

# Ejercicios prácticos

Actualización de GeneXus 16 a

# **GeneXus<sup>TM</sup> 17**

Marzo 2022

*© GeneXus. Todos los derechos reservados. Este documento no puede ser reproducido por ningún medio sin la autorización expresa de GeneXus<sup>TM</sup>. La información contenida en este documento es para uso personal únicamente.*

**Marca registrada:**

*GeneXus es una marca comercial o una marca registrada de GeneXus S.A. Todas las demás marcas comerciales mencionadas en este documento son propiedad de sus respectivos dueños.*

CONTNIDO	
OBJETIVO .....	3
KB DE PARTIDA .....	3
WEB PANELS .....	4
Múltiples Grids [OPCIONAL].....	4
Tipos de Web Panels.....	4
DESIGN SYSTEMS.....	7
Paginado en Grids .....	11
BASE STYLE Y USER CONTROL [OPCIONAL] .....	13
PANELS .....	18
STARTUP OBJECT .....	21
Menu de opciones y MasterPage.....	21

## OBJETIVO

En este práctico trataremos sobre el diseño de Web Panels con múltiples grids, aplicaremos el concepto de Design System aplicando y reusando clases, y diseñaremos un objeto Panel generado con Angular (Front end). Finalmente, crearemos el objeto Startup de la aplicación con un menú de opciones.

La letra de este práctico se ha actualizado de acuerdo al upgrade 8 de GeneXus 17. Si lo realiza con una versión posterior, considere que puede haber cambios entre lo que aquí se le muestra y lo que usted ve.

Utilizaremos una KB que corresponde a una aplicación simplificada de una app que gestiona Parques de Diversiones en diferentes países y sus empleados.

## KB DE PARTIDA

Se partirá de la versión de la KB que puede crear desde <http://samples.genexusserver.com/v17>, (KB de nombre UpgGX16\_GX17) eligiendo la versión de nombre AmusementParks\_UpgGX16\_GX17.

Observe os objetos presentes en la KB

Transacciones:

- AmusementPark
- Country
- Employee

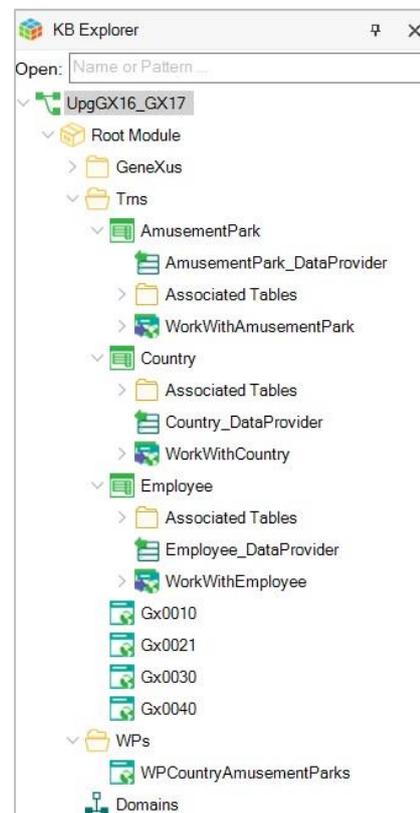
Data Providers asociados a las transacciones:

- AmusementPark\_DataProvider
- Country\_DataProvider
- Employee\_DataProvider

**Recuerde** que a partir de la versión 15 de GeneXus, las transacciones ofrecen la propiedad Data Provider que permite poblar de datos las tablas asociadas. De esa forma, los Data Providers que se observan generan los procesos de carga de datos.

Web Panels

- WPCountryAmusementParks



Ejecute la aplicación y verifique a través de las pantallas generadas por la aplicación del pattern Work With, que efectivamente los datos se han cargado en forma automática.

## WEB PANELS

### MÚLTIPLES GRIDS [OPCIONAL]

Se desea ver una pantalla que liste la información de todos los países con sus respectivos parques relacionados.

En el Web Panel a definir, coloque un Grid Free Style y dentro de éste, un Grid; cada uno de ellos con sus atributos correspondientes, de manera que la información aparezca de la siguiente manera:

Brazil	France
<b>Amusement Park Photo Amusement Park Name</b>  Beto Carrero World	<b>Amusement Park Photo Amusement Park Name</b>  Parc Du Bocasse
China <b>Amusement Park Photo Amusement Park Name</b>  Shanghai Disney Resor  Happy Valley	United States <b>Amusement Park Photo Amusement Park Name</b>  Disney Animal Kingdom

¿Cuál es la transacción base del Grid exterior?

¿Cuál es la transacción base del Grid anidado?

Agregue un nuevo país y vuelva a ejecutar el Web Panel.

¿Qué cambios debe realizar si solamente quiere ver en pantalla los países que tengan parques de diversiones relacionados?

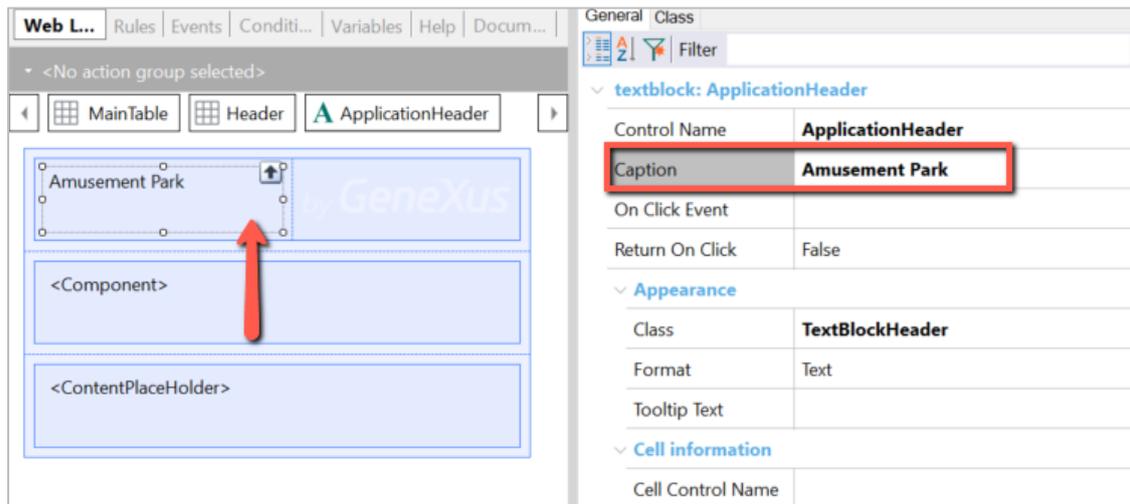
## TIPOS DE WEB PANELS

Diríjase al KB Explorer y busque en el filtro “RWD Master Page”. Esta es la Master Page asociada a todos los objetos GeneXus de nuestra KB.

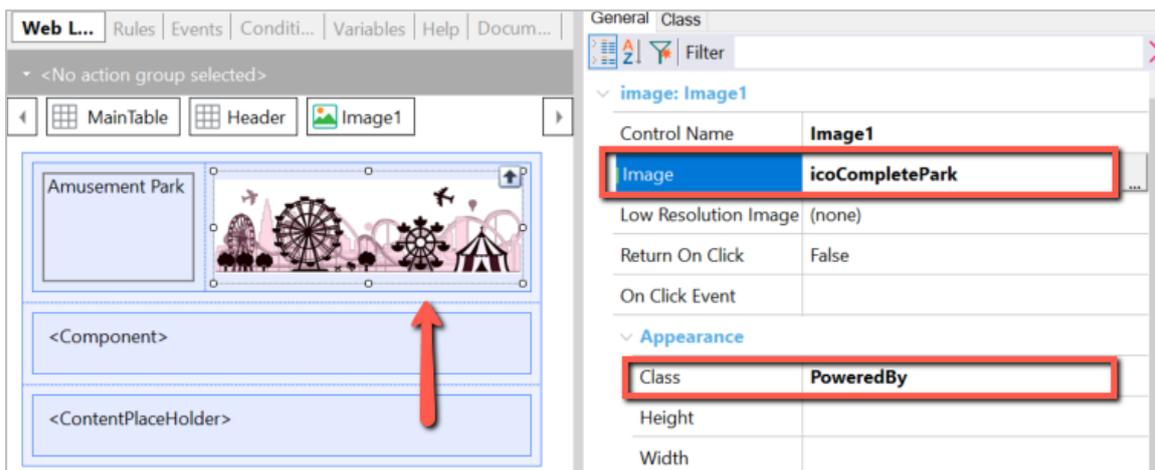
Observe que cuenta con tres secciones:

- Un Header donde podemos localizar el nombre de la aplicación y el logotipo.
- Un objeto Web Component que es el encargado de ofrecernos el menú de los links recientes.
- Un Content Place Holder donde se despliegan los objetos de nuestra aplicación.

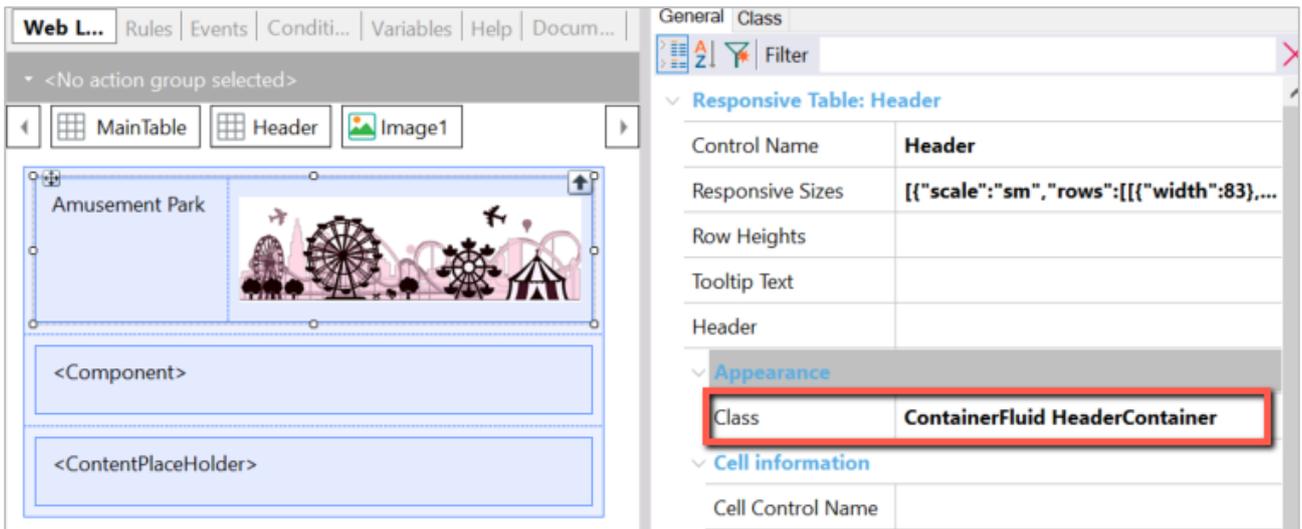
En el *Header*, seleccione el control *ApplicationHeader* y cambie la propiedad *Caption* por el nombre de la aplicación: Amusement Park:



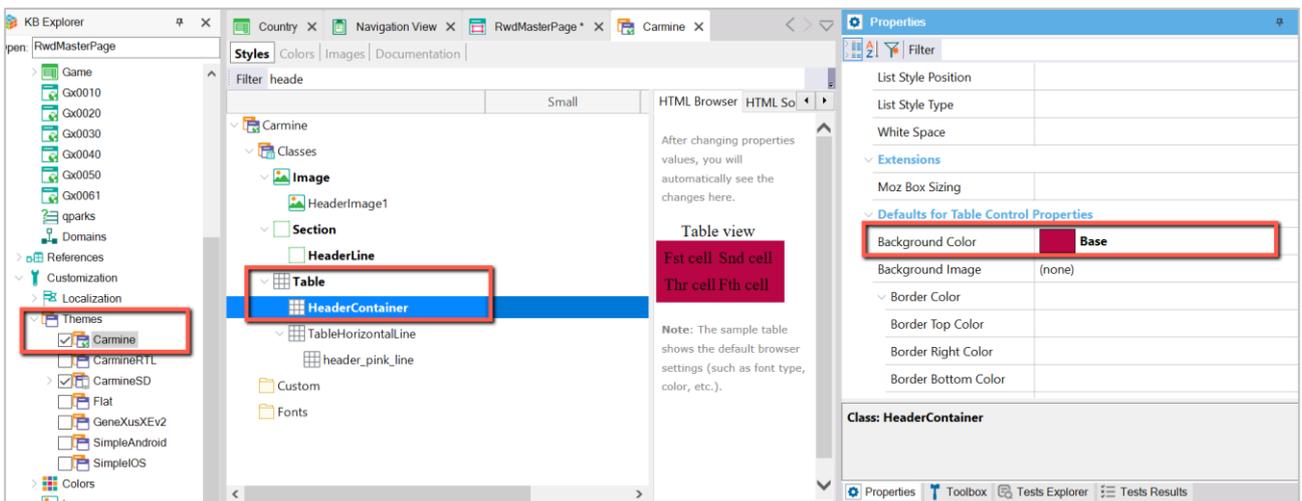
Seleccione el control *Image1* y en la propiedad *Image* seleccione una imagen como logotipo de la empresa. Verifique que tenga la clase *PoweredBy* aplicada:

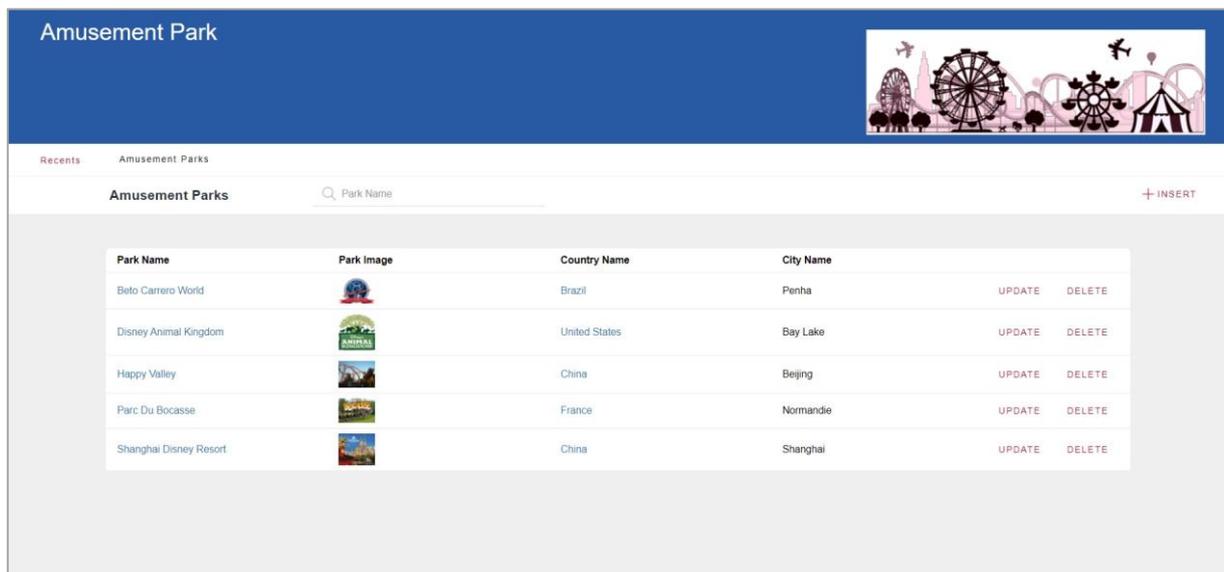


Ahora, seleccione el control *Header* y observe que tiene aplicada las clases *ContainerFluid* y *HeaderContainer*:



Vamos a editar el color de la aplicación. Para esto navegue en el KB Explorer dentro del nodo *Customization* → *Themes* → *Carmine*, dé doble clic y busque dentro de *Table* la clase *HeaderContainer*. Observe que en la propiedad *Background Color* se encuentra el color base del *Header*. Cámbielo por el de su preferencia para comenzar a dar estética a la aplicación:

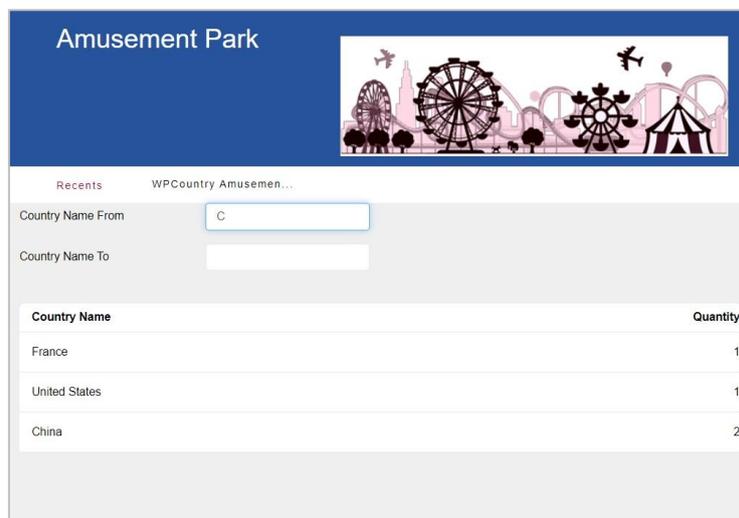




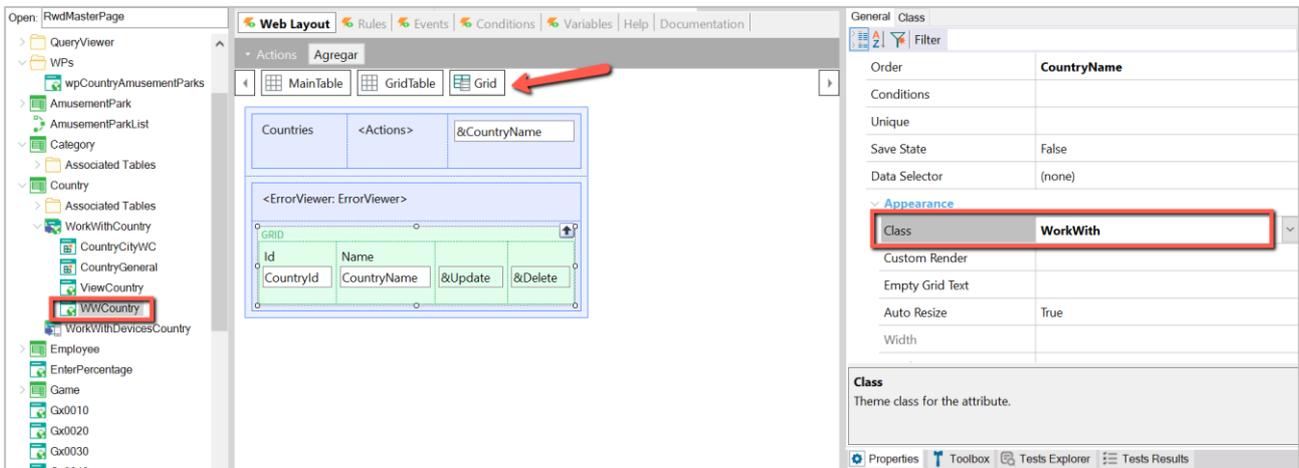
## DESIGN SYSTEMS

Abra el Web Panel donde se construyó la solución para mostrar todos los países y para cada uno, la cantidad de parques de diversiones que ofrece visitar (WPCountryAmusementParks).

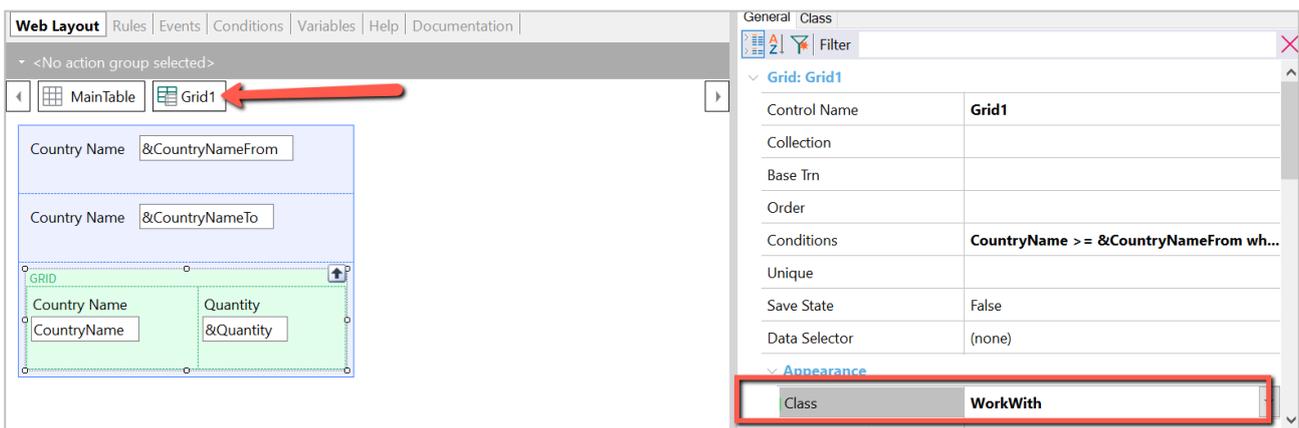
Se desea que este Web Panel tenga el mismo formato presentado en los Grids de los WorkWithForWeb:



En algún Web Panel *WW* asociado al *WorkWith*, observe que en las propiedades del Grid hay una clase asignada llamada *WorkWith*:

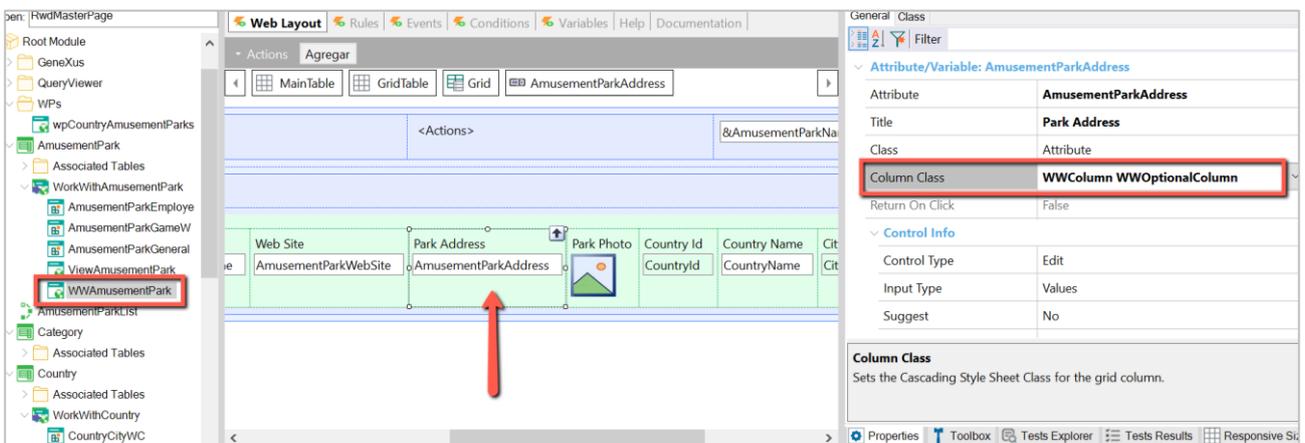
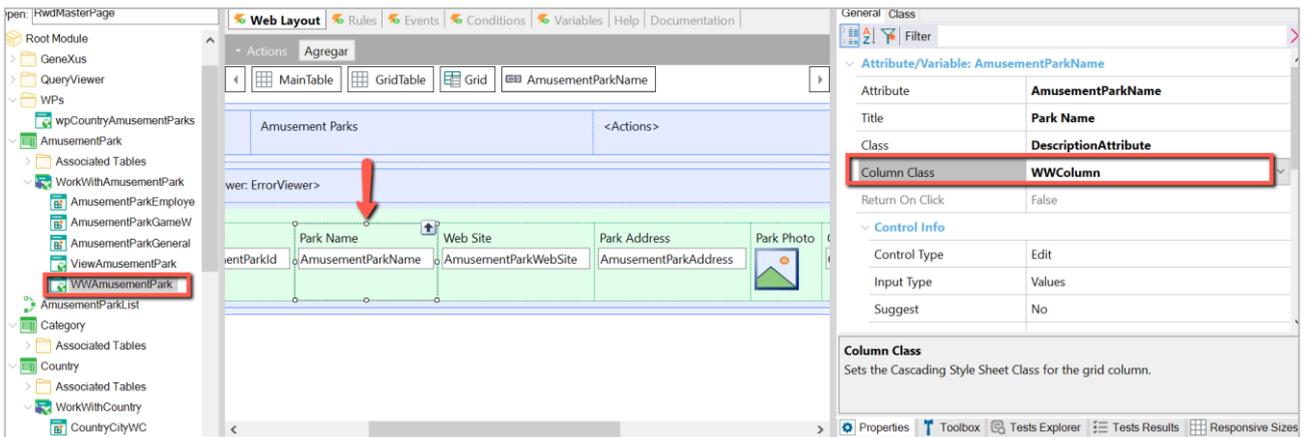


Esta misma clase aplíquela en el control Grid del Web Panel al que daremos formato:

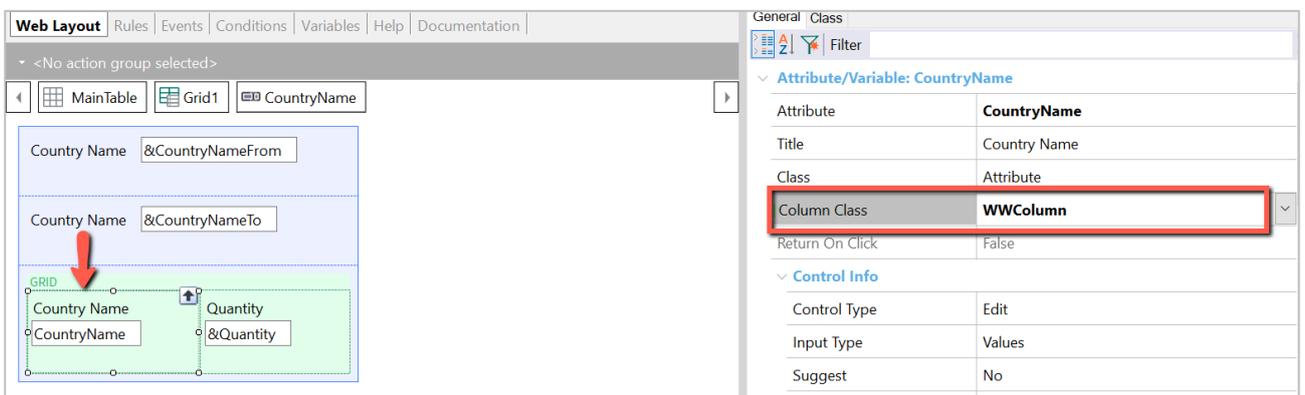


Para mantener la estética, observe que cada columna del Grid WorkWith tiene una clase asociada:

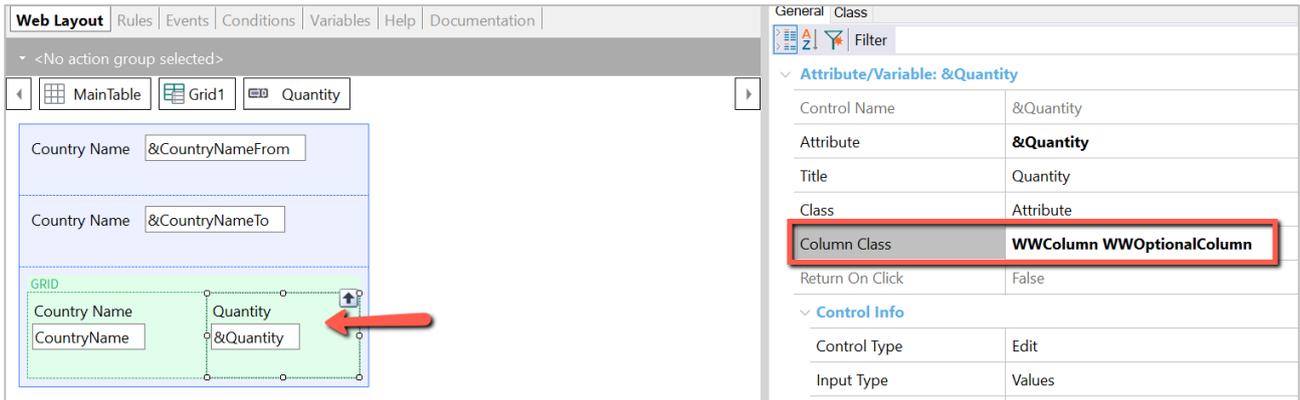
- *WWColumn* para los atributos que tienen mayor importancia que se muestran en todo momento, independientemente del tamaño de la pantalla.
- *WWColumn* y *WWOptionalColumn* para los atributos que se ocultan cuando la pantalla se vuelve más pequeña.



Para lograr un comportamiento igual en el Web Panel, escoja el nombre del país y en la propiedad *ColumnClass* coloque *WWColumn*:

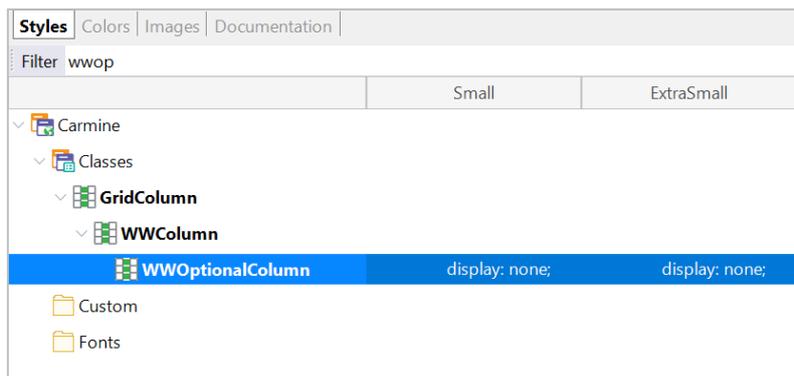


Para ocultar la cantidad de parques cuando la pantalla sea más pequeña, seleccione el control asociado a la variable y en la propiedad *ColumnClass* coloque *WWColumn* y *WWOptionalColumn*:

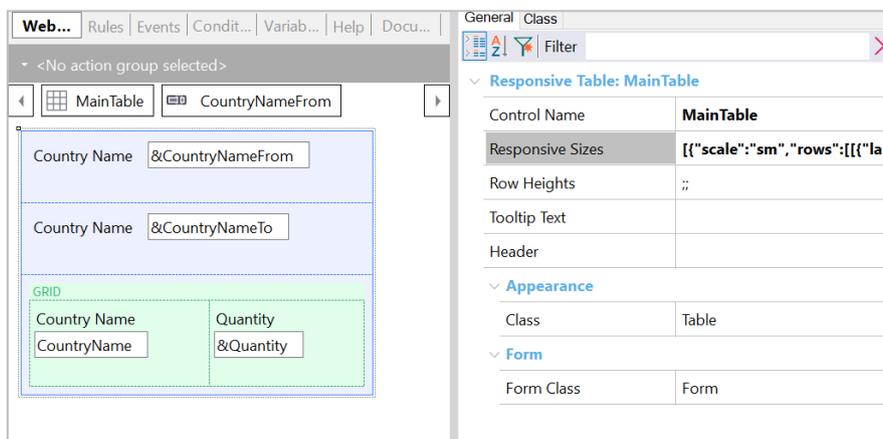


Ejecute la aplicación y pruebe el comportamiento. ¿Cómo se logran ocultar las columnas?

Gracias a la clase *WWOpcionalColumn*. En las columnas Small y ExtraSmall, indica que el atributo que tenga asignada dicha clase, no se mostrará en pantalla cuando tenga ese tamaño.

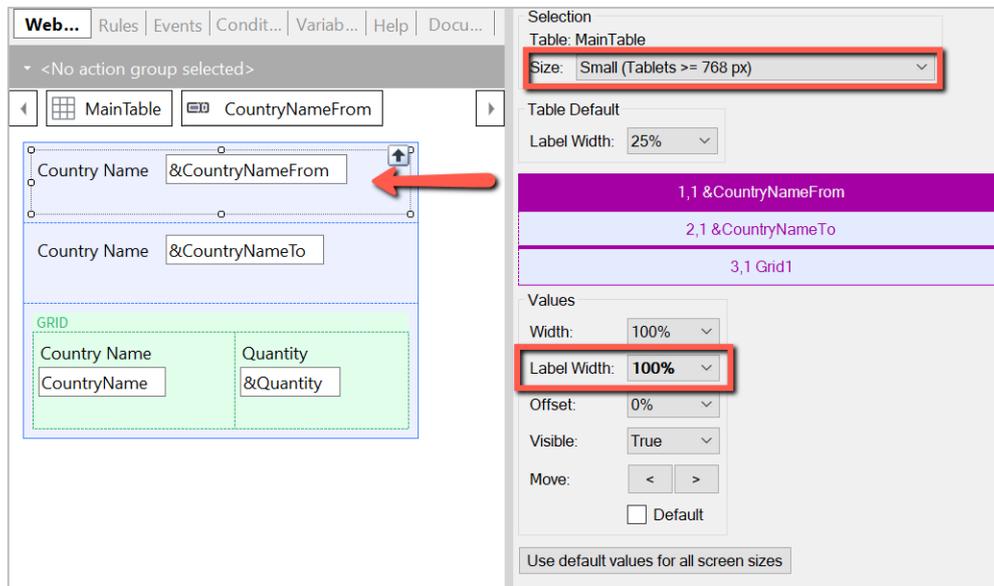


Por último, para que las etiquetas aparezcan arriba del cuadro editable, abra el recuadro de los tamaños responsivos. Si no aparecen, seleccione la MainTable del Web Panel y haga clic en la propiedad Responsive Sizes:



Seleccione la primera variable, en los tamaños responsivos seleccione el tamaño Small y en *Label Width* coloque 25%. Cambie el tamaño a Extra Small y configure lo mismo en *Label Width*.

Haga este mismo proceso para la segunda variable:



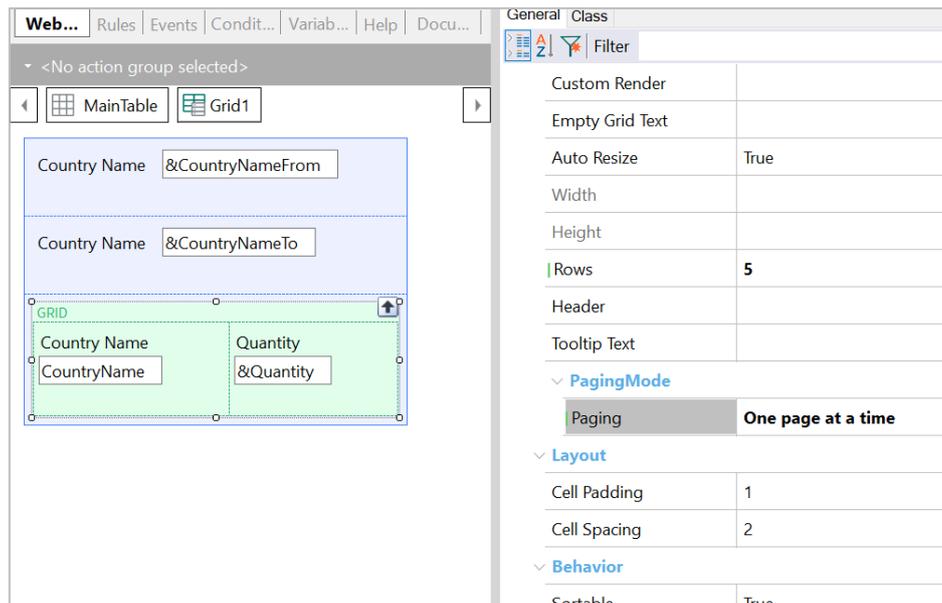
Ejecute la aplicación y verifique el comportamiento.

## PAGINADO EN GRIDS

Se requiere que la pantalla anterior cuente con un Paginado en el Grid para visualizar los países de una mejor manera.

Para esto, agregue más países.

En las propiedades del Grid del Web Panel, cambie la propiedad *Rows* por 5 y en *Paging* verifique que se encuentra activado *One page at a time*:

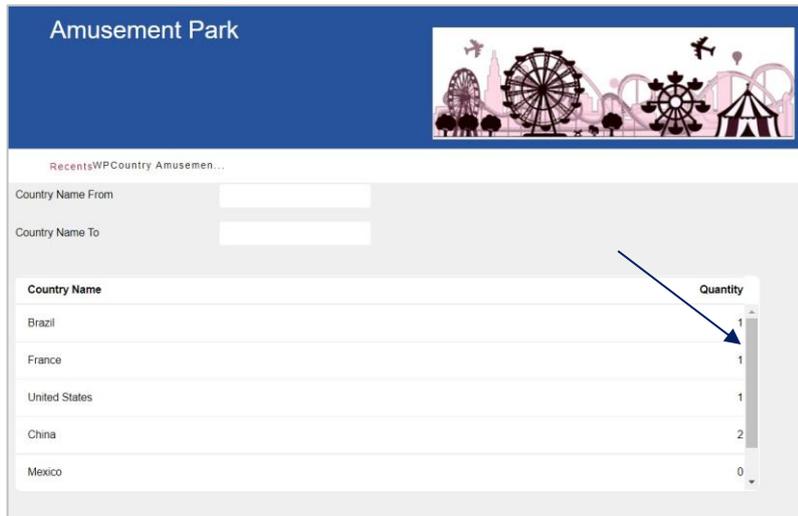


Ejecute y observe que ahora sólo se muestran 5 registros por página y gracias a las flechas de navegación del Grid puede desplazarse por los países:



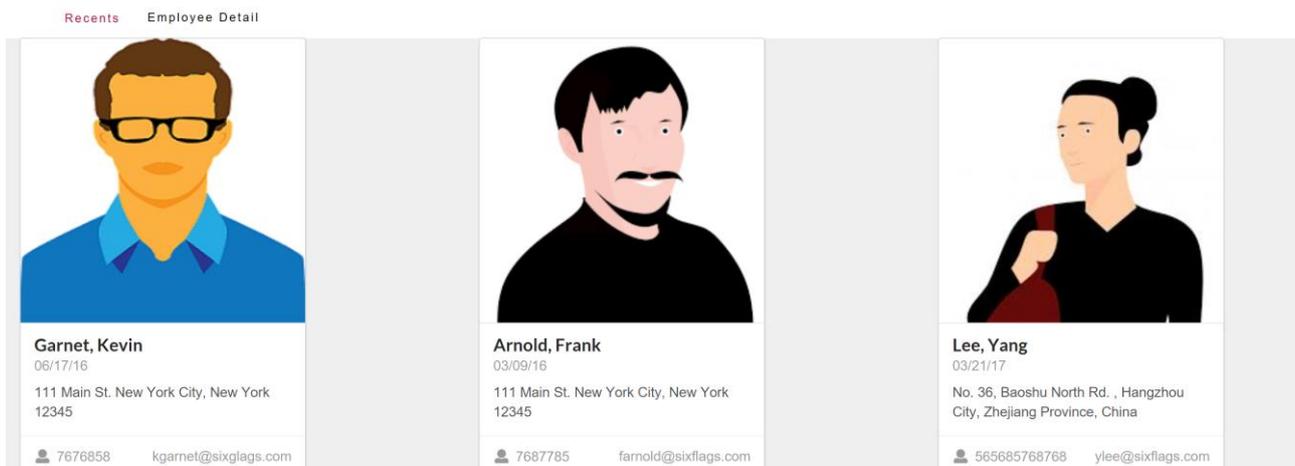
¿Qué sucedería si no quisiera ver la información en diferentes páginas, sino simular que el contenido se encuentra en una sola pantalla y hacer scroll para acceder a los registros? Regrese al Web Panel y cambie la propiedad *Paging* por *Infinite scrolling*.

Ejecute la aplicación y observe que a través de un scroll puede desplazarse a través de los registros:



### BASE STYLE Y USER CONTROL [OPCIONAL]

Se desea tener un Web Panel que liste los empleados del parque con un diseño como el que se muestra en la siguiente imagen:



### Solución:

Agregue el atributo EmployeePhoto de tipo Image a la transacción Employee con la finalidad de mostrar una foto para el empleado.

Será necesario crear un objeto del tipo User Control que permitirá presentar los datos del empleado siguiendo las pautas del diseño que nos solicitan.

#### Recuerde que:

- Es preciso contar con una biblioteca base la cual establece el estilo del User Control que se va generar.
- El estilo base es un zip que contiene todos los archivos css, js, assets, etc. provistos por los diseñadores o el Design System utilizado.

Para crear el Base Style:

Si no cuenta con la librería, descargue de la [página de SemanticUI](#) los archivos necesarios dando clic en *Download Zip*.

### Simpler Setup

If you are using Semantic UI as a dependency and just want to use our default theme, use our lighter `semantic-ui-css` or `semantic-ui-less` repo. If you just need the files you can download them as a zip from GitHub.

[Download Zip](#) [View all Semantic-Org Repos](#)

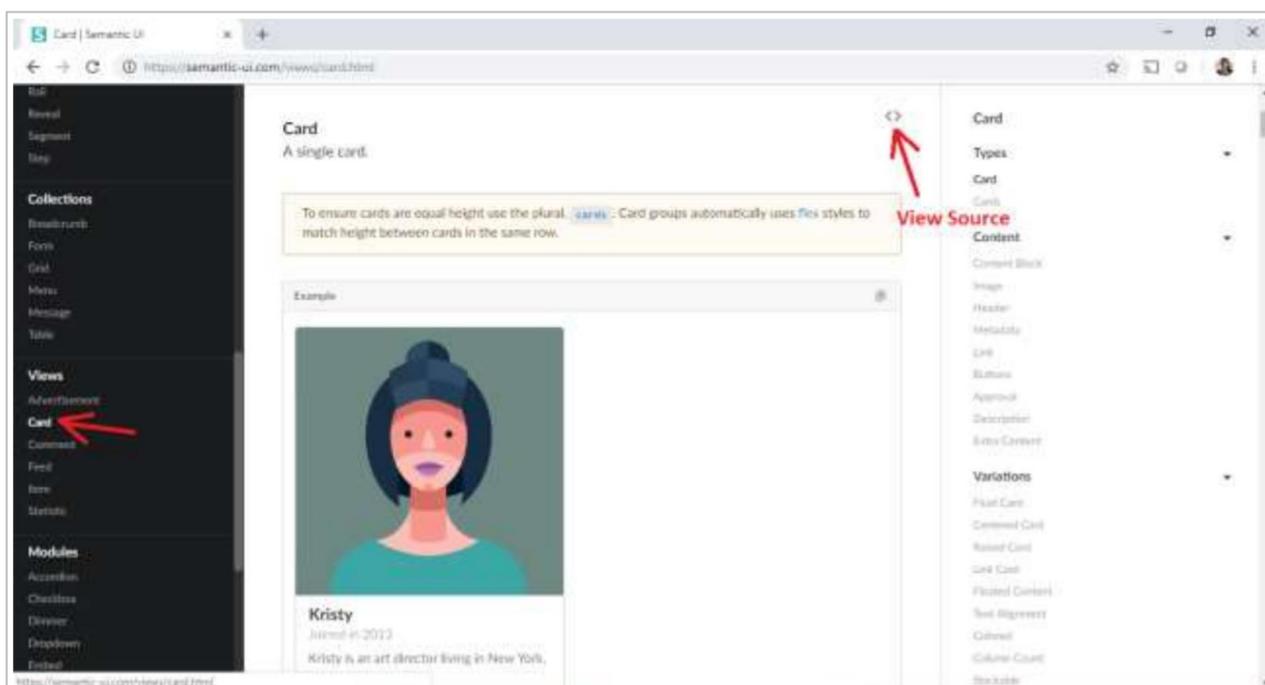
Guarde el .zip con la extensión *gxlibrary*.

Cree un objeto de tipo File de nombre **SemanticUI\_gxlibrary** e importe el archivo *SemanticUI.gxlibrary*.

Nota: si está trabajando con la versión Trial de GeneXus, diríjase al menú *Knowledge Manager* → *Import* para importar ese archivo, ya que no tiene disponible el nodo *Files* en su KB Explorer.

Para crear el User Control:

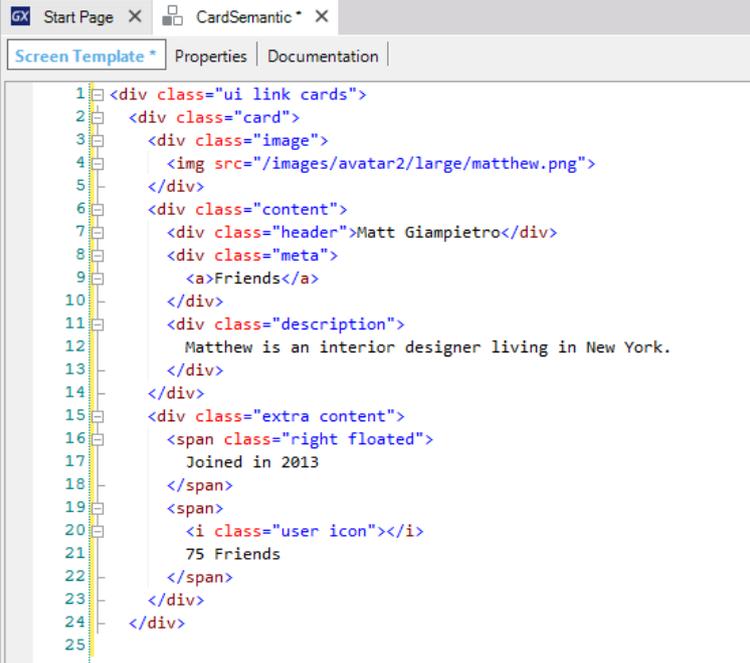
Será necesario crear un User Control basado en un “Card” del framework SemanticUI. Si va al sitio de SemanticUI, y busca por Card, encontrará distintos tipos de estos controles. Elija por ejemplo la Card más simple y visualice su código HTML, para poder copiarlo:



Para agilizar el desarrollo, le proveemos el código a utilizar.

- Cree un nuevo objeto del tipo User Control de nombre CardSemantic
- En el tab Screen Template copie el siguiente código:

```
<div class="ui link cards">
  <div class="card">
    <div class="image">
      
    </div>
    <div class="content">
      <div class="header">Matt Giampietro</div>
      <div class="meta">
        <a>Friends</a>
      </div>
      <div class="description">
        Matthew is an interior designer living in New York.
      </div>
    </div>
    <div class="extra content">
      <span class="right floated">
        Joined in 2013
      </span>
      <span>
        <i class="user icon"></i>
        75 Friends
      </span>
    </div>
  </div>
</div>
```



```
1 <div class="ui link cards">
2   <div class="card">
3     <div class="image">
4       
5     </div>
6     <div class="content">
7       <div class="header">Matt Giampietro</div>
8       <div class="meta">
9         <a>Friends</a>
10      </div>
11      <div class="description">
12        Matthew is an interior designer living in New York.
13      </div>
14    </div>
15    <div class="extra content">
16      <span class="right floated">
17        Joined in 2013
18      </span>
19      <span>
20        <i class="user icon"></i>
21        75 Friends
22      </span>
23    </div>
24  </div>
25
```

Parametrice las propiedades, de acuerdo a lo que desea ver. En nuestro caso quedará así:

The screenshot shows the GeneXus IDE interface. On the left, the 'Screen Template' editor displays the following code:

```

1 <div class="ui link cards">
2   <div class="card">
3     <div class="image">
4       
5     </div>
6     <div class="content">
7       <div class="header">{{Name}}</div>
8       <div class="meta">
9         <a>{{AddedDate}}</a>
10      </div>
11      <div class="description">
12        {{Address}}
13      </div>
14    </div>
15    <div class="extra content">
16      <span class="right floated">
17        {{Email}}
18      </span>
19      <span>
20        <i class="user icon"></i>
21        {{Phone}}
22      </span>
23    </div>
24  </div>
25

```

On the right, the 'Properties' window for the 'User Control: CardSemantic' is shown. The 'Base Style' property is highlighted with a red box and set to 'SemanticUI'.

Name	CardSemantic
Description	Card Semantic
Module/Folder	Design
Is Control Type	False
References	
Base Control Type	None
Base Style	<b>SemanticUI</b>
Qualified Name	CardSemantic
Object Visibility	Public

Revise que la propiedad Base Style del objeto User Control tenga el valor "SemanticUI" definido. De lo contrario, seleccione ese valor.

Aquí proveemos el código anterior para que pueda copiarlo:

```

<div class="ui link cards">
<div class="card">
<div class="image">
  
</div>
<div class="content">
<div class="header">{{Name}}</div>
<div class="meta">
  <a>{{AddedDate}}</a>
</div>
<div class="description">
  {{Address}}
</div>
</div>
<div class="extra content">

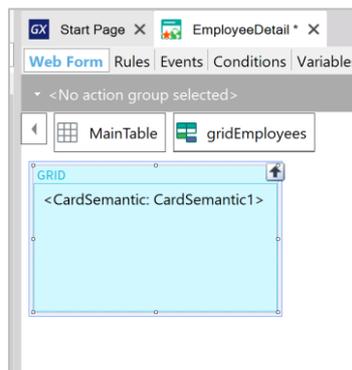
```

```

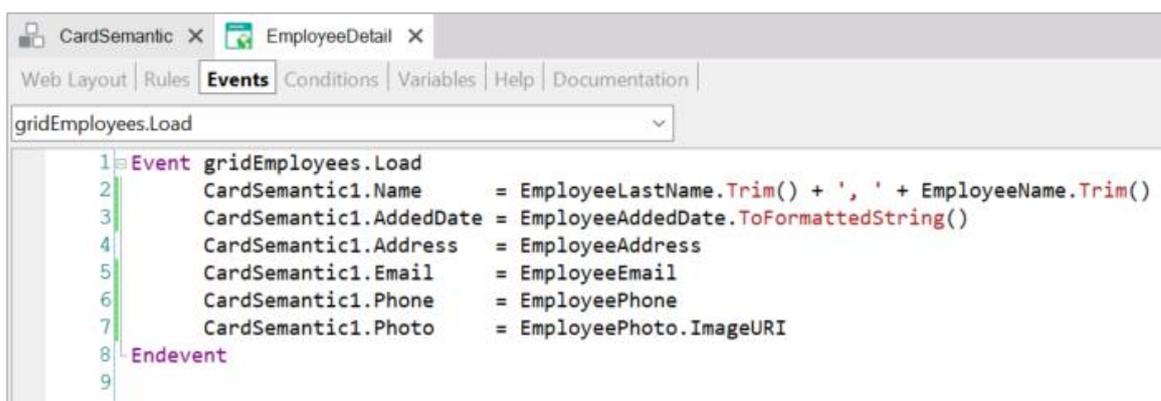
<span class="right floated">
  {{Email}}
</span>
<span>
  <i class="user icon"></i>
  {{Phone}}
</span>
</div>
</div>

```

- Cree un Web Panel de nombre EmployeeDetail
- Agregue un Grid Free Style e inserte el User Control que acaba de crear. En la Toolbox encontrará el elemento User Control, arrastre el control dentro del Grid y seleccione el control que acaba de crear.



- Inicialice las propiedades del User Control en base a los atributos:



- Ejecute.

## PANELS

Se solicita una pantalla, tanto para plataforma web como dispositivos móviles, que muestre la información de todos los parques.

Para eso, cree un objeto de tipo Panel de nombre AmusementPark\_Panel, y arrastre un control Grid sobre el form. Inserte las tablas y los siguientes atributos, como se muestra:

- AmusementParkImage – ocupando 3 filas (Propiedad Row Span = 3)
- AmusementParkName
- CountryName
- CityName

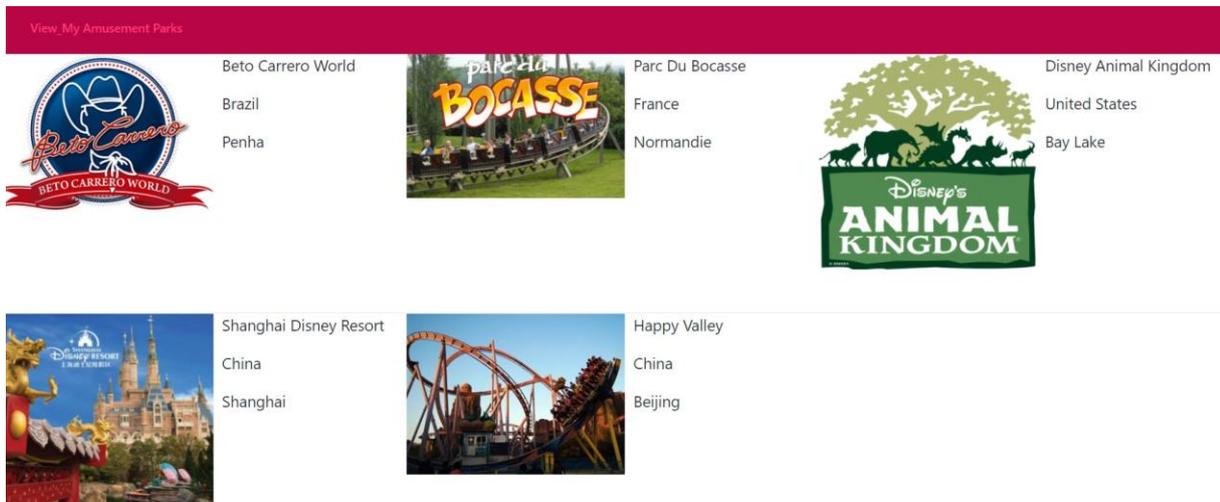
Table: Grid1Table	
Control Name	<b>Grid1Table</b>
Layout Name	
Appearance	
Columns Style	<b>250dip:10dip:100%</b>
Rows Style	300dip
Width	100%
Height	300dip
Auto Grow	True
Control Info	
Control Type	Grid
Auto Grow	False
Scroll Direction	Vertical
Snap To Grid	False
Items Layout Mode	<b>Multiple by Quantity</b>
Items Per Row	<b>3</b>

Definir el objeto Panel como Main Program (Propiedad Main Program = True)

Defina las siguientes propiedades a nivel del .Net Environment:

Generator: Frontend (Front end)	
Name	Frontend
Generate Android	<b>False</b>
Generate Apple	<b>False</b>
Main Platform	<b>Angular</b>
Dynamic Services URL	False
Services URL	https://trialapps3.genexus.com/lddc3710...
SSL Pinning Pin Set	
Smart Devices Cache Ma	On
Generate Angular	<b>True</b>

Ejecute el Panel (opción Run).



Se desea ahora modificar la apariencia del Grid (propiedad Control Type) para que los parques se visualicen como puntos en el mapa.

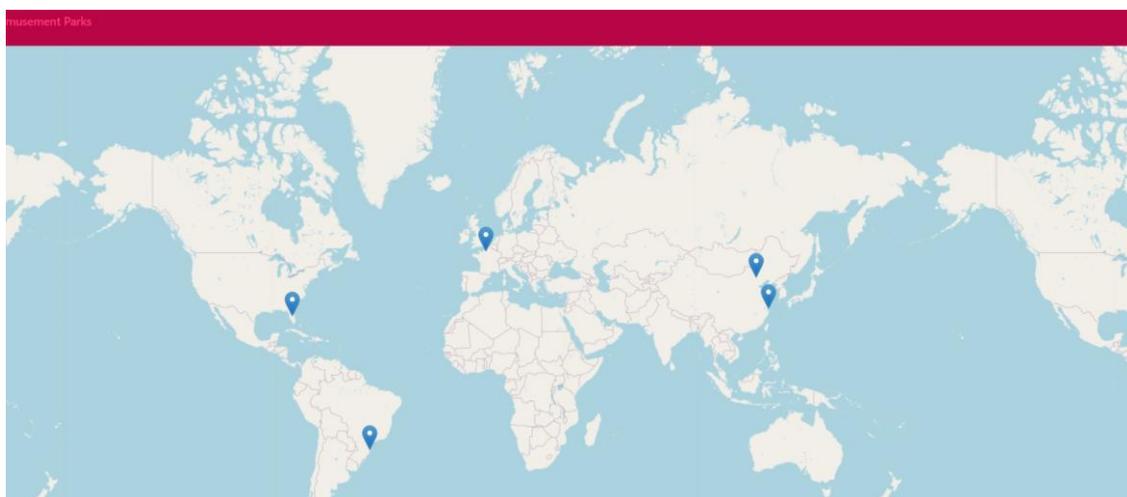
Application Bar

MainTable Grid1 Grid1Table

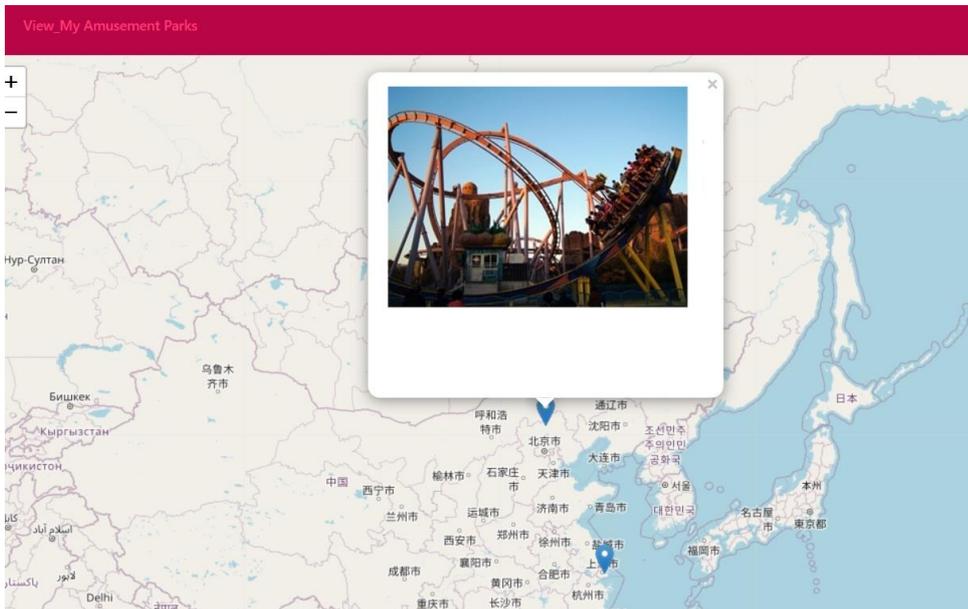
GRID

Control Info	
Control Type	<b>Maps</b>
Auto Grow	False
Show My Location	False
Map Type	Standard
User Can Choose Map Type	False
Location Attribute	<b>AmusementParkGeolocation</b>
Geometry Layer Id Attribute	

Ejecute el Panel. Visualizará los parques como puntos en el mapa.



Pruebe de ampliar el mapa y seleccionar uno de los puntos.



Para finalizar, pruebe de cambiar los valores de la propiedad Map type, por ejemplo a Satellite, y ejecute el Panel:

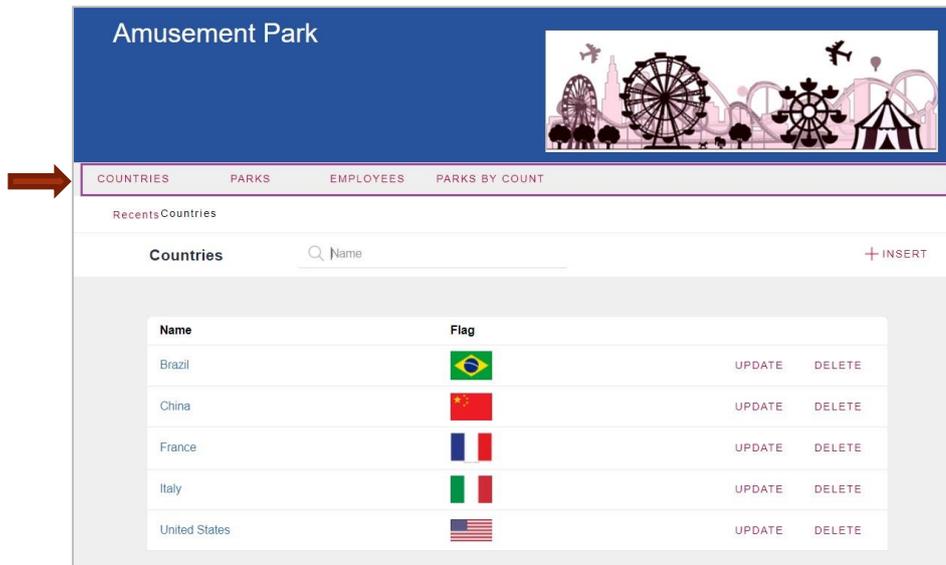
Control Info	
Control Type	Maps
Auto Grow	False
Show My Location	False
Map Type	Satellite
User Can Choose Map Type	Standard
Location Attribute	Satellite
Geometry Layer Id Attribute	Hybrid
Geometry Layer Id Field Specifier	



**STARTUP OBJECT**

**MENU DE OPCIONES Y MASTERPAGE**

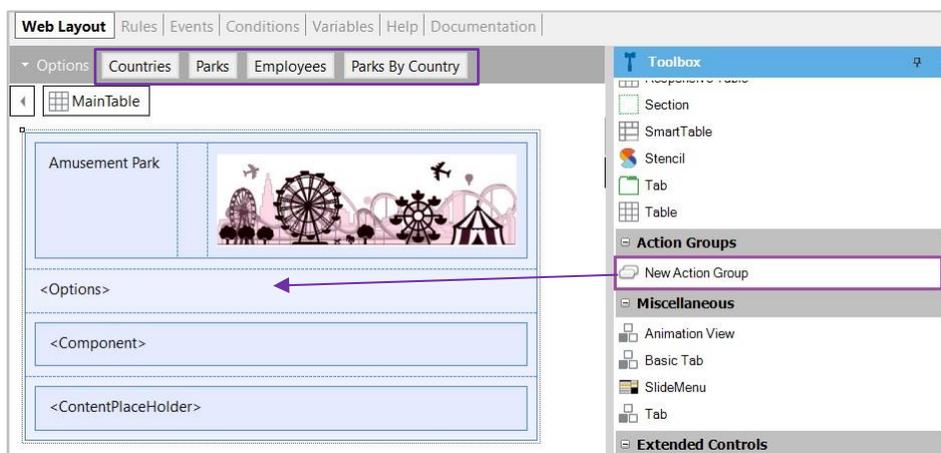
Para finalizar, se desea implementar un menú de opciones para poder acceder a las funcionalidades anteriormente desarrolladas.



La idea es que el menú de opciones se encuentre siempre visible en toda la aplicación, por lo tanto, un buen lugar para definirlo es la MasterPage utilizada.

Recuerde que, por defecto, se utiliza la RwdMasterPage. Edítala y arrastre el control Action group presente en la Toolbox. La ubicación en el form es la que se desee. En este caso, la ubicamos debajo del Header:

Para agregar las opciones, click derecho sobre el control Options, Insert Button



Al hacer doble click sobre cada uno de los botones insertados, se editará el evento asociado:

```

Layout | Rules | Events | Conditions | Variables | Help | Documentation |
57
58 Event 'Countries'
59     WWCountry()
60 Endevent
61
62 Event 'Parks'
63     WWAmusementPark()
64 Endevent
65
66 Event 'Employees'
67     WWEmployee()
68 Endevent
69
70 Event 'ParksByCountry'
71     WPCountryAmusementParks()
72 Endevent
73

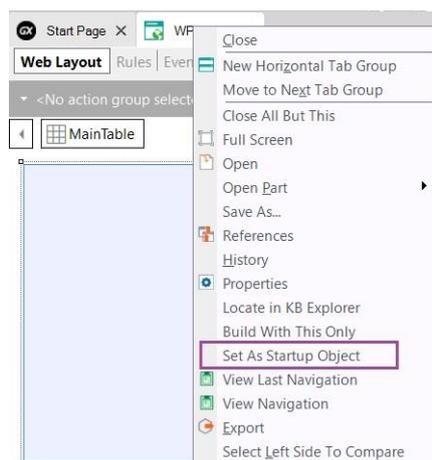
```

Ejecute la aplicación. ¿Qué apariencia tienen esos botones? ¿Se puede cambiar reusando alguna clase? Si. Pruebe por ejemplo la clase **BtnCancel**. Es la clase asociada a los botones Cancel y Delete de las transacciones.

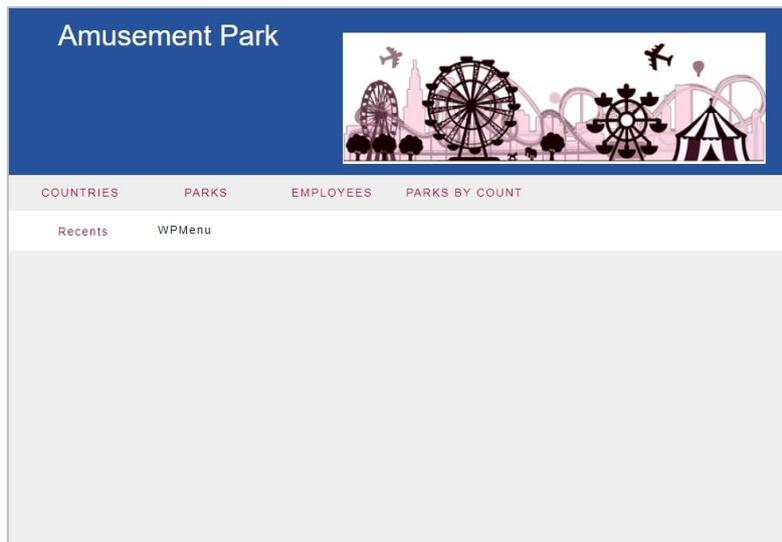
Ejecute nuevamente.

Llegado a este punto, solo resta definir el objeto Startup para que el Developer Menu deje de serlo.

Cree un nuevo Web Panel de nombre WPMenu, y defínalo como Startup object (click derecho, Set as Startup Object)



Presione F5.



El diseño de este Web Panel queda a gusto del desarrollador. Pruebe, por ejemplo, de editar nuevamente la MasterPage y eliminar el componente que muestra los Recent Links.

Piense ahora lo siguiente:

¿Es necesario que el menú de opciones se defina directamente en la MasterPage? No.

Otra posibilidad es definir un Web Panel de tipo Component y diseñar en él el menú de opciones. Luego se deberá referenciar a dicho component desde la MastarPage.

Pruebe también esta implementación.