

# Exercícios práticos

Curso GeneXus Core

# **GeneXus<sup>TM</sup> 18**

Outubro 2022

© GeneXus. Todos os direitos reservados. Este documento não pode ser reproduzido por nenhum meio sem a autorização expressa da GeneXus<sup>TM</sup>. A informação contida neste documento é única e exclusivamente para uso pessoal.

**Marca registrada:**

GeneXus é uma marca comercial ou registrada da GeneXus S.A. Todas as outras marcas registradas mencionadas neste documento são de propriedade de seus respectivos proprietários.

**CONTEÚDO**

O PROBLEMA.....	4
NOVO PROJETO, NOVA BASE DE CONHECIMENTO .....	4
PRIMEIRAS TRANSAÇÕES.....	5
TRANSAÇÃO “EMPLOYEE” .....	5
TRANSAÇÕES “AMUSEMENTPARK” E “COUNTRY”, RELACIONADAS.....	7
DADOS RELACIONADOS: COMO É MANTIDA A INTEGRIDADE? .....	10
TRANSAÇÃO ‘SHOW’ .....	11
TRANSAÇÃO ‘GAME’ .....	11
TRANSAÇÃO ‘CATEGORY’ .....	11
TRANSAÇÕES “EMPLOYEE” E “AMUSEMENTPARK”, RELACIONADAS.....	13
ADICIONEMOS AS CIDADES À TRANSAÇÃO ‘COUNTRY’ .....	14
TRANSAÇÃO “AMUSEMENTPARK”: ADICIONEMOS A CIDADE.....	15
ADICIONEMOS COMPORTAMENTO ÀS TRANSAÇÕES (RULES) .....	16
PATTERNS: MELHORANDO A INTERFACE PARA TRABALHAR COM A INFORMAÇÃO .....	17
TRANSAÇÕES “REPAIR” E “TECHNICIAN” E NECESSIDADE DE DEFINIR SUBTIPOS.....	20
FÓRMULAS .....	21
CRIAÇÃO DE SEGUNDO NÍVEL .....	22

ÍNDICES.....	23
LISTAS PDF.....	24
PASSAGEM DE PARÂMETROS .....	31
LISTA DE PARQUES EM UM INTEVALO DETERMINADO.....	31
BUSINESS COMPONENTS .....	31
AUMENTO DE PREÇO DOS REPAROS.....	31
TELA PARA ELIMINAÇÃO DE TODOS OS REPAROS.....	32
PROCEDIMENTOS PARA ATUALIZAR REGISTROS.....	33
AUMENTO DE PREÇOS DOS REPAROS .....	33
ELIMINAÇÃO DE TODOS OS REPAROS.....	34
INICIALIZAÇÃO DA INFORMAÇÃO DA BASE DE DADOS [OPCIONAL] .....	35
WEB PANELS .....	37
MÚLTIPLOS GRIDS [OPCIONAL] .....	38
TIPOS DE WEB PANELS.....	39
DESIGN SYSTEMS .....	40
PAGINAÇÃO EM GRIDS.....	44
GENEXUS SERVER .....	46

## O PROBLEMA

Uma multinacional encarregada de administrar parques de diversões o contrata para que desenvolva um sistema para armazenar e manipular as informações com que trabalha. Imagine que o sistema é composto por dois módulos:

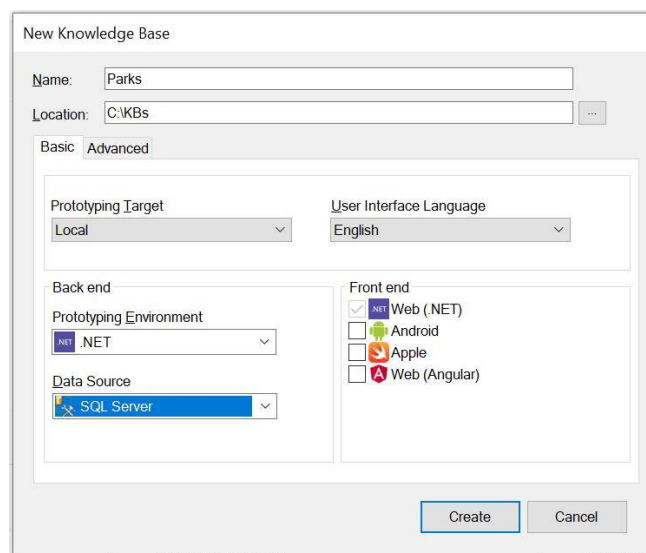
- **Backend:** parte da aplicação que deverá ser executada em um servidor web, de maneira tal que os funcionários da empresa possam manipular as informações a partir de qualquer lugar com conexão à internet.
- **Aplicação simples para dispositivos móveis:** parte da aplicação que será destinada a ser descarregada pelos clientes da empresa, a qual lhes permitirá consultar os países disponíveis, bem como os principais parques de diversões que oferece cada cidade e suas atrações.

## NOVO PROJETO, NOVA BASE DE CONHECIMENTO

Entrar no GeneXus e criar uma base de conhecimento chamada *Parks* para iniciar o desenvolvimento da aplicação.

Sugerimos:

- Escolher como o ambiente de desenvolvimento .NET. Certifique-se de ter instalado todo o necessário (incluindo o SQL Server). Se usa GeneXus Trial, o ambiente de geração com .NET e SQL Server já está predefinido, prototipando na nuvem da Amazon.
- Não criar a base de conhecimento na pasta “Meus Documentos” ou em qualquer outra pasta que fique abaixo de “Documents and Settings”, pois estas pastas têm permissões especiais concedidas pelo Windows.



Dedique alguns minutos para **familiarizar-se** com o **IDE (ambiente de desenvolvimento integrado de GeneXus)**. **Teste mover janelas, visualizar janelas específicas que você deseja** (View e View/Other Tool Windows) e observe cuidadosamente o conteúdo da janela **KB Explorer** (Knowledge Base Explorer). Verá que aparecem já inicializados **domínios**, alguns **objetos**, **imagens**, etc.

Sugestão: mantenha a janela de propriedades aberta (**F4**), pois você a utilizará continuamente.

## PRIMEIRAS TRANSAÇÕES

Nas reuniões com a empresa, transmitem o seguinte para você:

“Nós registramos os dados de parques de diversões, para realizar o gerenciamento, tanto de seus funcionários, quanto de suas atrações e atividades que oferecem aos visitantes”.

Para começar a construir a aplicação, devemos começar por identificar os atores da realidade, e representá-los por meio de **transações**. Que transações devemos criar então na base de conhecimento (KB)?

### TRANSAÇÃO “EMPLOYEE”

Perguntamos: quais dados registram dos funcionários da empresa? A resposta é a seguinte:

O **nome** (que não excede os 20 caracteres), **sobrenome** (que também não os excede), **endereço**, **telefone** e o **e-mail**.

Com estes dados, já pode criar a transação *Employee*.

#### Lembrar que:

- Para criar objetos, existem várias alternativas:
  - Fazê-lo a partir do menu: File/New/Object
  - Ctrl+N
  - Ícone da barra de ferramentas
- Digitando ponto (“.”) quando vai inserir um novo atributo, este é inicializado com o nome da transação.
- Precisará de um atributo que identifique cada funcionário (EmployeeId).

A estrutura da transação deveria ter ficado conforme mostrado:

Name	Type
Employee	Employee
EmployeeId	Numeric(4.0)
EmployeeName	Character(20)
EmployeeLastName	Character(20)
EmployeeAddress	Address, GeneXus
EmployeePhone	Phone, GeneXus
EmployeeEmail	Email, GeneXus

#### Lembrar que:

- Address, Phone e Email** são **domínios semânticos** que são atribuídos automaticamente aos atributos que são definidos contendo em seu nome os textos Address, Phone ou Email respectivamente.
- Ao definir o tipo de dados do atributo identificador, em vez de utilizar diretamente Numeric(4.0), defina o **domínio Id** de forma inline: **Id=Numeric(4.0)**. Configure a propriedade **Autonumber** desse domínio como **True**, para que todos os atributos baseados nele sejam numerados automaticamente, sem que o usuário deva se preocupar.

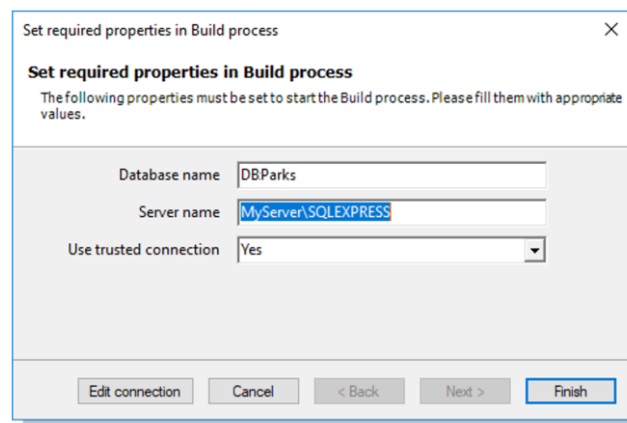
O próximo passo é testar a aplicação em execução. Certifique-se de ter a janela *Output* de GeneXus habilitada e visível (*View/Other Tool Windows/Output*).

Agora sim, **teste** a aplicação em execução pressionando **F5**.

O que acontecerá?

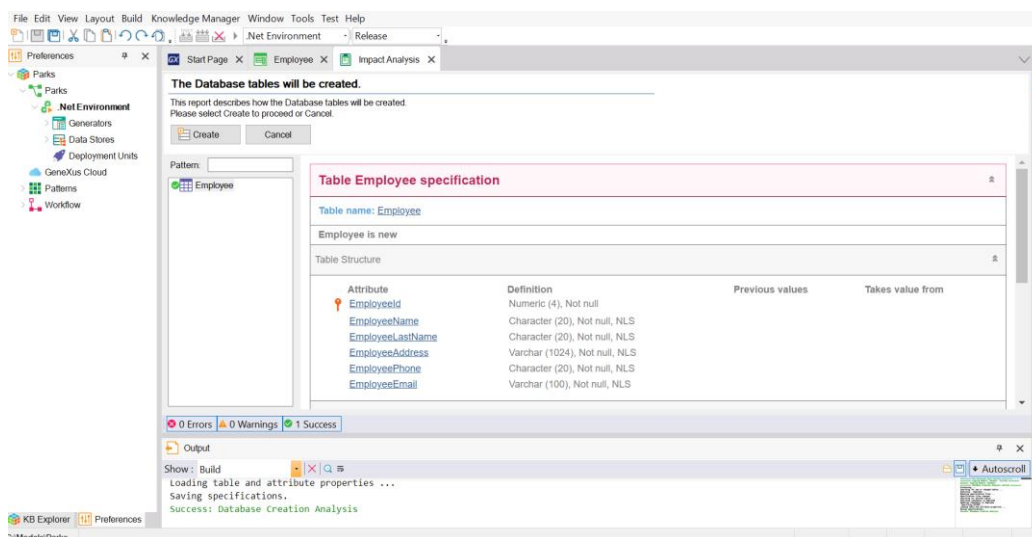
### Lembre:

Se está utilizando a versão *Full* ou a versão *Learning* de GeneXus e decide criar a base de dados e programas **localmente**, será aberta uma janela como a seguinte para que insira as informações de Base de Dados, Servidor e método de conexão. Lembre-se de que, se não existe uma base de dados com o nome que indicou nesse servidor, GeneXus a **criará**.



Se por outro lado a base de dados e programas serão criados na nuvem, o diálogo anterior não aparece, pois GeneXus conhece os dados do servidor na nuvem e configura automaticamente o nome da base de dados e todas as informações de conexão à mesma.

A seguir, é exibida uma **Análise de Impacto** que detalha que será criada a base de dados e a tabela *EMPLOYEE* dentro da mesma:



Se pressiona o botão **Create**, GeneXus continuará a executar o programa que realizará a criação. Ao finalizar o processo, será aberto no navegador que tenha configurado como padrão, o menu com links para executar os objetos definidos. Neste caso, apenas um: a transação *Employee*.

**Insira** alguns funcionários no sistema. Em seguida, **modifique** algum dado de algum dos funcionários previamente inseridos e **exclua** algum funcionário.

Teste também usar as setinhas oferecidas para passar de registro em registro de funcionários e a opção *SELECT*, que oferece uma “lista de seleção” para visualizar a lista de funcionários registrados e selecionar um.

Agora, passemos a identificar e criar a próxima transação. Lembremos do que nos haviam dito, ao qual foram feitos alguns acréscimos:

“Registramos os dados de **parques de diversões** de diferentes idades em diferentes **países**, para gerenciar tanto seus funcionários quanto suas atrações e atividades que oferece aos visitantes.”

Antes de continuar com o desenvolvimento, publiquemos a KB no GeneXus Server (<http://sandbox.genexusserver.com/v16>)

Lembre:

- Deve se autenticar com sua conta GeneXus.
- O alias com o qual se publique a KB deve ser único. Isto significa que não podem existir no Server duas KBs com o mesmo alias.

## TRANSAÇÕES “AMUSEMENTPARK” E “COUNTRY”, RELACIONADAS

Perguntamos aos funcionários da empresa: quais dados registram dos parques de diversões com os quais trabalham? A resposta é a seguinte:

**O nome** (que não excede os 40 caracteres), **site web** (que não excede os 60 caracteres), **endereço** e **foto representativa**.

Com estes dados, já pode criar a *AmusementPark*.

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

Vamos criar uma transação para registrar os países aos quais os parques de diversões pertencem.

**Lembre-se** que:

- pressionando ponto (".") quando está prestes a nomear um atributo na estrutura da transação, aparece inicializado com o nome da transação.
- Precisar de um atributo identificador, CountryId.

Defina o atributo *CountryName*, criando e utilizando um novo domínio: *Name = Character(50)*.

Name	Type
Country	Country
CountryId	Id
CountryName	Name

Agora, retornaremos à transação *AmusementPark*, para adicionar os atributos *CountryId* e *CountryName*.

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

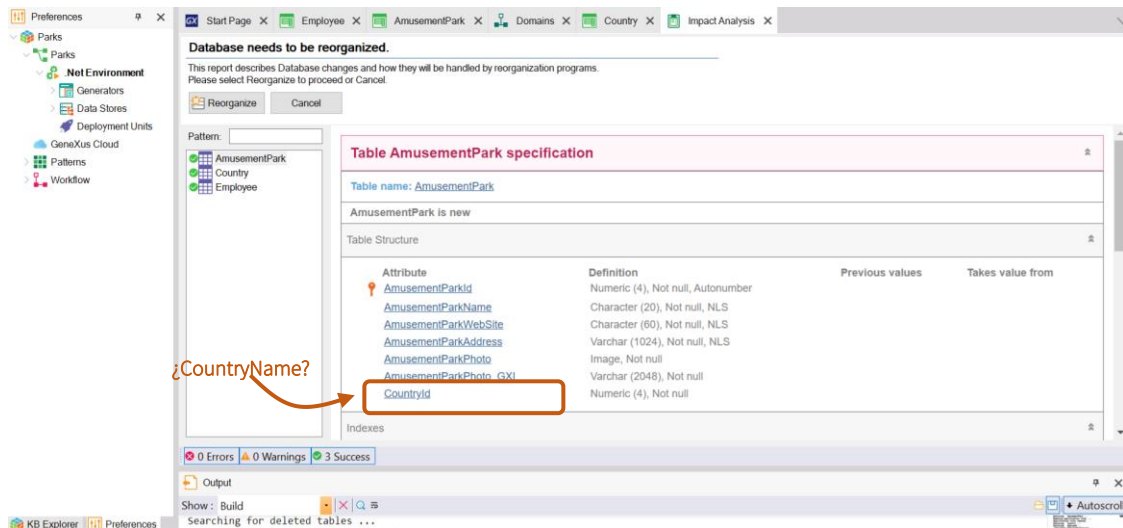
Name	Type
Country	Country
CountryId	Id
CountryName	Name

Por que colocou além de *CountryId*, o atributo *CountryName* em *AmusementPark*?



**Resposta:** O atributo *CountryName* é importante que apareça na tela da transação, para nos mostrar o nome do país, que é o dado que mais lembramos do país, em vez de ver apenas seu identificador. Também é necessário adicionar o atributo na estrutura da transação se quisermos utilizá-lo posteriormente, por exemplo, dentro de uma regra.

Execute para testar (F5) e o seguinte relatório será exibido:



Por que na tabela *AmusementPark*, que GeneXus informa que deve ser alterada na Base de Dados não aparece o atributo *CountryName*? Ou seja, por que a tabela física não o conterá, quando está na estrutura da transação?

Depois de estudar o relatório, se concordarmos, pressionamos **Reorganize** para efetivamente realizar isso que se informa. Será aberto o navegador com o menu com links para os 3 programas que correspondem a cada uma das transações (*AmusementPark*, *Country* e *Employee*).

Inserir como **países**: Brasil, França e China. Observar que, deixando o valor 0 como valor no identificador, ao gravar lhe é atribuído automaticamente o número posterior ao último atribuído (efetivamente, está sendo autonumerado).

Id	Name	Last Line
1	Brazil	0
2	France	0
3	China	0

Inserir como parque de diversões: “Beto Carrero World”, que fica no Brasil. Se não se lembra do identificador do Brasil no sistema, como insere o país? É oferecido um ícone com uma seta ao lado de *CountryId*, para abrir uma “Lista de seleção” de países, criada automaticamente por GeneXus. Isto ocorre porque *CountryId* tem o papel de chave estrangeira (foreign key) nesta transação (ou seja, está “apontando” para outra tabela).

#### DADOS RELACIONADOS: COMO É MANTIDA A INTEGRIDADE?

*AmusementPark* e *Country* estão relacionados. Ao colocar *CountryId* na estrutura de *AmusementPark*, por ter exatamente o mesmo nome do atributo que é chave primária na transação *Country*, GeneXus entende que em *AmusementPark* o atributo *CountryId* é chave estrangeira e mantém automaticamente a integridade da informação. Então, por exemplo:

- Tente inserir um parque de diversões com um Id de país que não exista. Permite gravar esse parque?
- Escolha um parque previamente inserido (por exemplo, ‘Beto Carrero World’) e altere o país, para um que não exista. Foi possível gravar a modificação?
- Tente eliminar um país (usando a transação *Country*) que tenha algum parque associado (por exemplo, Brasil). Permite isso?

**Conclusão:** os programas correspondentes às transações garantem a integridade dos dados.

Volte ao GeneXus e altere o tipo de dado associado ao atributo *AmusementParkId* para o domínio *Id*. Execute a aplicação. O que marca a Análise de Impacto? Pressione **Reorganize**.

### TRANSAÇÃO 'SHOW'

Nos informam que os parques de diversões podem oferecer diferentes shows (musicais, de magia, educativos, etc) aos seus participantes, em determinada data e hora. Dos shows, interessa saber seu nome e uma imagem representativa. Um mesmo show pode ser contratado por vários parques de diversões.

### TRANSAÇÃO 'GAME'

Conforme foi solicitado inicialmente, o sistema deve oferecer a possibilidade de inserir as atrações disponíveis em cada parque, portanto, devemos criar uma transação que contenha seu nome e o parque de diversões ao qual pertence.

### TRANSAÇÃO 'CATEGORY'

Nos falta completar as informações da transação *Game*. Os funcionários descreveram que também registram de cada atração, a **categoria** (infantil, radical, recreativa, etc) à qual pertence. Portanto, precisaremos criar uma transação para registrar esta informação e adicionar a categoria à transação *Game*.

Mas, além disso, nos informaram que não é obrigatório registrar sempre a categoria à qual pertence uma determinada atração que está sendo manipulada. Pode ser deixada vazia. Se sabemos que GeneXus controla automaticamente a integridade, como obtemos isso?

---

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 finalizar a definição da transação *Game*, adicionemos o dado que está faltando: a foto.

Para fazer isso, crie o atributo *GamePhoto* do tipo de dados *Image*.

Peça ao GeneXus que construa a aplicação, então pode testá-la em execução (F5).

**Observe** o que informa o relatório de Análise de Impacto. Deverão ser criadas as tabelas *Category* e *Game* (não se preocupe em entender por que precisa armazenar dois valores por imagem).

Reorganize e execute.

Insira categorias (como infantil e radical) e atrações (como montanhas russas e carrosséis).

Observe que, neste caso, pode deixar a categoria vazia (porque configurou a propriedade **Nullable** como **Yes** na estrutura da transação).

No entanto, se tentar colocar como valor de *CategoryId* para a atração um valor inexistente, não permitirá gravar.

Lembre-se de atualizar as mudanças no GeneXus Server.

GeneXus™
Application Name

### Game

Id	0
Name	<input type="text" value="Big Tower"/>
Amusement Park Id	<input type="text" value="1"/>
Amusement Park Name	Beto Carrero World
Category Id	<input type="text" value="2"/>
Category Name	Radical
Photo	<div style="position: absolute; top: 5px; left: 5px; background-color: #4a7ebb; color: white; padding: 2px 5px; font-size: 8px;">CHANGE</div>

## TRANSAÇÕES “EMPLOYEE” E “AMUSEMENTPARK”, RELACIONADAS

Como nos disseram no início do desenvolvimento da aplicação, o sistema gerenciará os parques de diversões e seus funcionários. Portanto, precisamos vincular os funcionários que são registrados com o parque no qual trabalham.

Para isso, vamos para a transação *Employee* e adicionamos os atributos *AmusementParkId* e *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)

Fomos informados de que não é obrigatório registrar no momento, o parque onde o funcionário trabalha, ou seja, pode ficar vazio. O que devemos fazer?

Execute para testar (F5) e aparecerá o seguinte relatório indicando que o atributo AmusementParkId agora permite deixar um valor não 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

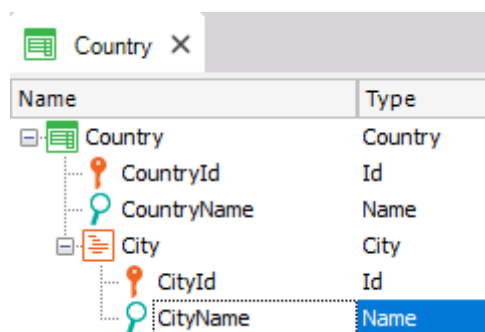
Attribute	Definition	Previous values	Takes value from
EmployeeId	Numeric (4), Not null, Autonumber		Employee. EmployeeId
EmployeeName	Character (20), Not null, NLS		Employee. EmployeeName
EmployeeLastName	Character (20), Not null, NLS		Employee. EmployeeLastName
EmployeeAddress	Varchar (1024), Not null,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

## ADICIONEMOS AS CIDADES À TRANSAÇÃO 'COUNTRY'

Além dos países, precisamos registrar a informação de suas cidades. Portanto, devemos adicionar um segundo nível à transação *Country*, com o identificador e o nome da cidade.



### **Lembre-se** de que:

- Posicionado no atributo *CountryName*, com botão direito, **Insert Level**, adiciona o subnível.
- Depois de dar um nome ao novo nível, digitando aspas (") em vez de ponto, o atributo que define se inicializará com o nome do nível.
- As cidades serão identificadas por seu próprio Id em combinação com o do país. Ou seja, não poderá identificar uma cidade sem fornecer antes a informação do país em questão. Assim, poderia haver uma cidade 1 Rosario tanto para Uruguai como para Argentina:
  - País: 1 (Uruguai) – Cidade: 1 (Rosario)
  - País: 2 (Argentina) – Cidade: 1 (Rosario)

- Ou inclusive poderia ser que Rosario para Argentina se identificava com outro número:  
País: 2 (Argentina) – Cidade: 4 (Rosario)

Reorganize e execute (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 a Lista de Navegação irá informá-lo de que:

- A propriedade **Autonumber** para o caso de **CityId** será ignorada. Isto significa que em execução o usuário deverá inserir manualmente os identificadores de cidade. A explicação é que a propriedade **Autonumber** somente autonumera chaves primárias simples e, neste caso, **CityId** é o segundo componente de uma chave composta.
- Será criada uma nova tabela **CountryCity** para armazenar a informação correspondente às cidades.

Insira cidades para os países que já tinha registrados.

TRANSAÇÃO “AMUSEMENTPARK”: ADICIONEMOS A CIDADE.

Na transação *AmusementPark* adicionemos a cidade do país à qual o parque pertence. O que deve fazer se a empresa nos informa que esse valor pode não ser conhecido ou relevante para um determinado parque em um determinado momento?

Construa a aplicação e teste-a (F5 e Reorganize).

Atualize as mudanças no GeneXus Server.

## ADICIONEMOS COMPORTAMENTO ÀS TRANSAÇÕES (RULES)

Depois de testar conosco a aplicação que estamos desenvolvendo, na empresa nos dizem que, para os funcionários, há algum comportamento específico que devemos fazer cumprir ao manipular as informações por meio do programa (transação *Employee*).

Qual é este comportamento?

Nos dizem:

- “O sistema não deve permitir inserir funcionários sem nome, nem sem sobrenome”.
- “O usuário deve ser avisado se estiver deixando o telefone sem atribuição, caso tenha sido um descuido”.
- “Deve ser registrada a data de inserção do funcionário no sistema (**EmployeeAddedDate**) e deve ser proposta como valor padrão para esse atributo, a data de hoje”.

Especifique este comportamento e teste-o (F5 e Reorganize).

**Lembre-se** de que:

- As regras terminam com ponto e vírgula “;”.
- O método **IsEmpty()** aplicado a um atributo devolve True quando o atributo está vazio e False caso contrário.
- A variável **&Today** é do sistema e tem carregado o valor da data do dia.

Para escrever uma variável na tela **Rules**, ao digitar “&” são exibidas todas as variáveis definidas até o momento para que selecione a que precisa. A outra possibilidade é utilizar **Insert / Variable**.

Teste inserir um novo funcionário, deixando o nome em branco. Permite salvar ou passar para o próximo campo?

Idem com o sobrenome. Ocorre o mesmo com o telefone?

Se posteriormente lhe informam que a data de inserção no sistema não deveria ser manipulada pelo usuário, mas apenas exibida, como estabelece este comportamento?

Especifique e teste-o em execução.

Na Transação Country, adicione uma regra que permita autonumerar o atributo CityId.

Especifique e teste-o em execução.



**PATTERNS: MELHORANDO A INTERFACE PARA TRABALHAR COM A INFORMAÇÃO**

Ao mostrar ao cliente o que foi feito até agora, ele nos diz que gostaria de poder manipular as informações de países, parques de diversões, funcionários, shows, categorias e atrações de uma forma mais potente e atraente (que ofereça consulta, possibilidade de filtrar, bem como inserir, modificar e excluir dados, etc.).

Para fazer isso, deverá aplicar o padrão *Work With for Web* às transações. Teste-o e veja em execução.

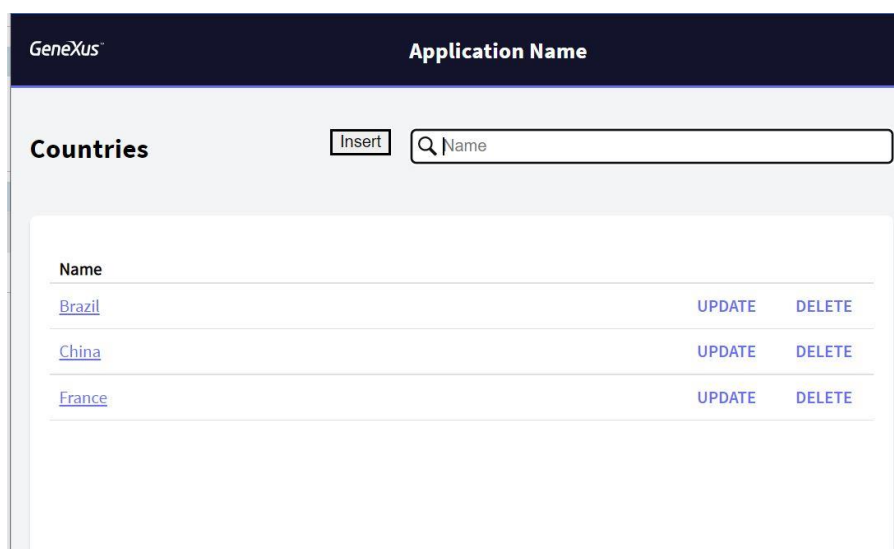
**Observar** que:

- existe também um *Work With* para Smart Devices. Mas o que você deverá aplicar é aquele que corresponde à aplicação web que está construindo.
- GeneXus criará automaticamente vários objetos por transação, para implementar o "Trabalhar com" essa entidade.

Por que não aparecem mais no *Developer Menu* as transações *Country*, *AmusementPark*, *Employee*, *Show*, *Category* e *Game*?

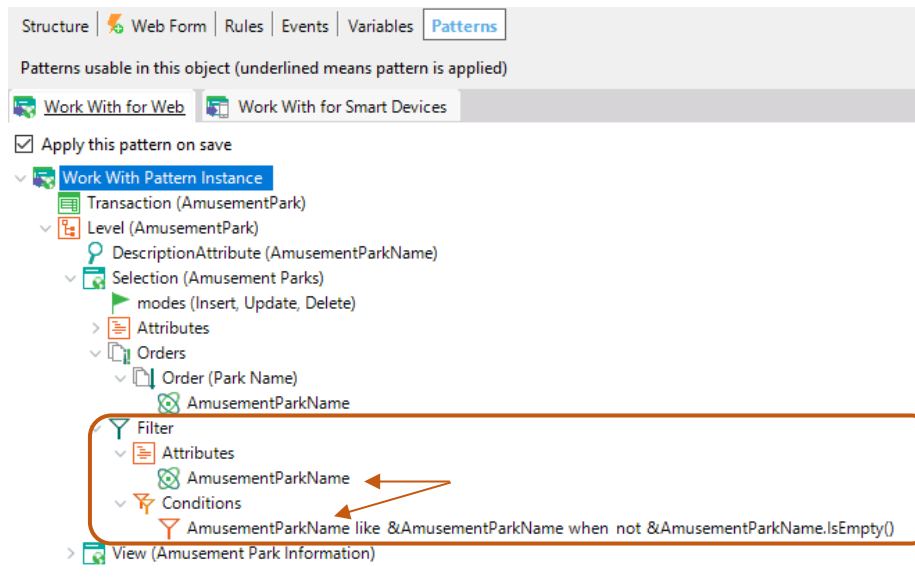
Tente fazer o seguinte:

1. Inserir um novo país.
2. Modificar um país existente (por exemplo, lhe adicionando uma cidade).
3. Eliminar um país existente.
4. Visualizar a informação de um país.
5. Realizar uma pesquisa por nome de país.



6. Insira alguns parques de diversões (Ex: Shangai Disney Resort, da China/Shangai, Parc Du Bocasse da França/Normandia, Happy Valley, da China/Beijing).
7. Filtre os parques de diversões cujo nome comece com P. E se agora quiser poder visualizar todos os parques da China? Esta possibilidade não está incluída, portanto, deveremos personalizar o pattern *Work With* desta transação, para incluí-la. Faça em **GeneXus** e teste em **execução**.

**Sugestão:** Observe como está especificado o filtro que existe, pelo nome do parque de diversões. Tente definir o filtro solicitado da mesma forma.







8. Agora remova os identificadores de país e cidade da tela do *Work With* e teste em execução.

GeneXus Application Name

Amusement Parks

Insert GAMES


Park Id	Park Name	Web Site	Park Address	Park Photo	Country Id	Country N...	City Id	City Name		
1	<a href="#">Beto Carrero World</a>	www.betorrero.br	<a href="#">Rua Inácio Francisco de Souza, 1597</a>		1	<a href="#">Brazil</a>	2	Sao Paulo	UPDATE	DELETE
4	<a href="#">Happy Valley</a>	bj.happyvalley.com	<a href="#">Beijing Olympic Tower</a>		3	<a href="#">China</a>	1	Beijing	UPDATE	DELETE
3	<a href="#">Parc Du Bocasse</a>	www.parcdubocasse.fr	<a href="#">225 Route de Clères, 76690</a>		2	<a href="#">France</a>	2	Normandia	UPDATE	DELETE
2	<a href="#">Shanghai Disney Resort</a>	shangaidisneyresort.com	<a href="#">Shanghai Disney Resort, Pudong, Xangai</a>		3	<a href="#">China</a>	2	Shangai	UPDATE	DELETE

9. Se agora deseja oferecer a possibilidade de que o usuário escolha se quer ver os parques de diversões ordenados por nome do parque ou por nome do país, implemente e teste.





GeneXus Application Name

Amusement Parks

Insert GAMES

Ordered By 

- Park Name
- Park Name**
- Country Name
- Country Name

Park Id	Park Na...	Web Site	Park Ad...	Park Ph...	Country...	City Name		
1	<a href="#">Beto Carrero World</a>	www.betorrero.br	<a href="#">Rua Inácio Francisco de Souza, 1597</a>		<a href="#">Brazil</a>	Sao Paulo	UPDATE	DELETE
4	<a href="#">Happy Valley</a>	bj.happyvalley.com	<a href="#">Beijing Olympic Tower</a>		<a href="#">China</a>	Beijing	UPDATE	DELETE
3	<a href="#">Parc Du Bocasse</a>	www.parcdubocasse.fr	<a href="#">225 Route de Clères, 76690</a>		<a href="#">France</a>	Normandia	UPDATE	DELETE
2	<a href="#">Shanghai Disney Resort</a>	shangaidisneyresort.com	<a href="#">Shanghai Disney Resort, Pudong, Xangai</a>		<a href="#">China</a>	Shangai	UPDATE	DELETE




## TRANSAÇÕES “REPAIR” E “TECHNICIAN” E NECESSIDADE DE DEFINIR SUBTIPOS

Agora é necessário registrar as atrações que entram em estado de reparo. Cada reparo possui um identificador, uma data a partir da qual a atração deixa de estar disponível para uso, a quantidade estimada de dias para seu conserto, o identificador da atração, seu nome, o técnico titular e o técnico substituto. Além disso, cada reparo tem um custo. Para o custo crie um domínio chamado *Cost*, do tipo Numeric(8.2).

Crie uma transação para registrar os técnicos que trabalharão nos reparos. Cada técnico possui um identificador, um nome e sobrenome, um telefone, um país e uma cidade na qual se encontra.

Como se define que cada reparo tem um técnico titular e outro substituto?

### **Lembre**

- 1) Que na estrutura da transação:
  - um ícone representando uma seta para cima  informa que o atributo é chave estrangeira (Foreign Key), isto é, que aponta para outra tabela.
  - Um ícone representado por uma seta para baixo  informa que o atributo é inferido de outra tabela.
  - Um ícone representando um  indica que o atributo é um subtipo.
- 2) Sobre os grupos de subtipos:
  - São definidos da mesma maneira que qualquer tipo de objeto.
  - Cada grupo de subtipos deve conter obrigatoriamente um subtipo de um atributo primário (que é a chave primária de uma tabela) ou conjunto de atributos que formam uma chave primária.
  - Em cada grupo de subtipos, é necessário incluir todos os atributos subtipos que precisam ser conhecidos, pertencentes à tabela base e/ou estendida da chave primária do grupo.

**Execute e verifique** que, ao tentar inserir um reparo, um erro será disparado se o técnico titular que está querendo atribuir ao reparo não existir. Idem para o técnico substituto.

Não deve ser permitido inserir um reparo cujo técnico titular coincida com o técnico substituto. Implemente esse comportamento e teste-o em execução.

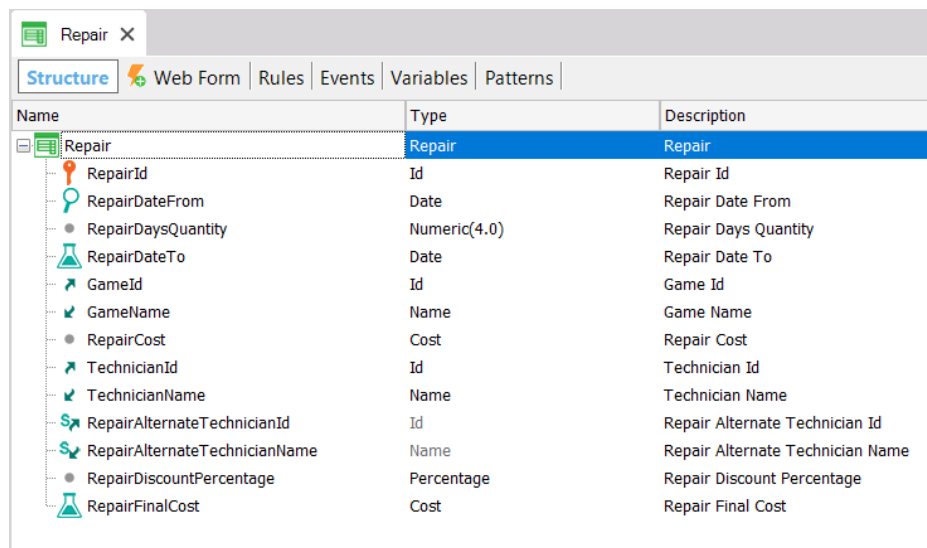
Atualize as mudanças no GeneXus Server.

## FÓRMULAS

É necessário poder registrar o **desconto atual que tem cada reparo**. Defina um novo atributo na transação Repair para armazenar este dado. Dê ao novo atributo o nome: *RepairDiscountPercentage* e que seu tipo de dados seja um domínio *Percentage*, numérico de 3.

Deseja-se visualizar o preço final do conserto com o desconto aplicado. Para resolver isto, defina outro atributo, chamado *RepairFinalCost*, que seja **fórmula global** e que calcule automaticamente o preço final do reparo.

Adicione um novo campo chamado *RepairDateTo*, que será uma soma entre a data de início do reparo e a quantidade de dias que levará o mesmo.



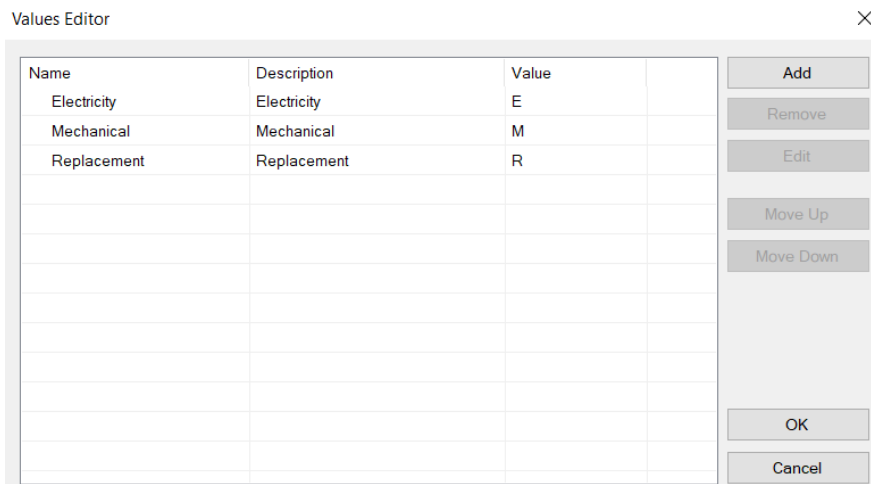
Name	Type	Description
Repair	Repair	Repair
RepairId	Id	Repair Id
RepairDateFrom	Date	Repair Date From
RepairDaysQuantity	Numeric(4.0)	Repair Days Quantity
RepairDateTo	Date	Repair Date To
GameId	Id	Game Id
GameName	Name	Game Name
RepairCost	Cost	Repair Cost
TechnicianId	Id	Technician Id
TechnicianName	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

Pressione F5, observe na Análise de Impacto qual atributo será criado fisicamente e qual não será, reorganize e teste a aplicação em operação.

## CRIAÇÃO DE SEGUNDO NÍVEL

Deseja-se criar um segundo nível na transação *Repair* para guardar um detalhe do tipo de problema encontrado para reparar.

Para fazer isso, primeiro deverá ser criado um domínio chamado *KindName*, *Character(1)*. Restrinja os valores possíveis para o domínio: que sejam válidos os valores “E”, “M” e “R” (editando a propriedade **Enum Values** do mesmo conforme visto abaixo).



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

Crie um segundo nível na transação *Repair* chamado *Kind* para registrar o tipo de reparo. Este nível terá estes três atributos:

- *RepairKindId* – *Numeric(4)* (será chave neste segundo nível)
- *RepairKindName* – Baseado no domínio *KindName* (Genexus o sugerirá automaticamente).
- *RepairKindRemarks* – *Character(120)*, conterà observações sobre o problema, um pequeno detalhe do problema encontrado ou a peça a ser substituída, por exemplo.

**Lembre-se de que** para definir que determinado atributo faz parte da chave primária, deve pressionar o botão **direito do mouse sobre o atributo** e o menu contextual oferecerá a opção **Toggle Key**. Neste caso, não será necessário, pois apenas o primeiro atributo faz parte da chave primária e este já aparece com o ícone da chave.

Para saber a quantidade de tipos de problemas envolvidos em um reparo, crie um novo atributo no primeiro nível da transação *Repair*, chamado *RepairProblems*, *Numeric(1)* e defina-o como fórmula global. Este atributo deverá contar os tipos de problemas encontrados.

Um reparo poderia envolver problemas elétricos, mecânicos ou que seja necessária a substituição de alguma peça. Poderia até ter mais de um problema do mesmo tipo, dois registros de eletricidade, por exemplo. Se houver muitos, pode detalhar algo mais com o uso do atributo de observações, que permite escrever um pequeno texto.

Se deseja ver a quantidade de problemas no formulário web e deve ser controlado que sejam inseridas entre 1 e 3 linhas de tipos de problema.

Este controle deverá ser realizado **quando terminar de inserir dados no segundo nível** e depois de ter pressionado o botão *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

O valor de *RepairKindId* vai sendo inserido manualmente. O normal seria ir dando valores a partir 1. Faça o correspondente para numerar automaticamente este atributo (lembre-se de que **não é possível autonumerá-lo** utilizando a propriedade *Autonumber*).

Atualize as mudanças no GeneXus Server.

## ÍNDICES

Vá ao menu *View* → *Tables* e coloque os controles necessários de forma que não seja possível repetir o nome de um país nem de um parque de diversões.

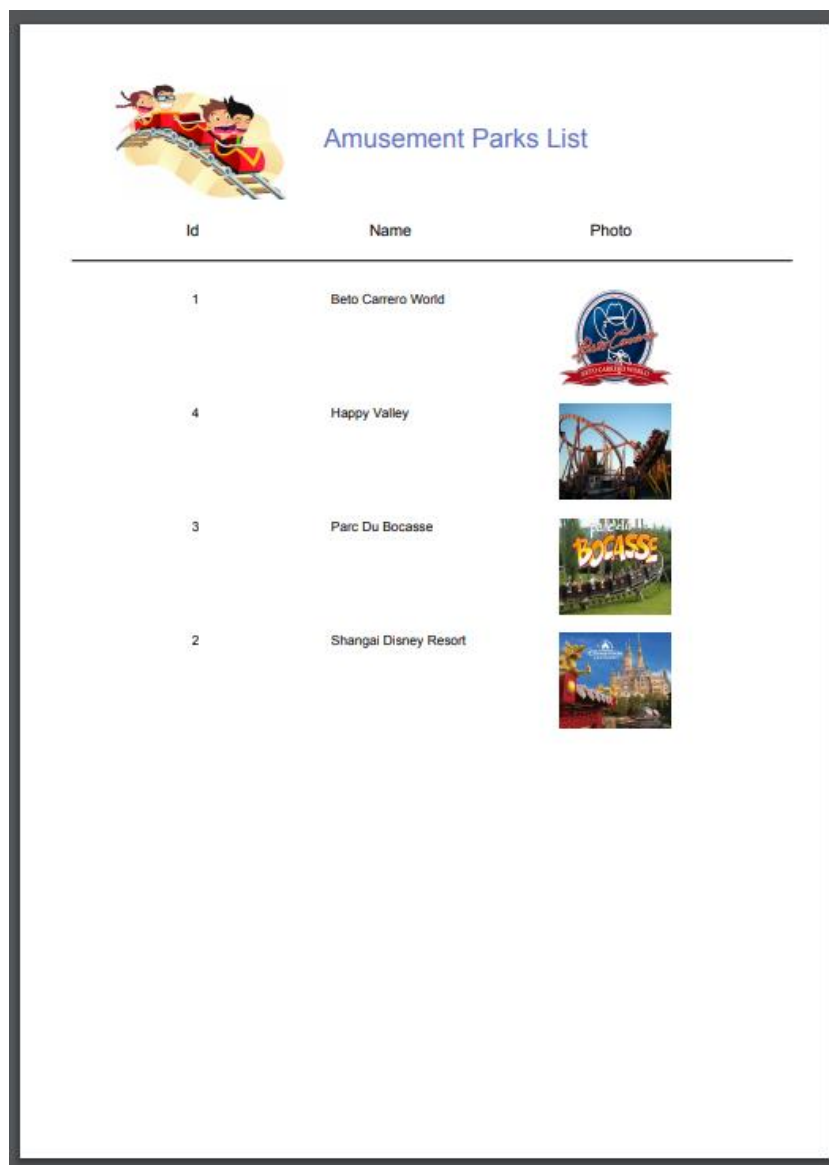
Em qual seção estes controles são especificados?





Que mensagem aparece no navegador se tenta adicionar um país/parque de diversões cujo nome se repete?

## LISTAS PDF

Agora suponhamos que, como parte da aplicação, deverá ser implementada a possibilidade de que, a pedido do usuário, sejam exibidas listas *PDF* com a informação solicitada. Por exemplo, suponha que seja necessária uma lista que mostre em ordem alfabética, os parques de diversões armazenados na base de dados.

Sabe-se que deve parecer mais ou menos como segue:

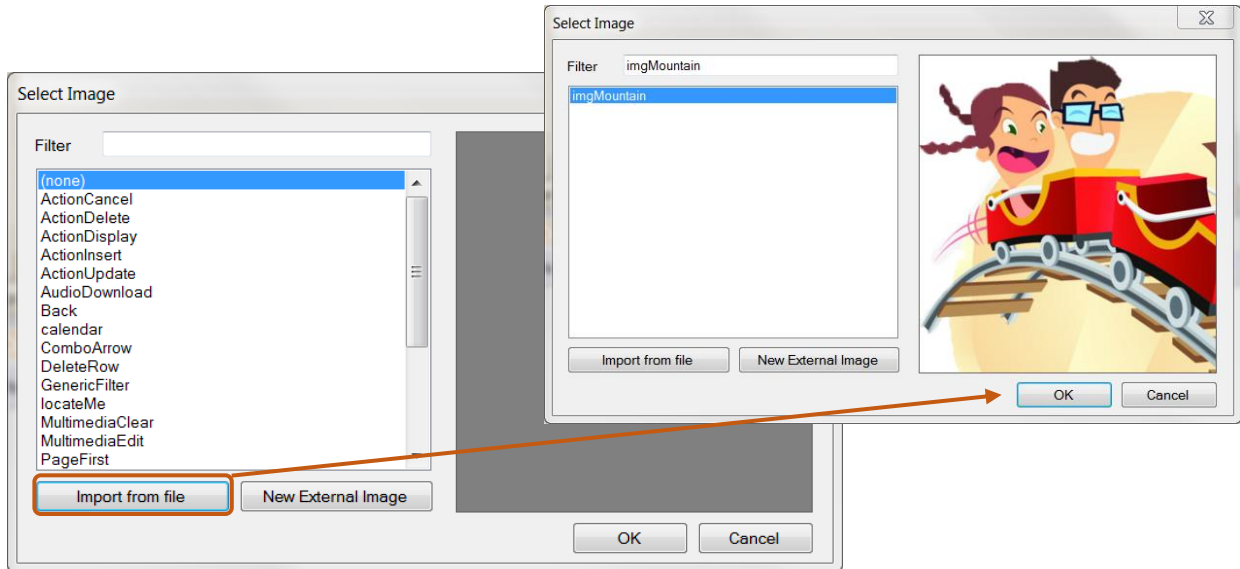


Id	Name	Photo
1	Beto Carrero World	
4	Happy Valley	
3	Parc Du Bocasse	
2	Shangai Disney Resort	

**Sugestão para colocar uma imagem no título:** Utilizar o control *Image* presente na *Toolbox* para exibir uma imagem junto ao título da lista.



Tal imagem deverá ser integrada à base de conhecimento. Para isso selecionar a opção *Import from File*, procurar a imagem e atribuir a ela um nome.



Implemente em GeneXus.

**Lembrar** que para poder visualizar uma lista diretamente a partir do browser, a mesma deve ser gerada como PDF. Para isto deve configurar as seguintes propriedades do objeto procedimento:

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

E a seguinte regra:


- Output\_file('nombre-archivo.pdf', 'PDF')

Para executar a lista, na guia do objeto, botão direito / **Run with this only**



Notou no que lhe informa a lista de navegação do procedimento?

E se agora necessita que a lista saia ordenada por nome de país? Implemente-o, observe o que é informado na lista de navegação e teste-o.

E se agora necessita somente listar os parques de diversões da China? Teste (observando a lista de navegação).




### Amusement Parks List

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



Em cada caso, descubra qual tabela da base de dados se está percorrendo para realizar a consulta do *for each*.

Também é necessária uma lista como a que segue (que mostre cada categoria, e para cada uma delas, suas atrações). Implemente-a e teste o que foi feito.






### 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	

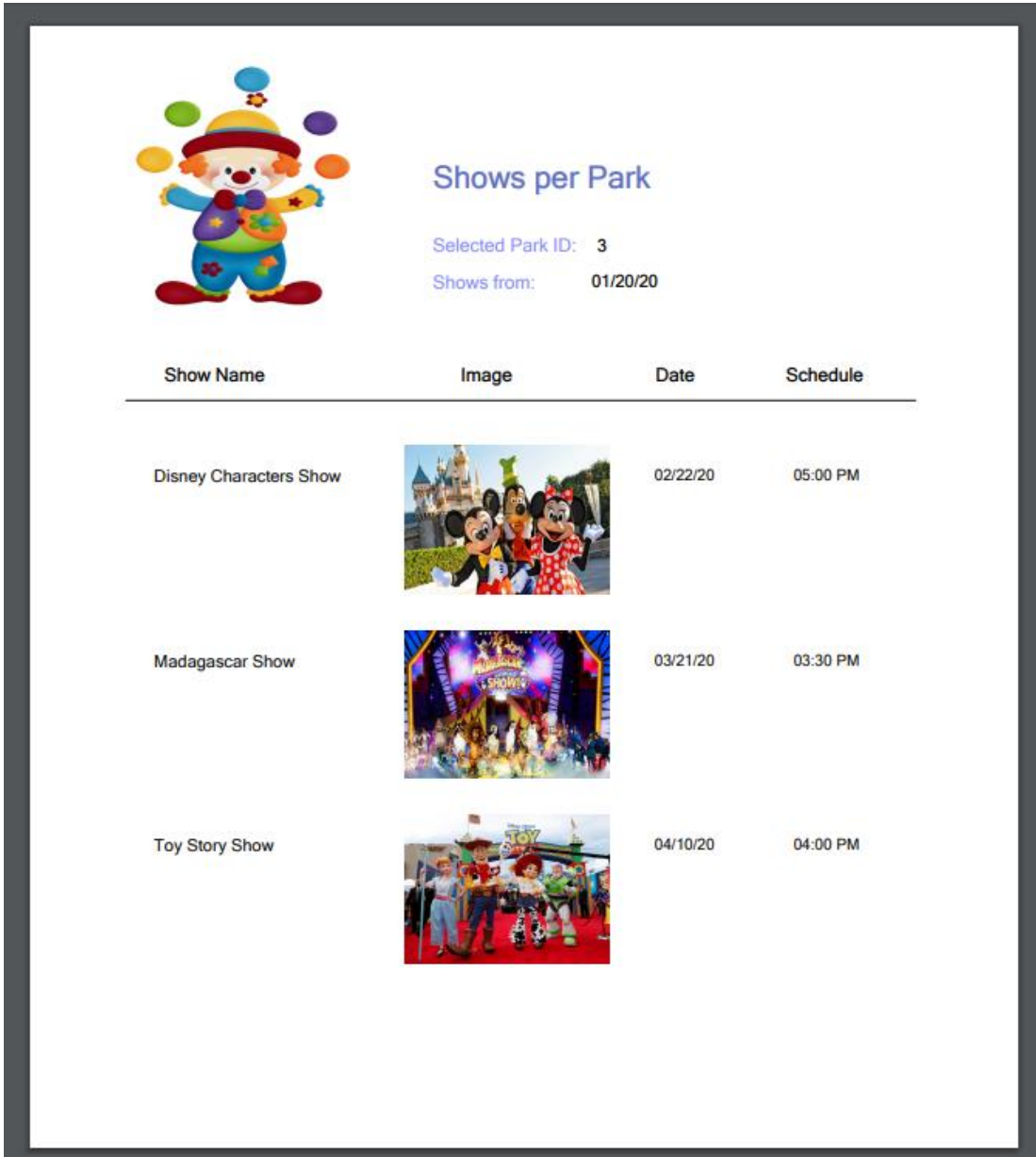
Adicione uma nova categoria ao sistema, por exemplo *Aquatic* (aquáticos). Execute novamente a lista anterior. Foi listada a categoria?

Modifique a lista anterior para que não sejam listadas categorias que não tenham atrações relacionadas.

Que modificações encontrou na lista de navegação?




**Resposta:** O surgimento da palavra **break** que indica que está sendo realizado um  **corte de controle**, mostrando também que a tabela base do *for each* externo é igual àquela do *for each* aninhado.

Outra lista necessária é a dos shows que são oferecidos a partir de uma data indicada pelo usuário, em determinado parque de diversões também selecionado pelo usuário.



**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

Outra solicitação da empresa é uma lista que mostre todos os nomes de países e, para cada país, a quantidade de parques de diversões que oferece:



Countries list

Country	Quantity
Brazil	2
France	1
China	2
United States	3

E nos pedem outra lista que mostre todos os países que têm mais de 2 parques para visitar:



Countries list

Country	Quantity
United States	3

## PASSAGEM DE PARÂMETROS

Muitas vezes necessitamos que um objeto receba parâmetros para que, com base neles, execute sua lógica. Por exemplo, uma coisa é fazer uma lista *PDF* de todos os parques de diversões da base de dados, e outra é fazer uma lista daqueles cujos nomes estão dentro de um determinado intervalo.

### LISTA DE PARQUES EM UM INTEVALO DETERMINADO

Grave com outro nome o procedimento que criou anteriormente para listar os parques de diversões e modifique-o para que agora liste apenas aqueles cujo nome esteja dentro de um intervalo recebido por parâmetro (o valor inicial e o final do intervalo serão os parâmetros recebidos).

Implemente uma tela que solicite os valores desse intervalo ao usuário e chame este objeto, passando-lhe esses valores por parâmetro. Teste em execução.

#### **Lembre-se** de que:

- Se define uma variável baseada em (**based on**) um atributo, ficará vinculada ao tipo de dados do atributo, ou seja, se este for modificado, o da variável também, de acordo com a alteração.
- As variáveis utilizadas para fazer uma chamada no objeto que chama e aquelas usadas para declarar os parâmetros que são recebidos no objeto chamado não precisam coincidir em nome, mas devem ter tipos de dados compatíveis.

#### *Sugestões:*

- Definir a regra Parm correspondente no procedimento *AmusementParksFromTo* para que receba os valores a consultar.
- Criar um Web Panel com duas variáveis para que o usuário final possa inserir os valores a consultar e sejam, então, enviadas por parâmetro ao procedimento.

## BUSINESS COMPONENTS

Realizaremos algumas operações na base de dados, através de *Business Components*.

### AUMENTO DE PREÇO DOS REPAROS

A empresa se reserva o direito de umentar os preços dos reparos em um determinado percentual a partir de determinado momento. Para fazer uma gravação massiva, testaremos esta funcionalidade. Para fazer isso, deveremos implementar uma **tela** que permita ao usuário

especificar esse **percentual de aumento** e dar a **ordem** para aplicá-lo a todos os reparos da base de dados. **Implemente e teste.**

**Lembre-se** de que:

- O objeto Web Panel permite implementar telas flexíveis para entrada e saída de informações.
- Para a inserção de informações (que o usuário possa inserir valores na tela), pode inserir a partir da *Toolbar* um controle *Attribute/Variable* no Form e atribuir a ele uma variável.
- Se quiser editar barras de menu, posicionando-se em GeneXus acima, na barra, com botão direito poderá inserir, por exemplo, a barra *Formatting*.
- Para que o Web Panel execute uma ação determinada, podem ser inseridos botões e programar o "evento" associado.
- Os *Business Components* são tipos de dados que são criados ao configurar a propriedade de nome *Business Component* do objeto transação com valor *Yes*. Ao fazer isso, para inserir, modificar ou excluir registros das tabelas correspondentes, poderá ser utilizada além da transação, uma variável do tipo de dados *Business Component* em qualquer outro objeto (por exemplo, um Web Panel) e realizar as mesmas operações por meio de dos métodos *Load()*, *Insert()*, *Update()*, *Save()* e *Delete()*.
- Para que as operações realizadas através do *Business Component* sejam efetuadas permanentemente, deverá executar em seguida o comando *Commit*.
- Se para um valor X quiser aumentá-lo em 20%, consegue-se fazendo  $X = X * (1 + 20/100) = X * 1,20$

**Sugestões:**

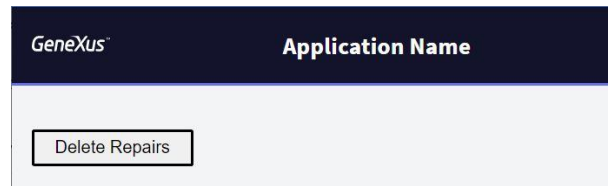
- Crie um Web Panel com uma variável *&percentage*, de tipo de dados: *Numeric(3.0)*, e arraste um controle do tipo *Button* no Form.
- No evento associado ao botão, programe a lógica que deseja que seja executada quando o usuário pressione o referido botão.
- Lembre-se que se deseja percorrer todos os reparos, e para cada um deles, utilizando o conceito de *Business Component*, modificar seu preço aumentando-o no percentual indicado pelo usuário na variável da tela.
- Lembre-se também que, como as operações de gravação ou exclusão da base de dados (por meio dos métodos *Insert()*, *Update()*, *Save()* e *Delete()*) podem causar erros, é importante saber o que aconteceu. Para isso temos o método *Success()* que devolverá *True* se não houve erros, e *False* caso contrário.
- Para indicar que "o que foi feito fique permanente", deve ser executado o comando *Commit*. Se pelo contrário deseja desfazer, deve ser executado o comando *Rollback*.

Atualize as mudanças no GeneXus Server.

**TELA PARA ELIMINAÇÃO DE TODOS OS REPAROS**

Salve o Web Panel anterior com outro nome (para isto basta posicionar-se na aba e com o botão direito, fazer **Save as**) e modifique o Form para que contenha apenas um botão com o texto *Delete Repairs*, de forma que em execução este Web Panel seja visto assim:





Quando o usuário pressione o botão, deverão ser eliminados todos os reparos da base de dados.

O que deve modificar do evento *Confirm* que tinha programado?

**Nota:** Se posiciona-se sobre o botão e vê suas propriedades, na de nome *Caption* pode modificar o texto do botão.

---

Sugestões:

- Utilize o método *Delete()* que se aplica a variáveis do tipo Business Component (simples ou coleções).
- Lembre-se de que utilizando o comando *Msg*, é possível exibir uma mensagem no Form do Web Panel.

Atualize as mudanças no GeneXus Server.

## PROCEDIMENTOS PARA ATUALIZAR REGISTROS

### AUMENTO DE PREÇOS DOS REPAROS

Suponha que os reparos para os quais deve aumentar o preço em uma determinada porcentagem sejam milhares. Sabendo que o aumento de preço é um procedimento simples que não fará falhar a integridade de forma alguma, exercite resolver com um procedimento, sem utilizar *Business Components*.

**Lembre-se** de que:

- Com o comando *for each* dentro de um procedimento, pode atualizar os atributos de sua tabela estendida através de atribuições simples.
- A atualização "direta" através de procedimentos não controla a integridade das informações.
- Todo objeto deve declarar os parâmetros que recebe e os parâmetros que retorna. Se não os declarar, não receberá nem retornará valores.
- Os parâmetros são declarados através da regra **parm**.
- As variáveis são locais para o objeto onde são utilizadas. Isto significa que se eu quiser receber um valor como parâmetro em uma variável &X, devo declará-la no objeto.

## ELIMINAÇÃO DE TODOS OS REPAROS

E se agora quisesse eliminar todos os reparos, como foi feito anteriormente no prático, mas desta vez através de um procedimento?

---

Sugestão:

- Criar um procedimento *RepairsDeletion* que não receba parâmetros, e que se encarregue de percorrer e eliminar todos os registros de reparos. Utilize o comando Delete.
- Faça um Save as do Web Panel que implementou e programe a eliminação através do Business Component.

E se agora quer excluir todas as informações da base de dados?

Sugestão

- Criar um procedimento *DeleteAll* que se encarregue de percorrer e apagar todos os registros das diferentes tabelas associadas às transações definidas.

Lembre-se de que os procedimentos não validam a consistência dos dados, mas a base de dados o faz. Ou seja, a base de dados valida a consistência dos dados inter-relacionados, portanto, a ordem na qual tente eliminar os dados é importante.

Por exemplo, se você tentar eliminar os países antes dos parques, a base de dados o impedirá, o programa será cancelado e não será amigável para o usuário.

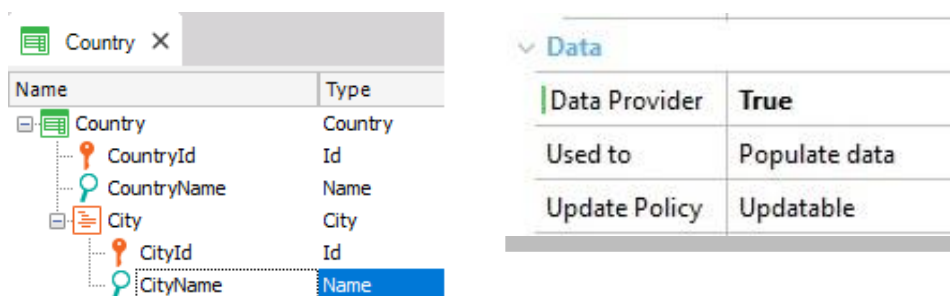
## INICIALIZAÇÃO DA INFORMAÇÃO DA BASE DE DADOS [OPCIONAL]

Quando a aplicação que você está desenvolvendo é colocada em produção (ou seja, comece a ser utilizada pela empresa) deverão ser carregados todos os dados de países, parques, shows, categorias, atrações, técnicos e reparos. Utilize as facilidades que o objeto *Transação* oferece para isto e inicialize essas tabelas com informações fornecidas pela empresa, e teste.

### **Levar em conta que:**

- As transações têm uma propriedade **Data Provider** que nos permite ter um Data Provider associado à transação, de modo que quando a tabela é criada na base de dados, seja inicializada com os valores que definamos no referido Data Provider. Para isso, também deverá configurar a propriedade **Used to** e **Update Policy**.
- Deve inserir as imagens na KB para poder utilizá-las. Uma maneira de fazer isso é na janela *KB Explorer* ir ao nó *Customization* e escolhendo *Images*, onde são listadas todas as imagens atualmente armazenadas na KB, inserir uma nova (a partir de um arquivo) e dar um nome a ela.

**Sugestão:** Como forma de testar diferentes opções, popularemos os países e as categorias utilizando a propriedade Data Provider das transações.



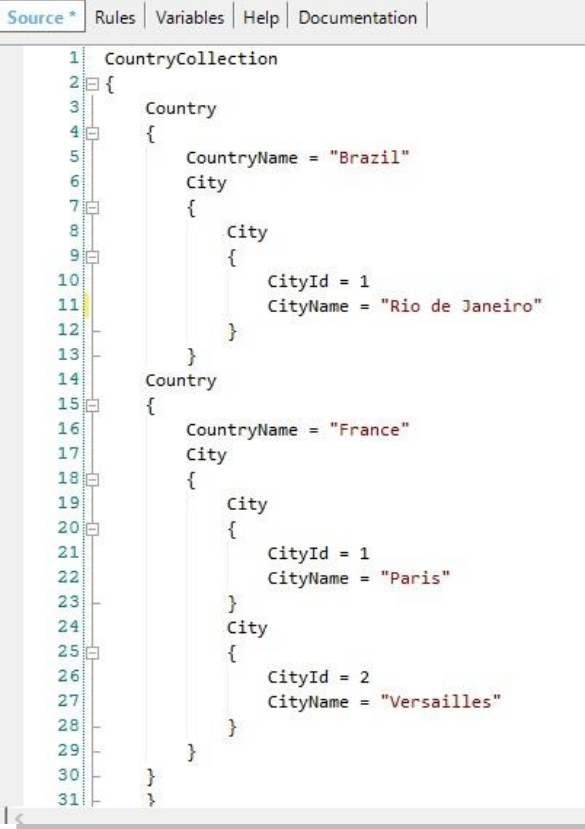
**Lembrar:** O Data Provider associado à transação voltará a disparar:

- Cada vez que a tabela correspondente seja reorganizada.
- Cada vez que seja editado o conteúdo do Data Provider.

Portanto, é necessário controlar que as informações não sejam duplicadas. Isto pode ser conseguido:

- Se a chave primária for autonumerada, pode ser definido um índice unique no atributo descritor.

- Deixando a chave primária sem autonumeração. Desta forma, seu valor será atribuído de forma fixa ao definir o Data Provider e não se duplicará.
- Carregue os dados desejados em Country\_DataProvider (pode vê-lo abaixo da transação Country na janela KB Explorer). No momento da criação, a tabela será carregada com os dados indicados.



```
Source * Rules Variables Help Documentation
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   }
```

- De maneira análoga, carregue as categorias.
- Atualize as mudanças no GeneXus Server.

## WEB PANELS

Solicita-se uma página que mostre todos os países e para cada um, a quantidade de parques de diversões que oferece visitar.

**Lembre-se** de que o evento **Load em um Web Panel com tabela base que possui um Grid** é executado antes de carregar cada linha no Grid. Esse evento é adequado para atribuir à variável o cálculo que retorna a conta de parques de diversões de cada país que está sendo navegado e prestes a carregar em uma linha de Grid.

Agora, adicione duas variáveis ao Web Panel definido (&CountryNameFrom e &CountryNameTo) e defina as condições necessárias para filtrar os países incluídos no referido intervalo.

Solicita-se outra página web que apresente a informação dos Shows: Nome do show, sua imagem representativa, o parque ao qual está associado, data, horários; ordenado pelo nome do Show.

Deseja-se poder filtrar a informação de acordo com o parque em que será apresentado o Show, caso não selecionar nenhum parque, deverá mostrar todos os Shows.

Outro requisito é um link que permita mostrar outra tela como pop-up onde seja visualizada a informação do show selecionado.

The screenshot displays a GeneXus application interface. At the top, the GeneXus logo and 'Application Name' are visible. The main content area is divided into two sections. On the left, there is a table titled 'Amusement Park Id' with columns for 'Show Name' and 'Show Image'. The table lists four rows: 'Dance' with a show image, 'Dance' with a show image, 'Kids' with a show image, and 'Magic' with a show image. On the right, there is a table titled 'Schedule Show Detail' with columns for 'Schedule' and 'Show Detail'. The table lists four rows: '00 PM' with a 'Detail' link, '30 PM' with a 'Detail' link, '00 AM' with a 'Detail' link, and '00 PM' with a 'Detail' link. A pop-up window titled 'One Show Detail' is overlaid on the right side of the interface. The pop-up window contains the following information: 'Amusement Park Name: Happy Valley', 'Amusement Park Address: Beijing Olympic Tower', 'Show Name: Dance', 'Show Image: [Image of a show]', and 'Amusement Park Show Date: [Date]'. The pop-up window has a close button (X) in the top right corner.

**SUGESTÃO:**

Para exibir um pop-up em tela, utilizar a seguinte sintaxe: Objeto.Popup(Parâmetros)

---

É desejado visualizar em uma tela web um ranking de acordo com os Shows que possui cada Parque.

NOTA: não é necessário definir um atributo fórmula no nível da estrutura da transação Parque.

Amusement Park Name	Shows Quantity
Beto Carrero World	4
Happy Valley	2
Par Du Bocasse	1

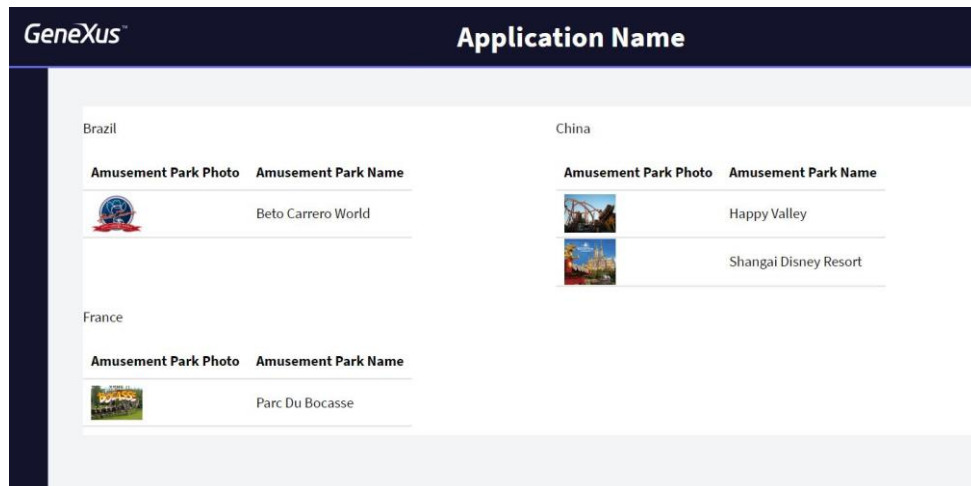
**SUGESTÃO:**

Definir uma estrutura de dados em memória que permita armazenar a coleção de parques para, então poder ordená-la do maior ao menor pela quantidade de Shows e exibi-la em um Web panel

**MÚLTIPLOS GRIDS [OPCIONAL]**

Deseja-se ver uma tela que liste as informações de todos os países com seus respectivos parques relacionados.

No Web Panel a ser definido, coloque um Grid Free Style e dentro dele, um Grid; cada um deles com seus atributos correspondentes, de maneira que as informações apareçam da seguinte forma:



Qual é a transação base do Grid externo?

Qual é a transação base do Grid aninhado?

Adicione um novo país e volte a executar o Web Panel.

Que alterações deve realizar se quer ver na tela apenas os países que tenham parques de diversões relacionados?

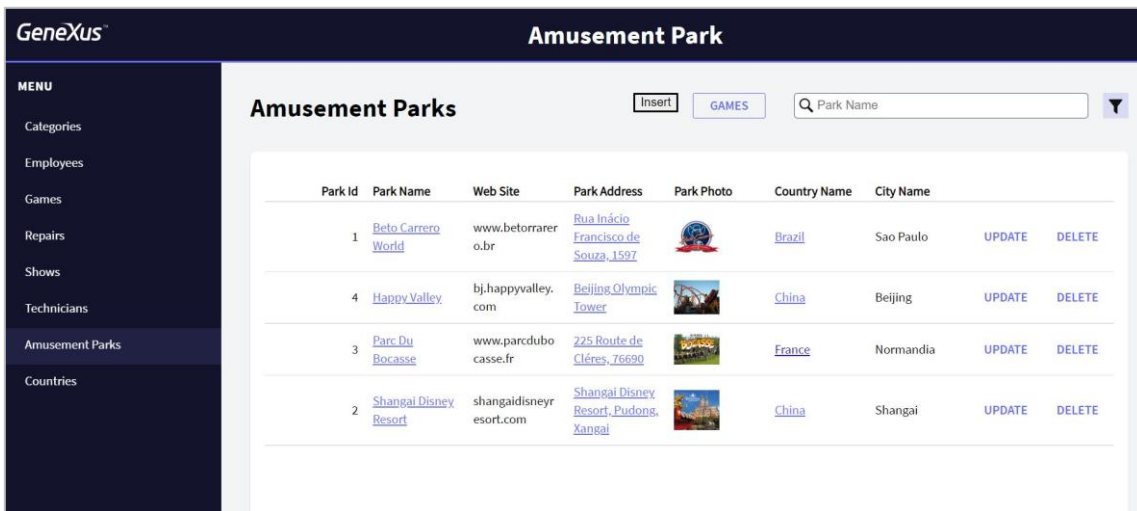
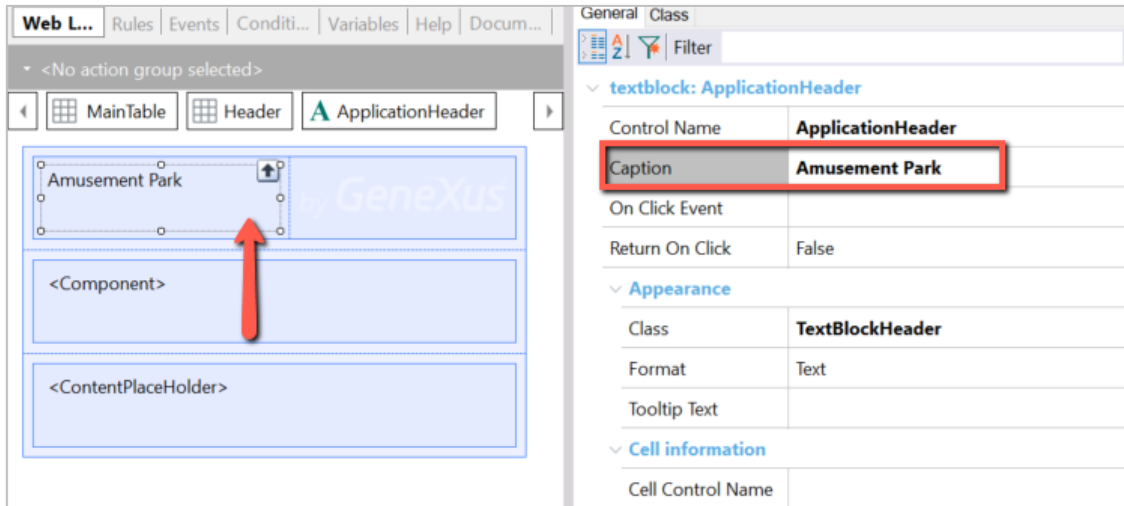
## TIPOS DE WEB PANELS

Dirija-se ao KB Explorer e procure no filtro “*RWD Master Page*”. Esta é a Master Page associada a todos os objetos GeneXus de nossa KB.

Observe que conta com três seções:

- Um Header onde podemos localizar o nome da aplicação e o logotipo.
- Um User Control SideBar que é o encarregado de nos oferecer o menu dos objetos que possuem o padrão WorkWith for Web
- Um Content Place Holder onde são exibidos os objetos de nossa aplicação.

No *Header*, selecione o controle *ApplicationHeader* e altere a propriedade *Caption* para o nome da aplicação: Amusement Park:

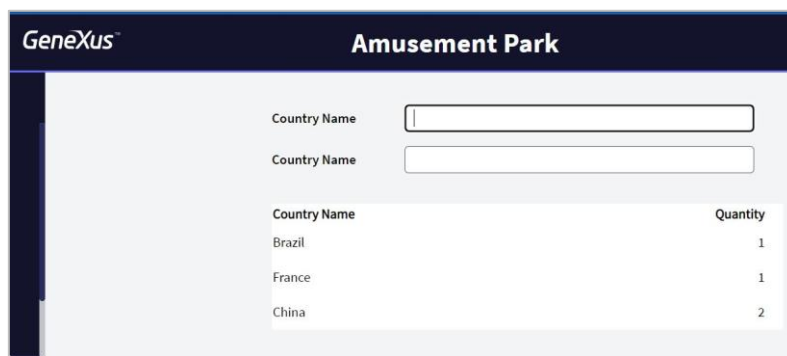
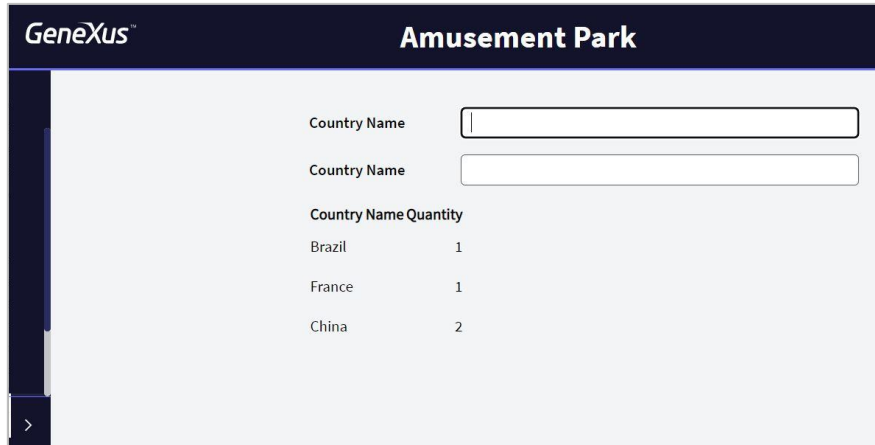


## DESIGN SYSTEMS

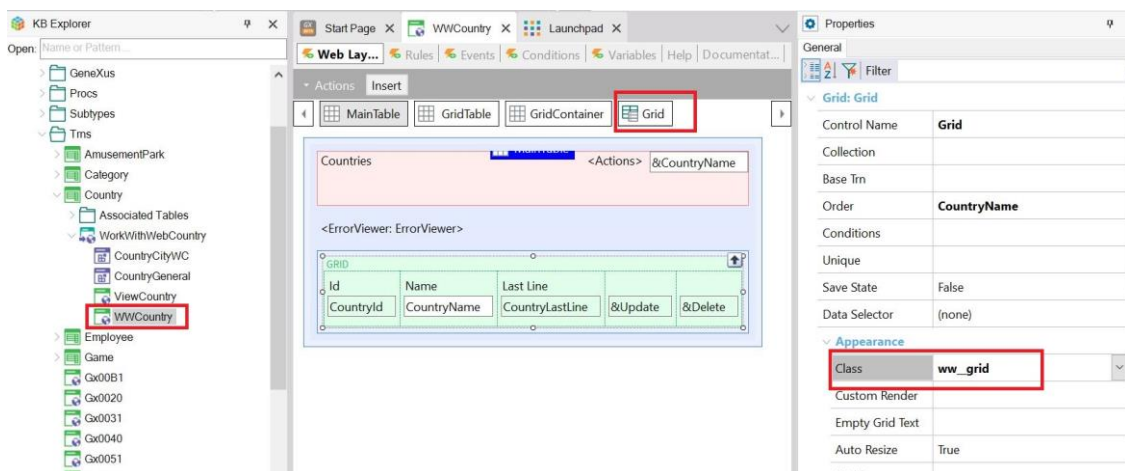
Abra o Web Panel onde foi construída a solução para mostrar todos os países e, para cada um, a quantidade de parques de diversões que oferece visitar.

Deseja-se que este Web Panel tenha o mesmo formato apresentado nos Grids do WorkWithForWeb:

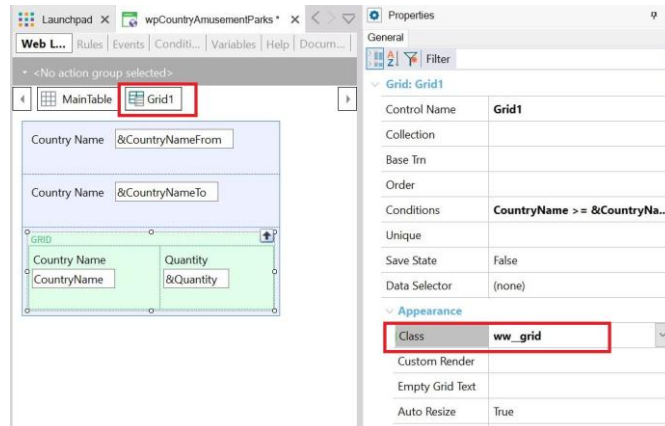




Em algum Web Panel *WW* associado ao *WorkWith*, observe que nas propriedades do Grid há uma classe atribuída chamada *ww\_grid*:

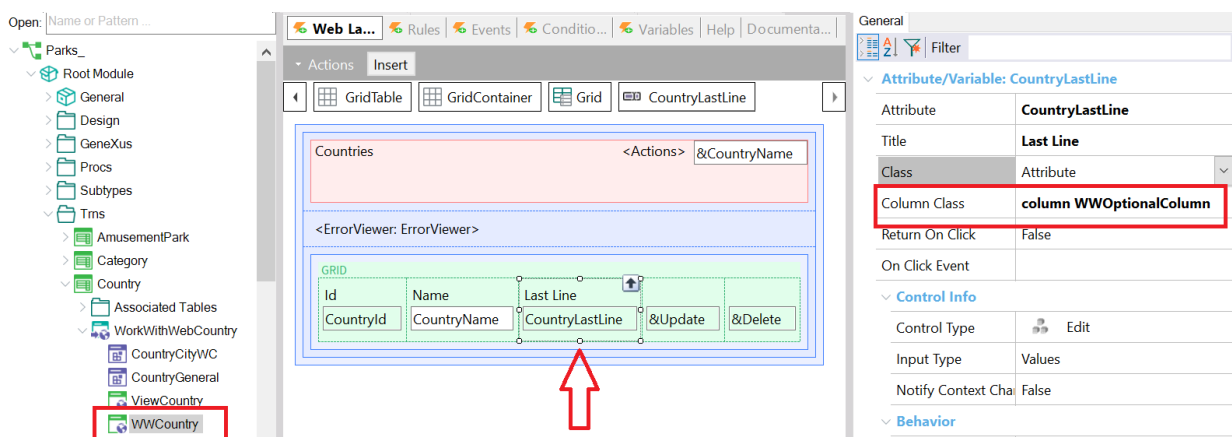
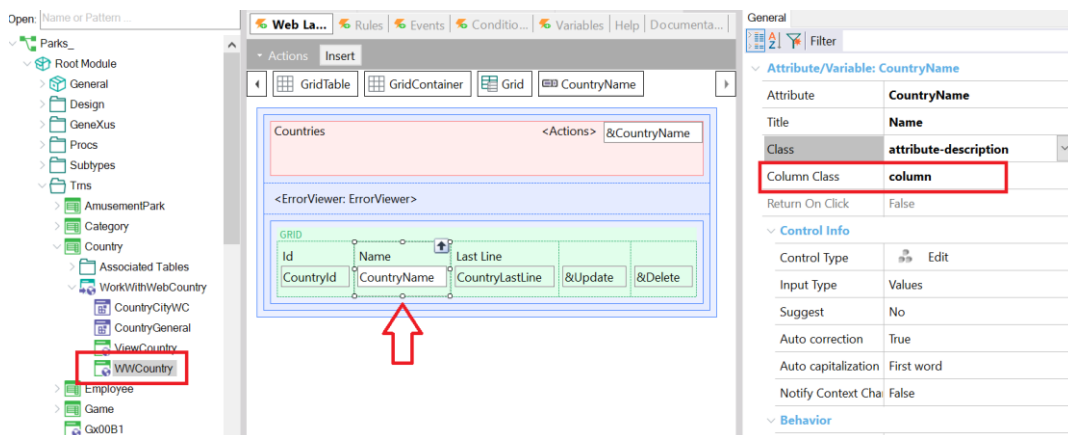


Aplice esta mesma classe ao controle Grid do Web Panel ao qual daremos formato:

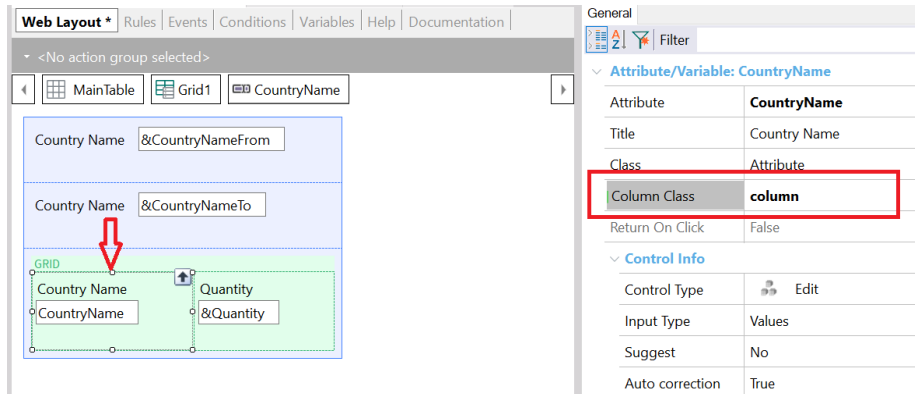


Para manter a estética, observe que cada coluna do Grid WorkWith tem uma classe associada:

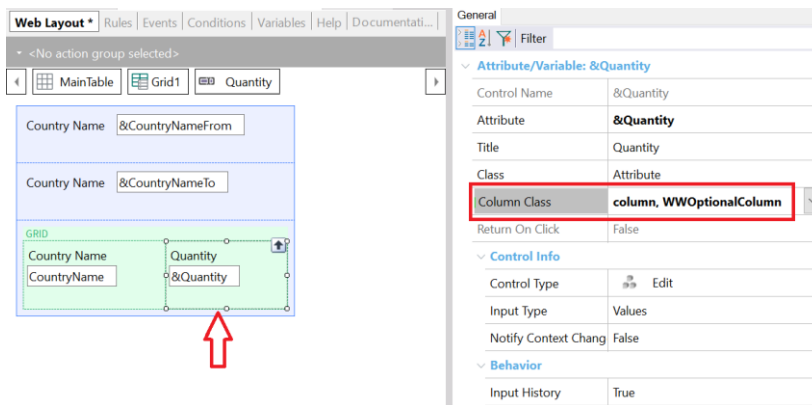
- *Column* para os atributos que têm maior importância que sejam exibidos em todos os momentos, independentemente do tamanho da tela.
- *Column* e *WVOptionalColumn* para atributos que ficam ocultos quando a tela fica menor.



Para obter o mesmo comportamento no Web Panel, escolha o nome do país e na propriedade *ColumnClass* coloque *Column*:

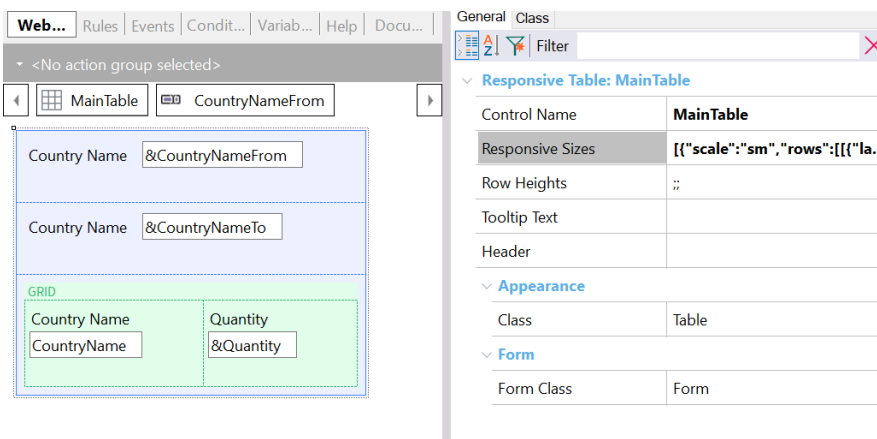


Para ocultar a quantidade de parques quando a tela for menor, selecione o controle associado à variável e na propriedade *ColumnClass* coloque *Column* e *WWOptionalColumn*:



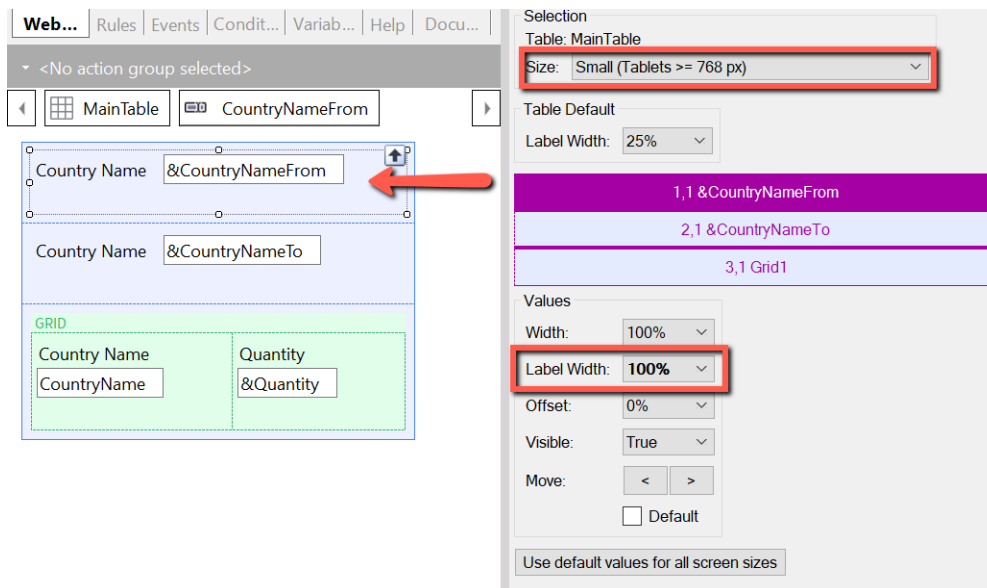
Execute a aplicação e teste o comportamento.

Por último, para que os rótulos apareçam acima da caixa editável, abra a caixa dos tamanhos responsivos. Caso não apareçam, selecione a Main Table do Web Panel e clique na propriedade Responsive Sizes:



Selecione a primeira variável, nos tamanhos responsivos selecione o tamanho Small e em *Label Width* defina 25%. Altere o tamanho para Extra Small e defina o mesmo para *Label Width*.

Faça este mesmo processo para a segunda variável:



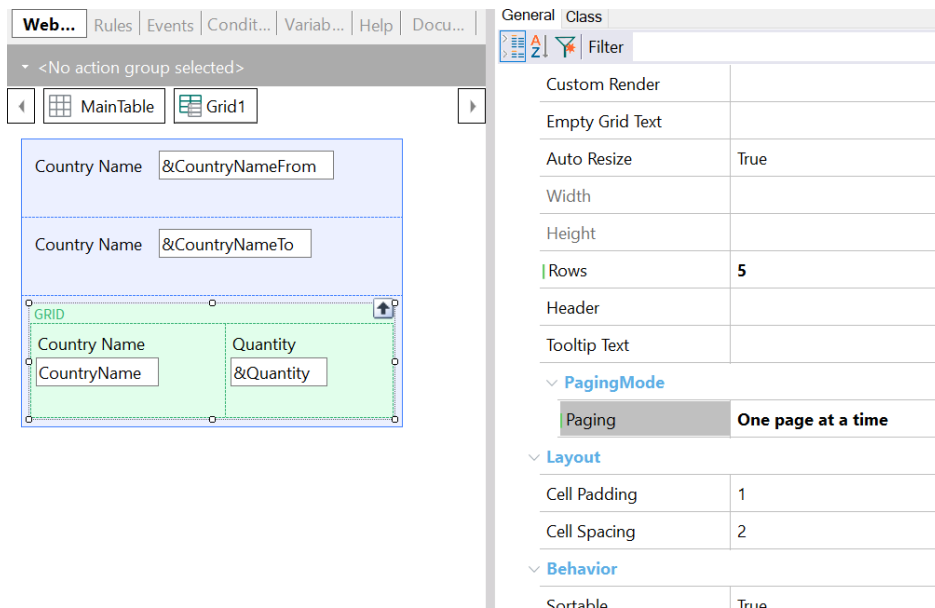
Execute a aplicação e verifique o comportamento.

## PAGINAÇÃO EM GRIDS

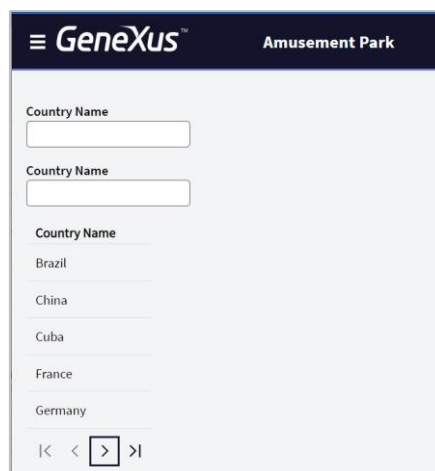
É requerido que a tela anterior tenha uma Paginação no Grid para visualizar os países de uma maneira melhor.

Para isto, adicione mais países.

Nas propriedades do Grid do Web Panel, altere a propriedade *Rows* para 5 e em *Paging* verifique que se encontra ativada *One page at a time*:



Execute e observe que agora são mostrados apenas 5 registros por página e graças às setas de navegação do Grid é possível percorrer os países:



O que aconteceria se não quisesse ver as informações em páginas diferentes, mas sim simular que o conteúdo está em uma única tela e fazer scroll para acessar os registros? Volte ao Web Panel e altere a propriedade *Paging* para *Infinite scrolling*.

Execute a aplicação e observe que através de um scroll pode se mover pelos registros:

GeneXus™ Amusement Park

Country Name

Country Name

Country Name

Brazil

China

Cuba

France

Germany

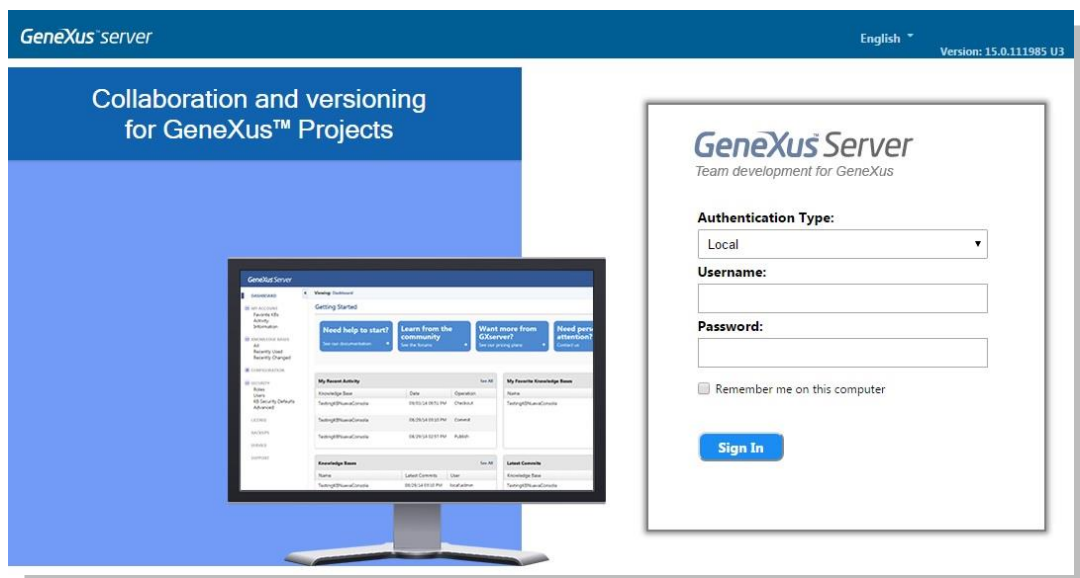
## GENEXUS SERVER

Publique a base de conhecimento no servidor <http://sandbox.genexusserver.com/v17>

Crie um novo Web Panel que mostre a lista de parques de diversões registrados.

Envie este novo objeto ao server para que seja integrado à base de conhecimento centralizada.

Entre no console web e verifique o estado final da KB.




Feche a base de conhecimento anterior e crie uma nova sincronizando-se com a KB publicada anteriormente. Desta forma, receberá localmente uma cópia da KB administrada pelo GeneXus Server.

Send Knowledge Base

### Send the Knowledge Base to GeneXus Server

To Send the Knowledge Base select the url and type an alias



Parks

Security

Authentication Type: GeneXus Account

Username:

Password:

Save Password

Alias:

Work with Lock Model

All versions in Knowledge Base

Nesta nova cópia local, edite a transação *Country* e defina o novo atributo *CountryFlagImage*, do tipo *Image*. Envie esta alteração para o servidor.

Feche esta *KB* e abra novamente a *KB* inicial. Execute a operação *Update* para receber a alteração realizada anteriormente.