

Ejercicios prácticos

Curso GeneXus Core

GeneXus™ 17

Abril 2022

© GeneXus. Todos los derechos reservados. Este documento no puede ser reproducido por ningún medio sin la autorización expresa de GeneXus™. 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.

CONTENIDO

EL PROBLEMA	4
NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO	4
PRIMERAS TRANSACCIONES	5
Transacción "Employee"	5
Transacciones "AmusementPARK" y "Country", relacionadas.....	7
Datos relacionados: ¿cómo se mantiene la integridad?.....	10
Transacción 'Show'.....	11
Transacción 'Game'	11
Transacción 'Category'.....	11
Transacciones "Employee" y "AmusementPark", relacionadas.....	13
Agreguemos las web a la transacción 'Country'	14
Transacción "AmusementPark": agreguemos la ciudad.	15
AGREGUEMOS COMPORTAMIENTO A LAS TRANSACCIONES (RULES)	16
PATTERNS: MEJORANDO LA INTERFAZ PARA TRABAJAR CON LA INFORMACIÓN.....	17
TRANSACCIONES "REPAIR" Y "TECHNICIAN" Y NECESIDAD DE DEFINIR SUBTIPOS.....	20
FÓRMULAS	21
CREACIÓN DE SEGUNDO NIVEL	22
ÍNDICES.....	23
LISTADOS PDF.....	24

PASAJE DE PARÁMETROS.....	31
Listado de parques en un rango determinado	31
BUSINESS COMPONENTS.....	31
Aumento de precio de las reparaciones.....	31
Pantalla para eliminación de todas las reparaciones	32
PROCEDIMIENTOS PARA ACTUALIZAR REGISTROS.....	33
Aumento de precios de las reparaciones.....	33
Eliminación de todas las reparaciones.....	34
Inicialización de la información de la base de datos [OPCIONAL].....	35
WEB PANELS	37
Múltiples Web Panels [OPCIONAL].....	37
Tipos de Web Panels.....	38
DESIGN SYSTEMS.....	40
Paginado en Grids	44
GENEXUS SERVER.....	46



EL PROBLEMA

Una multinacional encargada de gestionar parques de diversiones lo contrata para que desarrolle un sistema para almacenar y manipular la información con la que trabaja. Imagine que el sistema se compone de un módulo:

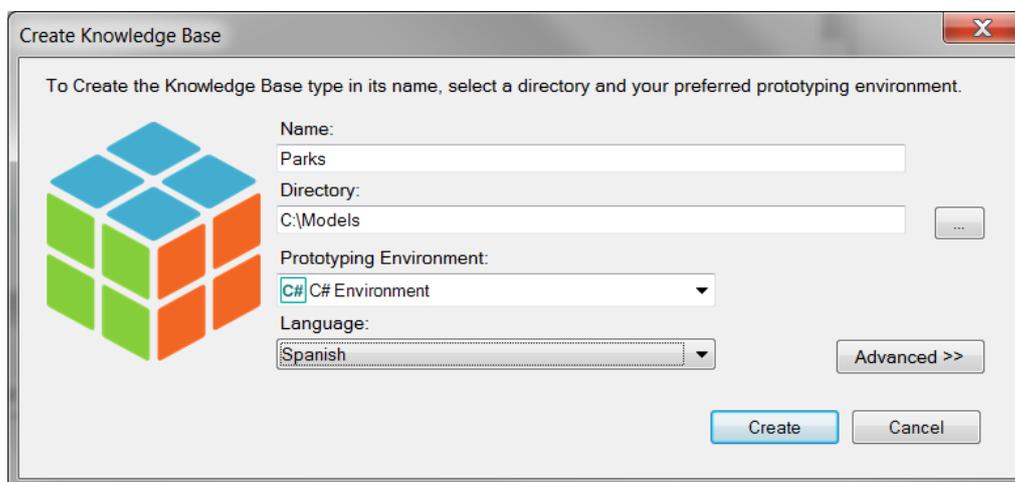
- **Backend:** parte de la aplicación que deberá correr en un servidor web, de manera tal que los empleados de la empresa puedan manipular la información desde cualquier lugar con conexión a internet.

NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO

Entrar a GeneXus y crear una base de conocimiento de nombre *Parks* para comenzar el desarrollo de la aplicación.

Sugerimos:

- Elegir como ambiente de desarrollo C#. Asegúrese de tener instalado todo lo necesario (incluyendo SQL Server). Si usa GeneXus Trial, el ambiente de generación con C# y SQL Server ya es predefinido, prototipando en la nube de Amazon.
- No crear la base de conocimiento en la carpeta “Mis Documentos” o cualquier otra carpeta que quede bajo “Documents and Settings”, debido a que estas carpetas tienen permisos especiales otorgados por Windows.



Tómese unos minutos para **familiarizarse** con el **IDE (ambiente de desarrollo integrado de GeneXus)**. Pruebe **mover ventanas, visualizar ventanas específicas que desee** (View y View/Other Tool Windows) y observe detenidamente el contenido de la ventana **KB Explorer** (Knowledge Base Explorer). Verá que aparecen ya inicializados **dominios**, algunos **objetos, imágenes**, etc.

Sugerencia: mantenga la ventana de propiedades abierta (**F4**), pues la utilizará continuamente.

PRIMERAS TRANSACCIONES

En las reuniones con la empresa, le transmiten lo siguiente:

“Nosotros registramos los datos de parques de diversiones, para realizar la gestión tanto de sus empleados como de sus juegos y actividades que ofrece a los visitantes”.

Para empezar a construir la aplicación, debemos empezar por identificar los actores de la realidad, y representarlos mediante **transacciones**. ¿Qué transacciones debemos crear entonces en la base de conocimiento (KB)?

TRANSACCIÓN “EMPLOYEE”

Preguntamos: ¿qué datos registran de los empleados de la empresa? La respuesta es la siguiente:

El **nombre** (que no supera los 20 caracteres), **apellido** (que tampoco los supera), **dirección**, **teléfono** y el **e-mail**.

Con estos datos ya puede crear la transacción *Employee*.

Recordar que:

- Para crear objetos existen varias alternativas:
 - Hacerlo desde el menú: File/New/Object
 - Ctrl+N
 - Ícono de la barra de herramientas
- Necesitará un atributo que identifique a cada empleado (EmployeeId).
- Digitando punto (“.”) cuando va a ingresar un nuevo atributo, éste se inicializa con el nombre de la transacción.

La estructura de la transacción debería haberle quedado como se muestra:

Name	Type
Employee	Employee
EmployeeId	Numeric(4,0)
EmployeeName	Character(20)
EmployeeLastName	Character(20)
EmployeeAddress	Address, GeneXus
EmployeePhone	Phone, GeneXus
EmployeeEmail	Email, GeneXus

Recordar que:

- **Address, Phone e Email** son **dominios semánticos** que se asignan automáticamente a los atributos que se definen conteniendo en su nombre los textos Address, Phone o Email respectivamente.
- Cuando esté definiendo el tipo de datos del atributo identificador, en vez de utilizar directamente Numeric(4,0), defina el **dominio Id** de forma inline: **Id=Numeric(4,0)**. Configure la propiedad **Autonumber** de ese dominio en **True**, para que todos los atributos basados en el mismo se numeren automáticamente, sin que el usuario deba preocuparse.

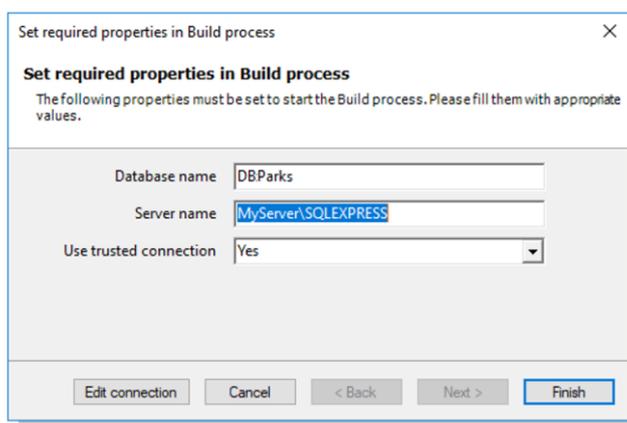
El próximo paso es probar la aplicación en ejecución. Asegúrese de tener la ventana *Output* de GeneXus habilitada y a la vista (*View/Other Tool Windows/Output*).

Ahora sí, **pruebe** la aplicación en ejecución presionando **F5**.

¿Qué sucederá?

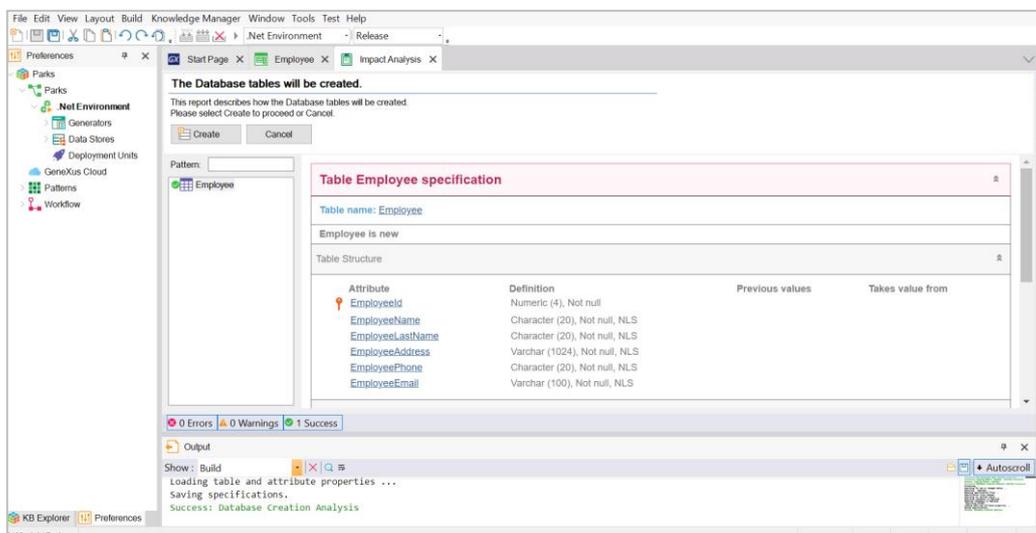
Recuerde:

Si está utilizando la versión *Full* o la versión *Learning* de GeneXus y decide crear la base de datos y programas **localmente**, se le abrirá una ventana como la siguiente para que ingrese la información de Base de Datos, Servidor y método de conexión. Recuerde que si no existe una base de datos con el nombre que indicó en ese servidor, GeneXus la **creará**.



Si en cambio la base de datos y programas se crearán en la nube, el diálogo anterior no aparece puesto que GeneXus conoce los datos del servidor en la nube y configura automáticamente el nombre de la base de datos y toda la información de conexión a la misma.

A continuación, se despliega un **Análisis de Impacto** que detalla que se creará la base de datos y la tabla *EMPLOYEE* dentro de la misma:



Si presiona el botón **Create**, GeneXus procederá a ejecutar el programa que llevará a cabo la creación. Al finalizar el proceso, se le abrirá en el navegador que tenga configurado como predeterminado, el menú con links para ejecutar los objetos definidos. En este caso sólo uno: la transacción *Employee*.

Ingrese algunos empleados al sistema. Luego, **modifique** algún dato de alguno de los empleados previamente ingresados y **elimine** algún empleado.

También pruebe usar las flechitas ofrecidas para pasar de registro en registro de empleados y la opción *SELECT*, que ofrece una “lista de selección” para ver la lista de empleados registrados y seleccionar uno.

Ahora pasemos a identificar y crear la siguiente transacción. Recordemos lo que nos habían enunciado, a lo cual se le hicieron algunos agregados:

“Nosotros registramos los datos de **parques de diversiones** de diferentes ciudades en diferentes países, para realizar la gestión tanto de sus empleados como de sus juegos y actividades que ofrece a los visitantes”.

Antes de continuar con el desarrollo, publiquemos la KB en GeneXus Server (<http://sandbox.genexusserver.com/v17>)

Recuerde:

- Debe autenticarse con su cuenta GeneXus.
- El alias con el cual se publique la KB debe ser único. Esto significa que no pueden existir en el Server dos KBs con el mismo alias.

TRANSACCIONES “AMUSEMENTPARK” Y “COUNTRY”, RELACIONADAS

Le preguntamos a los empleados de la empresa: ¿qué datos registran de los parques de diversiones con los que trabajan? La respuesta es la siguiente:

El nombre (que no supera los 40 caracteres), **sitio web** (que no supera los 60 caracteres), **dirección** y **foto representativa**.

Con estos datos ya puede crear la transacción *AmusementPark*.

Name	Type
AmusementPark	AmusementPark
AmusementParkId	Numeric(4,0)
AmusementParkName	Character(20)
AmusementParkWebsite	Character(60)
AmusementParkAddress	Address, GeneXus
AmusementParkPhoto	Image

Vamos a crear una transacción para registrar los países a los que los parques de diversiones pertenecen.

Recuerde que:

- presionando punto (".") cuando está por dar nombre a un atributo en la estructura de la transacción, aparece inicializado con el nombre de la transacción.
- Necesitará un atributo identificador, CountryId.

Defina el atributo *CountryName*, creando y utilizando un nuevo dominio: *Name* = *Character(50)*.

Name	Type
Country	Country
CountryId	Id
CountryName	Name

Ahora, volveremos a la transacción *AmusementPark*, para agregarle los atributos *CountryId* y *CountryName*.

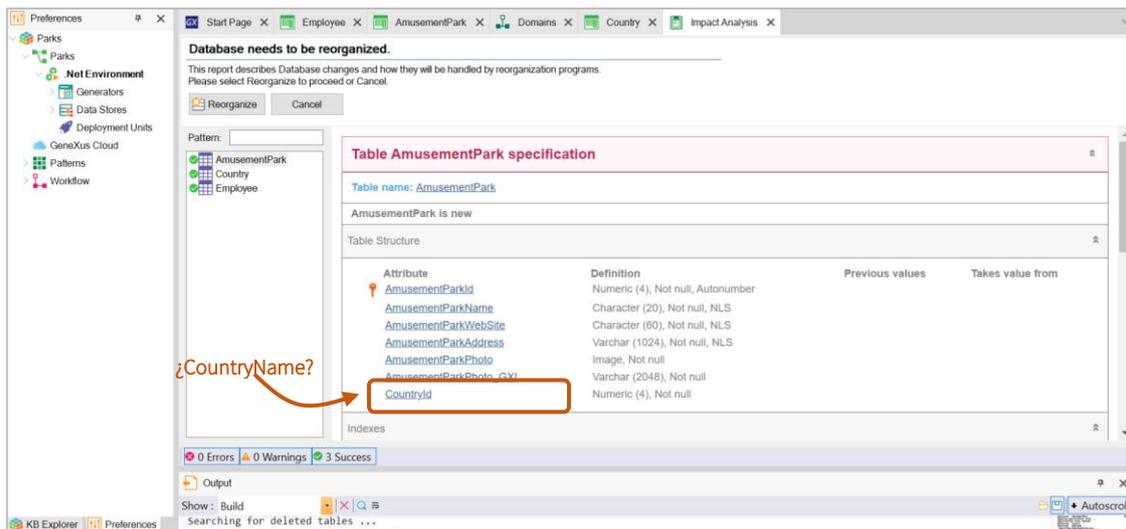
Name	Type
AmusementPark	AmusementPark
AmusementParkId	Id
AmusementParkName	Character(20)
AmusementParkWebsite	Character(60)
AmusementParkAddress	Address, GeneXus
AmusementParkPhoto	Image
CountryId	Id
CountryName	Name

Name	Type
Country	Country
CountryId	Id
CountryName	Name

¿Por qué colocó además de *CountryId*, el atributo *CountryName* en *AmusementPark*?

Respuesta: El atributo *CountryName* es importante que aparezca en la pantalla de la transacción, para mostrarnos el nombre del país, que es el dato que recordamos mejor del país, en lugar de ver solamente su identificador. También es necesario agregar al atributo en la estructura de la transacción si luego lo queremos utilizar, por ejemplo, dentro de una regla.

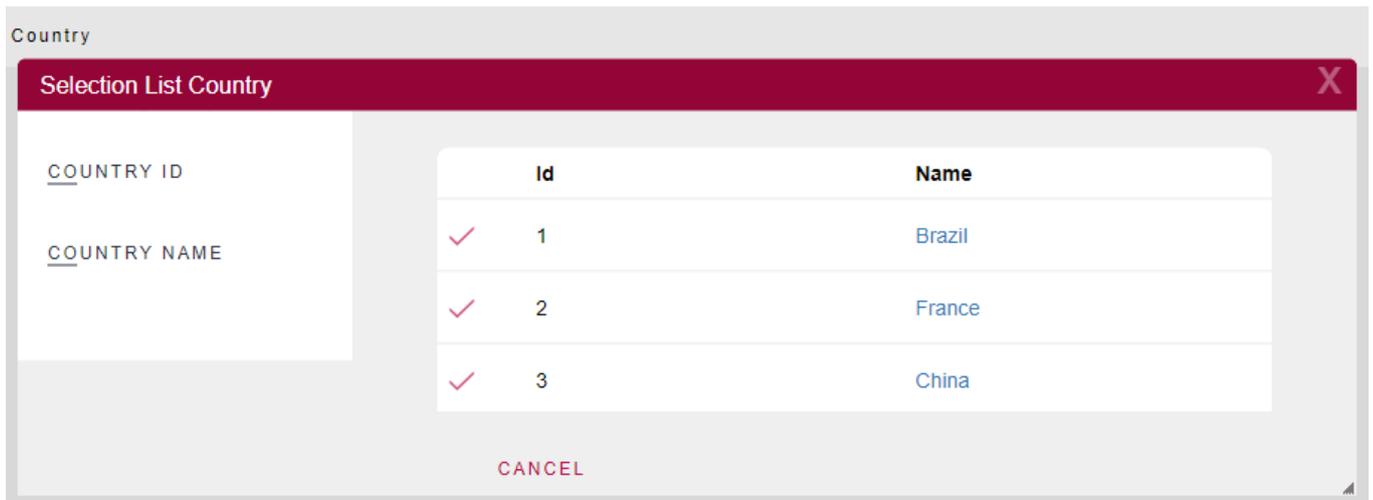
Ejecute para probar (F5) y le aparecerá el siguiente reporte:



¿Por qué en la tabla *AmusementPark*, que GeneXus informa que se debe alterar en la Base de Datos, no aparece el atributo *CountryName*? Es decir, ¿por qué la tabla física no lo contendrá, cuando sí está en la estructura de la transacción?

Después de estudiar el reporte, si estamos de acuerdo, presionamos *Reorganize* para que efectivamente se lleve a cabo eso que se informa. Se abrirá el navegador con el menú con links a los 3 programas que corresponden a cada una de las transacciones (*AmusementPark*, *Country* y *Employee*).

Ingresar como **países** a: Brasil, Francia y China. Observar que dejando el valor 0 como valor en el identificador, al grabar se le asigna automáticamente el número posterior al último asignado (efectivamente, se está autonumerando).



Ingresar como parque de diversiones: “Beto Carrero World”, que está en Brasil. Si no recuerda el identificador de Brasil en el sistema, ¿cómo ingresa el país? Se le ofrece un ícono con una flecha al lado de *CountryId*, para abrir una “Lista de selección” de países, creada automáticamente por GeneXus. Esto es porque *CountryId* tiene el rol de llave foránea (foreign key) en esta transacción (es decir, está “apuntando” a otra tabla).

DATOS RELACIONADOS: ¿CÓMO SE MANTIENE LA INTEGRIDAD?

AmusementPark y *Country* están relacionados. Al colocar *CountryId* en la estructura de *AmusementPark*, por tener exactamente el mismo nombre que el atributo que es llave primaria en la transacción *Country*, GeneXus entiende que en *AmusementPark* el atributo *CountryId* es llave foránea y mantiene automáticamente la integridad de la información. Así, por ejemplo:

- Intente ingresar un parque de diversiones con un Id de país que no exista. ¿Le permite grabar ese parque?
- Elija un parque previamente ingresado (por ejemplo, ‘Beto Carrero World’) y cambie el país, por uno que no exista. ¿Pudo grabar la modificación?
- Intente eliminar un país (usando la transacción *Country*) que tenga algún parque asociado (por ejemplo, Brasil). ¿Se lo permite?

Conclusión: los programas correspondientes a las transacciones aseguran la integridad de los datos.

Regrese a GeneXus y cambie el tipo de dato asociado al atributo *AmusementParkId* por el dominio *Id*. Ejecute la aplicación. ¿Qué marca el Análisis de Impacto? Presione **Reorganize**.

TRANSACCIÓN 'SHOW'

Nos informan que los parques de diversiones pueden brindar diferentes shows (musicales, de magia, didácticos, etc) a sus asistentes, en determinada fecha y horario. De los shows interesa conocer su nombre y una imagen representativa. Un mismo show puede ser contratado por varios parques de diversiones.

TRANSACCIÓN 'GAME'

Como nos fue solicitado inicialmente, el sistema debe brindar la posibilidad de ingresar los juegos disponibles en cada parque, por lo que debemos crear una transacción que contenga su nombre y el parque de diversiones al que pertenece.

TRANSACCIÓN 'CATEGORY'

Nos falta completar la información de la transacción *Game*. Los empleados describieron que también registran de cada juego la **categoría** (infantil, radical, recreativo, etc.) a la que pertenece. Así que necesitaremos crear una transacción para registrar esta información, y agregar la categoría a la transacción *Game*.

Pero, además, nos han informado que no es obligatorio registrar indefectiblemente la categoría a la que pertenece un juego dado que se está manipulando. Se puede dejar vacía. Si sabemos que GeneXus controla automáticamente la integridad, ¿cómo lo conseguimos?

Name	Type	Description	Formula	Nullable
Game	Game	Game		
GameId	Id	Game Id		No
GameName	Name	Game Name		No
AmusementParkId	Id	Amusement Park Id		No
AmusementParkName	Character(40)	Amusement Park Name		
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		

Name	Type
Category	Category
CategoryId	Id
CategoryName	Name

Para terminar la definición de la transacción *Game*, agreguemos el dato que nos está faltando: la foto.

Para ello, cree el atributo *GamePhoto* de tipo de datos *Image*.

Pídale a GeneXus que construya la aplicación, así puede probarla en ejecución (F5).

Observe lo que le informa el reporte de Análisis de Impacto. Deberán crearse las tablas *Category* y *Game* (no se preocupe en entender por qué requiere almacenar dos valores por imagen).

Reorganice y ejecute.

Ingrese categorías (como infantil y radical) y juegos (como montañas rusas y carruseles).

Observe que en este caso puede dejar la categoría vacía (debido a que configuró la propiedad **Nullable** en **Yes** en la estructura de la transacción).

Sin embargo, si intenta poner como valor de *CategoryId* para el juego un valor inexistente, no le dejará grabar.

Recuerde actualizar los cambios en GeneXus Server.

Application Name
by GeneXus

Recents Category — Game

Game

	SELECT
Id	0
Name	Big Tower
Amusement Park Id	1 ↑
Amusement Park Name	Beto Carrero World
Category Id	2 ↑
Category Name	Radical
Photo	

TRANSACCIONES “EMPLOYEE” Y “AMUSEMENTPARK”, RELACIONADAS

Como nos dijeron al comienzo del desarrollo de la aplicación, el sistema gestionará los parques de diversiones y sus empleados. Por lo tanto, necesitamos vincular a los empleados que son registrados con el parque en el cual trabajan.

Para eso, vayamos a la transacción *Employee* y agreguemos los atributos *AmusementParkId* y *AmusementParkName*.

Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Character(20)
EmployeeLastName	Character(20)
EmployeeAddress	Address, GeneXus
EmployeePhone	Phone, GeneXus
EmployeeEmail	Email, GeneXus
AmusementParkId	Id
AmusementParkName	Character(40)

Nos han informado que no es obligatorio registrar en el momento el parque en donde trabaja el empleado, es decir, se puede dejar vacío. ¿Qué debemos hacer?

Ejecute para probar (F5) y le aparecerá el siguiente reporte indicándonos que el atributo AmusementParkId ahora permite dejar un valor no especificado:

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs. Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern:

Employee

Table Employee specification

Table name: Employee

Employee needs conversion

Table Structure

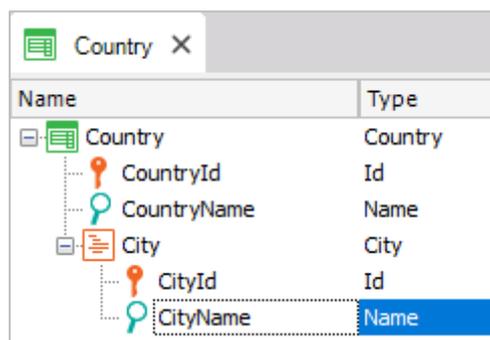
Attribute	Definition	Previous values	Takes value from
EmployeeId	Numeric (4), Not null, Autonumber		Employee. EmployeeId
EmployeeName	Character (20), Not null, NLS		Employee. EmployeeName
EmployeeLastName	Character (20), Not null, NLS		Employee. EmployeeLastName
EmployeeAddress	Varchar (1024), Not null, NLS		Employee. EmployeeAddress
EmployeePhone	Character (20), Not null, NLS		Employee. EmployeePhone
EmployeeEmail	Varchar (100), Not null, NLS		Employee. EmployeeEmail
New AmusementParkId	Numeric (4)		Null

Indexes

0 Errors 0 Warnings 1 Success

AGREGUEMOS LAS CIUDADES A LA TRANSACCIÓN 'COUNTRY'

Además de los países, necesitamos registrar la información de sus ciudades. Por lo tanto, debemos agregar un segundo nivel a la transacción *Country*, con el identificador y el nombre de ciudad.



Recuerde que:

- Posicionado en el atributo *CountryName*, con botón derecho, **Insert Level**, agrega el subnivel.
- Una vez que le dé un nombre al nuevo nivel, digitando comillas (") en lugar de punto, el atributo que defina se inicializará con el nombre del nivel.
- Las ciudades se identificarán por su propio Id en combinación con el del país. Es decir, no podrá identificar a una ciudad sin brindar antes la información del país del que se trata. Así, podría haber una ciudad 1 Rosario tanto para Uruguay como para Argentina:
 País: 1 (Uruguay) – Ciudad: 1 (Rosario)
 País: 2 (Argentina) – Ciudad: 1 (Rosario)

- O incluso podría ser que Rosario para Argentina se identificara con otro número:
País: 2 (Argentina) – Ciudad: 4 (Rosario)

Reorganice y ejecute (F5).

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs.
Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern: CountryCity

Table CountryCity specification

Table name: CountryCity

CountryCity is new

Warnings

rgz0009 AutoNumber=True ignored. Attribute CityId is not table CountryCity's primary key.

Table Structure

Attribute	Definition	Previous values	Takes value from
CountryId	Numeric (4), Not null		
CityId	Numeric (4), Not null		
CityName	Character (50), Not null, NLS		

Indexes

Name	Definition	Composition
ICOUNTRYCITY	primary key Clustered	CountryId CityId

0 Errors 1 Warnings 0 Success

Observe que el Listado de Navegación le informará que:

- La propiedad **Autonumber** para el caso de **CityId** será ignorada. Esto significa que en ejecución el usuario deberá ingresar manualmente los identificadores de ciudad. La explicación es que la propiedad **Autonumber** solamente autonumera llaves primarias simples y en este caso **CityId** es el segundo componente de una llave compuesta.
- Se creará una nueva tabla **CountryCity** para almacenar la información correspondiente a las ciudades.

Ingrese ciudades para los países que ya tenía registrados.

TRANSACCIÓN “AMUSEMENTPARK”: AGREGUEMOS LA CIUDAD.

En la transacción *AmusementPark* agreguemos la ciudad del país a la que el parque pertenece. ¿Qué debe hacer si la empresa nos informa que ese valor puede no ser conocido o relevante para un parque dado en un momento dado?

Construya la aplicación y pruébela (F5 y Reorganize).

Actualice los cambios en GeneXus Server.

AGREGUEMOS COMPORTAMIENTO A LAS TRANSACCIONES (RULES)

Después de probar con nosotros la aplicación que venimos desarrollando, en la empresa nos cuentan que para los empleados hay algún comportamiento específico que debemos hacer cumplir a la hora de manipular la información a través del programa (transacción *Employee*).

¿Cuál es este comportamiento?

Nos dicen:

- “El sistema no debe permitir ingresar empleados sin nombre, ni sin apellido”.
- “Debe advertirse al usuario si está dejando el teléfono sin asignar, por si fue un descuido”.
- “Se debe registrar la fecha de ingreso del empleado al sistema (**EmployeeAddedDate**) y se debe proponer como valor predeterminado para ese atributo, la fecha de hoy”.

Especifique ese comportamiento y pruébelo (F5 y Reorganize).

Recuerde que:

- Las reglas finalizan con punto y coma “;”.
 - El método **IsEmpty()** aplicado a un atributo devuelve True cuando el atributo está vacío y False en caso contrario.
 - La variable **&Today** es del sistema y tiene cargado el valor de la fecha del día.
- Para escribir una variable dentro de la pantalla **Rules**, cuando digita “&” se le despliegan todas las variables definidas hasta el momento para que seleccione la que necesita. La otra posibilidad es utilizar **Insert / Variable**.

Pruebe ingresar un nuevo empleado dejando vacío el nombre. ¿Le permite grabar o pasar al siguiente campo?

Ídem con el apellido. ¿Sucede lo mismo con el teléfono?

Si luego le informan que la fecha de ingreso al sistema no debería ser manipulada por el usuario, sino únicamente visualizada, ¿cómo establece este comportamiento?

Especifíquelo y pruébelo en ejecución.

En la Transacción Country agregue una regla que permita autonumerar el atributo CityId.

Especifique y pruébelo en ejecución.

PATTERNS: MEJORANDO LA INTERFAZ PARA TRABAJAR CON LA INFORMACIÓN

Al mostrarle al cliente lo realizado hasta ahora, nos dice que quisiera poder manipular la información de países, parques de diversiones, empleados, shows, categorías y juegos de un modo más potente y vistoso (que ofrezca consulta, posibilidad de filtrar, así como insertar, modificar y eliminar datos, etc.).

Para ello deberá aplicar el patrón *Work With for Web* a las transacciones. Pruébalo y véalo en ejecución.

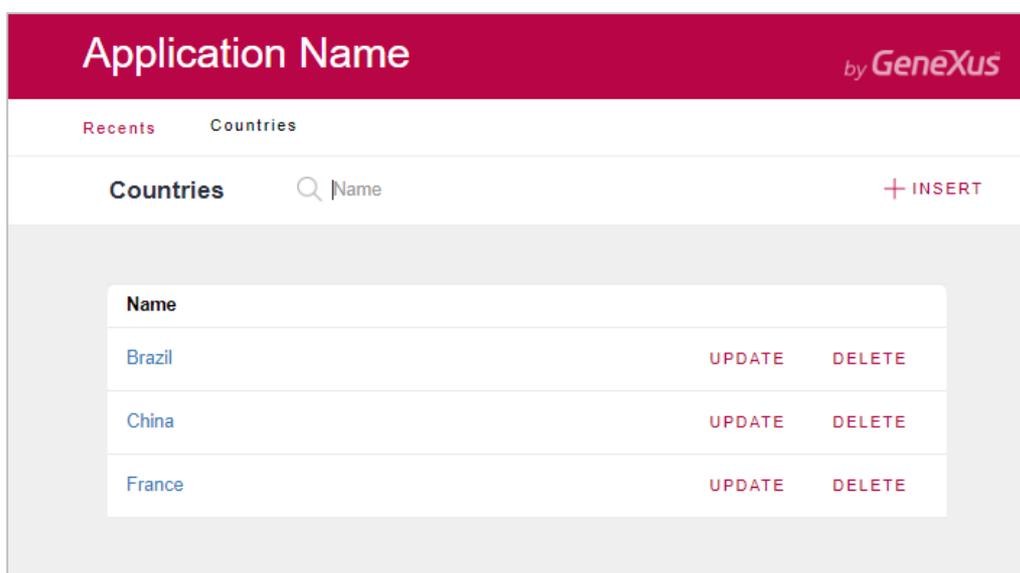
Observar que:

- existe un *Work With* para Smart Devices, también. Pero el que usted deberá aplicar es el que corresponde a la aplicación web que está construyendo.
- GeneXus creará automáticamente varios objetos por transacción, para implementar el "Trabajar con" esa entidad.

¿Por qué no aparecen más en el *Developer Menu* las transacciones *Country*, *AmusementPark*, *Employee*, *Show*, *Category* y *Game*?

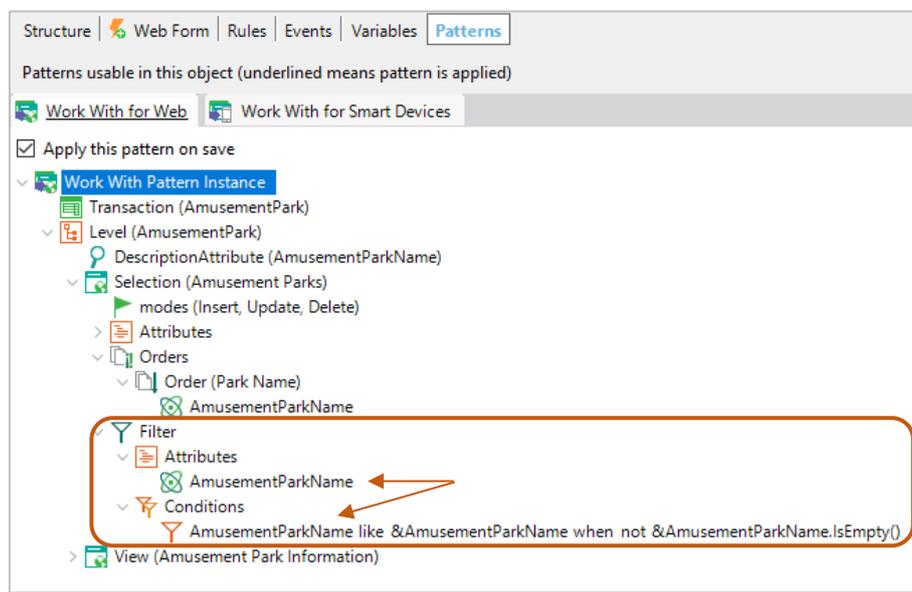
Pruebe hacer lo siguiente:

1. Ingresar un nuevo país.
2. Modificar un país existente (por ejemplo, agregándole una ciudad).
3. Eliminar un país existente.
4. Visualizar la información de un país.
5. Realizar una búsqueda por nombre de país.



6. Ingrese un par de parques de diversiones (Ej: Shangai Disney Resort, de China/Shangai, Parc Du Bocasse de Francia/Normandía, Happy Valley, de China/Beijing).
7. Filtre los parques de diversiones cuyo nombre empiece con P. ¿Y si ahora quiere poder visualizar todos los parques de China? No está incluida esta posibilidad, por lo que deberemos personalizar el pattern *Work With* de esta transacción, para agregárselo. Hágalo en GeneXus y pruebe en ejecución.

Sugerencia: Observe cómo está especificado el filtro que sí existe, por nombre del parque de diversiones. Intente definir el filtro solicitado de igual forma.



8. Ahora quite los identificadores de país y ciudad de la pantalla del *Work With* y pruébelo en ejecución.

Application Name by GeneXus

Recents Amusement Parks

Amusement Parks + INSERT

Park Id	Park Name	Park Website	Park Address	Park Photo	Country Id	Country Name	City Id	City Name		
1	Beto Carrero World	www.betocarrero.com.br	Rua Inácio Francisco de Souza, 1597		1	Brazil	2	São Paulo	UPDATE	DELETE
4	Happy Valley	bj.happyvalley.cn	Beijing Olympic Tower		3	China	1	Beijing	UPDATE	DELETE
3	Parc Du Bocasse	www.parcubocasse.fr	226 Route de Clères, 76690		2	France	2	Normandia	UPDATE	DELETE
2	Shangai Disney Resor	www.shangaidisneyresort.com	Shanghai Disney Resort, Pudong, Xangai		3	China	2	Shangai	UPDATE	DELETE

9. Si ahora quiere brindar la posibilidad de que el usuario elija si quiere ver los parques de diversiones ordenado por nombre del parque o por nombre de país, impleméntelo y pruebe.

Application Name by GeneXus

Recents Amusement Parks

HIDE FILTERS Amusement Parks + INSERT

Ordered By : Park Name

Park Name

Country Name

Park Id	Park Name	Park Website	Park Address	Park Photo	Country Name	City Name		
1	Beto Carrero World	www.betocarrero.com.br	Rua Inácio Francisco de Souza, 1597		Brazil	São Paulo	UPDATE	DELETE
4	Happy Valley	bj.happyvalley.cn	Beijing Olympic Tower		China	Beijing	UPDATE	DELETE
3	Parc Du Bocasse	www.parcubocasse.fr	226 Route de Clères, 76690		France	Normandia	UPDATE	DELETE
2	Shangai Disney Resor	www.shangaidisneyresort.com	Shanghai Disney Resort, Pudong, Xangai		China	Shangai	UPDATE	DELETE

TRANSACCIONES “REPAIR” Y “TECHNICIAN” Y NECESIDAD DE DEFINIR SUBTIPOS

Se necesita ahora registrar los juegos que entran en estado de reparación. Cada reparación tiene un identificador, una fecha desde la cual el juego deja de estar disponible para su uso, la cantidad de días estimada para su reparación, el identificador del juego, su nombre, el técnico titular y el técnico suplente. También cada reparación tiene un costo. Para el costo cree un dominio llamado *Cost*, del tipo Numeric(8.2).

Cree una transacción para registrar a los técnicos que trabajarán en las reparaciones. Cada técnico tiene un identificador, un nombre y apellido, un teléfono, un país y una ciudad en la cual se encuentra.

¿Cómo se define que cada reparación tiene un técnico titular y otro suplente?

Recuerde

- 1) Que en la estructura de la transacción:
 - un ícono representando una flecha hacia arriba  informa que el atributo es clave foránea (Foreign Key), es decir que apunta a otra tabla.
 - Un ícono representado una flecha hacia abajo  informa que el atributo es inferido de otra tabla.
 - Un ícono representando una  indica que el atributo es un subtipo.
- 2) Sobre los grupos de subtipos:
 - Se definen de la misma manera que cualquier tipo de objeto.
 - Cada grupo de subtipos debe contener obligatoriamente un subtipo de un atributo primario (que es llave primaria de una tabla) o conjunto de atributos que forman una llave primaria.
 - En cada grupo de subtipos, hay que incluir todos los atributos subtipos que se necesiten conocer, pertenecientes a la tabla base y/o extendida de la llave primaria del grupo.

Ejecute y verifique que al intentar ingresar una reparación, se dispare un error si el técnico titular que está queriendo asignarle a la reparación no existe. Ídem para el técnico suplente.

No debe permitirse ingresar una reparación cuyo técnico titular coincida con el técnico suplente. Implemente ese comportamiento y pruébelo en ejecución.

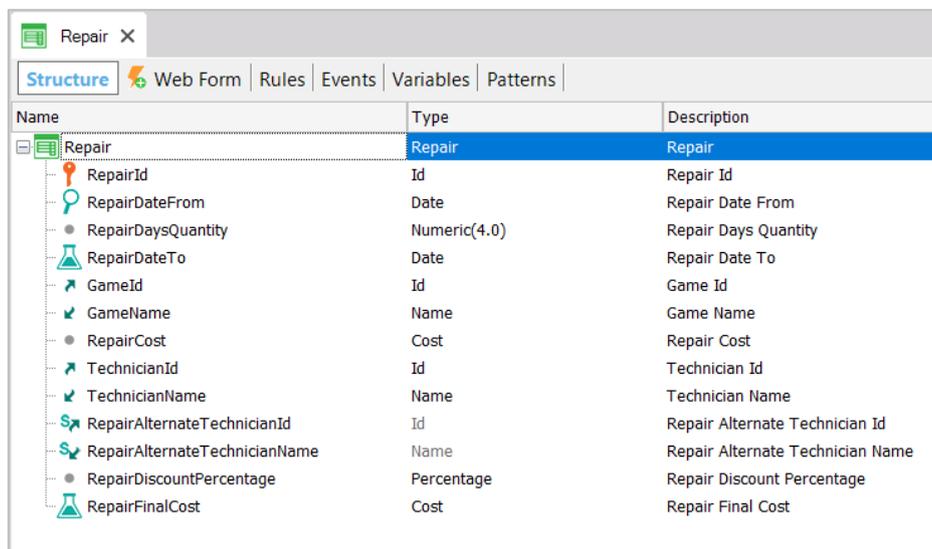
Actualice los cambios en GeneXus Server.

FÓRMULAS

Se necesita poder registrar **el descuento actual que tiene cada reparación**. Defina un nuevo atributo en la transacción *Repair* para **almacenar este dato**. Darle al nuevo atributo el nombre: *RepairDiscountPercentage* y que su tipo de datos sea un dominio *Percentage*, numérico de largo 3.

Se desea visualizar el precio final de la reparación con el descuento aplicado. Para resolver esto, defina otro atributo más, de nombre *RepairFinalCost*, que sea **fórmula global** y que calcule automáticamente el precio final de la reparación.

Agregue un nuevo campo llamado *RepairDateTo*, el cual será una suma entre la fecha de inicio de reparación y la cantidad de días que la misma llevará.



Name	Type	Description
Repair	Repair	Repair
RepairId	Id	Repair Id
RepairDateFrom	Date	Repair Date From
RepairDaysQuantity	Numeric(4.0)	Repair Days Quantity
RepairDateTo	Date	Repair Date To
GameId	Id	Game Id
GameName	Name	Game Name
RepairCost	Cost	Repair Cost
TechnicianId	Id	Technician Id
TechnicianName	Name	Technician Name
RepairAlternateTechnicianId	Id	Repair Alternate Technician Id
RepairAlternateTechnicianName	Name	Repair Alternate Technician Name
RepairDiscountPercentage	Percentage	Repair Discount Percentage
RepairFinalCost	Cost	Repair Final Cost

Presione F5, observe en el Análisis de Impacto cuál atributo se creará físicamente y cuál no, reorganice y pruebe la aplicación en funcionamiento.

CREACIÓN DE SEGUNDO NIVEL

Se desea crear un segundo nivel en la transacción *Repair* para allí guardar un detalle del tipo de problema encontrado para reparar.

Para ello, antes que nada se deberá crear un dominio llamado *KindName*, Character(1). Restrinja los valores posibles para el dominio: que sean válidos los valores “E”, “M” y “R” (editando la propiedad **Enum Values** del mismo como se ve a continuación).

Name	Description	Value
Electricity	Electricity	E
Mechanical	Mechanical	M
Replacement	Replacement	R

Cree un segundo nivel en la transacción *Repair* de nombre *Kind* para registrar el tipo de reparación. Este nivel tendrá estos tres atributos:

- *RepairKindId* – Numeric(4) (será clave en este segundo nivel)
- *RepairKindName* – Basado en dominio *KindName* (Genexus lo sugerirá automáticamente).
- *RepairKindRemarks* – Character(120), contendrá observaciones sobre el problema, un pequeño detalle del problema encontrado o la pieza a remplazar por ejemplo.

Recuerde que para definir que cierto atributo es parte de la llave primaria, debe presionar el botón derecho del mouse sobre el atributo y el menú contextual le ofrecerá la opción **Toggle Key**. En este caso, no será necesario ya que sólo el primer atributo forma parte de la clave primaria y éste ya aparece con el ícono de llave.

A efectos de conocer la cantidad de tipos de problema que implica una reparación, cree un nuevo atributo en el primer nivel de la transacción *Repair*, de nombre *RepairProblems*, Numeric(1) y defínalo como fórmula global. Este atributo deberá contar los tipos de problemas encontrados.

Una reparación podría implicar problemas de electricidad, de mecánica o que sea necesario el remplazo de alguna pieza. Podría incluso tener más de un problema del mismo tipo, dos registros de electricidad, por ejemplo. Si hubiera muchos, se puede detallar algo más con el uso del atributo de observaciones, el cual permite escribir un pequeño texto.

Se desea ver la cantidad de problemas en el formulario web y se debe controlar que se ingresen entre 1 y 3 líneas de tipos de problema.

Este control deberá realizarse cuando se termine de ingresar datos en el segundo nivel y luego de haber presionado el botón *Confirm*.

Name	Type	Description
Repair	Repair	Repair
RepairId	Id	Repair Id
RepairDateFrom	Date	Repair Date From
RepairDaysQuantity	Numeric(4.0)	Repair Days Quantity
RepairDateTo	Date	Repair Date To
GameId	Id	Game Id
GameName	Name	Game Name
RepairCost	Cost	Repair Cost
TechnicianId	Id	Technician Id
TechnicianName	Name	Technician Name
RepairAlternateTechnicianId	Id	Repair Alternate Technician Id
RepairAlternateTechnicianName	Name	Repair Alternate Technician Name
RepairDiscountPercentage	Percentage	Repair Discount Percentage
RepairFinalCost	Cost	Repair Final Cost
RepairProblems	Numeric(1.0)	Repair Problems
Kind	Kind	Kind
RepairKindId	Numeric(1.0)	Repair Kind Id
RepairKindName	KindName	Repair Kind Name
RepairKindRemarks	Character(120)	Repair Kind Remarks

El valor de *RepairKindId* se va ingresando manualmente. Lo normal sería ir dándole valores a partir de 1. Realice lo correspondiente para autonumerar este atributo (recuerde que **no es posible autonumerarlo** utilizando la propiedad *Autonumber*).

Actualice los cambios en GeneXus Server.

ÍNDICES

Diríjase al menú *View* → *Tables* y coloque los controles necesarios de manera que no se pueda repetir el nombre de un país ni de un parque de diversiones.

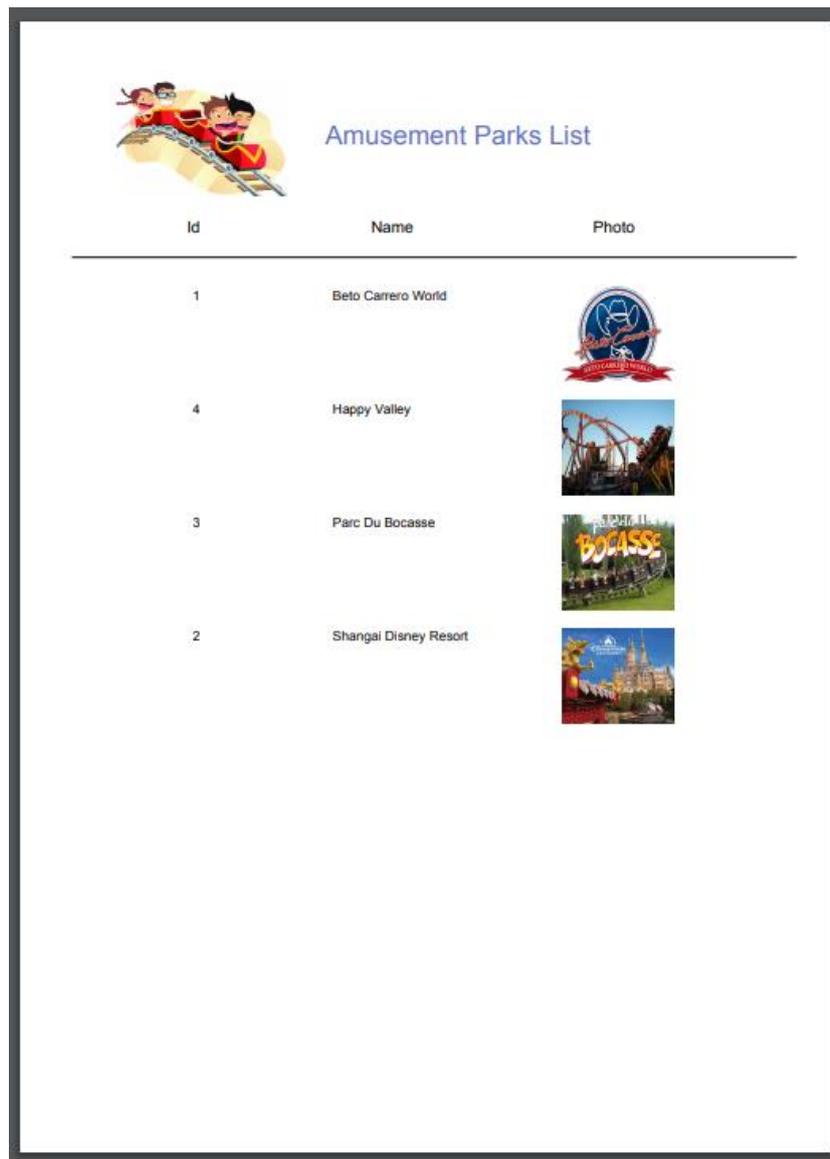
¿En qué sección se especifican estos controles?

¿Qué mensaje aparece en el navegador si intenta agregar un país/parque de diversiones cuyo nombre se repite?

LISTADOS PDF

Ahora supongamos que como parte de la aplicación, deberá implementarse la posibilidad de que a pedido del usuario, se le despliegan listados *PDF* con la información requerida. Por ejemplo, suponga que se necesita un listado que muestre en orden alfabético, los parques de diversiones almacenados en la base de datos.

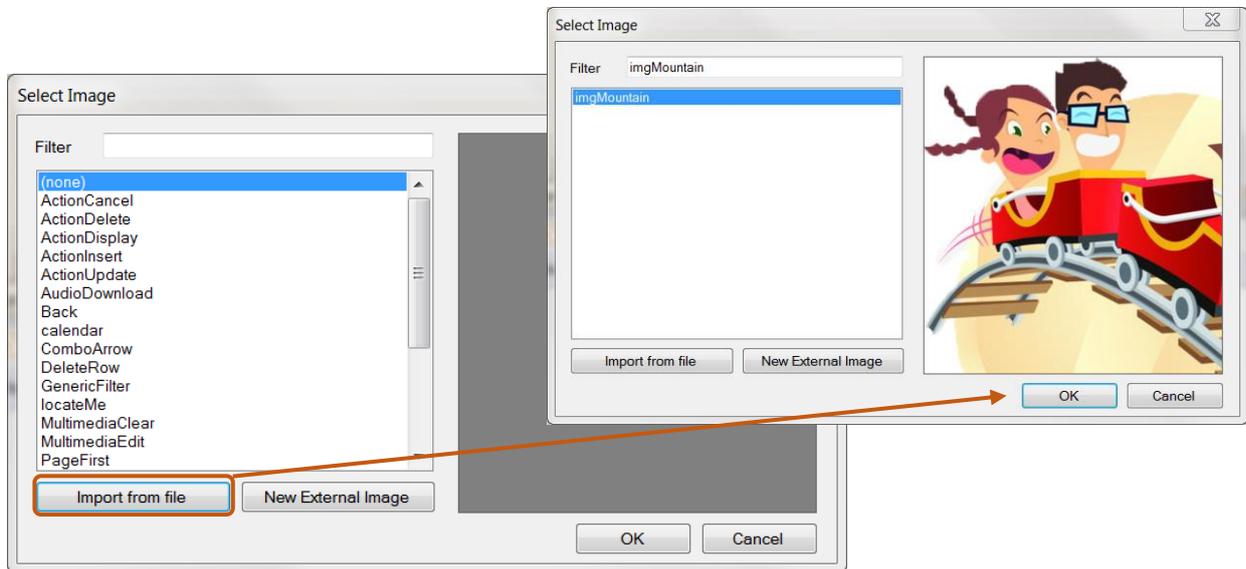
Se sabe que debe lucir más o menos como sigue:



Id	Name	Photo
1	Beto Carrero World	
4	Happy Valley	
3	Parc Du Bocasse	
2	Shangai Disney Resort	

Sugerencia para poner una imagen en el título: Utilizar el control *Image* presente en la *Toolbox* para desplegar una imagen junto al título del listado.

Dicha imagen deberá ser integrada a la base de conocimiento. Para eso seleccionar la opción *Import from File*, buscar la imagen y asignarle un nombre.



Implementelo en GeneXus.

Recordar que para poder visualizar directamente desde el browser un listado, el mismo debe ser generado como PDF. Para esto debe configurar las siguientes propiedades del objeto procedimiento:

- Main program = 'True'
- Call Protocol = HTTP

Y la siguiente regla:

- Output_file('nombre-archivo.pdf' , 'PDF')

Para ejecutar el listado, sobre la pestaña del objeto, botón derecho / **Run with this only**

¿Reparó en lo que le informa el listado de navegación del procedimiento?

¿Y si ahora se necesita que el listado salga ordenado por nombre de país? Implementelo, observe lo informado en el listado de navegación y pruébelo.

¿Y si ahora necesita solamente listar los parques de diversiones de China? Pruébelo (observando el listado de navegación).



Amusement Parks List

Id	Name	Country	Photo
2	Shangai Disney Resort	China	
4	Happy Valley	China	

En cada caso, deduzca qué tabla de la base de datos se está recorriendo para realizar la consulta del *for each*.

También se necesita un listado como el que sigue (que muestre cada categoría, y por cada una de ellas, sus juegos). Impleméntelo y pruebe lo realizado.

Categories and their games

Category: Childish

Games	Name	Park Name	Photo
	Baby Elefant	Beto Carrero World	
	Carousel	Parc Du Bocasse	

Category: Radical

Games	Name	Park Name	Photo
	Big Tower	Beto Carrero World	
	Fire Whip	Beto Carrero World	
	Hulk	Shangai Disney Resort	

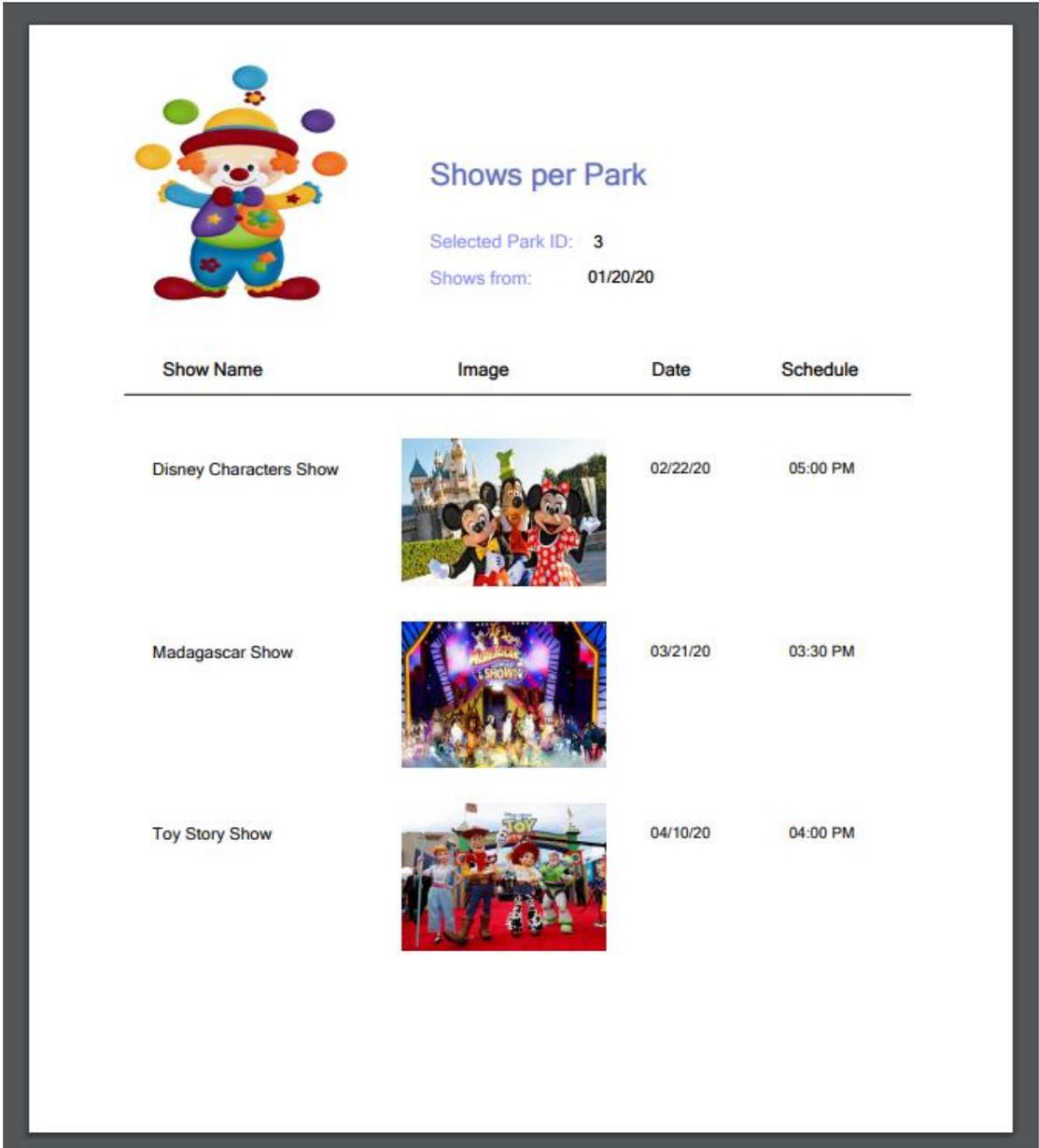
Agregue una nueva categoría al sistema, por ejemplo *Aquatic* (acuáticos). Vuelva a ejecutar el listado anterior. ¿Salió listada la categoría?

Modifique el listado anterior para que no salgan listadas categorías que no tengan juegos relacionados.

¿Qué modificaciones encontró en el listado de navegación?

Respuesta: La aparición de la palabra *break* que indica que se está realizando un corte de control, mostrándose además que la tabla base del *for each* externo es igual a la del *for each* anidado.

Otro listado necesario es el de los shows que se ofrecen a partir de una fecha indicada por el usuario, en determinado parque de diversiones también seleccionado por el usuario.



Shows per Park

Selected Park ID: 3
Shows from: 01/20/20

Show Name	Image	Date	Schedule
Disney Characters Show		02/22/20	05:00 PM
Madagascar Show		03/21/20	03:30 PM
Toy Story Show		04/10/20	04:00 PM

Otra solicitud de la empresa es un listado que muestre todos los nombres de países y para cada país, la cantidad de parques de diversiones que ofrece:



Countries list

Country	Quantity
Brazil	2
France	1
China	2
United States	3

Y nos piden otro listado que muestre todos los países que tienen más de 2 parques para visitar:



Countries list

Country	Quantity
United States	3

PASAJE DE PARÁMETROS

Muchas veces necesitamos que un objeto reciba parámetros para que, en base a los mismos, ejecute su lógica. Por ejemplo, una cosa es realizar un listado *PDF* de todos los parques de diversiones de la base de datos, y otra es realizar un listado de aquellos cuyo nombre se encuentra dentro de un rango dado.

LISTADO DE PARQUES EN UN RANGO DETERMINADO

Grabe con otro nombre el procedimiento que había creado anteriormente para listar los parques de diversiones, y modifíquelo para que ahora solamente liste aquellos cuyo nombre se encuentra dentro de un rango recibido por parámetro (el valor inicial y el final del rango serán los parámetros recibidos).

Implemente una pantalla que pida los valores de ese rango al usuario, e invoque a este objeto, pasándole esos valores por parámetro. Pruebe en ejecución.

Recuerde que:

- Si define una variable basada en (**based on**) un atributo, quedará ligada al tipo de datos del atributo, es decir, si éste se modifica, el de la variable también, acorde al cambio.
- Las variables utilizadas para realizar una invocación en el objeto que llama y las utilizadas para declarar los parámetros que se reciben en el objeto llamado, no tienen por qué coincidir en nombre, pero sí deben tener tipos de datos compatibles.

Sugerencias:

- Definir la regla Parm correspondiente en el procedimiento *AmusementParksFromTo* para que reciba los valores a consultar.
- Crear un Web Panel con dos variables para que el usuario final pueda ingresar los valores a consultar y sean luego enviados por parámetro al procedimiento.

BUSINESS COMPONENTS

Realizaremos algunas operaciones sobre la base de datos, a través de *Business Components*.

AUMENTO DE PRECIO DE LAS REPARACIONES

La empresa se reserva el derecho de incrementar los precios de las reparaciones en un porcentaje determinado a partir de cierto momento. A los efectos de realizar una grabación masiva probaremos esta funcionalidad. Para ello, deberemos implementar una **pantalla** que permita al usuario

especificar ese porcentaje de aumento, y dar la orden de aplicarlo a todas las reparaciones de la base de datos. Implementélo y pruebe.

Recuerde que:

- El objeto Web Panel le permite implementar pantallas flexibles para entrada y salida de información.
- Para el ingreso de información (que el usuario pueda ingresar valores a la pantalla), puede insertar desde la *Toolbar* un control *Attribute/Variable* en el Form y asignarle una variable.
- Si quiere editar barras de menú, posicionándose en GeneXus arriba, sobre la barra, con botón derecho podrá insertar, por ejemplo, la barra *Formatting*.
- Para que el Web Panel tome alguna acción determinada, pueden insertarse botones y programar el "evento" asociado.
- Los *Business Components* son tipos de datos que se crean al configurar la propiedad de nombre *Business Component* del objeto transacción con valor *Yes*. Al hacerlo, para insertar, modificar o eliminar registros de las tablas correspondientes, podrá utilizarse, además de la transacción, una variable de tipo de datos *Business Component* en cualquier otro objeto (por ejemplo, un Web Panel) y realizar las mismas operaciones a través de los métodos *Load()*, *Insert()*, *Update()*, *Save()* y *Delete()*.
- Para que las operaciones realizadas a través del *Business Component* queden efectuadas de manera permanente, deberá ejecutar a continuación el comando *Commit*.
- Si a un valor X se le quiere incrementar un 20%, alcanza con hacer $X = X * (1 + 20/100) = X * 1,20$

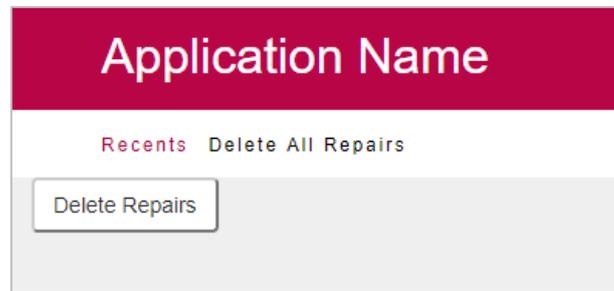
Sugerencias:

- Cree un Web Panel con una variable &percentage, de tipo de datos: Numeric(3,0), y arrastre un control de tipo Button sobre el Form.
- En el evento asociado al botón programe la lógica que se desea que se ejecute cuando el usuario presione dicho botón.
- Recuerde que se desea recorrer todas las reparaciones, y para cada una de ellas, utilizando el concepto de Business Component, modificar su precio aumentándolo en el porcentaje indicado por el usuario en la variable de pantalla.
- Recuerde también que como las operaciones de grabación o eliminación de la base de datos (a través de los métodos *Insert()*, *Update()*, *Save()* y *Delete()*) pueden provocar errores, es importante saber qué ocurrió. Para ello tenemos el método *Success()* que devolverá True si no hubo errores, y False en caso contrario.
- Para indicar que "lo hecho quede permanente" se debe ejecutar el comando Commit. Si por el contrario se quiere deshacer, se debe ejecutar el comando Rollback.

Actualice los cambios en GeneXus Server.

PANTALLA PARA ELIMINACIÓN DE TODAS LAS REPARACIONES

Grabe el Web Panel anterior con otro nombre (para ello alcanza con posicionarse sobre la pestaña y con botón derecho, hacer *Save as*) y modifique el Form para que sólo contenga un botón con el texto *Delete Repairs*, de modo que en ejecución este Web Panel se vea así:



Cuando el usuario presione el botón, deberán eliminarse todas las reparaciones de la base de datos.

¿Qué debe modificar del evento *Confirm* que tenía programado?

Nota: Si se posiciona sobre el botón y ve sus propiedades, en la de nombre *Caption* puede modificar el texto del botón.

Sugerencias:

- Utilice el método *Delete()* que aplica a variables de tipo Business Component (simples o colecciones).
- Recuerde que utilizando el comando *Msg*, es posible desplegar un mensaje en el Form del Web Panel.

Actualice los cambios en GeneXus Server.

PROCEDIMIENTOS PARA ACTUALIZAR REGISTROS

AUMENTO DE PRECIOS DE LAS REPARACIONES

Suponga que las reparaciones a las que debe aumentar el precio en un porcentaje dado son miles. Sabiendo que el aumento de precio es un procedimiento sencillo que no hará fallar la integridad de ningún modo, ejercite resolverlo con un procedimiento, sin utilizar *Business Components*.

Recuerde que:

- Con el comando *for each* dentro de un procedimiento, puede actualizar los atributos de su tabla extendida a través de simples asignaciones.
- La actualización "directa" a través de procedimientos no controla la integridad de la información.
- Todo objeto debe declarar los parámetros que recibe y los parámetros que devuelve. Si no los declara, ni recibirá ni devolverá valores.
- Los parámetros se declaran a través de la regla **parm**.
- Las variables son locales al objeto donde se utilizan. Esto significa que si quiero recibir un valor como parámetro en una variable &X, debo declararla en el objeto.

ELIMINACIÓN DE TODAS LAS REPARACIONES

¿Y si ahora quisiera eliminar todas las reparaciones, tal como se hizo anteriormente en el práctico, pero esta vez a través de un procedimiento?

Sugerencia:

- Crear un procedimiento *RepairsDeletion* que no reciba parámetros, y que se encargue de recorrer y eliminar todos los registros de reparaciones. Utilice el comando Delete.
- Haga un Save as del Web Panel que tenía implementado y programe la eliminación a través de Business Component.

¿Y si ahora quiere borrar toda la información de la base de datos?

Sugerencia

- Crear un procedimiento *DeleteAll* que se encargue de recorrer y eliminar todos los registros de las distintas tablas asociadas a las transacciones definidas.

Recuerde que los procedimientos no validan la consistencia de los datos, pero la base de datos sí lo hace. Es decir, la base de datos valida la consistencia de los datos interrelacionados, por lo tanto, el orden en el que intente eliminar los datos es importante.

Por ejemplo, si intenta eliminar los países antes que los parques, la base de datos lo impedirá, el programa cancelará y no resultará amigable para el usuario.

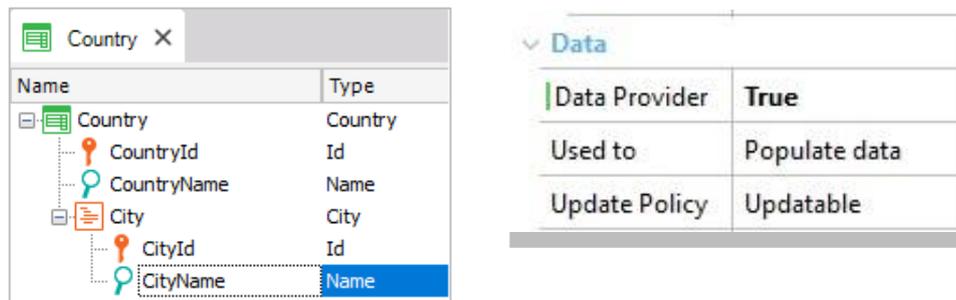
INICIALIZACIÓN DE LA INFORMACIÓN DE LA BASE DE DATOS [OPCIONAL]

Cuando la aplicación que usted está desarrollando se ponga en producción (es decir, empiece a ser utilizada por la empresa) deberán cargarse todos los datos de países, parques, shows, categorías, juegos, técnicos y reparaciones. Utilice las facilidades que el objeto *Transacción* ofrece para esto e inicialice esas tablas con información que nos ha brindado la empresa y pruébelo.

Tener en cuenta que:

- Las transacciones cuentan con una propiedad **Data Provider** que nos permite que dispongamos de un Data Provider asociado a la transacción, de modo que al crearse la tabla en la base de datos, se inicialice con los valores que definamos en dicho Data Provider. Para eso deberá configurar también la propiedad **Used to** y **Update Policy**.
- Hay que insertar las imágenes en la KB para poder utilizarlas. Una forma de hacerlo es en la ventana *KB Explorer* ir al nodo *Customization* y eligiendo *Images*, donde se listan todas las imágenes actualmente almacenadas en la KB, insertar una nueva (desde un archivo) y darle un nombre.

Sugerencia: A modo de probar diferentes opciones, poblar los países y las categorías utilizando la propiedad Data Provider de las transacciones.



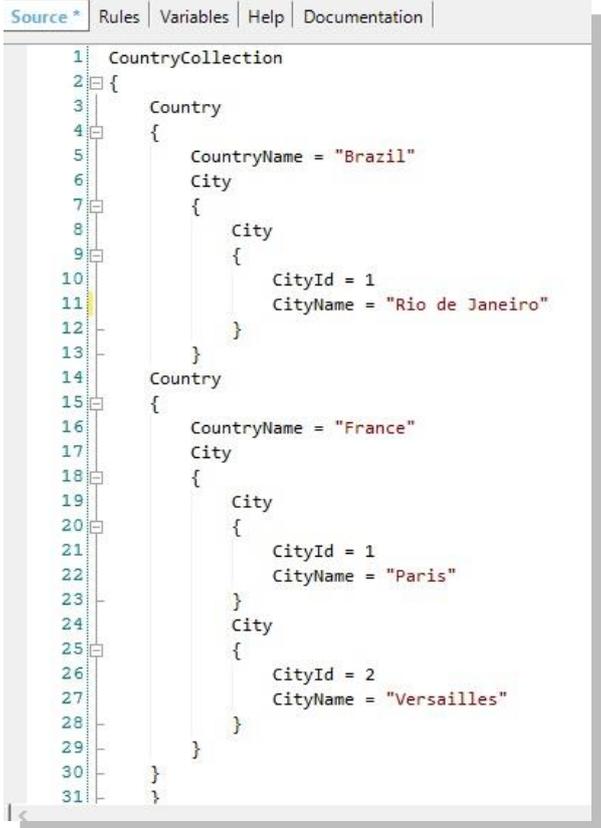
Recordar: El Data Provider asociado a la transacción se volverá a disparar:

- Cada vez que se reorganice la tabla correspondiente.
- Cada vez que se edite el contenido del Data Provider.

Por lo tanto, es necesario controlar que la información no se duplique. Esto se puede lograr:

- Si la llave primaria es autonumerada, se puede definir un índice único sobre el atributo descriptor.

- Dejando la llave primaria sin autonumerar. De esta forma su valor se asignará en forma fija al definir el Data Provider y no se duplicará.
- Cargue los datos deseados en Country_DataProvider (puede verlo bajo la transacción Country en la ventana KB Explorer). Al momento de crearse, la tabla se cargará con los datos indicados.



```
1 CountryCollection
2 {
3   Country
4   {
5     CountryName = "Brazil"
6     City
7     {
8       City
9       {
10        CityId = 1
11        CityName = "Rio de Janeiro"
12      }
13    }
14   }
15   Country
16   {
17     CountryName = "France"
18     City
19     {
20       City
21       {
22        CityId = 1
23        CityName = "Paris"
24      }
25       City
26       {
27        CityId = 2
28        CityName = "Versailles"
29      }
30     }
31   }
```

- En forma análoga cargue las categorías.
- Actualice los cambios en GeneXus Server.

WEB PANELS

Se solicita una página que muestre todos los países y para cada uno, la cantidad de parques de diversiones que ofrece visitar.

Recuerde que el **evento Load en un Web Panel con tabla base que tiene un Grid**, se ejecuta justo antes de cargar cada línea en el Grid. Este evento es adecuado para asignar a la variable el cálculo que devuelve la cuenta de parques de cada país que se está navegando y a punto de cargar en una línea del Grid.

Agregue ahora al Web Panel definido, dos variables (*&CountryNameFrom* y *&CountryNameTo*) y defina las condiciones necesarias para filtrar los países incluidos en dicho rango.

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
Amusement Park Photo Amusement Park Name  Beto Carrero World	Amusement Park Photo Amusement Park Name  Parc Du Bocasse
China Amusement Park Photo Amusement Park Name  Shanghai Disney Resor  Happy Valley	United States Amusement Park Photo Amusement Park Name  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?

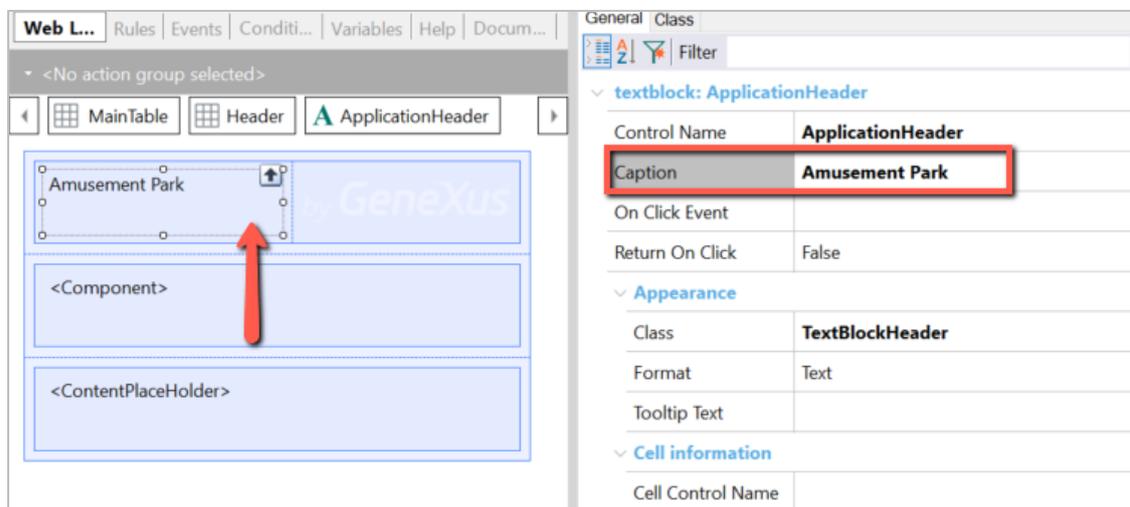
TIPOS DE WEB PANELS

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.

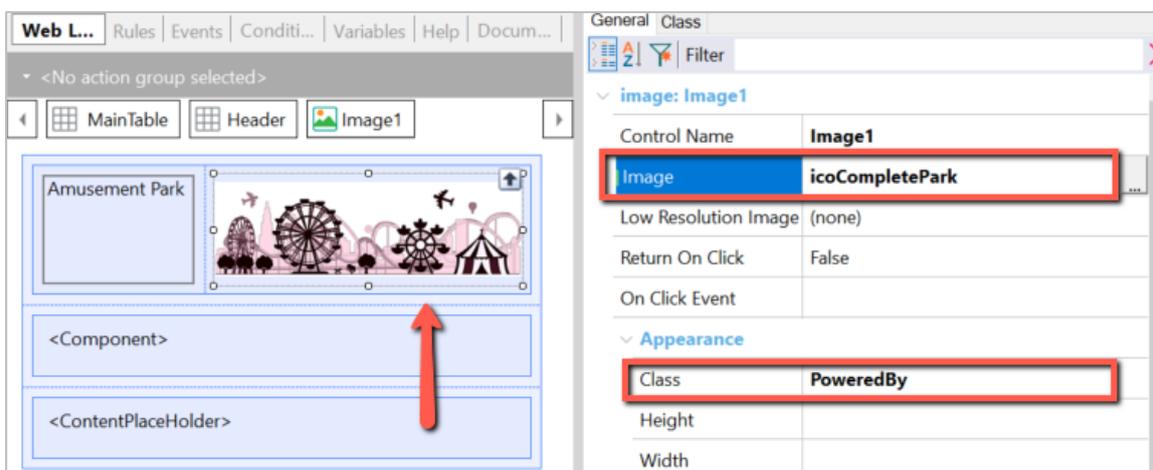
Observe que cuenta con tres secciones:

- 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.

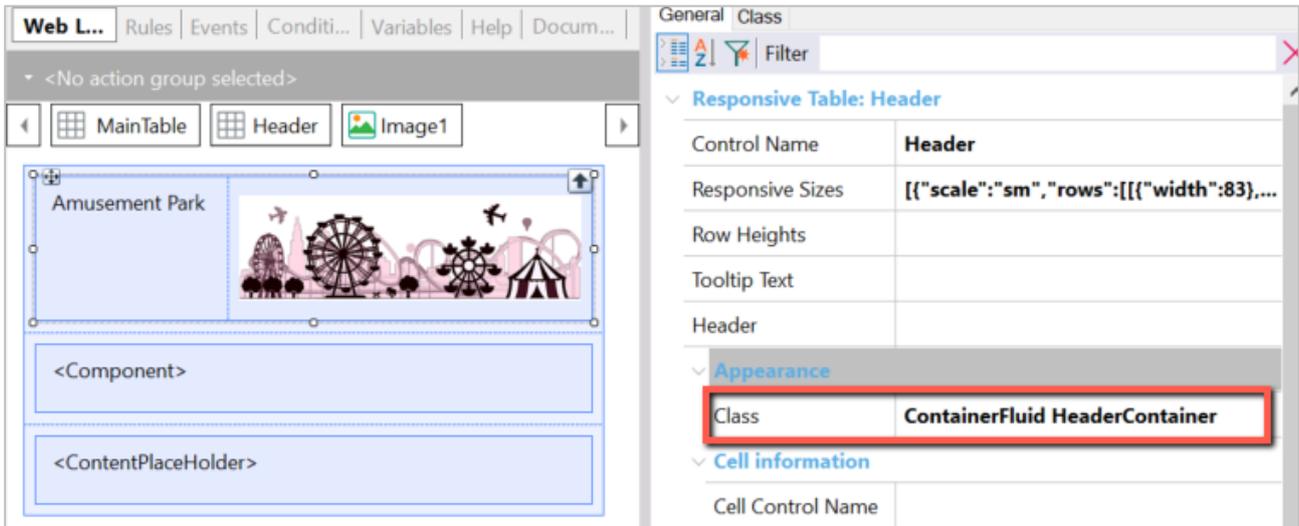
En el *Header*, seleccione el control *ApplicationHeader* y cambie la propiedad Caption por el nombre de la aplicación: Amusement Park:



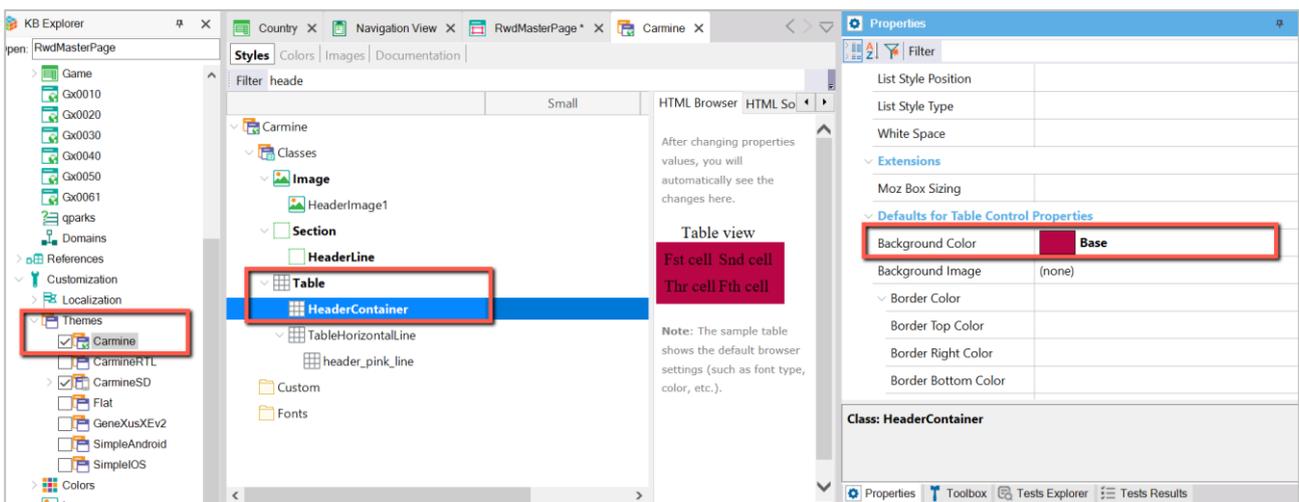
Seleccione el control *Image1* y en la propiedad Image seleccione una imagen como logotipo de la empresa. Verifique que tenga la clase *PoweredBy* aplicada:



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:



Amusement Park

Recents wp Countries And P... — Employee Detail — Employee — Employees — Amusement Park — Amusement Parks

X HIDE FILTERS Amusement Parks Q Park Name + AGREGAR GAMES

Ordered By Park Name

COUNTRY NAME

Park Id	Park Name	Web Site	Park Address	Park Photo	Country Name	City Name	
1	Beto Carrero World	https://www.betocarrero.com.br/bries	Rod. Beto Carrero World - Praia de Armação do Itapocoróí, Penha - SC, 88365-000, Brasil		Brazil	Sao Paulo	MODIFICAR ELIMINAR
4	Happy Valley	bj.happyvalley.com	China, North Xiaowuji Road, East Fourth Ring, Chaoyang District, Beijing		China	Beijing	MODIFICAR ELIMINAR
3	Parc Du Bocasse	https://www.parcubocasse.fr/	226 Route de Clères, 76690 Le Bocasse, Francia		France	Normandia	MODIFICAR ELIMINAR
2	Shanghai Disney Resort	https://www.shanghaidisneyresort.com/en/	China, Shanghai Shi, Pudong, 川沙新镇, Huangzhao Rd, No. 310		China	Shangai	MODIFICAR ELIMINAR

DESIGN SYSTEMS (OPCIONAL)

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.

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

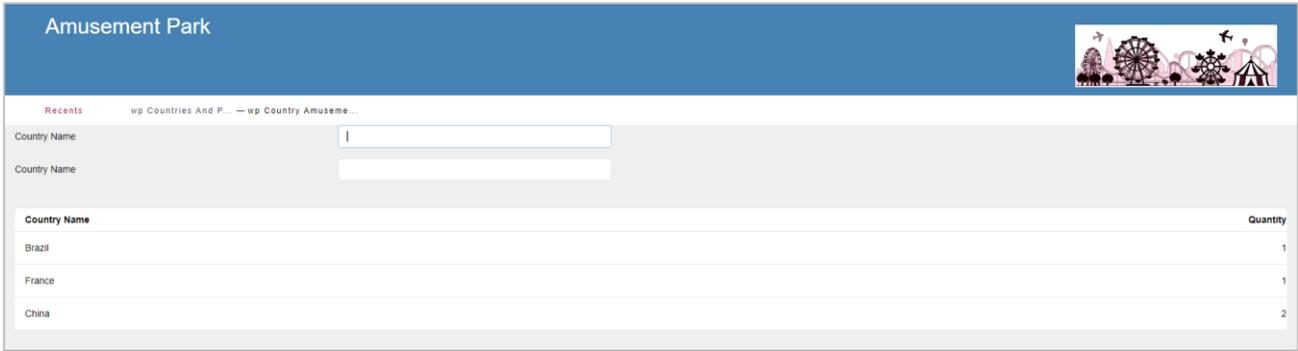
Amusement Park

Recents wp Countries And P... — wp Country Amuseme...

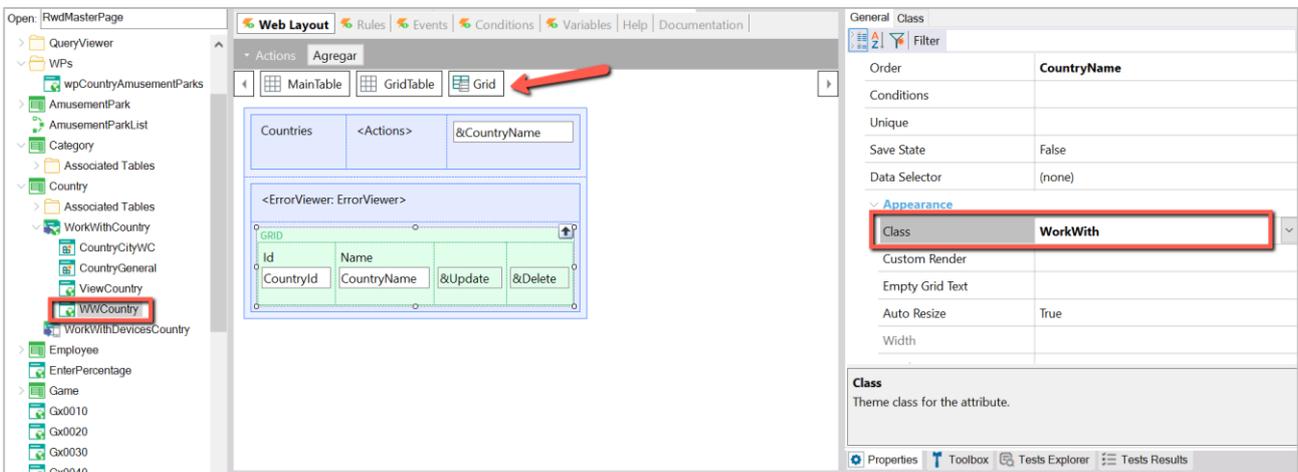
Country Name

Country Name

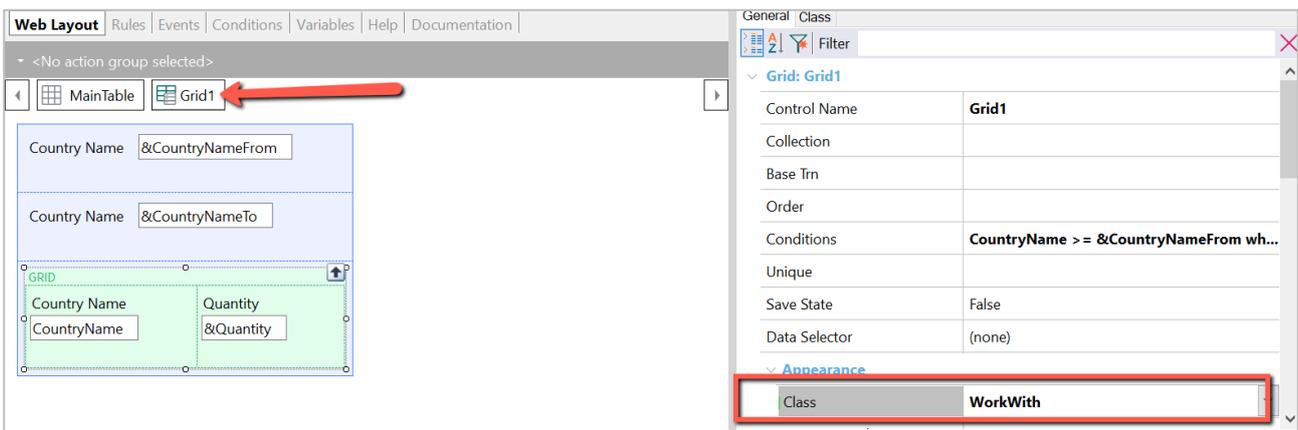
Country Name
Brazil
France
China



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



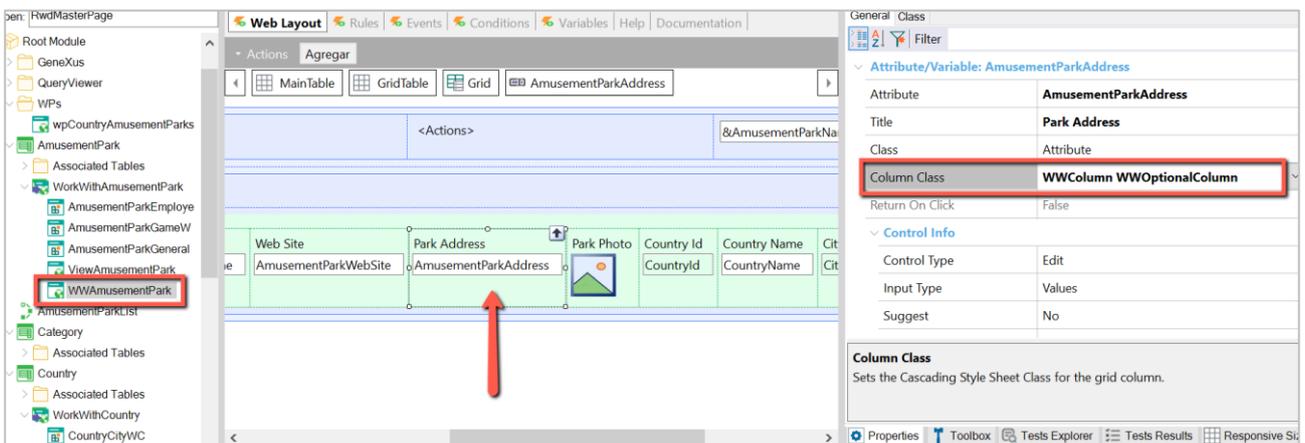
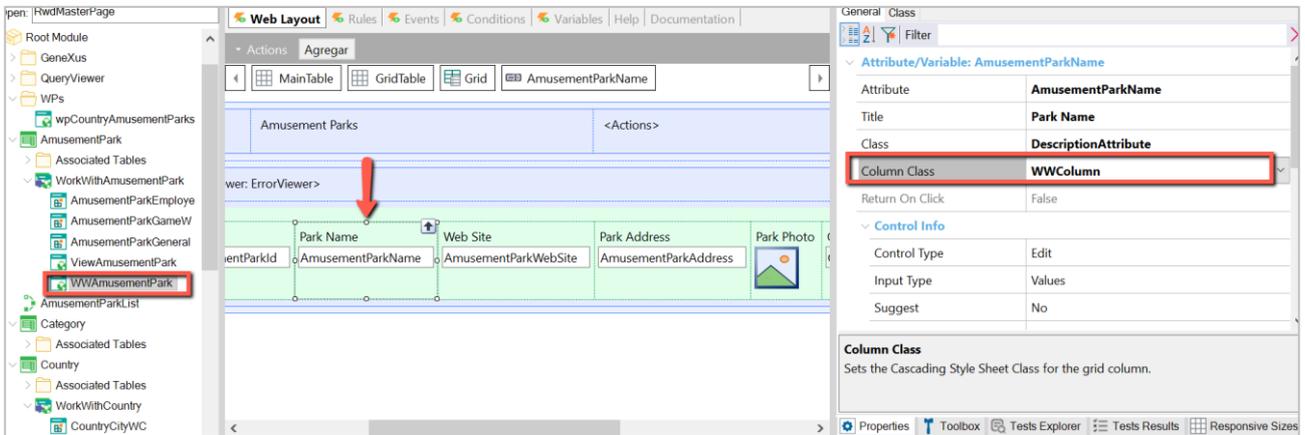
Esta misma clase aplíquela en el control Grid del Web Panel al que daremos formato:



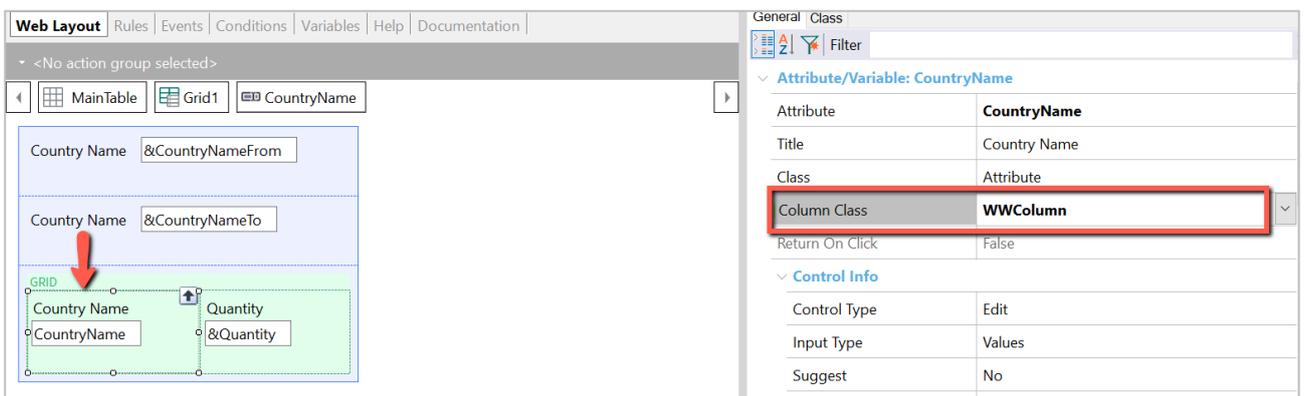
Para mantener la estética, observe que cada columna del Grid *WorkWith* tiene una clase asociada:

- *WWColumn* para los atributos que tienen mayor importancia que se muestran en todo momento, independientemente del tamaño de la pantalla.

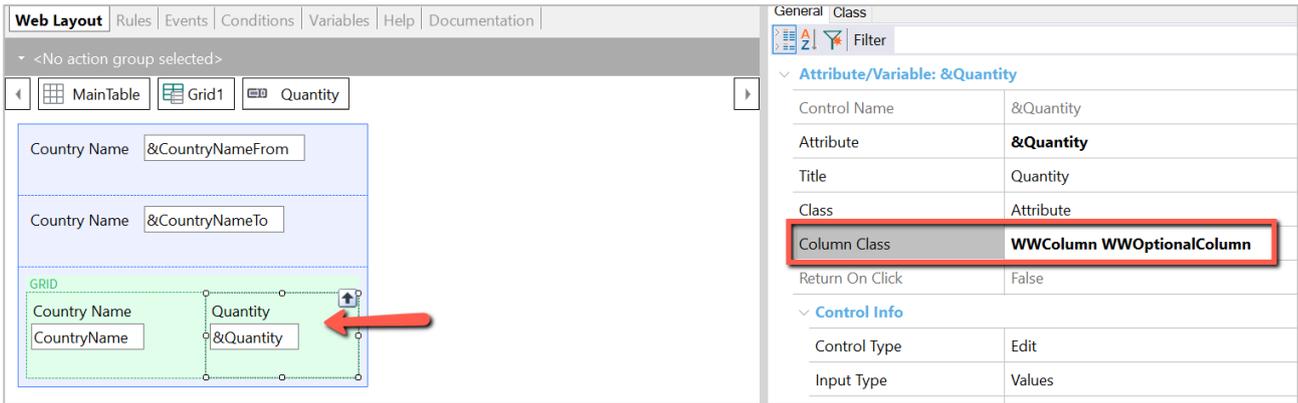
- *WWColumn* y *WWOptionalColumn* para los atributos que se ocultan cuando la pantalla se vuelve más pequeña.



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

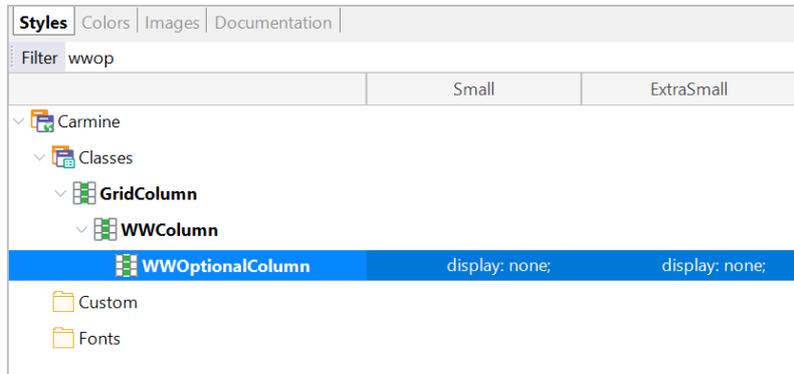


Para ocultar la cantidad de parques cuando la pantalla sea más pequeña, seleccione el control asociado a la variable y en la propiedad *ColumnClass* coloque *WWColumn* y *WWOptionalColumn*:

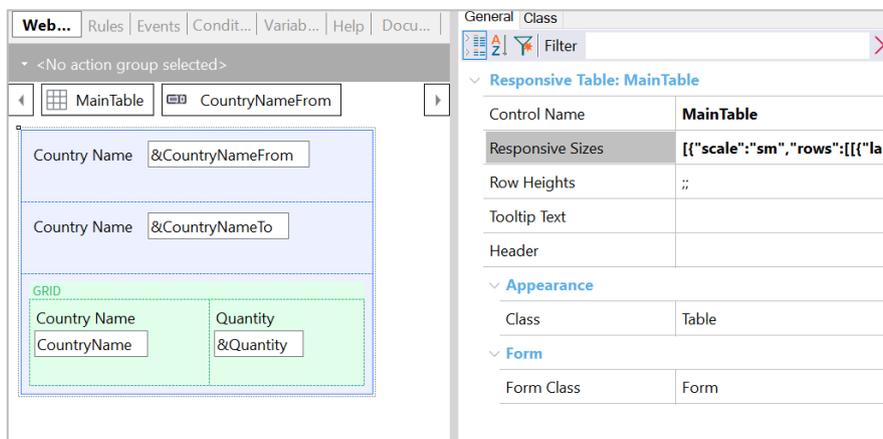


Ejecute la aplicación y pruebe el comportamiento. ¿Cómo se logran ocultar las columnas?

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.

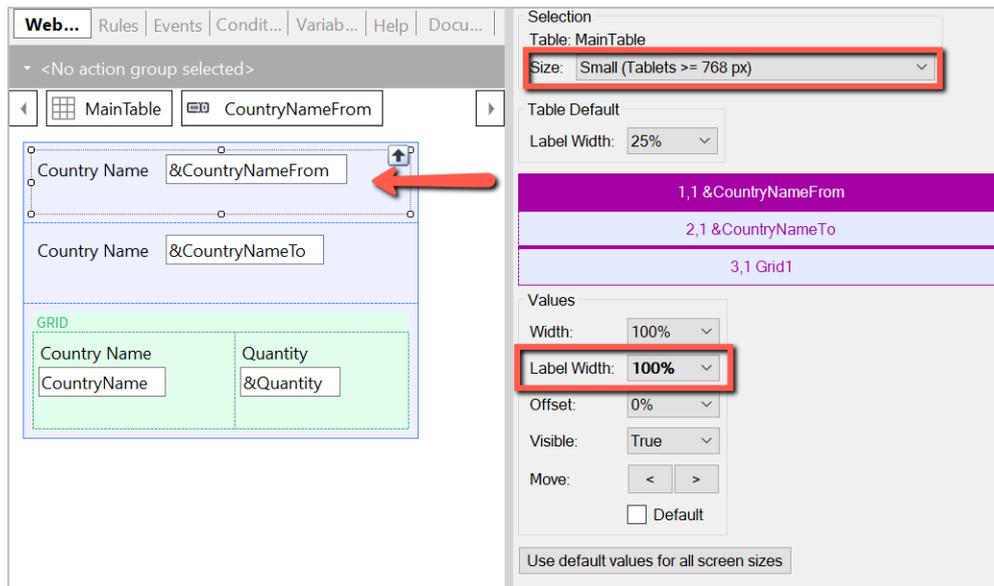


Por último, para que las etiquetas aparezcan arriba del cuadro editable, abra el recuadro de los tamaños responsivos. Si no aparecen, seleccione la MainTable del Web Panel y haga clic en la propiedad Responsive Sizes:



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:



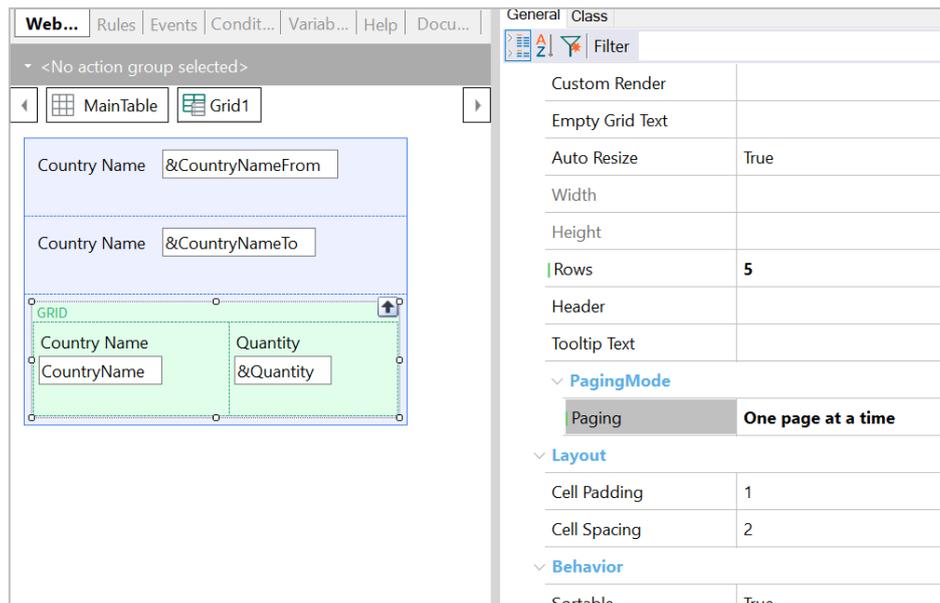
Ejecute la aplicación y verifique el comportamiento.

PAGINADO EN GRIDS

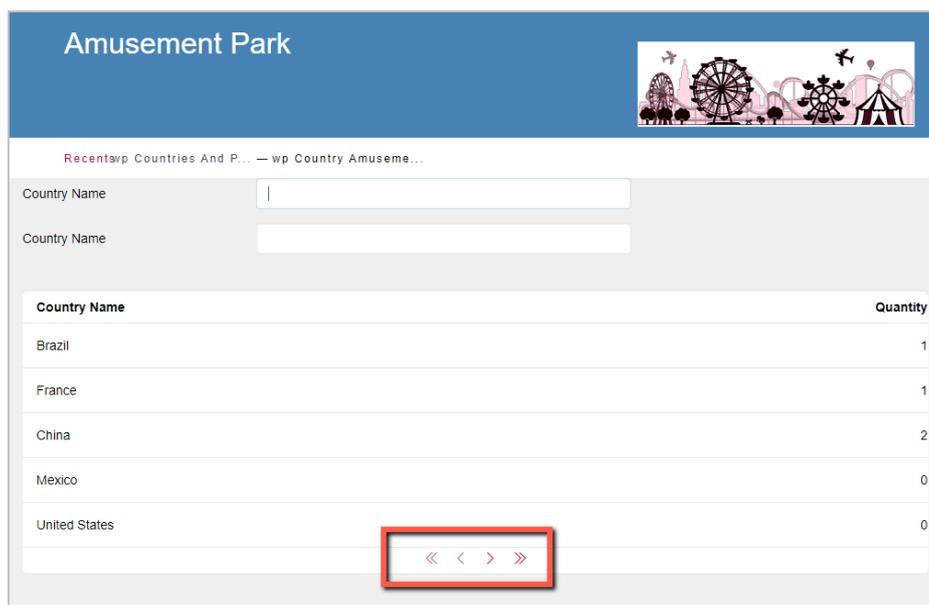
Se requiere que la pantalla anterior cuente con un Paginado en el Grid para visualizar los países de una mejor manera.

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:

Amusement Park

Recents wp Countries And P... — wp Country Amuseme...

Country Name

Country Name

Country Name	Quantity
Brazil	1
France	1
China	2
Mexico	0
United States	0

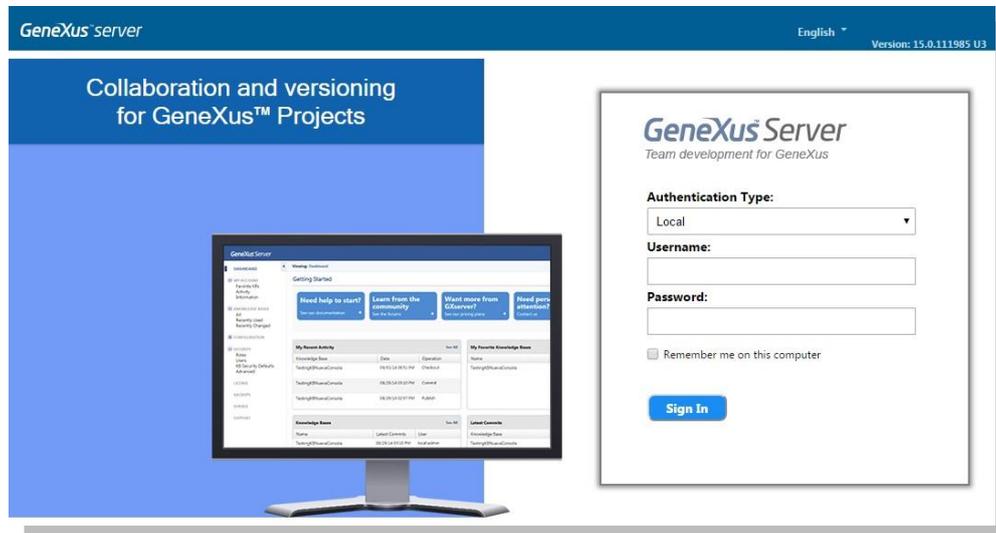
GENEXUS SERVER

Publique la base de conocimiento en el servidor <http://sandbox.genexusserver.com/v17>

Cree un nuevo Web Panel que muestre la lista de parques de diversiones registrados.

Envíe este nuevo objeto al servidor para que se integre a la base de conocimiento centralizada.

Ingrese a la consola web y verifique el estado final de la KB.



Cierre la base de conocimiento anterior, y cree una nueva sincronizándose con la KB previamente publicada. De esta forma recibirá localmente una copia de la KB administrada por GeneXus Server.

En esta nueva copia local, edite la transacción *Country* y defina el nuevo atributo *CountryFlagImage*, de tipo *Image*. Envíe este cambio al servidor.

Cierre esta *KB* y abra nuevamente la *KB* inicial. Realice la operación *Update* para recibir el cambio realizado anteriormente.