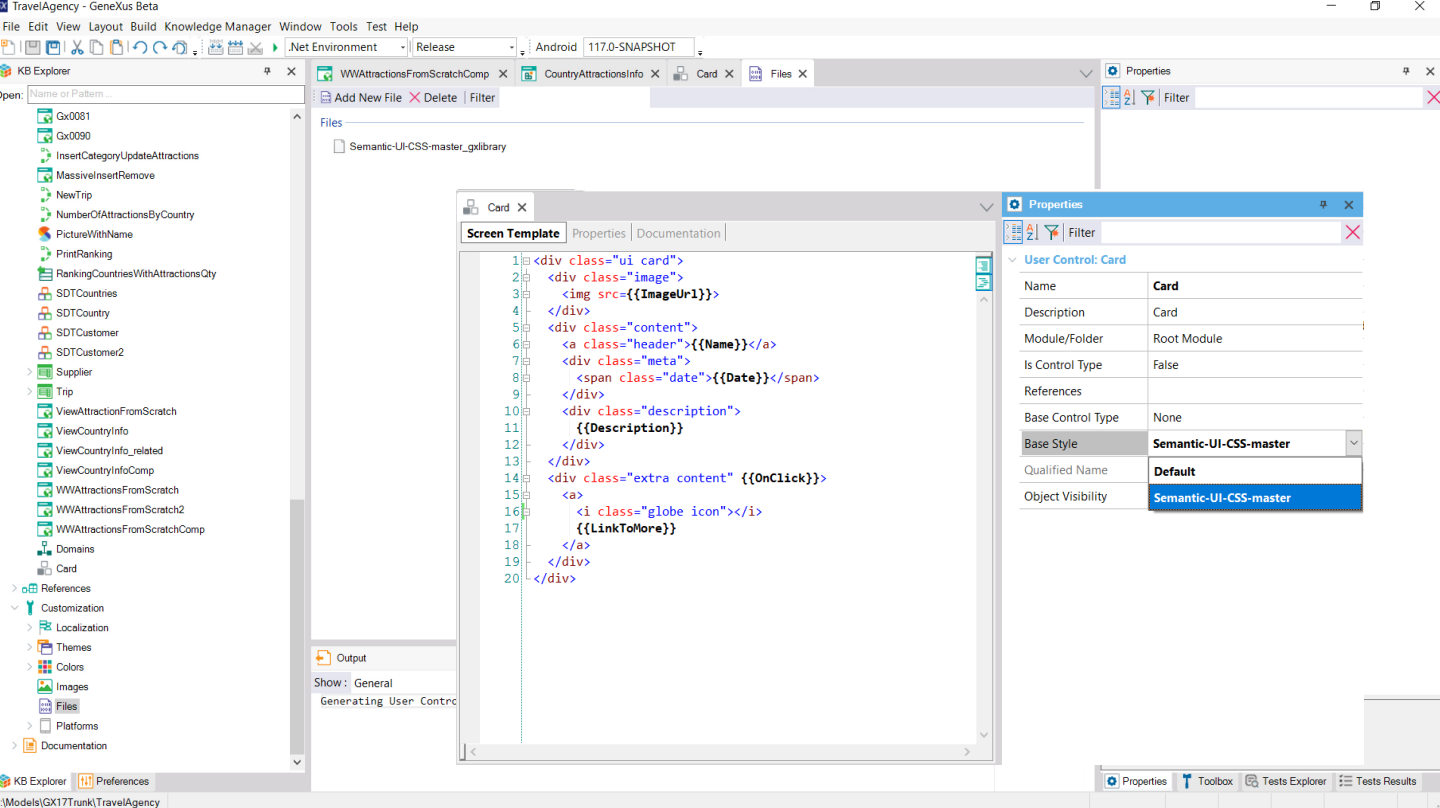
Cards

Content

Variations

Ahora lo que vamos a hacer es descargar el CSS, que es el que verdaderamente implementa el diseño de todos estos tags del HTML.

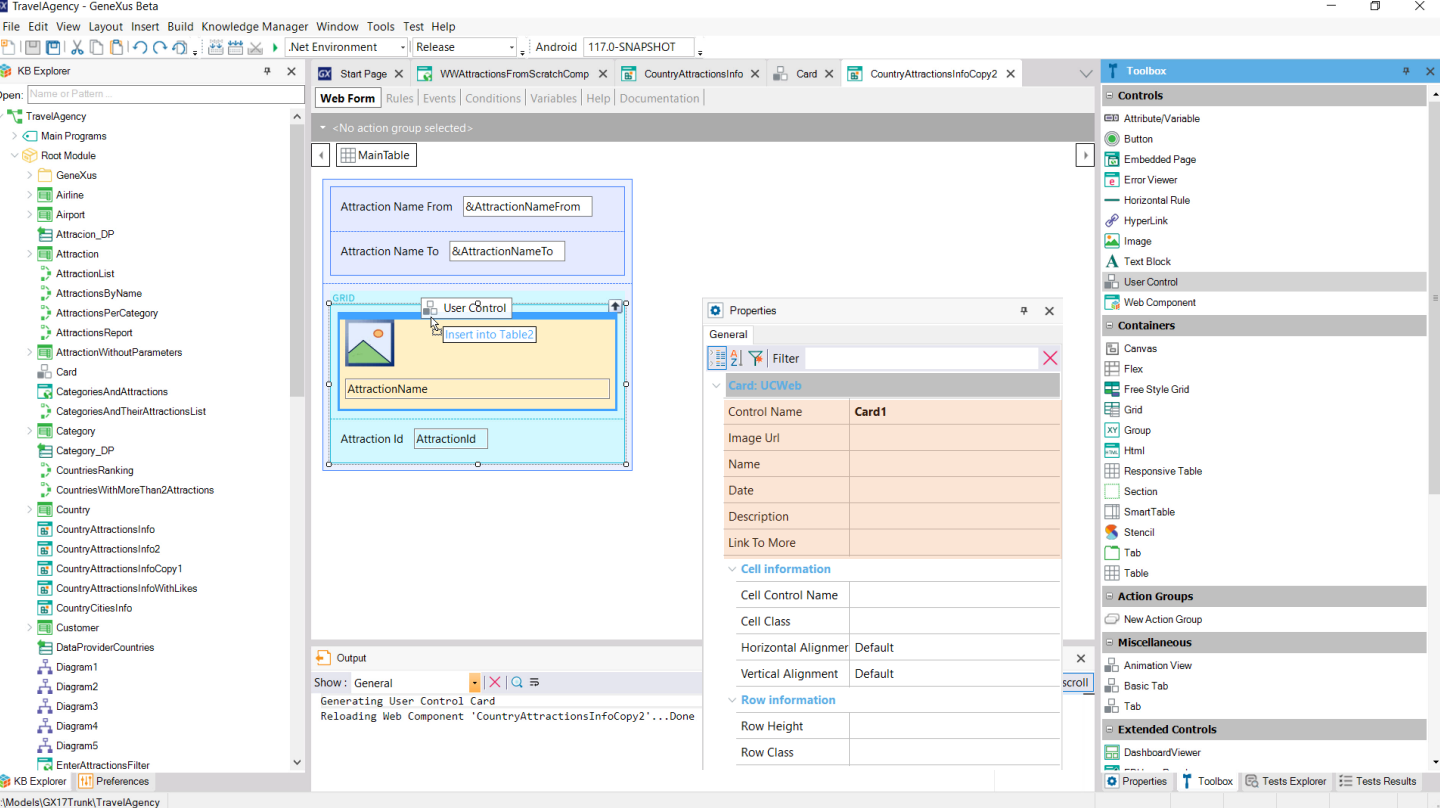
Aquí lo tenemos. Lo descargamos como un zip. Vamos a cambiarle la extensión por `gxl`, que es la extensión que es necesaria en GeneXus para interpretar este CSS.



Ahora lo que vamos a hacer es insertar este CSS como un archivo.

Y por último vamos a decirle al user control Card que su Base Style lo tome de allí.

Bien, tan simple como esto.



Ya tenemos el User Control para ser utilizado en nuestra KB. Ahora lo que vamos a hacer es grabar este componente con otro nombre y vamos a insertar dentro del grid este User Control, Card.

Si vemos sus propiedades vemos que están apareciendo todas aquellas que modificamos, y aparecen tanto en forma estática, como para asignarles un nombre, como –como vamos a ver- en forma dinámica.

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 a quitar todo lo que ya no nos interesa. Vamos a agregar el atributo CountryId, porque lo vamos a querer pasar como parámetro. AttractionId ya no nos interesa. Y vamos a colocar a CountryId como invisible.

Bien, entonces ahora lo que vamos a tener que hacer es cargar en el evento Load ese User Control con los valores de cada atracción. Entonces programaremos el evento Load del Grid. Y vemos que al escribir el nombre del control, Card1, ya nos aparecen todas las propiedades que vimos antes en el User Control.

A la propiedad Name le vamos a asignar el valor del atributo AttractionName. A la propiedad ImageUrl le tenemos que decir la url donde se encontrará la imagen que queremos que se cargue en esa Card. Y es la del atributo AttractionPhoto. Para la descripción vemos que hemos agregado a la transacción Attraction el atributo AttractionDescription, que contiene entonces la descripción de la atracción turística. Cargamos entonces ese elemento del User Control con el valor del atributo.

Y por último este va a ser el link. Queremos que se cargue el valor de CountryName, es decir, que diga el nombre del país, y lo que vamos a hacer después es establecer un link allí, es decir, que CountryName sea un link al panel que teníamos definido para listar la información del país, y sus atracciones y ciudades. Por lo tanto el evento que teníamos antes, que era el que se ejecutaba al hacer clic sobre AttractionName ya no lo necesitamos más, sino que ahora tenemos que programar el evento OnClick del user control. Y lo que haremos será invocar al panel que teníamos, pasándole como parámetro

el CountryId.



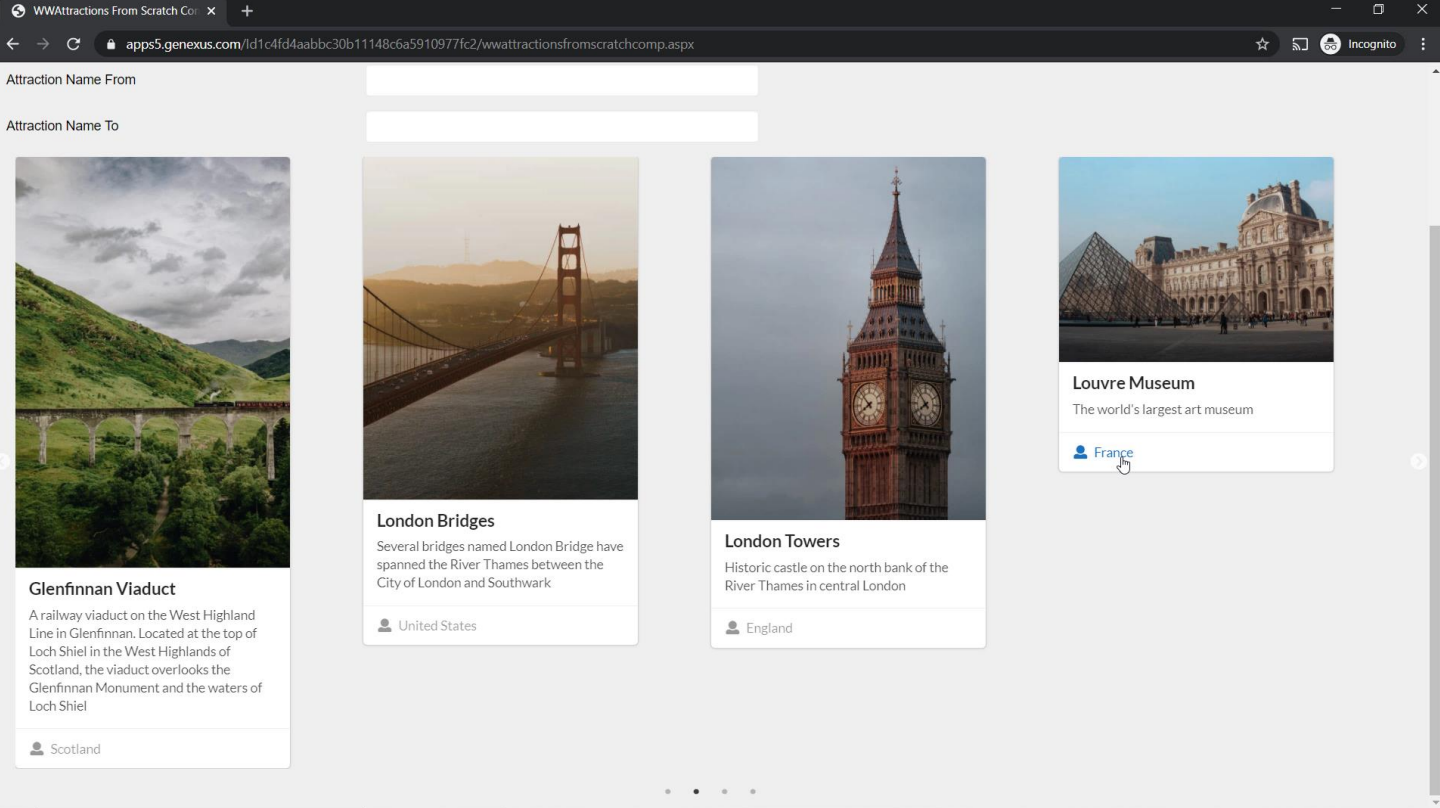
The screenshot displays the GeneXus IDE interface. On the left, the design view shows a web form with a 'Country Id' dropdown menu and a component placeholder '<Component: CountryAttractionsInfoCopy2>'. The properties window on the right shows the following details for 'Web Component: Component1':

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	

Below the design view, the following code is shown:

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

Por último vamos a modificar aquí. Queremos que el componente se cargue con el web component que acabamos de crear. Ahora ejecutemos para probar.

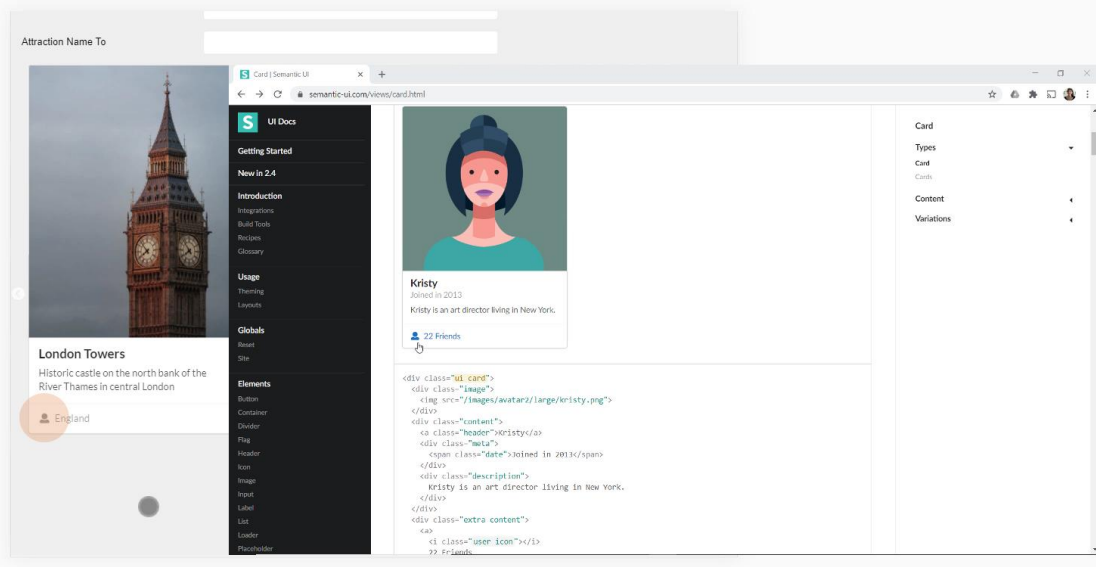


Y vemos que están saliendo las tarjetas.

Y aquí vemos el view del país.

Aquí estamos viendo 4 tarjetas por página porque habíamos especificado que para ese tamaño, para el tamaño Large, en el grid horizontal se mostrarán 4 columnas, mientras que si el tamaño era Small se mostrarán 2, y 3 cuando era Medium.

Attraction Name To



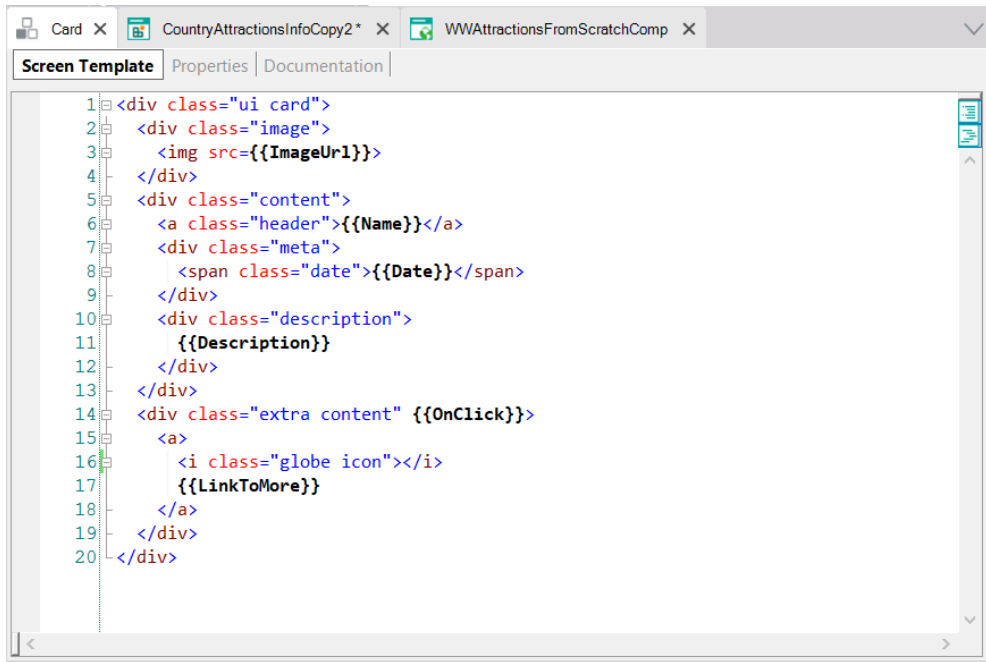
```
<div class="ui card">
  <div class="image">
    
  </div>
  <div class="content">
    <div class="header">kristy</div>
    <div class="meta">
      <span class="date">joined in 2013</span>
    </div>
    <div class="description">
      Kristy is an art director living in New York.
    </div>
  </div>
  <div class="extra content">
    <a
      <!-- user icon -->
    >> friends
  </div>
</div>
```

London Towers  
Historic castle on the north bank of the River Thames in central London

England

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

Vemos que el ícono que se está mostrando es un ícono de usuario que no es el que nos interesa. Nosotros queremos un ícono que refleje la realidad de un país. O sea que vamos a quedarnos con este, con el ícono de Globo.



```
1 <div class="ui card">
2   <div class="image">
3     <img src={{ImageUrl}}>
4   </div>
5   <div class="content">
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 ello lo único que tenemos que hacer es cambiar aquí la clase por Globe icon, porque está especificada de esa manera en Semantic UI.



Recents WWAttractions From...

Country Id

Attraction Name From

Attraction Name To



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

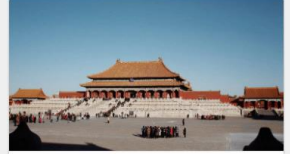


**Cinque Terre**  
A string of centuries-old seaside villages on the rugged Italian Riviera coastline

 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

Probemos y vemos cómo está saliendo el ícono.

User Control object | Category

wiki.genexus.com/commwiki/servlet/wiki?39356,Category%3AUser+Control+object

GeneXus Community Wiki MENU PAGE INFO PAGE TOOLS Login Register

User control object All

Recents Search Wiki Home

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:

Con esto tenemos una muestra de lo sencillo que es utilizar un User Control externo. De hecho podríamos crear nuestros propios User Controls, escribiendo de esta manera el html y brindando la biblioteca css que corresponda.

Para saber más de User Controls le recomendamos acceder a nuestro wiki.

# GeneXus™

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

[wiki.genexus.com](http://wiki.genexus.com)

[training.genexus.com/certifications](http://training.genexus.com/certifications)