

# Objecto User Control

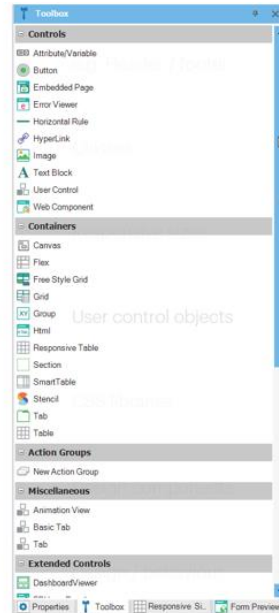
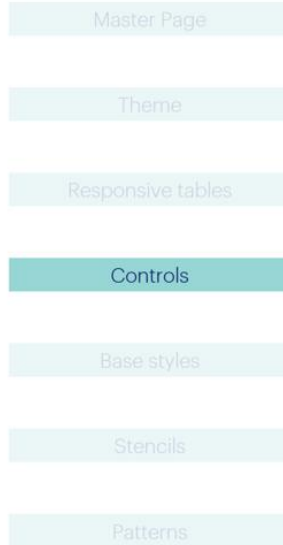
GeneXus™

DESIGN  
SYSTEM

Master Page	e.g. Header / footer
Theme	Classes
Responsive tables	Responsive sizes
Controls	User control objects
Base styles	CSS libraries
Stencils	Design components
Patterns	Design / behaviour

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.

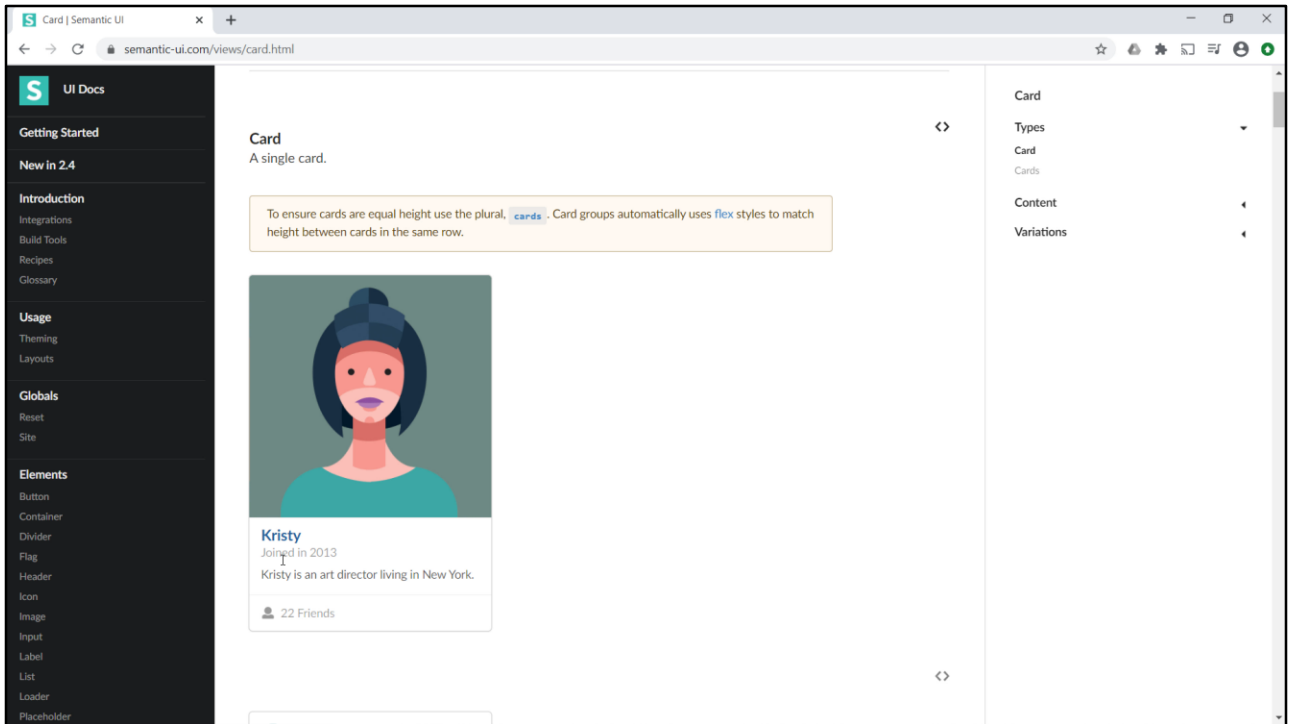
DESIGN  
SYSTEM



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

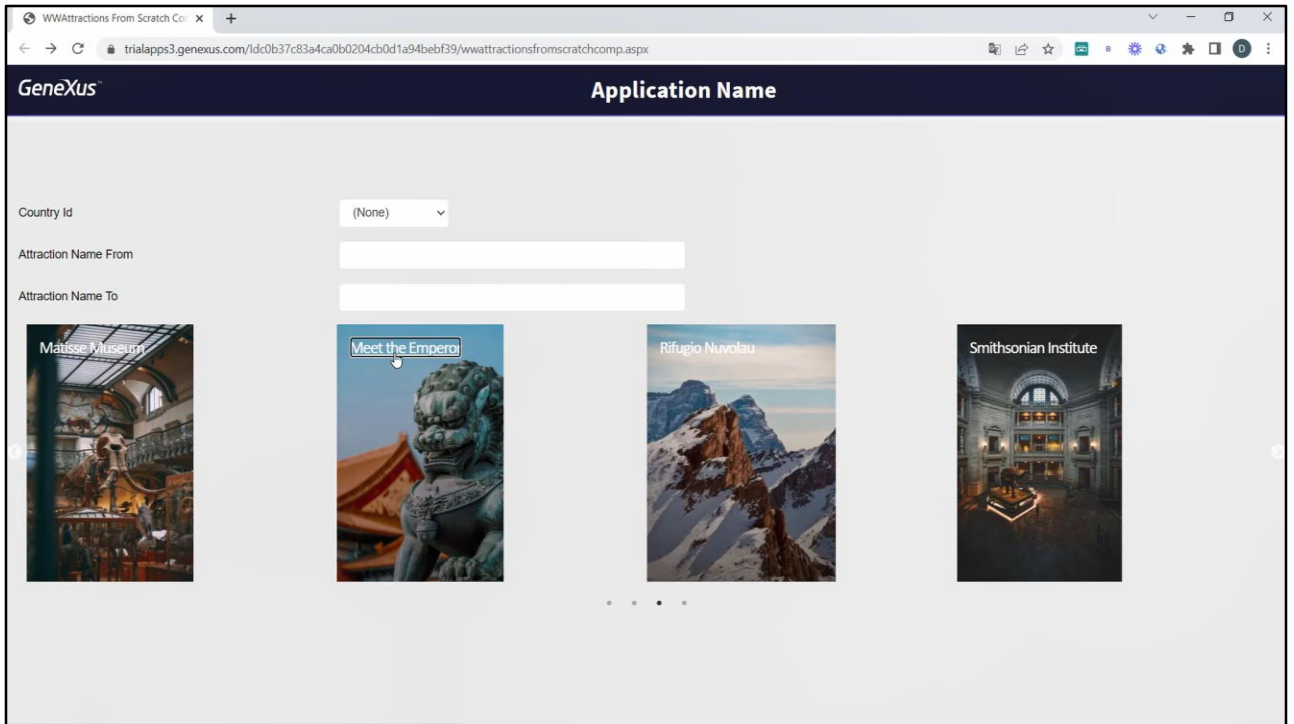
DESIGN SYSTEM	Master Page	e.g. Header / footer
	Theme	Classes
	Responsive tables	Responsive sizes
	Controls	User control objects
	Base styles	CSS libraries
	Stencils	Design components
	Patterns	Design / behaviour

...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-).

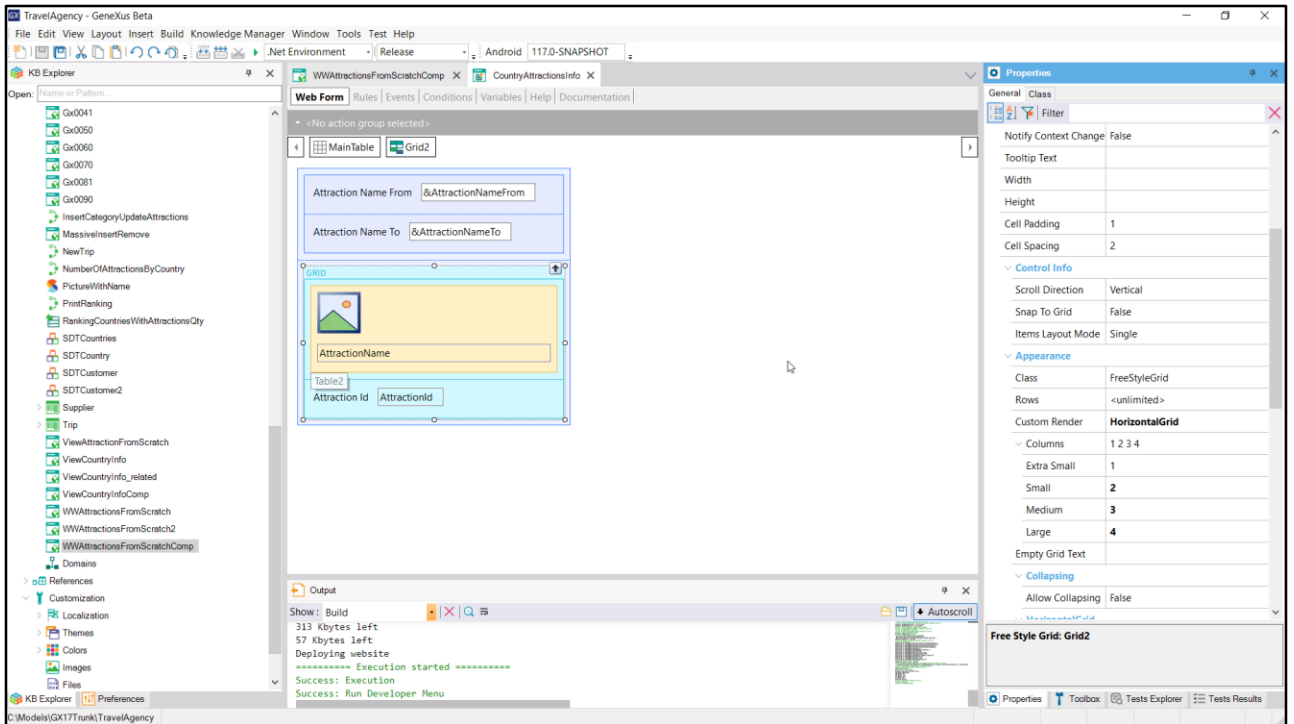


Por ejemplo, esta, Semantic UI.

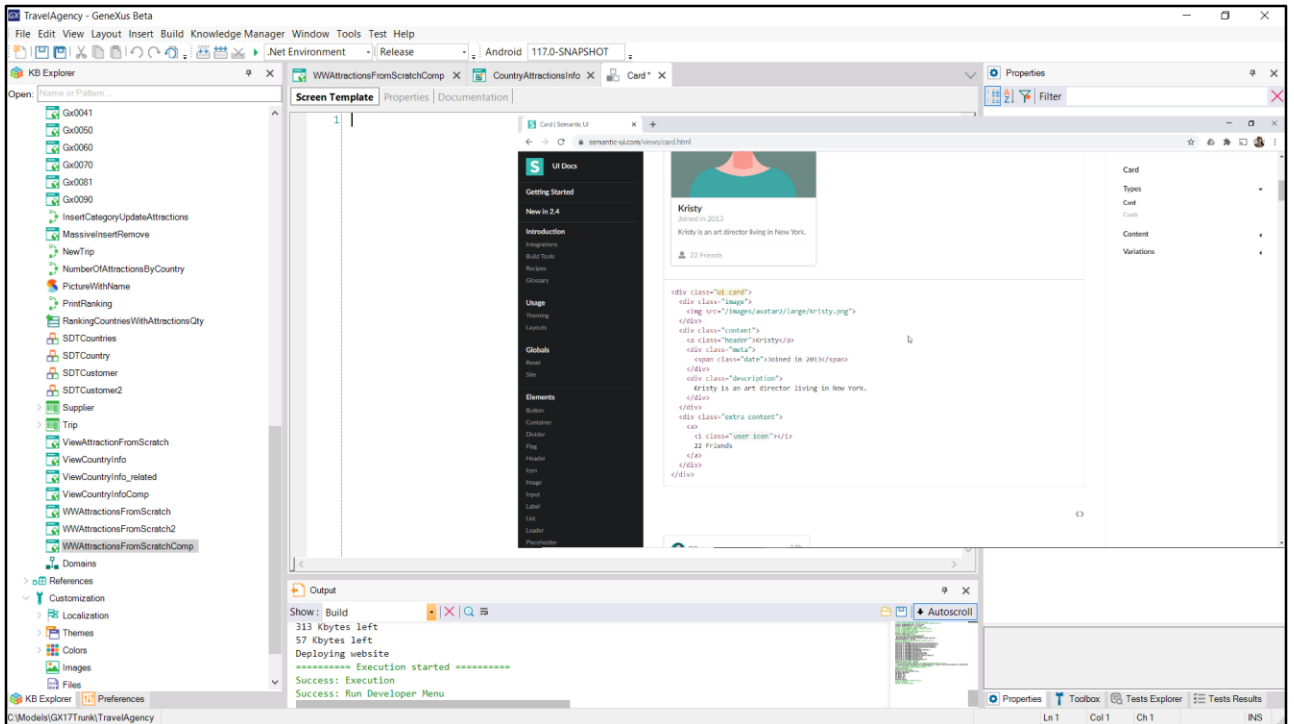
Podíamos querer utilizar un control Card como este, por ejemplo...



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.



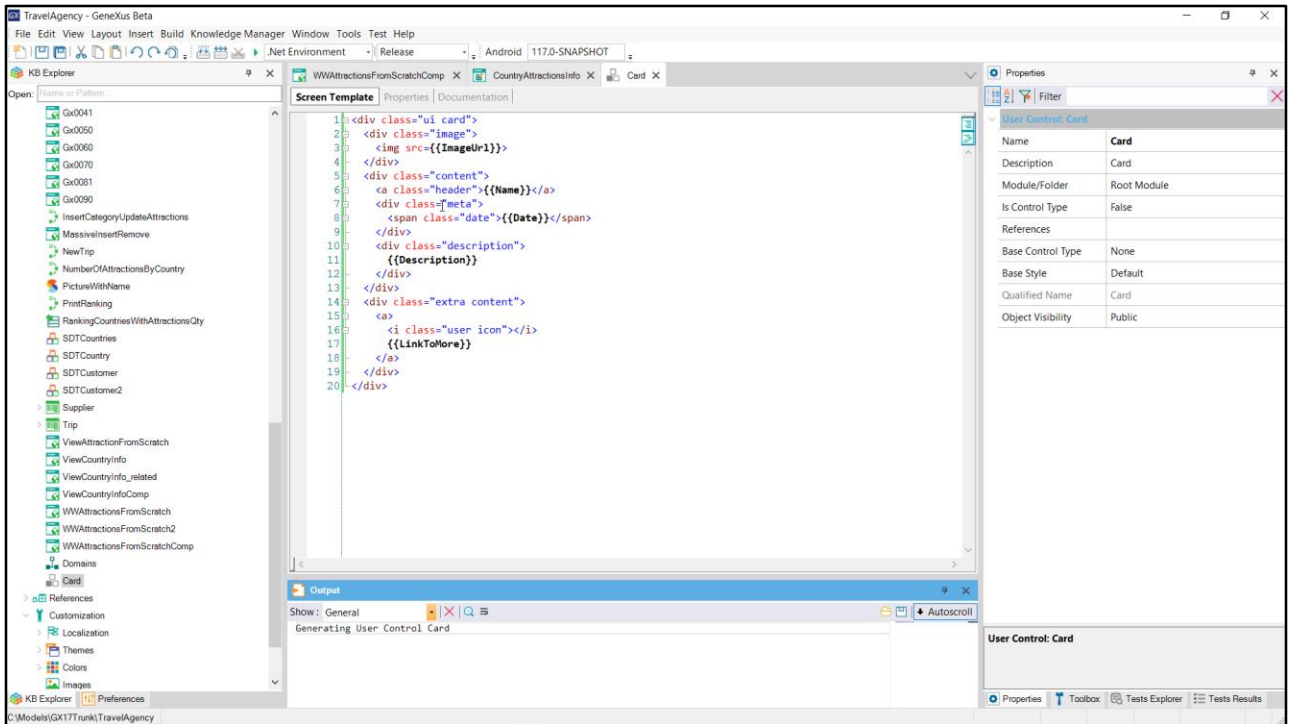
Allí vemos el grid y aquí vemos la tabla canvas.



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

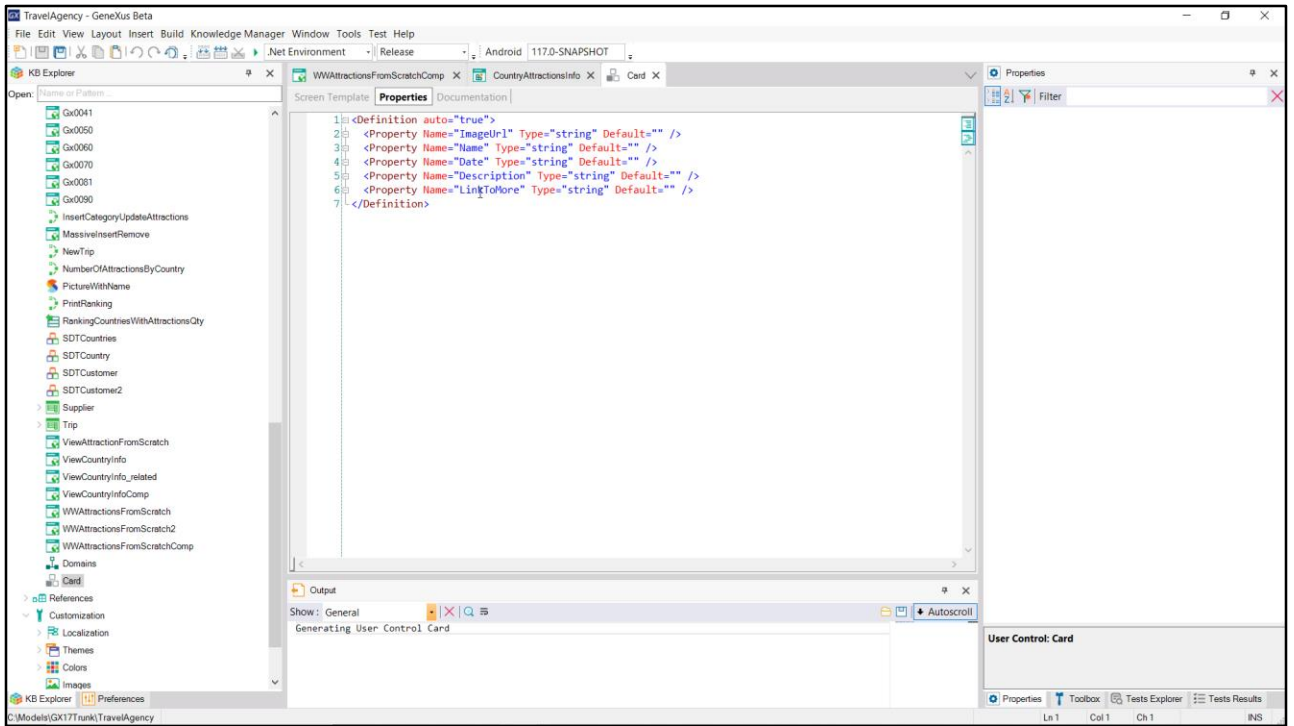
...copiar y pegar el html que nos ofrece Semantic UI...



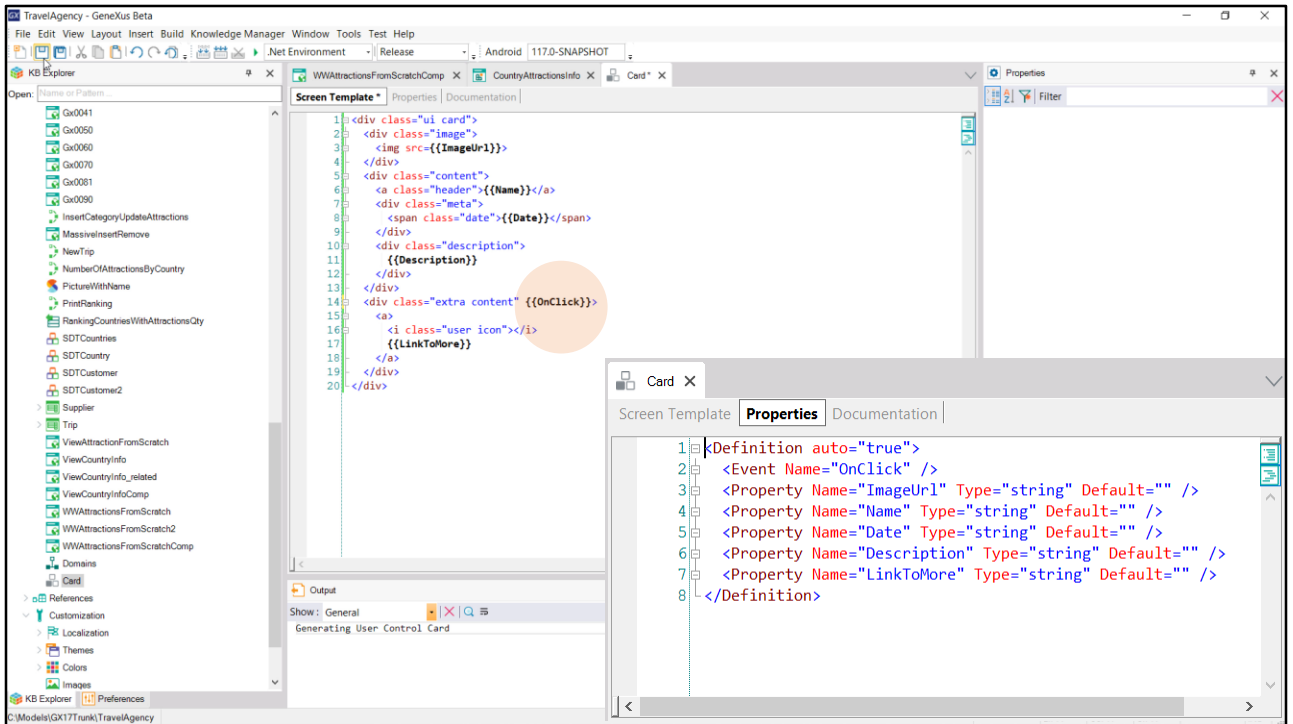


...cambiar estos datos fijos por algo así como nombres de elementos de un SDT (para poder utilizarlos en forma dinámica, es decir, cargar dinámicamente ese control con datos que vamos a poder especificar por ejemplo de la base de datos).

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.

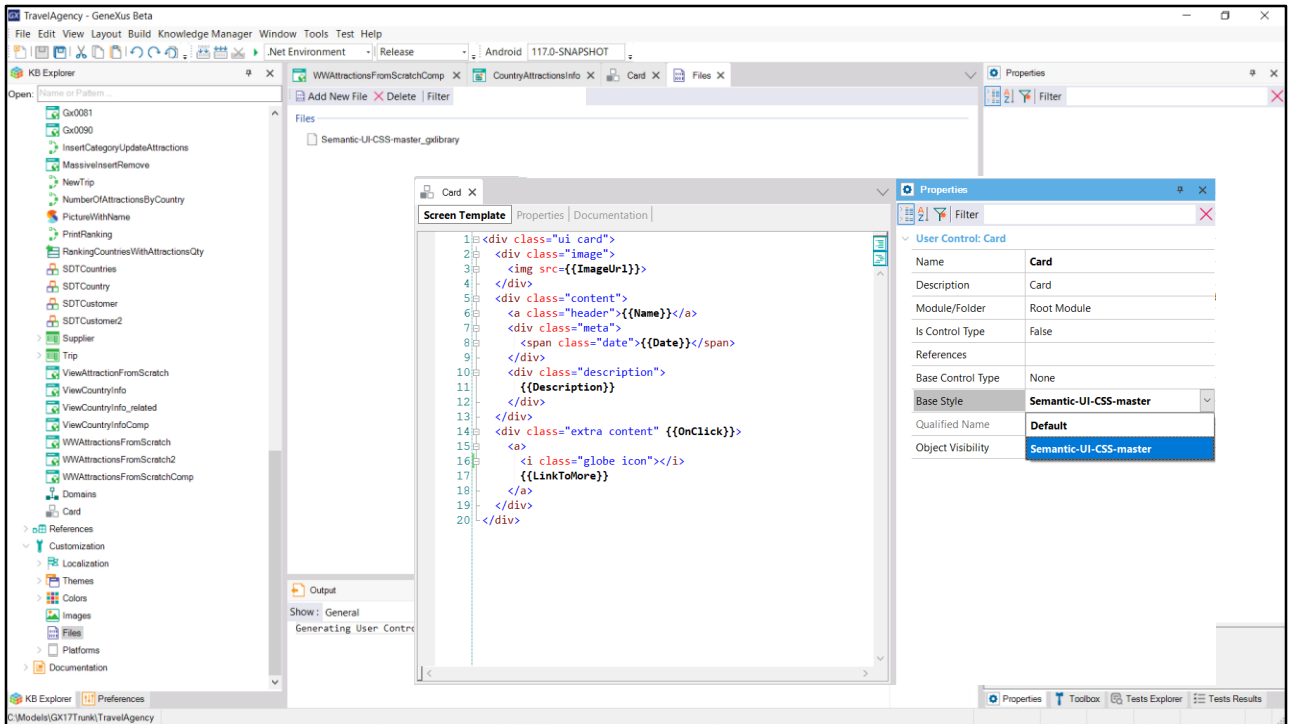
Y vemos que está apareciendo aquí.

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

The screenshot shows the Semantic UI website's documentation for the 'Card' component. The page is titled 'Card' and describes it as a component that displays site content in a manner similar to a playing card. It features a 'Download' button and a 'Package Managers' section with instructions for NPM and Git. An example card is shown at the bottom, and a right sidebar lists 'Card' types and 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.

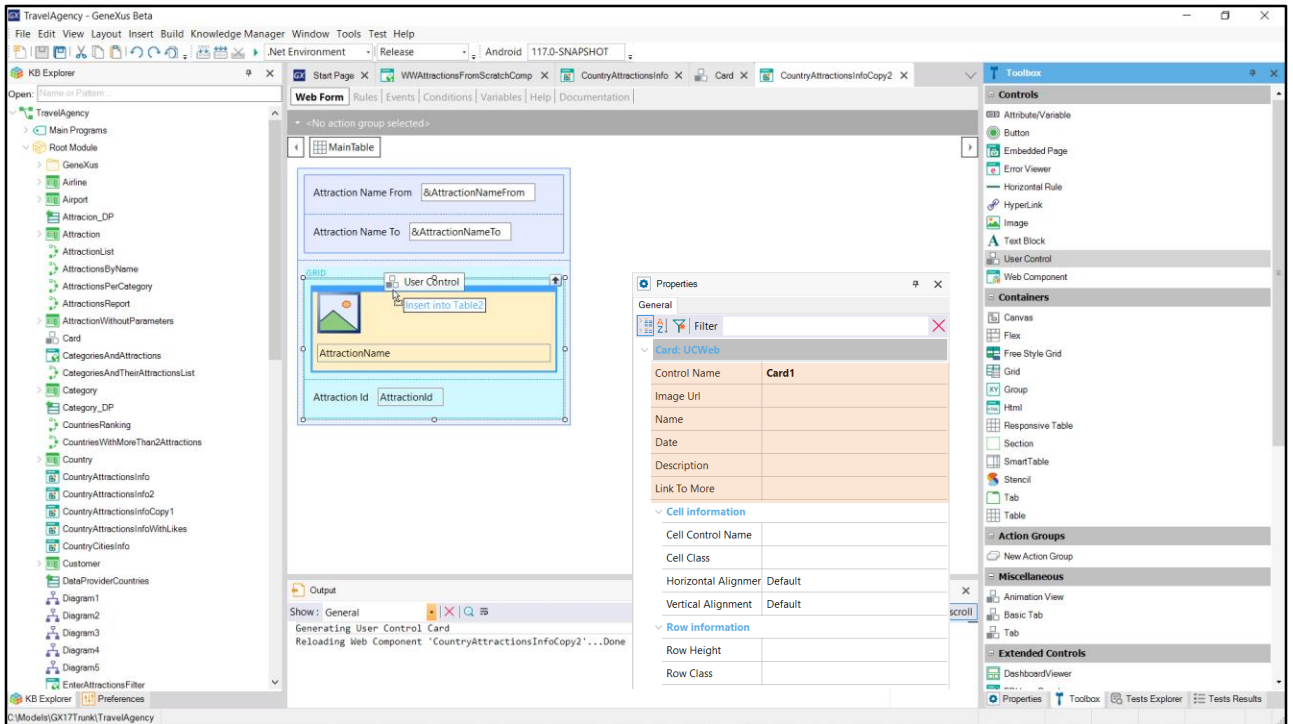
Aquí lo tenemos. Lo descargamos como un zip. Vamos a cambiarle la extensión por gxlibrary, que es la extensión que es necesaria en GeneXus para interpretar este CSS.



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 valor, como -como vamos a ver- en forma dinámica.

Attraction Name From	<input type="text" value="&amp;AttractionNameFrom"/>
Attraction Name To	<input type="text" value="&amp;AttractionNameTo"/>
GRID	
<Card: Card1>	
Country Id	<input type="text" value="CountryId"/>

```

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.



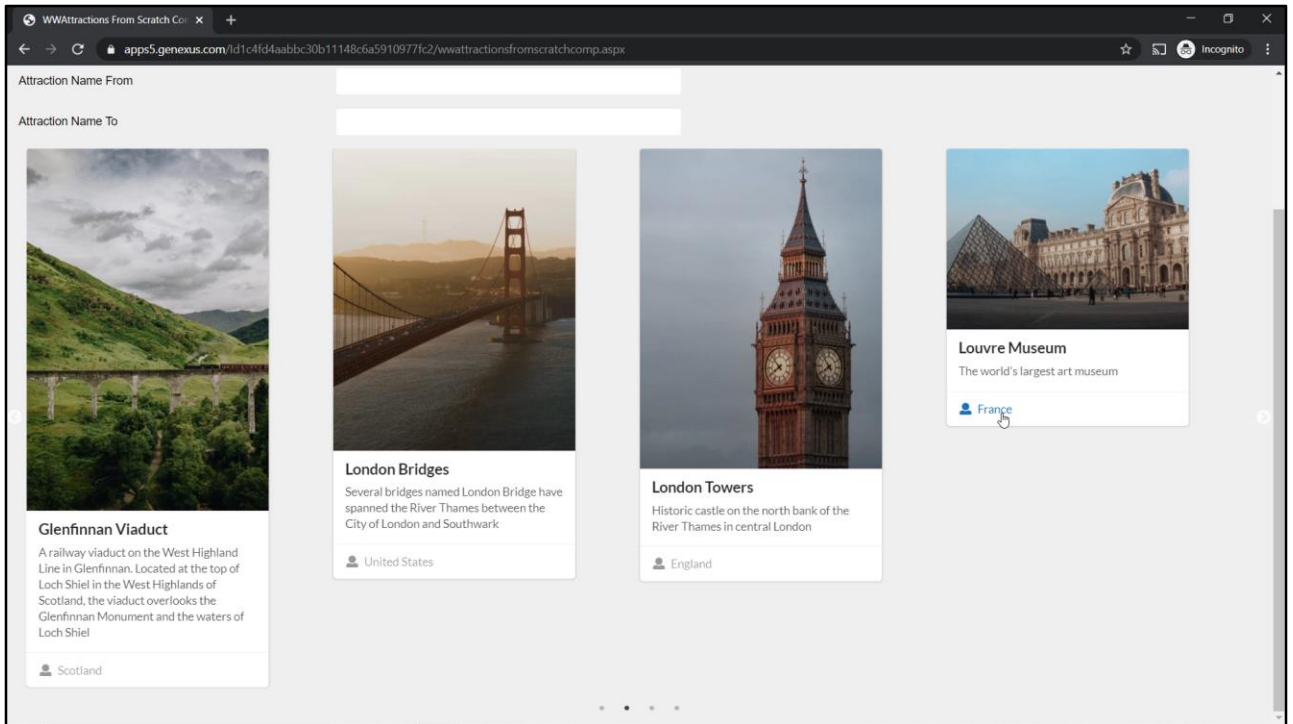
The screenshot displays the GeneXus IDE interface. The main workspace shows a web form with a table containing a dropdown menu for 'Country Id' with the value '&CountryId'. Below the table, a component is being added, represented by the text '<Component: CountryAttractionsInfoCopy2>'. The Properties window on the right is open to the 'General' tab, showing the following details for the selected component:

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	

At the bottom of the workspace, the following event code is visible:

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

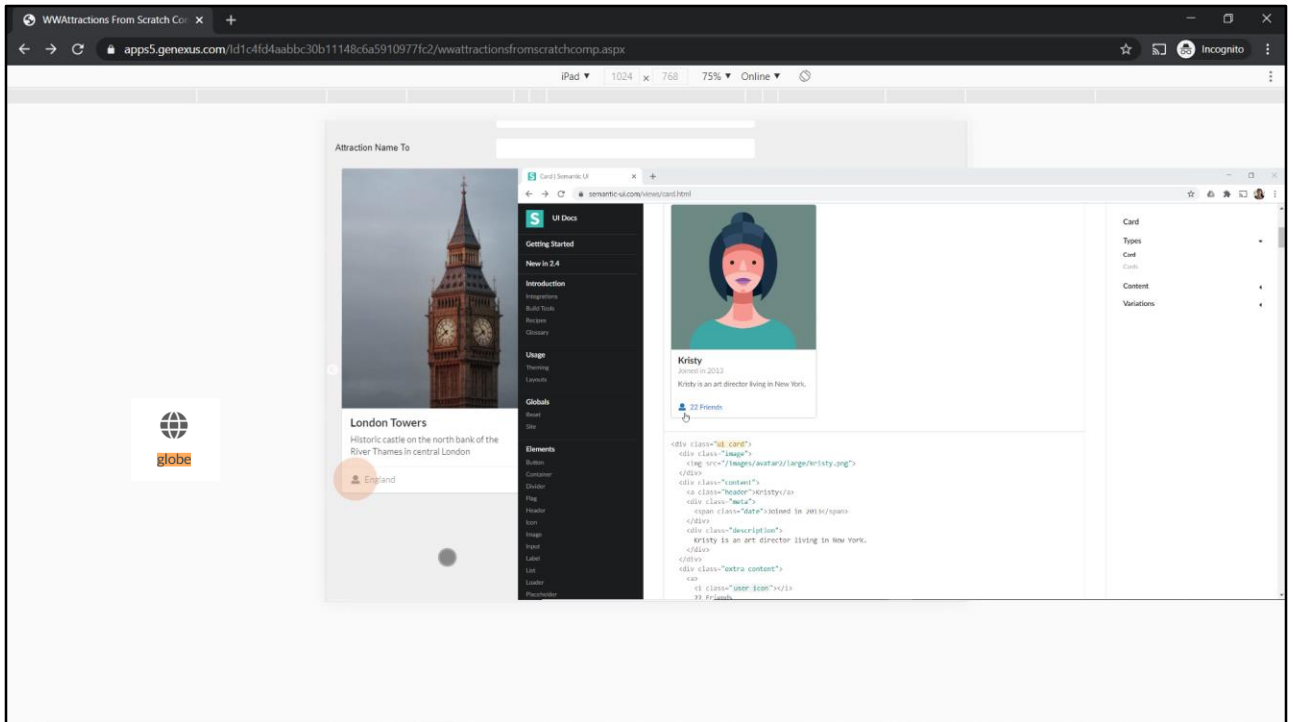
Por último vamos a modificar aquí. Queremos que el componente se cargue con el web component que acabamos de crear. Ahora ejecutemos para probar.



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.



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     
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.

WWAttractions From Scratch Co. x +


trialapps3.genexus.com/ldc0b37c83a4ca0b0204cb0d1a94bebf39/wwattractionsfromscratchcomp.aspx

GeneXus Application Name

Country Id (None) v


Attraction Name From

Attraction Name To




**Louvre Museum**  
The world's largest art museum

France




**The Great Wall**  
Series of fortification systems built across the historical northern borders of China

China




**Eiffel Tower**  
Wrought-iron lattice tower on the Champ de Mars in Paris.

France



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

Brazil



**Smithsonian Institute**  
Group of museums and research administered by the government of the United States

Probemos y vemos cómo está saliendo el ícono.

The screenshot shows a web browser window displaying the GeneXus Community Wiki page for 'User Control object'. The page has a dark red header with the site name 'GeneXus Community Wiki' and navigation links for 'MENU', 'PAGE INFO', and 'PAGE TOOLS'. There are also links for 'Login' and 'Register'. A search bar contains the text 'User control object'. Below the header, there are social media icons for Facebook, Google+, Twitter, and LinkedIn. The main content area is divided into two columns. The left column is a sidebar with a 'DESIGN SYSTEMS' section containing a tree view of topics: 'Web Master Panel object', 'Themes', 'Stencil object', 'User Control object' (highlighted in red), 'Related topics to work with it' (with sub-items 'Properties', 'Events', 'Scripts definition and usage'), 'Advanced usages' (with sub-item 'KB Sample'), 'Patterns', and 'See Also'. The right column features the main article title '<User Control object' with a blue checkmark icon. Below the title, it states 'This documentation is valid for:' followed by links to 'GeneXus 16 Help' and 'GeneXus 17 Help'. A definition follows: 'Defines a specific UI control (in addition to the predefined in the GeneXus toolbox)'. The next section is 'Most important benefits', which includes a bulleted list: '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.', and 'Generates server-side HTML code, which means power, since it is easier to scale a server than the clients.'. The final section is 'User control object sections', which begins with the text '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*<sup>™</sup>

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