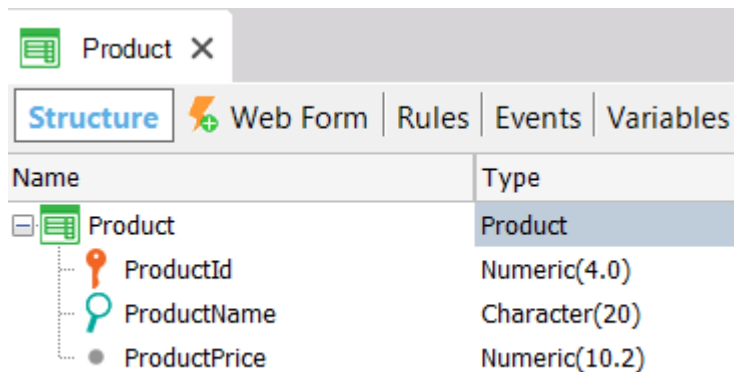


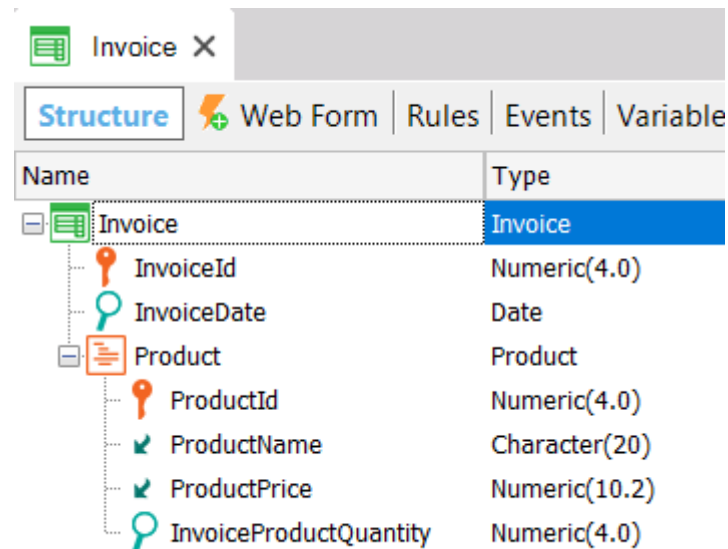


# Transações

## Quais nomes de atributos usamos?



Name	Type
Product	Product
ProductId	Numeric(4.0)
ProductName	Character(20)
ProductPrice	Numeric(10.2)



Name	Type
Invoice	Invoice
InvoiceId	Numeric(4.0)
InvoiceDate	Date
Product	Product
ProductId	Numeric(4.0)
ProductName	Character(20)
ProductPrice	Numeric(10.2)
InvoiceProductQuantity	Numeric(4.0)

## Nomenclatura GIK

Entidade + Categoria [+ Qualificador]

### Product

```
{  
  ProductId* (PK)  
  ProductName (S)  
  ProductPrice (S)  
}
```

### Invoice

```
{  
  InvoiceId* (PK)  
  InvoiceDate (S)  
  -----  
  Product  
  {  
    ProductId* (PK, FK)  
    ProductName (I)  
    ProductPrice (I)  
    InvoiceProductQuantity (S)  
    -----  
  }  
}
```

# Desenho de Transações

## 1 - N forte

Cada cliente pertence a um pais e um pais tem muitos clientes



COUNTRY
CountryId*
CountryName



**Country**  
{  
CountryId\* (PK)  
CountryName  
}

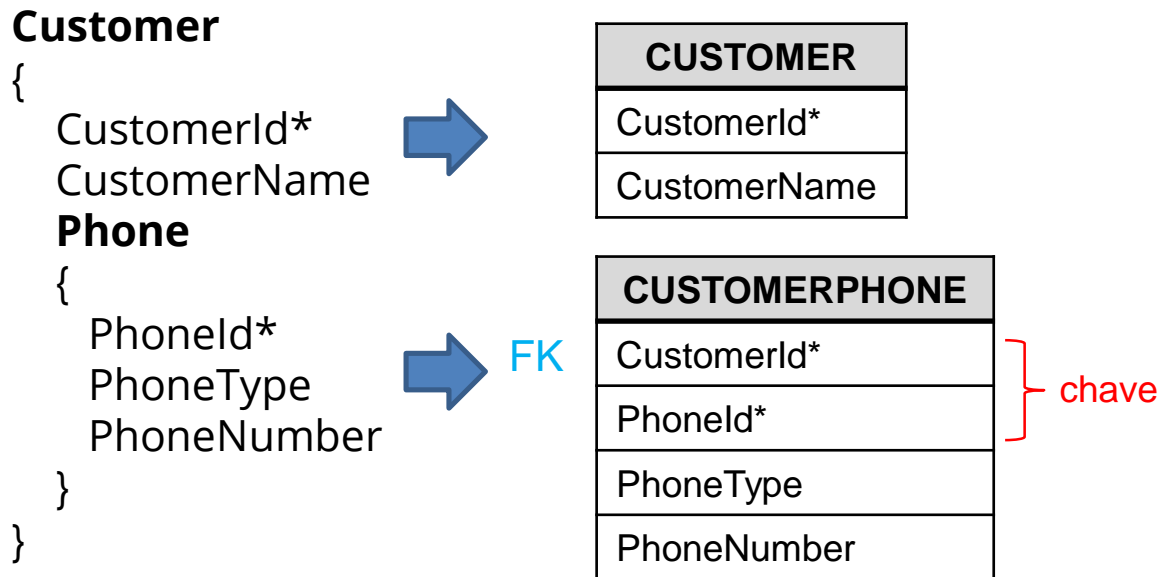
**Customer**  
{  
CustomerId\*  
CustomerName  
CountryId (FK)  
CountryName  
}



CUSTOMER
CustomerId*
CustomerName
CountryId

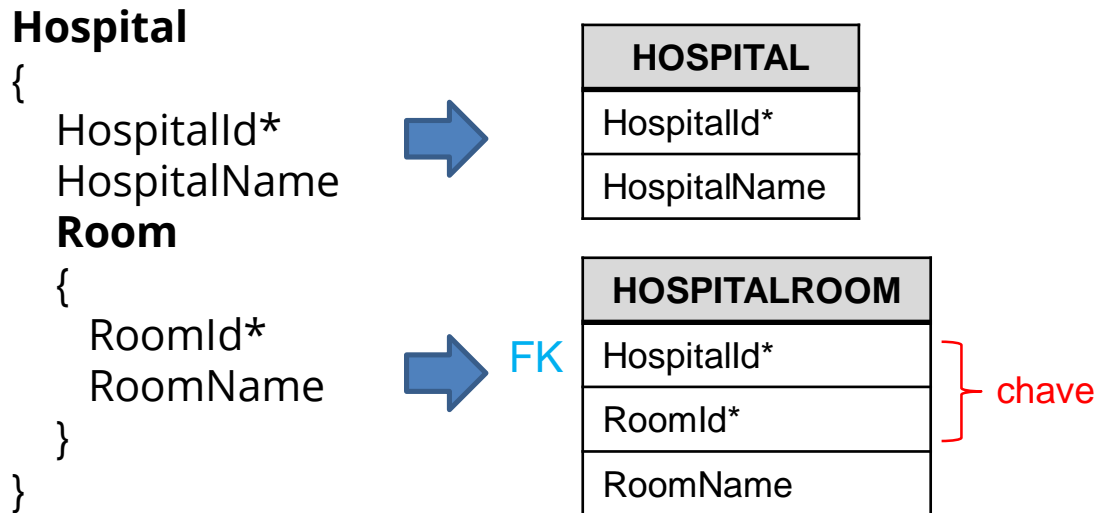
## 1 - N fraca

Cada cliente tem muitos telefones e cada telefone pertence a um único cliente



## 1 - N fraca

Cada hospital tem muitas salas e cada sala pertence a um único hospital





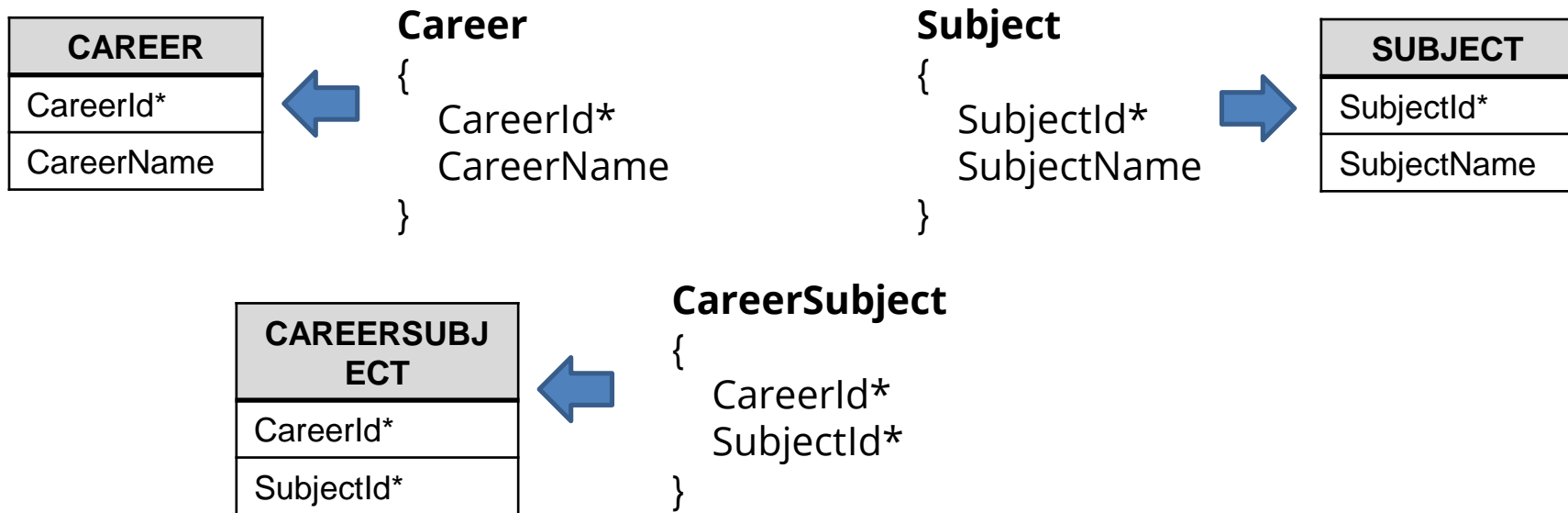
## N - N (M)

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



## N - N: Opção 1 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



## N - N: Opção 1 - Tabelas geradas

CAREER

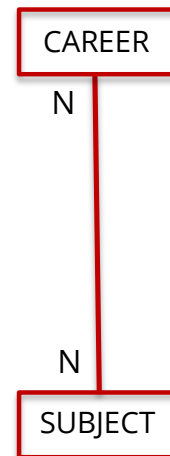
CareerId	CareerName
1	Computer Science
2	Data Science for Health

SUBJECT

SubjectId	SubjectName
1	Computer Logic
2	Programming Fundamentals

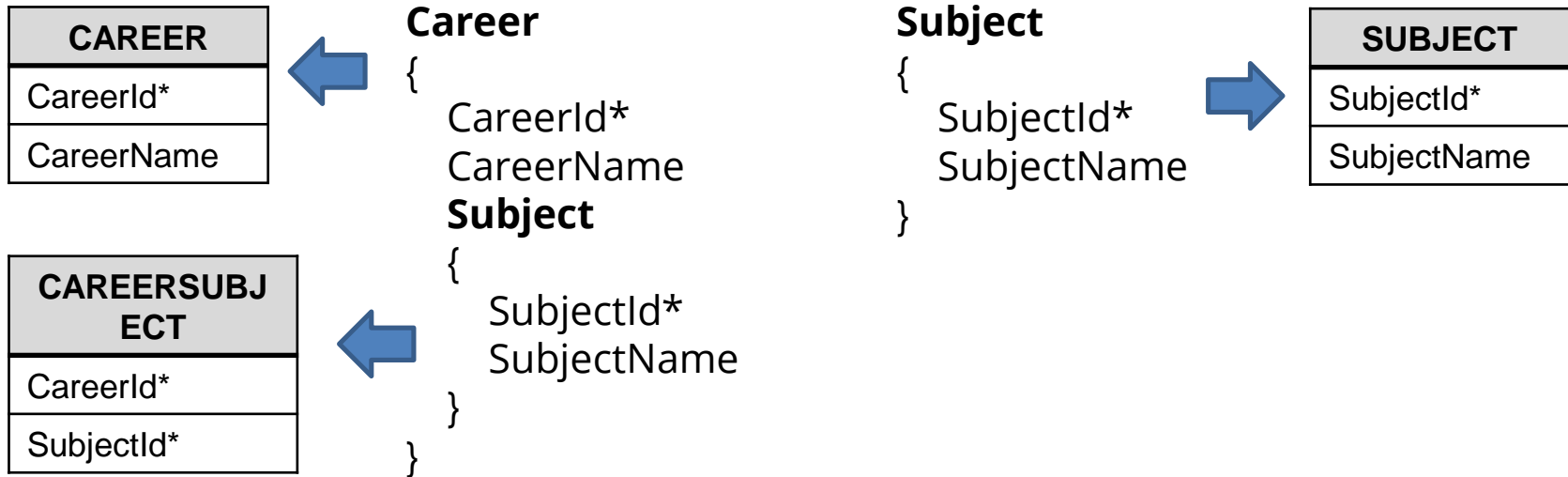
CAREERSUBJECT

CareerId	SubjectId
1	1
1	2
2	2



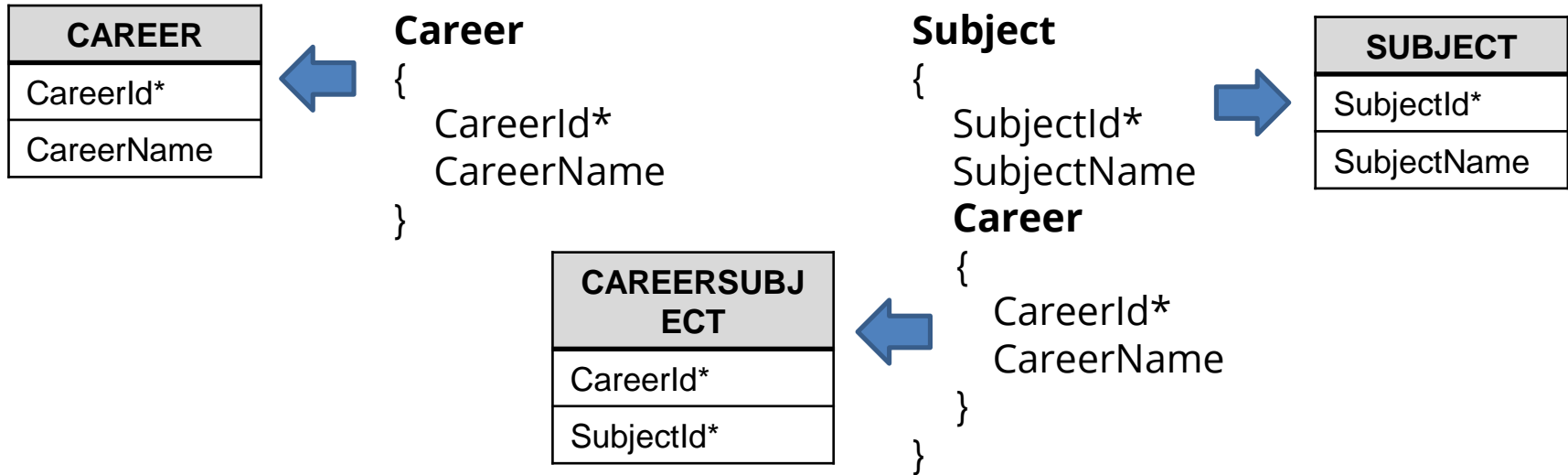
## N - N: Opção 2 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



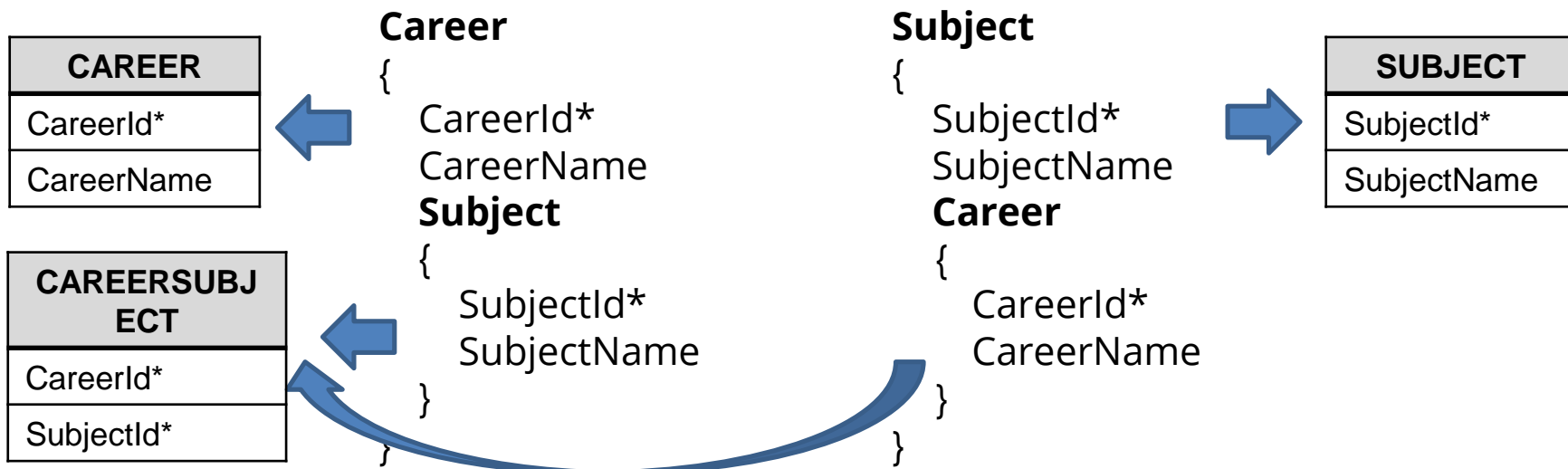
## N - N: Opção 3 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



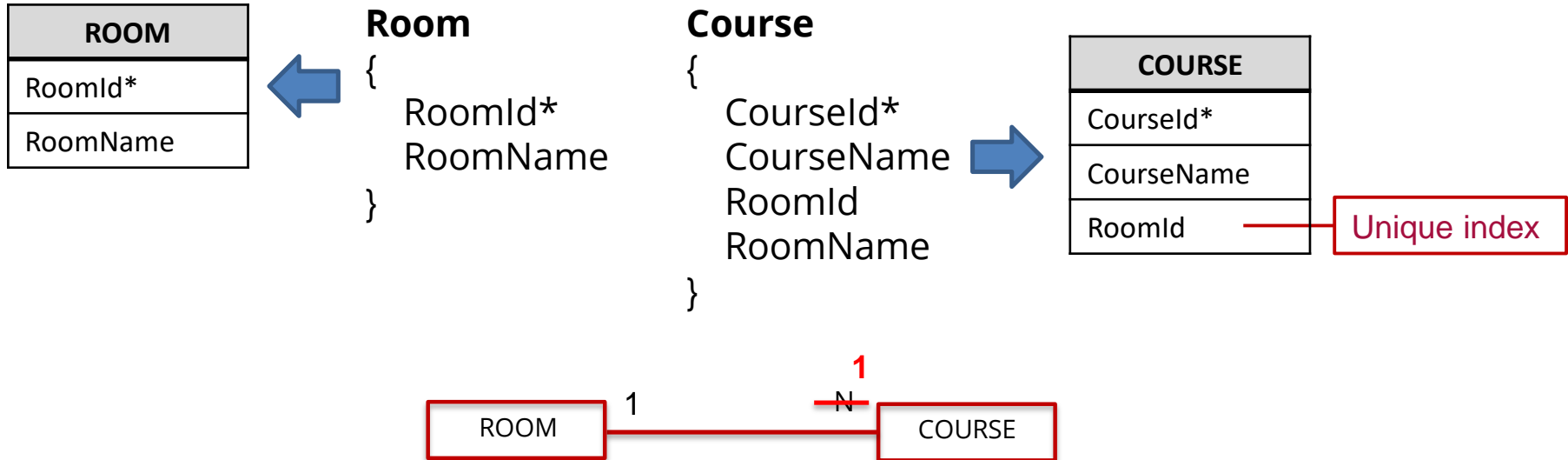
## N - N: Opção 4 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



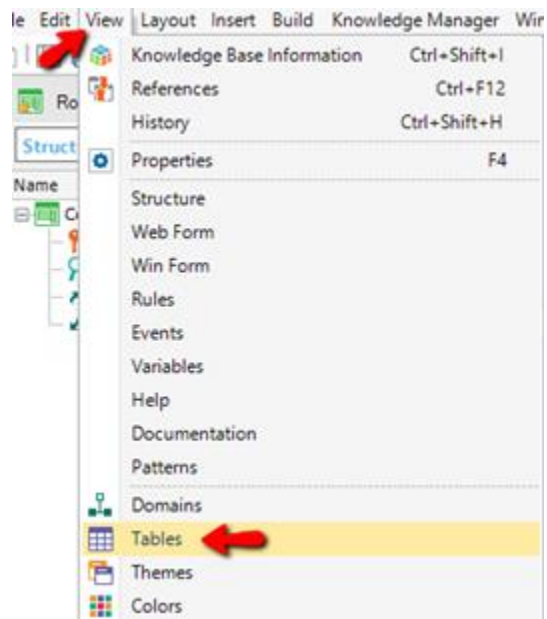
1 - 1

Cada curso é ministrado em uma sala e esta sala só pode ser usada para ministrar esse curso

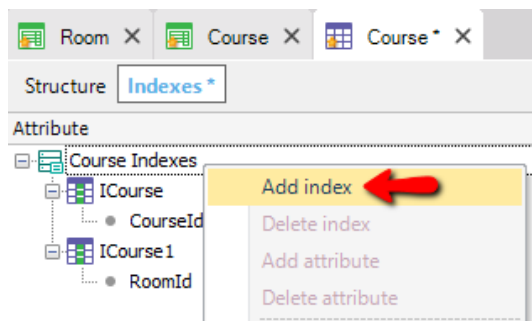


# Criação de um índice

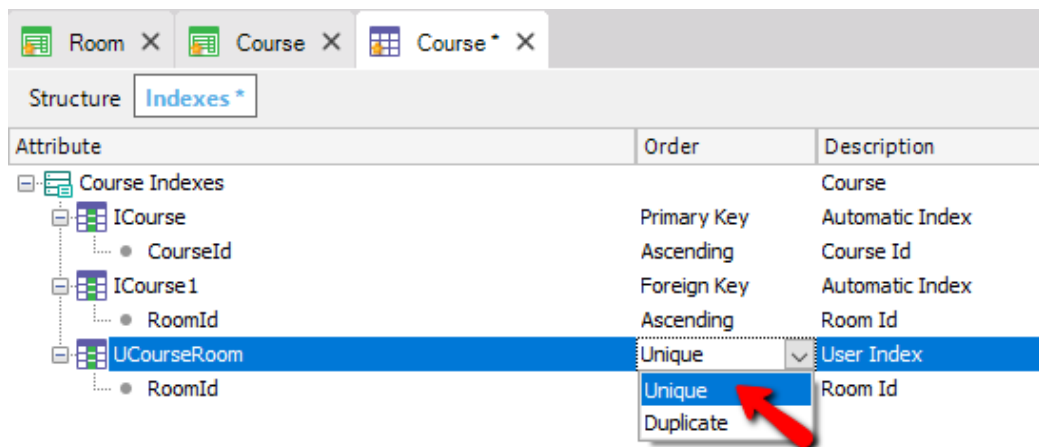
1



2



3





# Normalização

## GeneXus normaliza as tabelas na Terceira Forma Normal (3NF)

- Atributos inferidos em uma transação não são incluídos na tabela gerada

### Continent

```
{  
  ContinentId* (PK)  
  ContinentName  
}
```



CONTINENT
ContinentId*
ContinentName

### Country

```
{  
  CountryId* (PK)  
  CountryName  
  ContinentId (FK)  
  ContinentName (INF)  
}
```



COUNTRY
CountryId*
CountryName
ContinentId

### Customer

```
{  
  CustomerId* (PK)  
  CustomerName  
  CountryId (FK)  
  CountryName (INF)  
  ContinentId (INF)  
  ContinentName (INF)  
}
```



CUSTOMER
CustomerId*
CustomerName
CountryId

# Integridade referencial

## Integridade referencial

### Country

```
{  
  CountryId* (PK)  
  CountryName  
}
```

CountryId	CountryName
1	URUGUAI
2	ARGENTINA

### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId (FK)  
  CountryName  
}
```


CustomerId	CustomerName	CountryId
1	ANA	1
2	PEDRO	2
3	LUIS	2
4	JOSE	3



O registro não é inserido ←

## Integridade referencial

**Country**  
{  
  CountryId\* (PK)  
  CountryName  
}

CountryId	CountryName
1	URUGUAI 
2	ARGENTINA



O registro não é eliminado

**Customer**  
{  
  CustomerId\*  
  CustomerName  
  CountryId (FK)  
  CountryName  
}

CustomerId	CustomerName	CountryId
1	ANA	1
2	PEDRO	2
3	LUIS	2
4	JOSE	3

# Tabela Base e Estendida

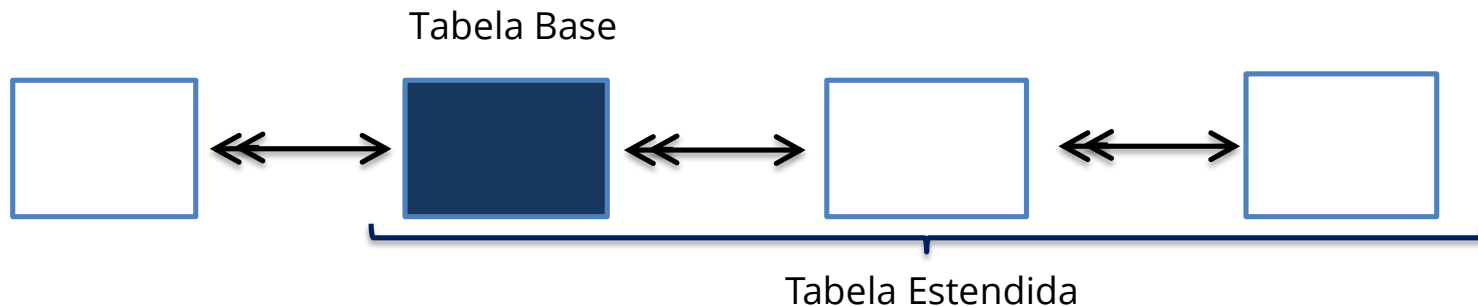
- **Tabela Base**

É qualquer tabela do banco de dados em que estamos posicionados trabalhando em um determinado momento.

- **Tabela Estendida**

Dada uma tabela, sua tabela estendida é um conceito que nos permite pensar todas as informações às quais temos acesso, a partir de suas chaves externas.

É o conjunto de atributos da própria tabela + todos os atributos das tabelas com as quais ela tem um relacionamento N-1 direta ou indiretamente.



## Exemplo

### Customer

```
{
  CustomerId*
  CustomerName
}
```

### Product

```
{
  ProductId*
  ProductName
  ProductPrice
}
```

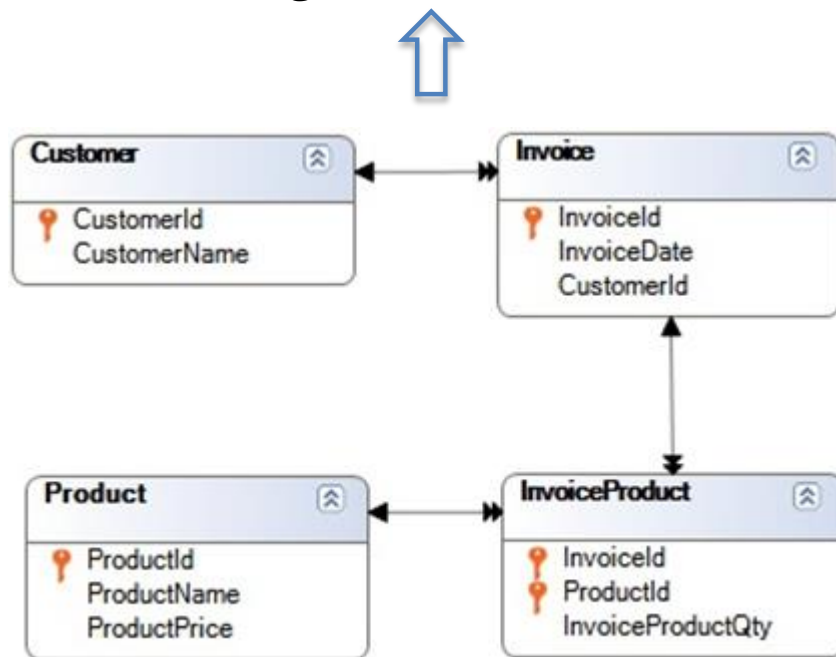
### Invoice

```
{
  InvoiceId*
  InvoiceDate
  CustomerId
  CustomerName
}
```

### Product

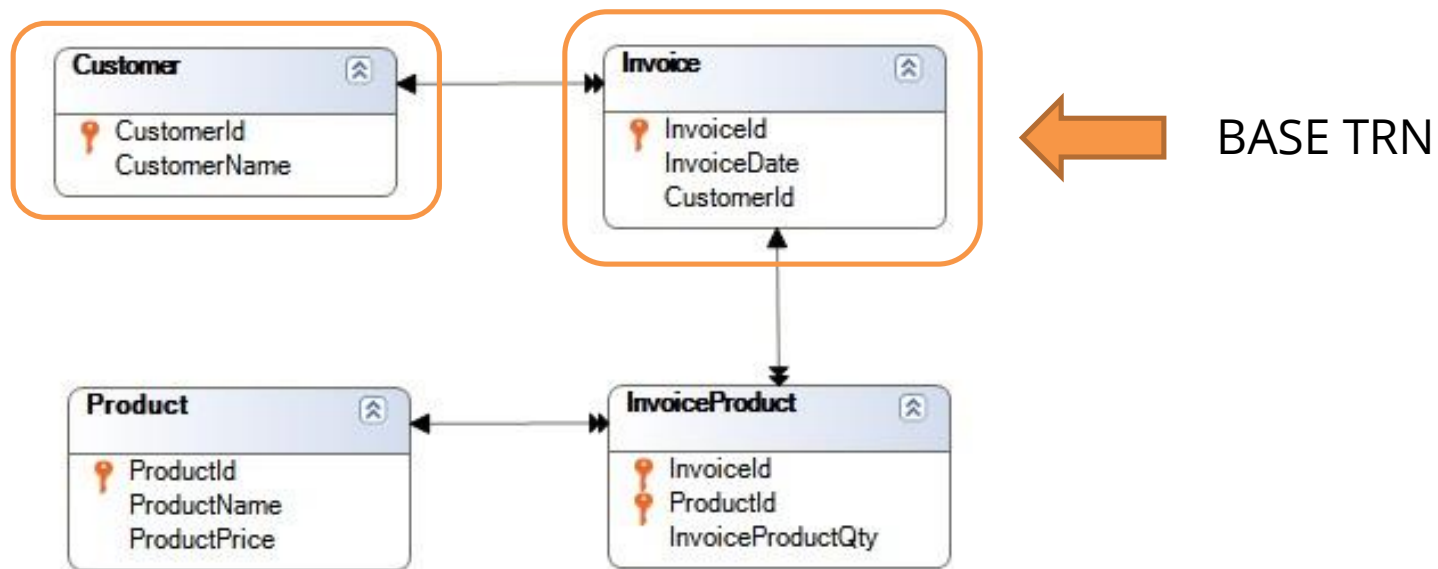
```
{
  ProductId*
  ProductName
  ProductPrice
  InvoiceProductQuantity
  -----
}
```

## Diagrama de Tabelas (diagrama de Bachman)

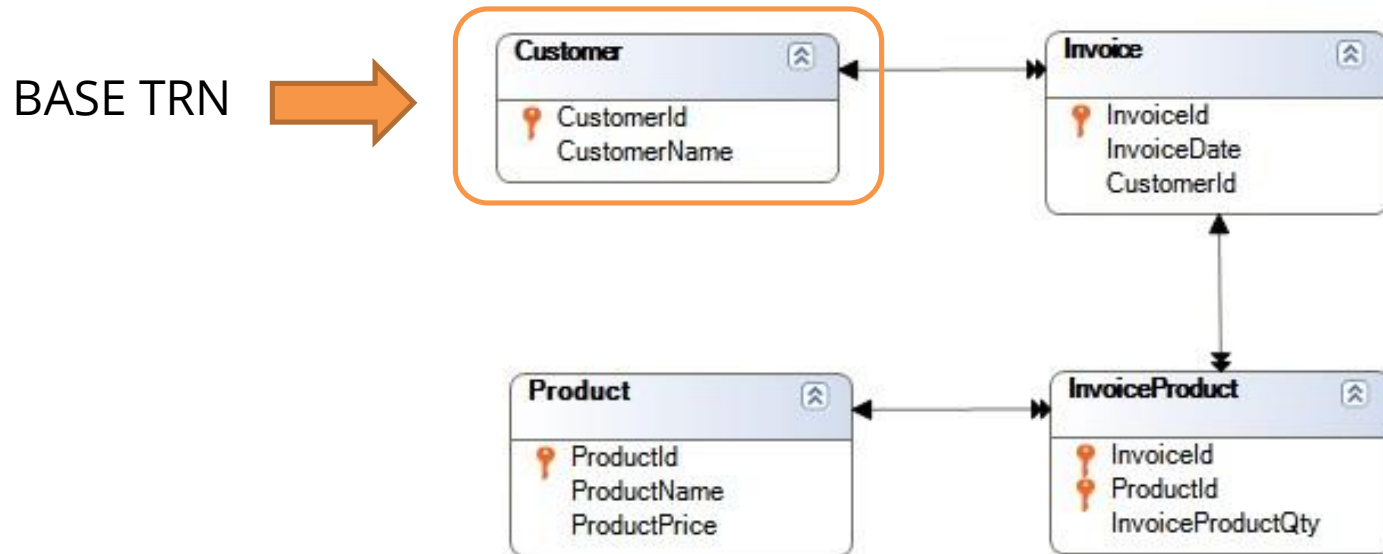




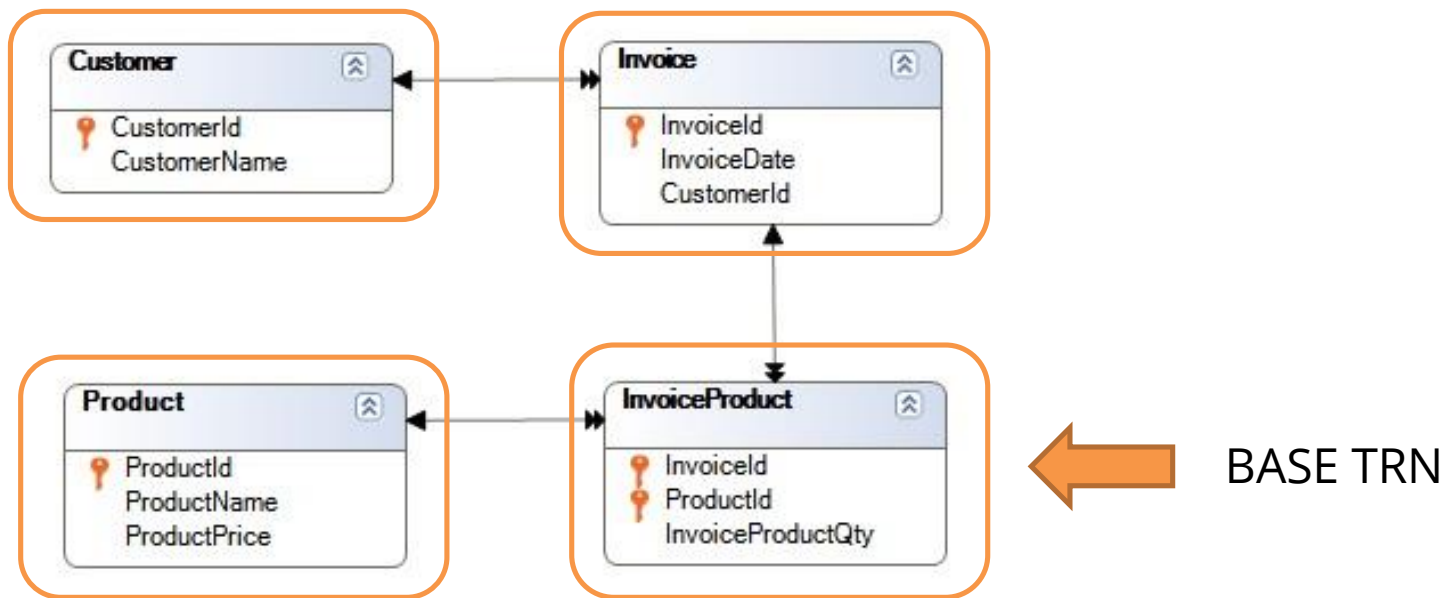
## Exemplo: Tabela Estendida de Invoice



## Exemplo: Tabela Estendida de Customer

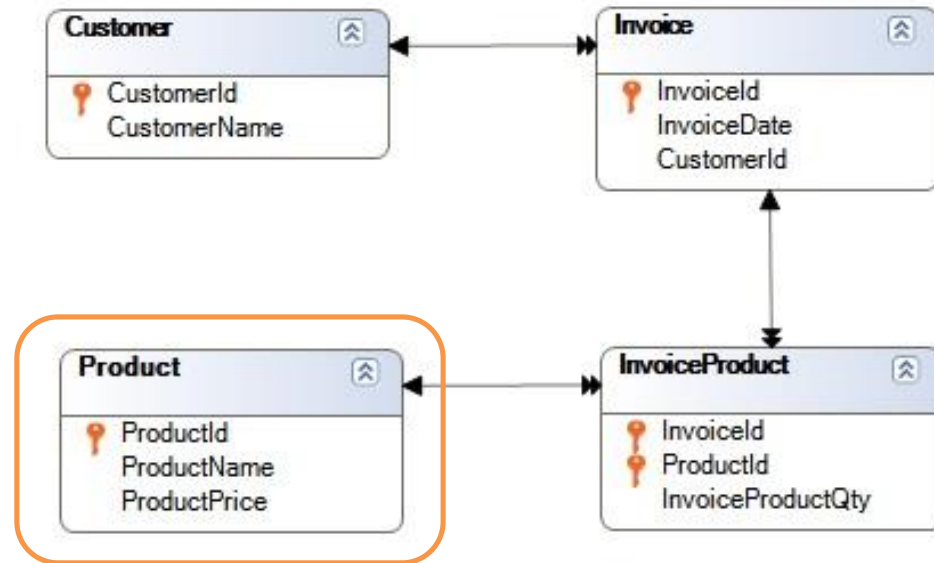


## Exemplo: Tabela Estendida de InvoiceProduct



## Exemplo: Tabela Estendida de Product

BASE TRN



# Subtipos

## Múltiplas Referências :

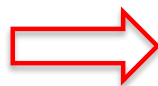
É preciso salvar, para cada vôo, o aeroporto de partida e o aeroporto de chegada

### Airport

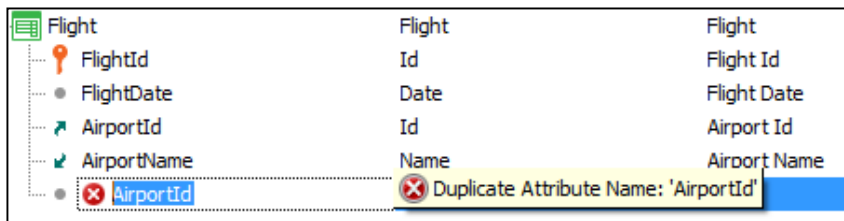
```
{  
  AirportId*  
  AirportName  
}
```

### Flight

```
{  
  FlightId*  
  FlightDate  
  AirportId  
  AirportName  
  AirportId  
  AirportName  
}
```



Erro por nome de atributos duplicados



## Solução 1 de 3: criar dois grupos de subtipos, um para o aeroporto de partida e outro para o aeroporto de chegada

### Airport

```
{
  AirportId*
  AirportName
}
```

### Flight

```
{
  FlightId*
  FlightDate
  DepartureAirportId
  DepartureAirportName } Subtype group: DepartureAirport
  ArrivalAirportId
  ArrivalAirportName } Subtype group: ArrivalAirport
}
```

Subtype	Description	Supertype
DepartureAirport		
DepartureAirportId	Departure Airport Id	AirportId
DepartureAirportName	Departure Airport Name	AirportName

Subtype	Description	Supertype
ArrivalAirport		
ArrivalAirportId	Arrival Airport Id	AirportId
ArrivalAirportName	Arrival Airport Name	AirportName




## Solução 2 de 3: criar um grupo de subtipos, somente para o aeroporto de partida

### Airport

```
{
  AirportId*
  AirportName
}
```

### Flight

```
{
  FlightId*
  FlightDate
  DepartureAirportId
  DepartureAirportName } Subtype group: DepartureAirport
  AirportId
  AirportName
}
```

Subtype	Description	Supertype
 DepartureAirport		
 DepartureAirportId	Departure Airport Id	AirportId
 DepartureAirportName	Departure Airport Name	AirportName






## Solução 3 de 3: criar um grupo de subtipos, somente para o aeroporto de chegada

### Airport

```
{
  AirportId*
  AirportName
}
```

### Flight

```
{
  FlightId*
  FlightDate
  AirportId
  AirportName
  ArrivalAirportId
  ArrivalAirportName } Subtype group: ArrivalAirport
}
```

Subtype	Description	Supertype
 ArrivalAirport		
 ArrivalAirportId	Arrival Airport Id	AirportId
 ArrivalAirportName	Arrival Airport Name	AirportName

## Múltiplas Referências :

É preciso salvar, além do país do cliente, o país onde foi emitida a Fatura

### Country

```
{
  CountryId*
  CountryName
}
```

### Customer

```
{
  CustomerId*
  CustomerName
  CountryId
  CountryName
}
```

### Invoice

```
{
  InvoiceId*
  InvoiceDate
  CustomerId
  CustomerName
  CountryId
  CountryName
  InvoiceCountryId
  InvoiceCountryName
}
```

Atributos inferidos

Subtype group: InvoiceCountry

Subtype	Description	Supertype
InvoiceCountry		
<ul style="list-style-type: none"> <li>InvoiceCountryId</li> </ul>	Invoice Country Id	CountryId
<ul style="list-style-type: none"> <li>InvoiceCountryName</li> </ul>	Invoice Country Name	CountryName

# Regras

## Regras

Name	Type
Student	Student
StudentId	Id
StudentName	Name
StudentAddress	Address, GeneXus
StudentBirthDate	Date
StudentAddedDate	Date

**Error**("Enter the student name") if  
StudentName.isEmpty();

**Msg**("The address is empty") if  
StudentAddress.isEmpty();

**Default**(StudentAddedDate, &Today);

**Noaccept**(StudentAddedDate);

& Variables	
& Standard Variables	
GxRemove	Numeric(1.0)
Mode	Character(3)
Pgmdesc	Character(256)
Pgmname	Character(128)
Time	Character(8)
Today	Date

## Regras

Name	Type
Country	Country
CountryId	Id
CountryName	Name
CountryLastLine	Numeric(4.0)
City	City
CityId	Id
CityName	Name

**Serial**(CityId, CountryLastLine, 1);

**Parm**(attribute1, &variable1, ....);



**Variável:** Espaço em memória que possui um nome e o tipo de dados que pode guardar. São referenciadas através do “&”.

# Momentos de disparo

### Browser

Transaction

http://trialapps3.genexus.com/ld8562acf4/

Attribute-1

Attribute-2

...

Attribute-n

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	⊙
data2-1	data2-2	...	☑
data3-1	data3-2	...	☐
data4-1	data 4-2	...	☑

**CONFIRM** CANCEL

### Server

Attribute-1  ✓

Attribute-2  ✓

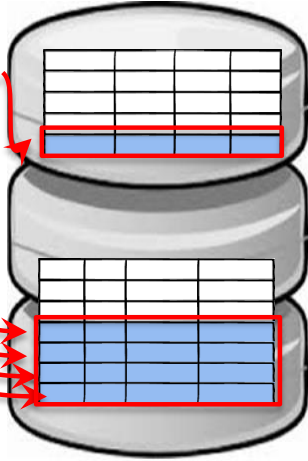
...

Attribute-n  ✓

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	⊙
data2-1	data2-2	...	☑
data3-1	data3-2	...	☐
data4-1	data 4-2	...	☑

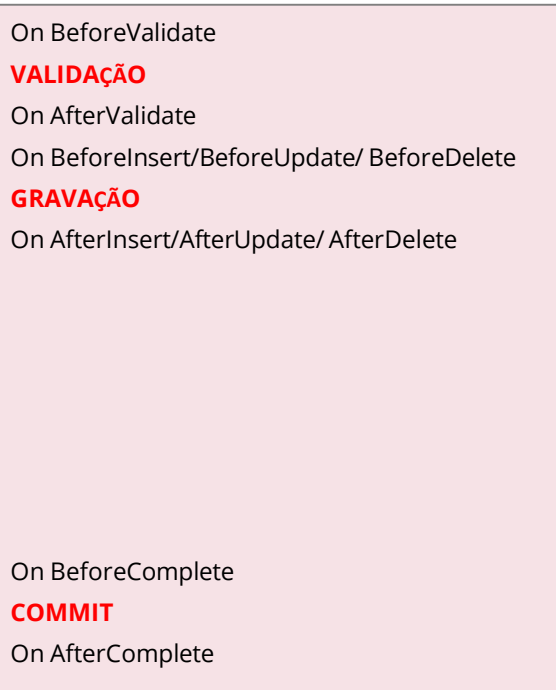
### Database



**Commit**

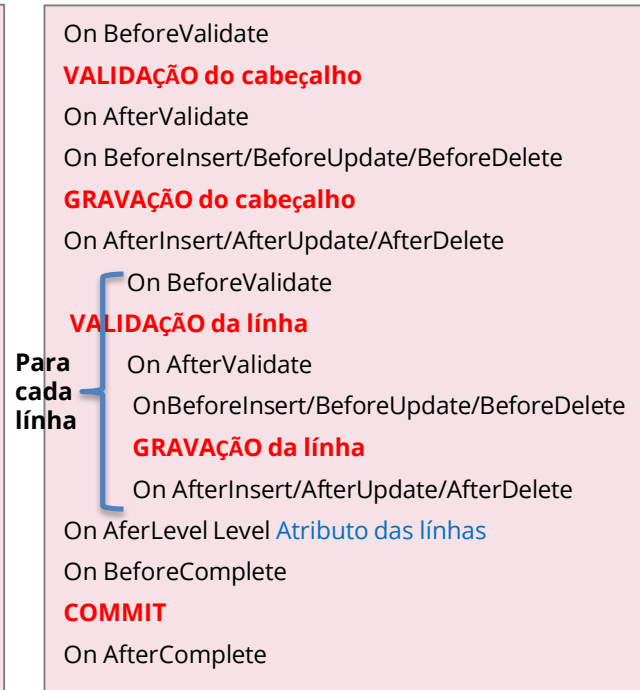
## Momentos de disparo de regras

Em transações de um só nível:



**No  
servidor,  
depois de  
pressionar  
Confirm**

Em transações de dois níveis:





## Momentos de disparo de regras

PrintCustomer(CustomerId) on AfterInsert, AfterUpdate; ¿Está certo ou não?



Está  
correto!

Transaction

http://trialapps3.genexus.com/1d8562acf4/

Attribute-1

Attribute-2

...

Attribute-n

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	<input type="radio"/>
data2-1	data2-2	...	<input checked="" type="checkbox"/>
data3-1	data3-2	...	<input type="checkbox"/>
data4-1	data 4-2	...	<input checked="" type="checkbox"/>

CONFIRM CANCEL

### VALIDAÇÃO

On Aftervalidate

### GRAVAÇÃO

On AfterInsert / On AfterUpdate / On AfterDelete

## Momentos de disparo de regras

**PrintCustomer(CustomerId) on  
AfterDelete;**

¿Está certo ou não?



Não está correto porque é chamado APÓS a exclusão e o cliente não será encontrado com esse ID na tabela

Transaction

http://trialapps3.genexus.com/Id8562acf4/

Attribute-1

Attribute-2

...

Attribute-n

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	<input type="radio"/>
data2-1	data2-2	...	<input checked="" type="checkbox"/>
data3-1	data3-2	...	<input type="checkbox"/>
data4-1	data 4-2	...	<input checked="" type="checkbox"/>

**CONFIRM** CANCEL

### VALIDAÇÃO

On Aftervalidate

### GRAVAÇÃO

On AfterInsert / On AfterUpdate / On AfterDelete

## Momentos de disparo de regras

```
Error('The seat quantity should be equal or greater than 8') if FlightCapacity<8
on AfterLevel
Level FlightSeatChar;
```

Transaction

http://trialapps3.genexus.com/Id8562acf4/

Attribute-1

Attribute-2

...

Attribute-n

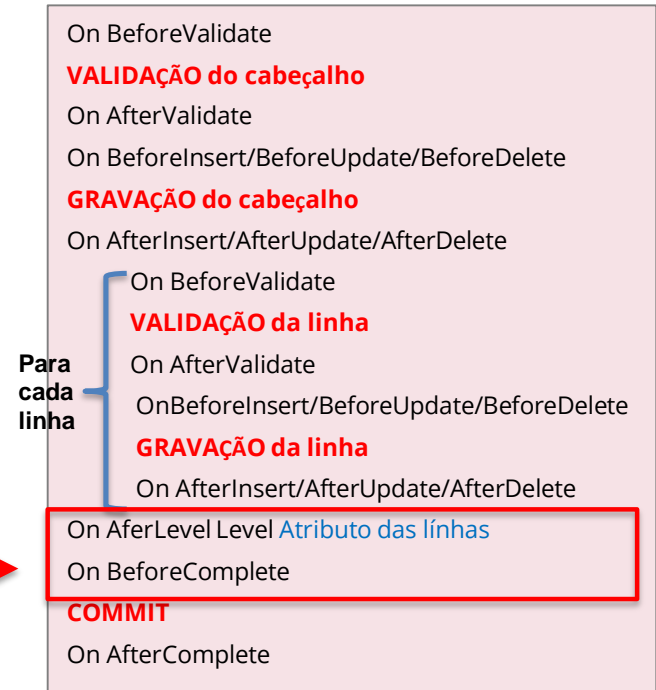
Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	<input type="radio"/>
data2-1	data2-2	...	<input checked="" type="checkbox"/>
data3-1	data3-2	...	<input type="checkbox"/>
data4-1	data 4-2	...	<input checked="" type="checkbox"/>

< 8

CONFIRM CANCEL

Em transações de dois níveis:

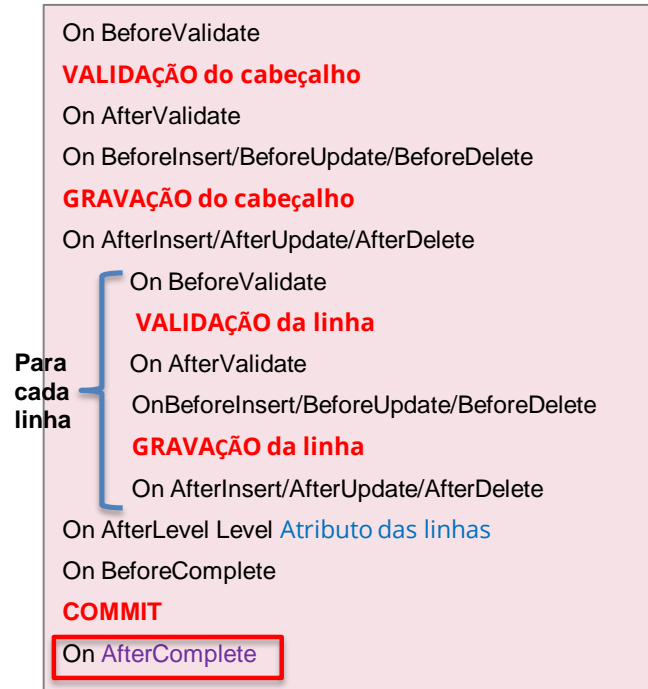


## Momentos de disparo de regras

```
PrintFlight(FlightId) on AfterComplete;
```

✓ on **AfterComplete**:  
Imediatamente depois de  
realizar o **Commit** no  
banco de dados

Em transações de dois níveis:



## Exemplos

### Invoice

```
{
  InvoicId*
  InvoiceDate
  -----
  Product
  {
    ProductId*
    ProductName
    ProductPrice
    InvoiceProductQuantity
    -----
  }
}
```

Determine se está correto ou não:

PrintInvoiceDetail(InvoicId) on **AfterComplete**;



ProductControl(ProductId) on **BeforeInsert**;

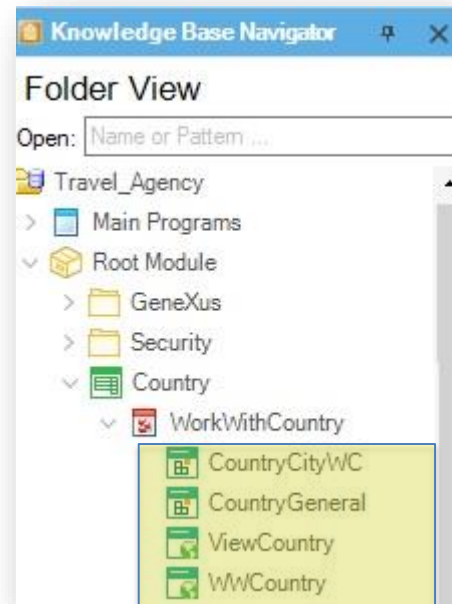
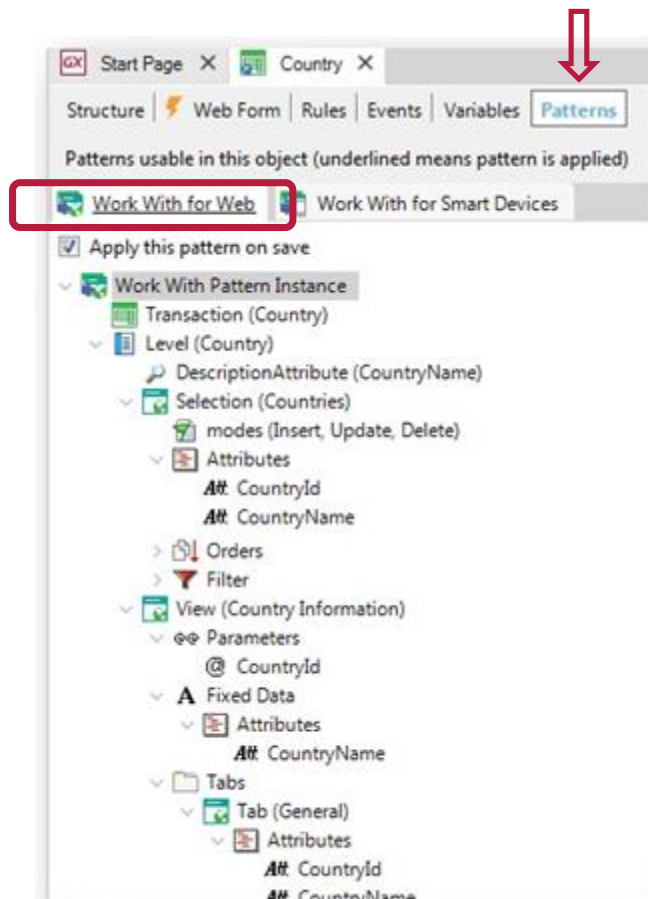


ProductControl(ProductId) on **AfterComplete**;



Posso atribuir um valor a um atributo on  
AfterInsert?  **NÃO**

# Patterns



**Gerados automaticamente por GeneXus**

## Dinamismo



**Example:** New attribute CountryFlag

The image illustrates the dynamic generation process in GeneXus. It shows three main components:

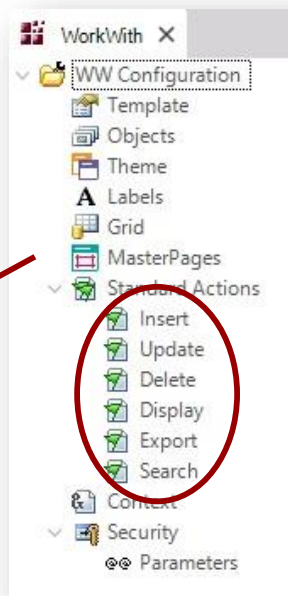
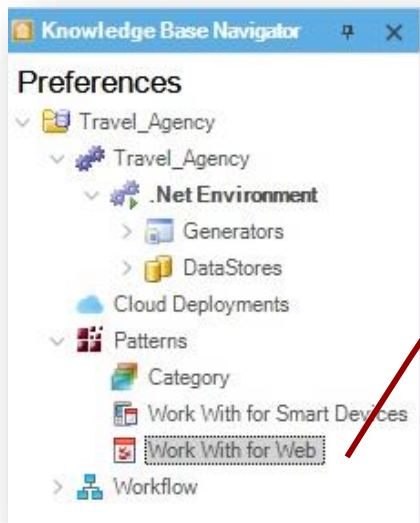
- Entity Structure:** A table defining the 'Country' entity with attributes: CountryId (Id), CountryName (Name), CountryFlag (Image), and City (City). The City entity has CityId (Id) and CityName (Name).
- Work With Pattern Instance Configuration:** A configuration window showing the 'Apply this pattern on save' checkbox checked. Under 'Work With Pattern Instance', the 'Attributes' section is expanded, showing the newly added attribute 'CountryFlag' alongside 'CountryId' and 'CountryName'.
- Generated Web Application:** A screenshot of the 'Travel Agency' web application showing a 'Countries' table with columns for ID, Name, Flag, and actions (UPDATE, DELETE). The table contains data for Brazil, China, and France.

Red arrows indicate the flow of data: a solid arrow points from the 'CountryFlag' attribute in the entity structure to the 'Attributes' list in the configuration window, and a dashed arrow points from the 'Attributes' list to the 'Flag' column in the web application table.



## Configurações Gerais

Como são inicializadas todas as instâncias?



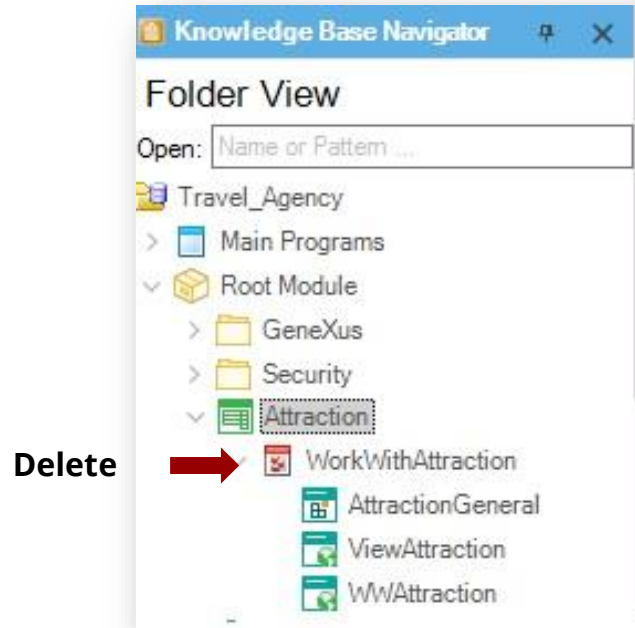
Em Country...

Properties

Filter

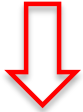
modes: Ins: default, Upd: default, ...	
Insert	default
Update	default
Delete	default
Display	default
Export	default

## Exclusão da aplicação do pattern



# Fórmulas

## Fórmulas Globais



- É um cálculo que definimos associado a um atributo
- São acessíveis em toda a KB

```

Product
{
  ProductId*
  ProductName
  ProductPrice
}
  
```

```

Invoice
{
  InvoiceId*
  InvoiceDate
  InvoiceAmount
  Product
  {
    ProductId*
    ProductName
    ProductPrice
    InvoiceProductQuantity
    InvoiceProductAmount
  }
}
  
```

Sum(InvoiceProductAmount)

Atributos "virtuais"

```

ProductPrice*InvoiceProductQuantity*0.9 if ProductId = 1;
ProductPrice*InvoiceProductQuantity*0.8 if ProductId = 3;
ProductPrice*InvoiceProductQuantity otherwise;
  
```

**Country**

```
{  
  CountryId*  
  CountryName  
  CountryCustomersQuantity  
}
```

**Customer**

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```



Count(CustomerName)

Esta fórmula contará clientes por país ou o número total de clientes?

Contará os clientes por cada país já que GeneXus realiza um filtro automático pelo atributo em comum (*CountryId*).

**Customer**

```
{  
  CustomerId*  
  CustomerName  
  CustomerTotal  
}
```

**Invoice**

```
{  
  InvoicId*  
  InvoiceDate  
  InvoiceType  
  CustomerId  
  CustomerName  
  InvoiceAmount  
}
```

Dominio InvType  
que oferece  
dois Enum Values

Credit

Cash

**Sum(InvoiceAmount, InvoiceType = InvType.Credit)**

**If CustomerId = 3**

condição de cálculo

condição de disparo

## Fórmulas Inline



- Estão escritas no código
- São acessíveis apenas no objeto em que foram definidas

**Requisito:** Nos pedem um relatório de países com a quantidade de atrações de cada país

Country	Quantity
Argentina	2
Uruguay	3
Paraguay	1
Estados Unidos	5



```
Country  
{  
  CountryId*  
  CountryName  
}
```

```
Attraction  
{  
  AttractionId*  
  AttractionName  
  CountryId  
  CountryName  
}
```

## Formula inline no código de um objeto Procedimento

```
Print Header
```

```
For each Country
```

```
    &AttractionsQuantity = Count(AttractionName)
```

```
    Print Countries
```

```
Endfor
```

Tabela base do For Each: COUNTRY

Tabela navegada pela fórmula: ATTRACTION

Esta fórmula contará as atrações por país ou o número total de atrações?

Contará as atrações por cada país já que é feito um filtro automático pelo atributo em comum *CountryId* (ambas tabelas estão relacionadas).



For each

## Transação Base

```
Flight  
{  
  FlightId*  
  FlightDate  
  ----  
  Seat  
  {  
    FlightSeatId*  
    FlightSeatChar  
  }  
}
```

For each **Flight**

-----  
Endfor

For each **Flight.Seat**

-----  
Endfor

**Transação Base**

Nome da transação cuja  
tabela física associada  
deseja-se percorrer

## Ordem

```

Customer
{
  CustomerId*
  CustomerName
  -----
}

```

**Requisito:** nos pedem um relatório de todos os clientes listados em ordem alfabética.

```

For each Customer order CustomerName
  -----
Endfor


```

**Requisito:** nos pedem um relatório de todos os clientes ordenados de forma decrescente pelo nome.

```

For each Customer order (CustomerName)
  -----
Endfor

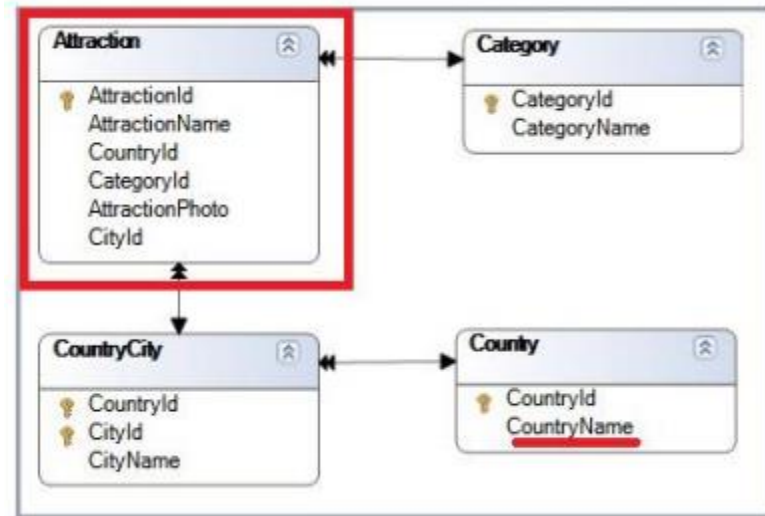
```



## Ordem

GeneXus permite ordenar pelo valor de um atributo que não está na tabela que estamos percorrendo, mas em sua tabela estendida.

```
Print Header
For each Attraction order CountryName
  Print Attractions
Endfor
```



## Filtros com Where

### Flight

```
{  
  FlightId*  
  FlightDate  
  ----  
  Seat  
  {  
    FlightSeatId*  
    FlightSeatChar  
  }  
}
```

### For each Flight

```
  Where FlightDate = Today()  
  -----
```

### Endfor

### For each Flight.Seat

```
  Where FlightId = 1  
  -----
```

### Endfor

## Filtros + Ordem

### Customer

```
{
  CustomerId*
  CustomerName
  CustomerAddress
}
```

Index



**For each Customer order CustomerName**  
**Where** CustomerName >= &NameFrom  
 -----  
**Endfor**

Warnings

 **spc0038** There is no index for order CustomerName; poor performance may be noticed in group starting at line 2.

Attribute	Order	Description
Customer Indexes		Customer
ICustomer	Primary Key	Automatic Index
• CustomerId	Ascending	Customer Id
UCustomerName	Duplicate	User Index
• CustomerName	Ascending	Customer Name

¡Consulta optimizada!



## For each Aninhados + Tabela base diferente + Tabelas NÃO relacionadas = PRODUTO CARTESIANO

### Country

```
{  
  CountryId*  
  CountryName  
}
```

### Room

```
{  
  RoomId*  
  RoomName  
}
```

### Produto Cartesiano

```
For each Country  
  Print Country
```

```
  For each Room  
    Print Room
```

```
  Endfor
```

```
Endfor
```

```
1 - Brasil  
    RoomA  
    RoomB  
    RoomC  
2 -Uruguay  
    RoomA  
    RoomB  
    RoomC  
3 - Argentina  
    RoomA  
    RoomB  
    RoomC  
4 - Estados Unidos  
    RoomA  
    RoomB  
    RoomC
```

## For each Aninhados + Tabela base diferente + Tabelas relacionadas = JOIN

### Country

```
{  
  CountryId*  
  CountryName  
}
```



### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```

Join

```
For each Country  
  Print Country
```

```
  For each Customer  
    Print Customer  
  Endfor
```

```
Endfor
```

1 - Brasil	LUIS JORGE
2 - Uruguay	
3 - Argentina	
4 - Estados Unidos	ANA



## For each Aninhados + Mesma Tabela base + Tabelas relacionadas = CORTE DE CONTROLE

### Country

```
{  
  CountryId*  
  CountryName  
}
```

### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```

CORTE DE CONTROLE

```
For each Customer order CountryId  
  Print Country
```

```
For each Customer  
  Print Customer  
Endfor
```

```
Endfor
```

1 - Brasil

LUIS  
JORGE

4 - Estados Unidos

ANA

## Resumo

### Produto Cartesiano

**For each Country**

Print Country

**For each Room**

Print Room

**Endfor**

**Endfor**

Tabelas  
diferentes, sem  
relacionamento

### Join

**For each Country**

Print Country

**For each Customer**

Print Customer

**Endfor**

**Endfor**

Tabelas  
diferentes, com  
relacionamento

### Corte de Controle

**For each Customer order CountryId**

Print Country

**For each Customer**

Print Customer

**Endfor**

**Endfor**

Mesma tabela,  
Agrupando com  
order

## Resumo

### Produto Cartesiano

1 - Brasil	RoomA
	RoomB
	RoomC
2 -Uruguay	RoomA
	RoomB
	RoomC
3 - Argentina	RoomA
	RoomB
	RoomC

*“Não existe relacionamento entre ambas entidades, mostra todas as possibilidades para cada país”*

### Join

1 - Brasil	LUIS
	JORGE
2 - Uruguai	
3 - Argentina	
4 - Estados Unidos	ANA

*“Todos os países e seus clientes, independentemente que tenha clientes ou não”*

### Corte de Controle

1 - Brasil	LUIS
	JORGE
4 - Estados Unidos	ANA

*“Somente aqueles países que tenham clientes”*

# Comunicação entre objetos

## Exemplo: passagem de parâmetros

### Country

```
{  
  CountryId*  
  CountryName  
}
```

### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```

Por exemplo, nas Regras da trn Country:

**CustomerList(CountryId)** on AfterComplete;



### CustomerList

**Parm(in: &CountryId);**

For each Customer  
 Where CountryId = &CountryId  
 -----  
Endfor

Variável

Filtro explícito



## Exemplo: passagem de parâmetros

### Country

```
{  
  CountryId*  
  CountryName  
}
```

### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```

Por exemplo, nas Regras da trn Country:

**CustomerList(CountryId)** on AfterComplete;



### CustomerList

**Parm(in: CountryId);**

For each Customer

~~Where CountryId = &CountryId~~

-----

Endfor

Atributo

Filtro implícito



## Exemplo: retorno de um valor

### Country

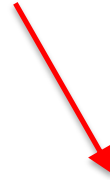
```
{  
  CountryId*  
  CountryName  
}
```

### Customer

```
{  
  CustomerId*  
  CustomerName  
  CountryId  
  CountryName  
}
```

Por exemplo, nas Regras da trn Customer:

```
&Control = CustomerControl(CustomerId);
```



### CustomerControl

```
Parm(in: &CustomerId, out: &Control);
```

```
For each Customer
```

```
  Where CustomerId = &CustomerId
```

```
  -----
```

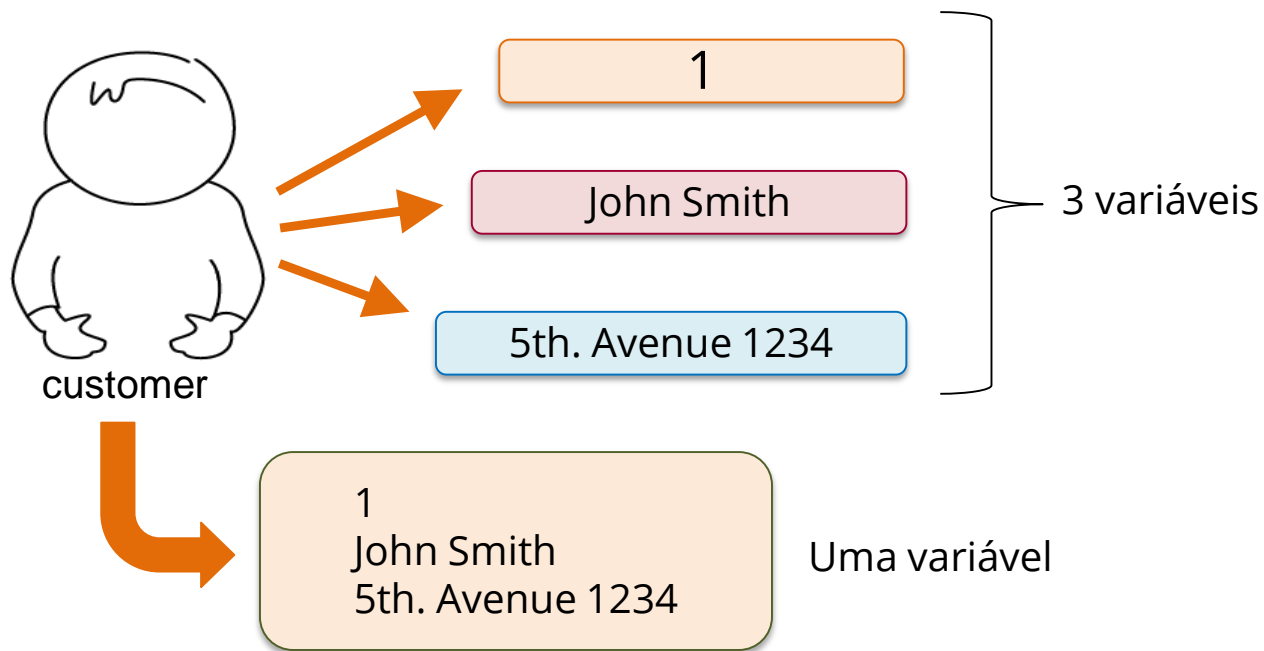
```
    &Control = True
```

```
Endfor
```

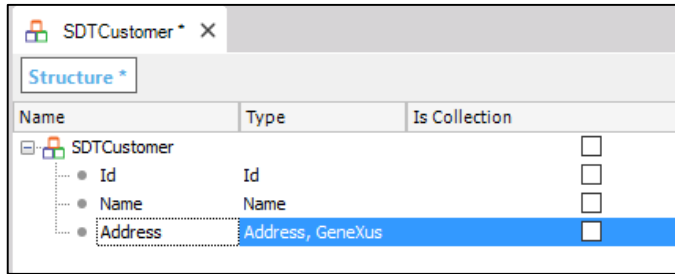
# Tipos de Dados Compostos



## Conceito



## Definição



The screenshot shows a window titled 'SDTCustomer \* X' with a 'Structure \*' tab. It displays a tree view of the structure with the following table:

Name	Type	Is Collection
SDTCustomer		<input type="checkbox"/>
● Id	Id	<input type="checkbox"/>
● Name	Name	<input type="checkbox"/>
● Address	Address, GeneXus	<input type="checkbox"/>

Objeto Structured Data Type



customer



&OneCustomer: SDTCustomer

```
&OneCustomer.Id = 1  
&OneCustomer.Name = 'John Smith'  
&OneCustomer.Address = '5th. Avenue 1234'
```

# Data Providers

**Id: 1**  
**Name: John Smith**  
**Address: 5th. Avenue 1234**

← Um cliente

## SDTCustomer

Id: 1  
Name: John Smith  
Address: 5th. Ave.

Id: 2  
Name: Susan  
Brown  
Address: 7th.Ave.

Id: 3  
Name: Robert Hill  
Address: 81th. St..

Id: 4  
Name: Peter Jensen  
Address: St,Paul Rd.

↑  
**Coleção de Clientes**

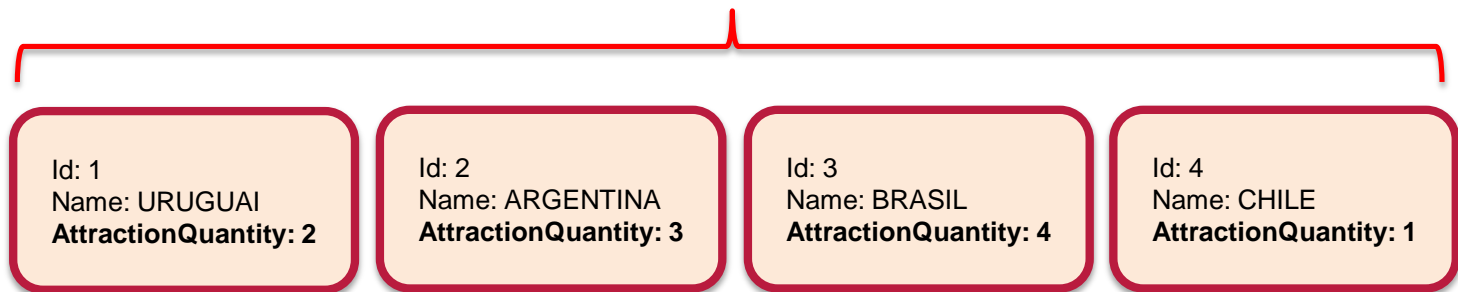
## Exemplo: Ranking de quantidade de atrações por país

<b>País</b>	<b>Qtde atrações</b>
BRASIL	4
ARGENTINA	3
URUGUAI	2
CHILE	1
....	
...	

## Exemplo: Ranking de quantidade de atrações por país

### SDTCountries

#### Uma coleção de países



## Exemplo: Ranking de quantidade de atrações por país

### Country

```
{
  CountryId*
  CountryName
}
```

Name	Type	Is Collection
SDTCountries		<input checked="" type="checkbox"/>
SDTCountriesItem		
Id	Id	<input type="checkbox"/>
Name	Name	<input type="checkbox"/>
CountryAttractionsQuantity	Numeric(4,0)	<input type="checkbox"/>

### Attraction

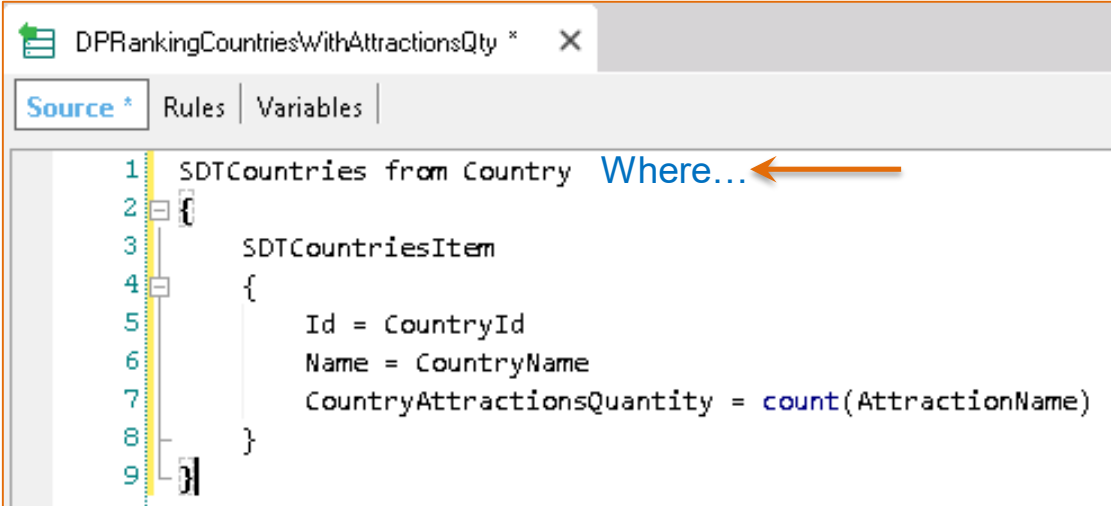
```
{
  AttractionId*
  AttractionName
  CountryId
  CountryName
}
```

```

1 SDTCountries from Country
2 {
3   SDTCountriesItem
4   {
5     Id = CountryId
6     Name = CountryName
7     CountryAttractionsQuantity = count(AttractionName)
8   }
9 }
```

Output	
Infer Structure	No
Output	SDTCountries
Collection	False

## Exemplo: Ranking de quantidade de atrações por país



The screenshot shows a window titled "DPRankingCountriesWithAttractionsQty \*". The "Source" tab is active, displaying the following code:

```
1 SDTCountries from Country Where... ←
2 {
3   SDTCountriesItem
4   {
5     Id = CountryId
6     Name = CountryName
7     CountryAttractionsQuantity = count(AttractionName)
8   }
9 }
```

An orange arrow points to the "Where..." text in line 1, indicating a filter or condition to be applied to the data source.



## Exemplo: Ranking de quantidade de atrações por país

### Country

```
{
  CountryId*
  CountryName
}
```

### Attraction

```
{
  AttractionId*
  AttractionName
  CountryId
  CountryName
}
```



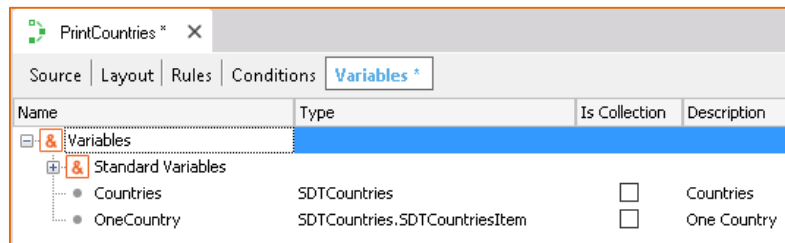
```
PrintCountries *
Source *
Layout | Rules | Conditions | Variables
Subroutines
1  &Countries = DPRankingCountriesWithAttractionsQty()
2  &Countries.Sort("[CountryAttractionsQuantity]")
3
4  Print Title
5
6  For &OneCountry in &Countries
7    print Country
8  Endfor
9
```

### País

BRASIL  
ARGENTINA  
URUGUAI  
CHILE

### Qtde Atrações

4  
3  
2  
1



Name	Type	Is Collection	Description
&Variables		<input checked="" type="checkbox"/>	
Standard Variables		<input checked="" type="checkbox"/>	
Countries	SDTCountries	<input type="checkbox"/>	Countries
OneCountry	SDTCountries.SDTCountriesItem	<input type="checkbox"/>	One Country

# Variáveis Coleção

&amp;numbers

Name	Type	Is Collection	Description
& Variables			
+ & Standard Variables			
numbers	Numeric(4.0)	<input checked="" type="checkbox"/>	numbers

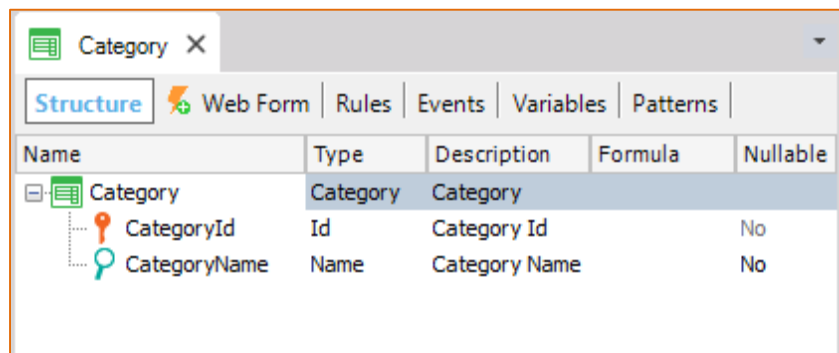
Name	Type	Description	Is Collection
SDTCountry		SDTCountry	<input type="checkbox"/>
• Id	Numeric(4.0)	Id	<input type="checkbox"/>
• Name	Character(20)	Name	<input type="checkbox"/>

&amp;Countries

Name	Type	Is Collection	Description
& Variables			
+ & Standard Variables			
Countries	SDTCountry	<input checked="" type="checkbox"/>	Countries

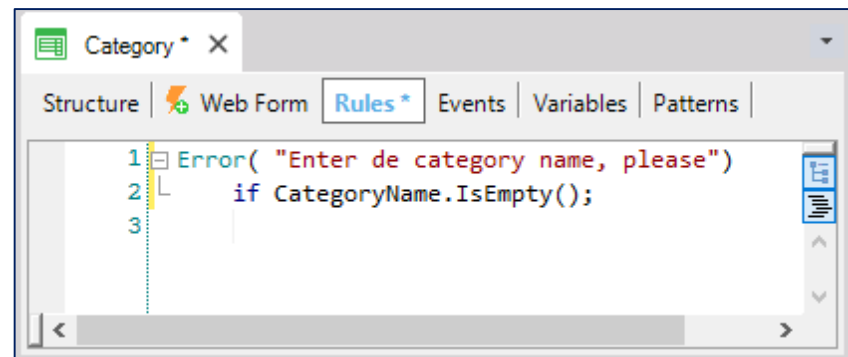
# Business Components

## Conceito: tipo de dados especial baseado em uma transação



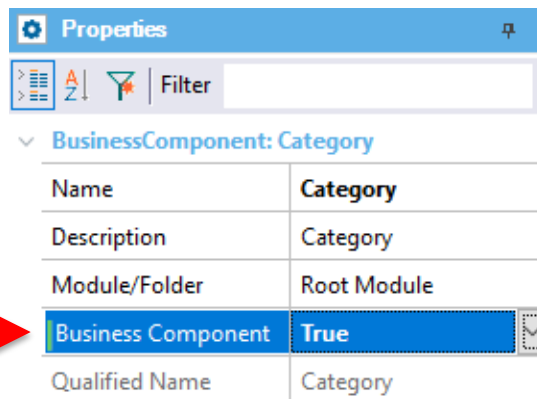
Structure | Web Form | Rules | Events | Variables | Patterns

Name	Type	Description	Formula	Nullable
Category	Category	Category		
CategoryId	Id	Category Id		No
CategoryName	Name	Category Name		No



Structure | Web Form | Rules \* | Events | Variables | Patterns

```
1 Error( "Enter de category name, please")
2 |   if CategoryName.IsEmpty();
3
```



Properties

BusinessComponent: Category

Name	Category
Description	Category
Module/Folder	Root Module
Business Component	True
Qualified Name	Category

## Conceito: tipo de dados especial baseado em uma transação

Web Form | Rules | Events | Conditions | **Variables**

Name	Type
& Variables	
+ & Standard Variables	
• Category	Category

Properties

Filter

BusinessComponent: Category

Name	Category
Description	Category
Module/Folder	Root Module
Business Component	True
Qualified Name	Category

### Variable: &Category

Name	Category
Description	Category
Column title	Category
Class	Attribute

[Help](#)

### Type Definition

Based on	(none)
<b>Data Type</b>	<b>Category</b>
Collection	
Initial value	
<b>Validation</b>	
Value range	
Validation Failed M	
<b>Control Info</b>	
Control Type	
Input Type	

Image

LongVarChar

Numeric

VarChar

Video

Extended Types

Structured Data Types

**Business Components**

Airline

Attraction

**Category**

Country

Country.City

External Objects

## Exemplos: inserção e alteração

### Category

```
{  
  CategoryId*  
  CategoryName  
}
```

**Insert**



```
Source * | Layout | Rules | Conditions | Variables * |  
Subroutines  
1  &Category.CategoryId = 1  
2  &Category.CategoryName = "Tourist site"  
3  &Category.Save()  
4  commit
```

**Update**



```
Source * | Layout | Rules | Conditions | Variables * |  
Subroutines  
1  &Category.Load(1)  
2  &Category.CategoryName = "New site"  
3  &Category.Save()  
4  commit
```

## Exemplo: exclusão

### Category

```
{  
  CategoryId*  
  CategoryName  
}
```

**Delete**

A screenshot of a code editor window with a tab labeled 'Source \*'. The editor has a menu bar with 'Layout', 'Rules', 'Conditions', and 'Variables \*'. Below the menu bar is a dropdown menu labeled 'Subroutines'. The main area of the editor shows three lines of code:

```
1 &Category.Load(1)  
2 &Category.Delete()  
3 commit
```



## Métodos Insert e Update

### Category

```
{  
  CategoryId*  
  CategoryName  
}
```

### Insert



```
Source * | Layout | Rules | Conditions | Variables * |  
Subroutines  
1  &Category.CategoryId = 1  
2  &Category.CategoryName = "Tourist site"  
3  &Category.Insert()  
4  commit
```

### Update



```
Source * | Layout | Rules | Conditions | Variables * |  
Subroutines  
1  &Category.CategoryId = 1  
2  &Category.CategoryName = "Tourist site"  
3  &Category.Update()  
4  commit
```

## Método InsertOrUpdate

### Category

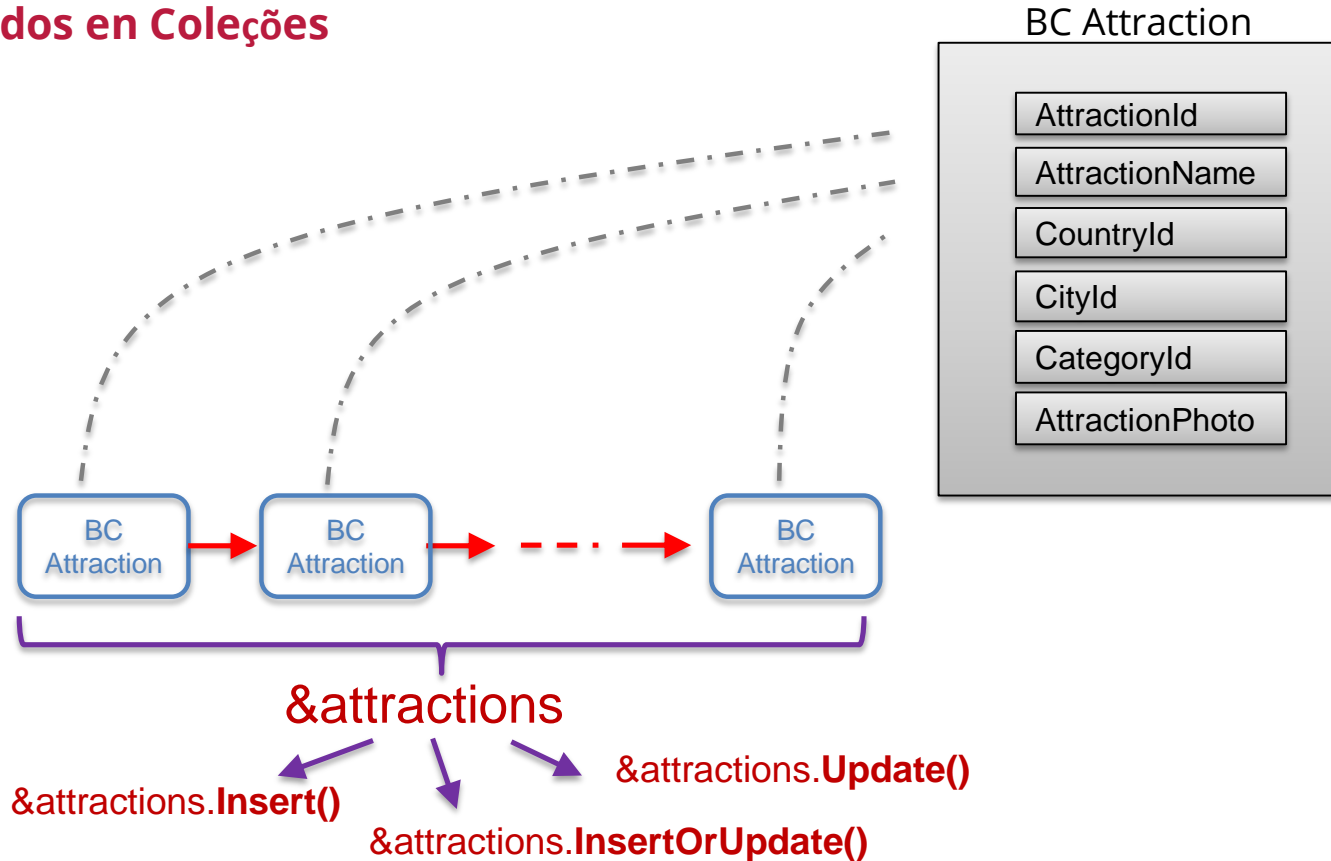
```
{  
  CategoryId*  
  CategoryName  
}
```

**InsertOrUpdate**



```
Source * | Layout | Rules | Conditions | Variables * |  
Subroutines ▾  
1  &Category.CategoryId = 2  
2  &Category.CategoryName = "Tourist site"  
3  &Category.InsertOrUpdate()  
4  commit
```

## Métodos en Coleções



## Insert / Update / InsertOrUpdate

Recomendamos o uso dos métodos Insert, Update e InsertOrUpdate uma vez que:

- O uso dos métodos Load e Save para modificar o banco de dados é acessado duas vezes - o que diminui o desempenho - enquanto no Update ou InsertOrUpdate é acessado apenas uma vez.
- Os nomes desses novos métodos já são claramente auto-explicativos.

## Manuseio de erros trabalhando com BC

Para cada variável Business Component, uma coleção é carregada na memória com todas as mensagens de aviso ou erro que ocorrem como resultado das operações.

```
Source | Layout | Rules | Conditions | Variables | Help |
Subroutines
1 &Country.CountryName = "Brasil"
2 &Country.Save()
3
4 &Messages = &Country.GetMessages() ←
5
6 For &oneMessage in &Messages ←
7     msg(&oneMessage.Description)
8 Endfor
```

Messages [Read-only] X

Structure

Name	Type
Messages	
Message	
• Id	VarChar(128)
• Type	MessageTypes, GeneXus
• Description	VarChar(256)

Source | Layout | Rules | Conditions | Variables | Help | Documentation

Name	Type	Is Collection
& Variables		
+ & Standard Variables		
• Country	Country	<input type="checkbox"/>
• Messages	Messages, GeneXus.Common	<input type="checkbox"/>
• oneMessage	Messages.Message, GeneXus.Common	<input type="checkbox"/>

# Carregamento de Dados

## Transação

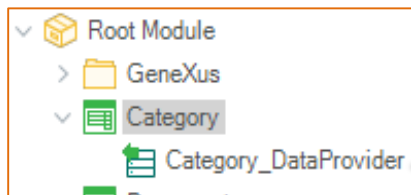
## Inicializar dados de forma automática

GeneXus 15 oferece uma alternativa muito fácil para definir os dados com os quais deseja-se popular as tabelas físicas criadas associadas às transações, o que evita ter que criar as cargas de dados de outras formas.

Name	Type	Description
Category	Category	Category
CategoryId	Id	Category Id
CategoryName	Name	Category Name

Data	
Data Provider	True
Used to	Populate data
Update Policy	Updatable

## Inicializar dados



```
Category_DataProvider * X
Source * Rules Variables
1 /*
2 CategoryCollection
3 {
4     Category
5     {
6         CategoryId =
7         CategoryName =
8     }
9 }
10 */
```

Não carregamos o CategoryId  
já que foi definido como autonumber

```
Category_DataProvider X
Source Rules Variables
1 CategoryCollection
2 {
3     Category
4     {
5         CategoryName = "Museum"
6     }
7     Category
8     {
9         CategoryName = "Monument"
10    }
11    Category
12    {
13        CategoryName = "Tourist Site"
14    }
15 }
```



























## Inicializar dados: Somente leitura

### Country

```
{  
  CountryId*  
  CountryName  
}
```

Data	
Data Provider	True
Used to	Populate data
Update Policy	Read Only

 United States
 Brazil
 Mexico
 Colombia
 Argentina
 Canada
 Peru
 Venezuela
 Chile
 Ecuador
 Guatemala
 Cuba
 Haiti
 Bolivia
 Dominican Republic
 Honduras
 Paraguay
 Nicaragua
 El Salvador
 Costa Rica
 Panama
 Puerto Rico
 Uruguay
 Jamaica
 Trinidad and Tobago

# Carregamento de Dados

## Business Components e Data Providers

## Exemplo

### Country

```
{
  CountryId* ← Autonumber = True
  CountryName
}
```

Web Form | Rules | Events | Conditions | Variables

<No action group selected>

MainTable

Initialize Countries

Web Form | Rules | Events | Conditions | Variables

Events

```
1 Event 'Initialize Countries'
2   &Countries = DPCountries()
3   &Countries.Insert()
4   commit
5 Endevent
```

Source | Rules | Variables

```
1 Country
2 {
3   CountryName = "Uruguay"
4 }
5 Country
6 {
7   CountryName = "Argentina"
8 }
9 Country
10 {
11  CountryName = "Brazil"
12 }
```

Output

Infer Structure	No
Output	<b>Country</b>
Collection	<b>True</b>
Collection Name	<b>Countries</b>

Web Form | Rules | Events | Conditions | Variables

Name	Type	Is Collection	Description
Variables			
Standard Variables			
Countries	Country	<input checked="" type="checkbox"/>	Countries

# Carregamento de Dados com Procedimentos Comandos New / For each / Delete

## Inserção - Comando NEW

### Category

```
{  
  CategoryId*  
  CategoryName  
}
```

```
New  
  CategoryId = 5  
  CategoryName = "Tourist Site"  
Endnew
```

```
New  
  CategoryName = "Tourist Site"  
Endnew
```



Se o atributo é autonumber  
não o informamos

## Alteração / Comando FOR EACH

### Category

```
{
  CategoryId*
  CategoryName
}
```

```
For each Attraction
Where CityName = "Beijing" and CategoryName = "Monument"
  CategoryId = find( CategoryId, CategoryName = "Tourist site")
Endfor
```

### Country

```
{
  CountryId*
  CountryName
  City
  {
    CityId*
    CityName
  }
}
```

### Attraction

```
{
  AttractionId*
  AttractionName
  CategoryId
  CategoryName
  CountryId
  CountryName
  CityId
  CityName
}
```

## Exclusão – Comando DELETE

### Attraction

```
{  
  AttractionId*  
  AttractionName  
  CategoryId  
  CategoryName  
  CountryId  
  CountryName  
  CityId  
  CityName  
}
```

```
For each Attraction  
  Delete  
Endfor
```

## Considerações

Os comandos New / For Each / Delete são usados para inserir, alterar e excluir informações do banco de dados, mas eles só podem ser usados em Procedimentos.

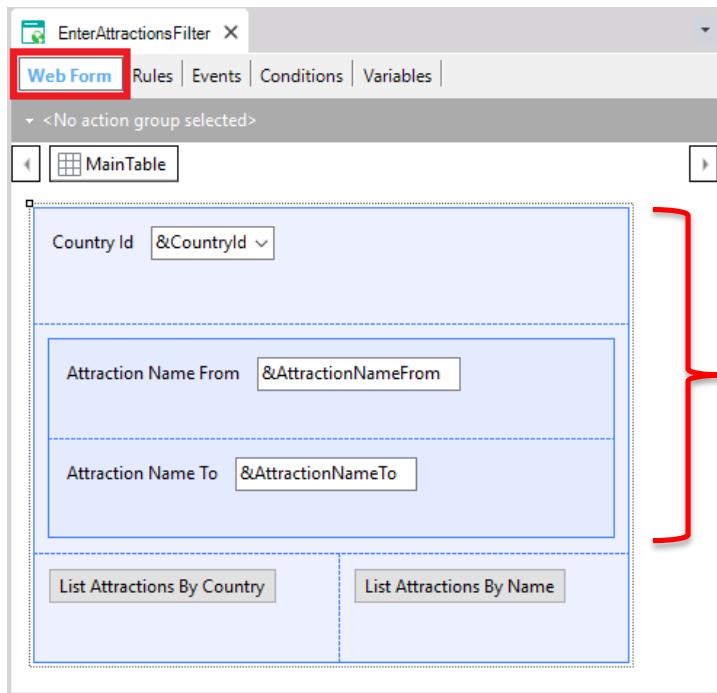
Embora os comandos a seguir sejam usados para inserir, alterar e excluir informações do banco de dados, recomendamos o uso do Business Component, uma vez que eles:

- Controlam a integridade referencial
- Disparam as regras definidas na transação



# Web Panels

## Web Panel sem grid, com variáveis no form



EnterAttractionsFilter X

Web Form | Rules | Events | Conditions | Variables

<No action group selected>

MainTable

Country Id &CountryId ▾

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

List Attractions By Country

List Attractions By Name

Variáveis: **de entrada**  
(não readonly)

## Web Panel sem grid mas com atributos no form

Parm(in: **AttractionId**);

ViewAttractionFromScratch X

Web Form | Rules | Events | Conditions | Variables

<No action group selected>

Id

Name

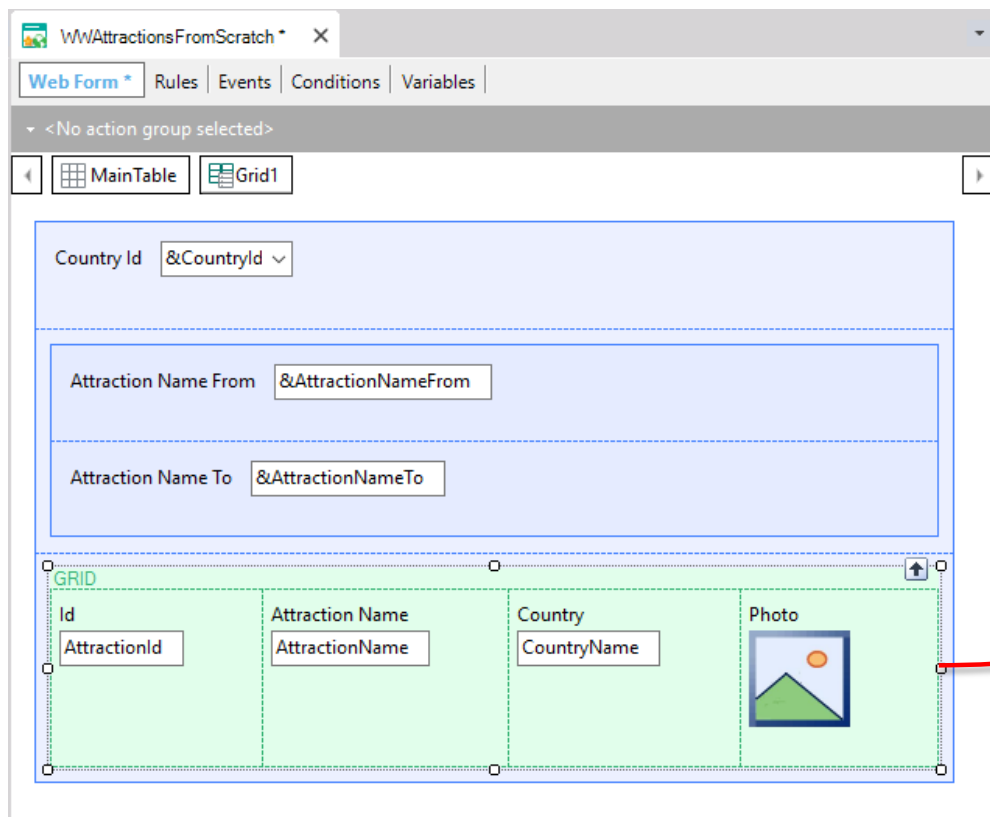
Country Name

Category Name

City Name

É carregado só **um** registro

## Grid: COM TABELA BASE



WWAttractionsFromScratch \* x

Web Form \* Rules Events Conditions Variables

<No action group selected>


MainTable Grid1

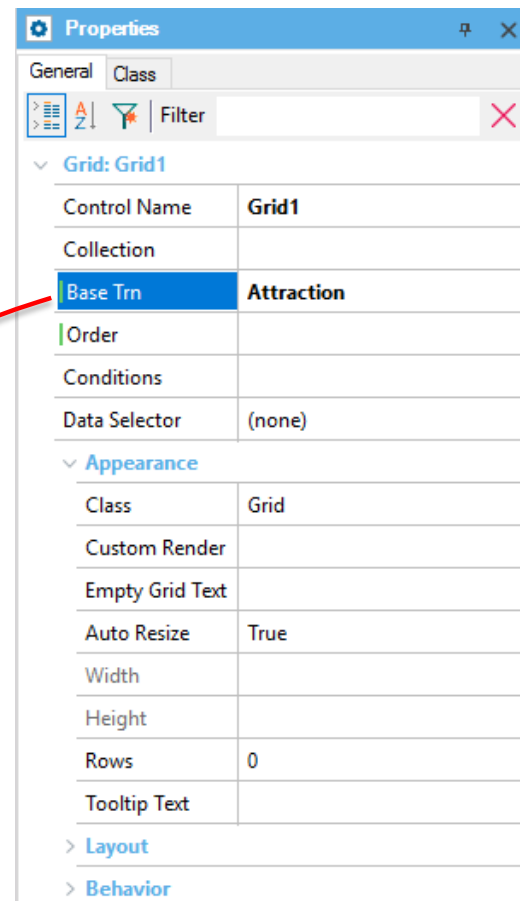
Country Id &CountryId v

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Id	Attraction Name	Country	Photo
AttractionId	AttractionName	CountryName	



Properties

General Class

Filter

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	
Conditions	
Data Selector	(none)

Appearance

Class	Grid
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	0
Tooltip Text	

> Layout

> Behavior

## Transação base

The screenshot displays the GeneXus IDE interface for a web form named 'AttractionsByName'. The interface is divided into three main sections:

- Web Form (Left):** Shows a form with three input fields: 'Country Id' (with a dropdown arrow), 'Attraction Name From', and 'Attraction Name To'. Below these fields is a grid control labeled 'GRID'. The grid has four columns: 'Id' (with a sub-label 'AttractionId'), 'Attraction Name' (with a sub-label 'AttractionName'), 'Country' (with a sub-label 'CountryName'), and 'Photo' (with a small image icon).
- Source Code (Middle):** Shows the following code in the 'Subroutines' view:

```
1 print Title
2 print ColumnTitles
3 For each Attraction and CountryName
4 where AttractionName = &AttractionFrom
5 where AttractionName = &AttractionTo
6   print Attractions
7 endfor
```

Lines 3, 4, and 5 are highlighted with a red box. A red arrow points from this box to the grid control in the web form, with an equals sign (=) next to the arrow's tail.
- Properties Panel (Right):** Shows the 'Properties' window for 'Grid: Grid1'. The 'Collection' property is set to 'Attraction', and the 'Base Trn' property is also highlighted with a red box.

# Ordem

The screenshot displays the GeneXus IDE interface with three main windows:

- Web Form:** Shows a form with fields for "Country Id" (dropdown), "Attraction Name From" (text), and "Attraction Name To" (text). Below these is a grid named "GRID" with columns: "Id" (AttractionId), "Attraction Name" (AttractionName), "Country" (CountryName), and "Photo".
- Source:** Contains the following code:

```
1 print Title
2 print ColumnTitles
3 For each Attraction order CountryName
4   print AttractionName &AttractionNameFrom
5   print AttractionName &AttractionNameTo
6   print Attractions
7 endfor
```
- Properties:** Shows the "Grid: Grid1" properties. The "Order" property is highlighted, and a red arrow points from it to the "Grid1's Order" dialog box.

The "Grid1's Order" dialog box is open, showing a list with "CountryName" selected. The dialog has "OK" and "Cancel" buttons.

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	
Conditions	
Data Selector	(none)

Property	Value
Height	
Rows	0
Tooltip Text	

## Grid: COM TABELA BASE

WWAttractionsFromScratch \* x

Web Form \* Rules Events Conditions Variables

<No action group selected>

MainTable Grid1

Country Id &CountryId v

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Id	Attraction Name	Country	Photo
AttractionId	AttractionName	CountryName	

Grid's Conditions

CountryId = &CountryId when not &CountryId.IsEmpty();

OK Cancel

Properties

General Class

Filter

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryName
Conditions	CountryId = &CountryId; ...
Data Selector	(none)

Appearance

Grid

True

Height

Rows 0

Tooltip Text

Layout


## Condições de filtro

The screenshot displays the GeneXus IDE interface with three main panes:

- Web Form:** Shows a form with three input fields: "Country Id" (with a dropdown arrow), "Attraction Name From", and "Attraction Name To". Below these is a grid control labeled "GRID".
- Source:** Contains the following code:

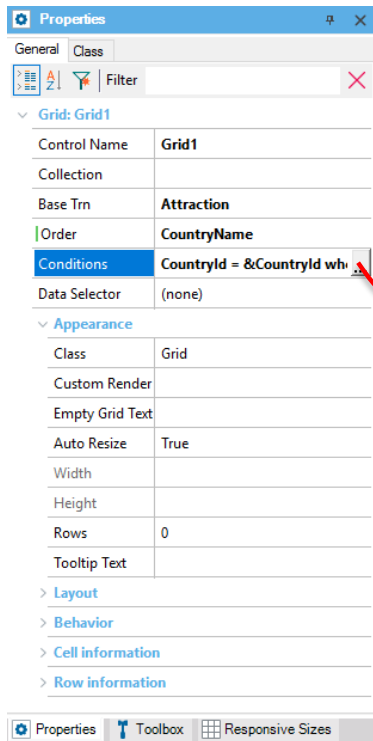
```
1 print Title
2 print ColumnTitles
3 For each Attraction order CountryName
4   where CountryId = &CountryId
5   print Attractions
6 endfor
```
- Properties:** Shows the configuration for "Grid: Grid1". The "Conditions" property is highlighted with a red box and contains the text "CountryId = &CountryId; ...". A red arrow points from this property to the "where" clause in the source code.

The grid control in the web form has the following columns:

Id	Attraction Name	Country	Photo
AttractionId	AttractionName	CountryName	



## Muitas condições



Properties

General Class

Filter

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryName
Conditions	CountryId = &CountryId wh...
Data Selector	(none)

Appearance

Class	Grid
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	0
Tooltip Text	

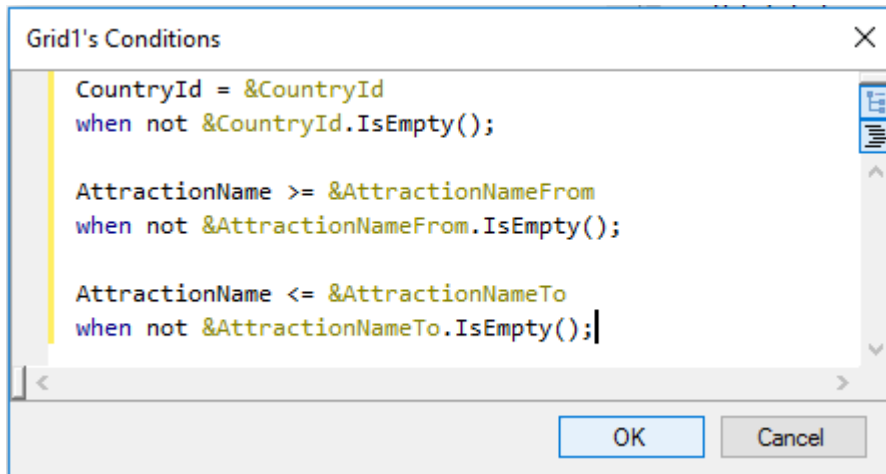
Layout

Behavior

Cell information

Row information

Properties | Toolbox | Responsive Sizes



Grid1's Conditions



```
CountryId = &CountryId
when not &CountryId.IsEmpty();

AttractionName >= &AttractionNameFrom
when not &AttractionNameFrom.IsEmpty();

AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();
```

OK Cancel

# Eventos

Country Id	&CountryId ▾					
Attraction Name From	&AttractionNameFrom					
Attraction Name To	&AttractionNameTo					
<b>GRID</b>						
Id	Attraction Name	Country	Photo	Trips		
AttractionId	AttractionName	CountryName		&trips		&newTrip
Total Trips	&totalTrips					

First time

Start

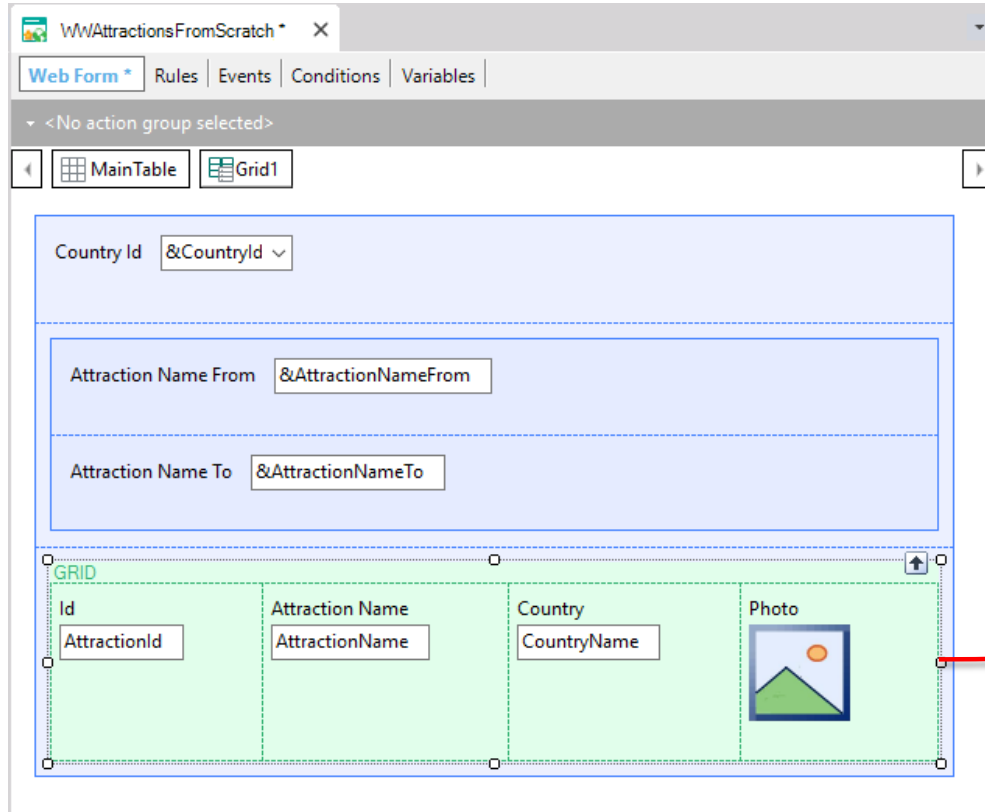
Refresh

Load

User / Control

Event

## Evento Load em Web Panel COM tabela base



The screenshot shows the GeneXus development environment for a web panel. The window title is "WWAttractionsFromScratch". The interface includes a menu bar with "Web Form", "Rules", "Events", "Conditions", and "Variables". Below the menu, there is a breadcrumb trail: "<No action group selected>". The main workspace contains a form with three input fields: "Country Id" with a dropdown menu labeled "&CountryId", "Attraction Name From" with a text input labeled "&AttractionNameFrom", and "Attraction Name To" with a text input labeled "&AttractionNameTo". Below the form is a grid component labeled "GRID". The grid has four columns: "Id" (with a sub-label "AttractionId"), "Attraction Name" (with a sub-label "AttractionName"), "Country" (with a sub-label "CountryName"), and "Photo" (with a small image icon). The grid is currently empty.

Evento LOAD

“N vezes,  
tantas  
quantos  
registros  
existirem na  
tabela  
percorrida”



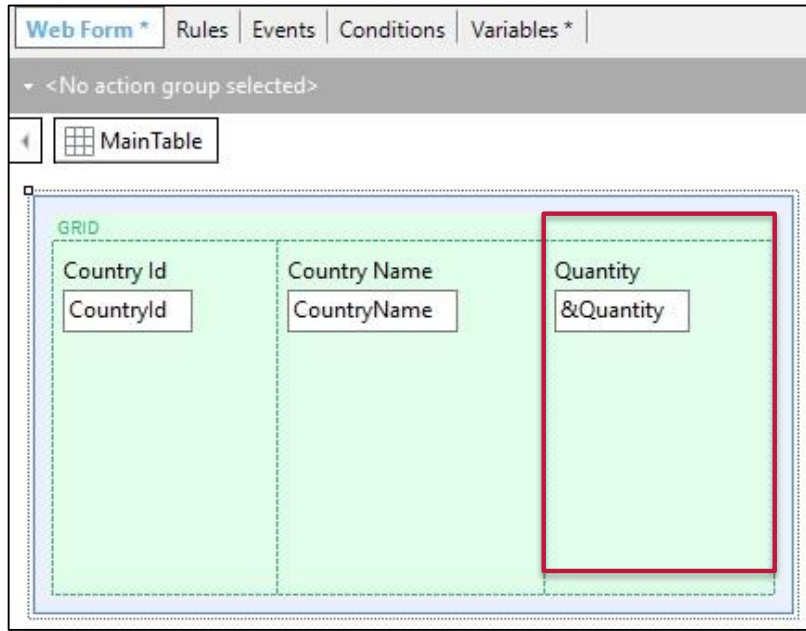
## Outro exemplo

### Country

```
{
  CountryId*
  CountryName
}
```

### Attraction

```
{
  AttractionId*
  AttractionName
  CountryId
  CountryName
}
```



### Event Load

&Quantity = Count(AttractionName)  
 endevent

# Evento Refresh

**Travel Agency**

CATEGORIES COUNTRIES ATTRACTIONS ▾

Country Id: (None) ▾

Attraction Name From:

Attraction Name To:

Attraction Name	Country	Attraction Photo	Trips
Christ the Redemmer	Brazil		1
Eiffel Tower	France		2
Forbidden city	China		0
Matisse Museum	France		1
Meet the Emperor	China		0
<b>Total Trips</b>			<b>4</b>

**Travel Agency** by GeneXus

CATEGORIES COUNTRIES ATTRACTIONS ▾

Country Id: France ▾

Attraction Name From:

Attraction Name To:

**Attraction Name Country Attraction Photo Trips**

Eiffel Tower	France		2
Matisse Museum	France		1
<b>Total Trips</b>			<b>7</b>

Refresh (once)  
Load (2 times)

```

Event Load
    &trips = count( TripDate )
    &totalTrips = &totalTrips + &trips
Endevent

Event Refresh
    &totalTrips = 0
Endevent
    
```

**Travel Agency**

CATEGORIES COUNTRIES ATTRACTIONS ▾


Country Id: France ▾

Attraction Name From:

Attraction Name To:

**Attraction Name Country Attraction Photo Trips**

Eiffel Tower	France		2
Matisse Museum	France		1
<b>Total Trips</b>			<b>3</b>



## Atributos na Grid

The screenshot displays the GeneXus IDE interface. The main workspace shows a web form with several fields: "Country Id" (dropdown), "Attraction Name From" (text), "Attraction Name To" (text), a "GRID" containing columns for "AttractionId", "AttractionName", "CountryName", "Photo", and "Trips", and "Total Trips" (text). A red arrow points from the "Visible" property in the Properties panel to the "AttractionId" column in the grid.

The Properties panel on the right shows the following configuration for the selected "AttractionId" attribute:

Attribute/Variable: AttractionId	
Attribute	AttractionId
Title	Id
Class	Attribute
Column Class	
Return On Click	False
On Click Event	
Control Info	
Control Type	Edit
Input Type	Values
Notify Context Char	False
Behavior	
Input History	True
Is Password	False
Read Only	True
Empty as null	Yes
Appearance	
Auto Resize	True
Format	Text
Visible	False
Tooltip Text	
Invite Message	

# Web panels sem tabela base

## Web Panels SEM TABELA BASE



Country Id

---

Attraction Name From

Attraction Name To

---

Attraction Id	Attraction Name	Country	Photo	Trips	
<input type="text" value="&amp;AttractionId"/>	<input type="text" value="&amp;AttractionName"/>	<input type="text" value="&amp;CountryName"/>		<input type="text" value="&amp;trips"/>	<input type="text" value="&amp;newTrip"/>

Total Trips



Evento LOAD  
"1 vez"

```

For each Attraction
  order CountryId, AttractionName when not &CountryId.IsEmpty()
  order AttractionName
  where CountryId = &CountryId when not &CountryId.IsEmpty()
  where AttractionName >= &AttractionNameFrom when not &AttractionNameFrom.IsEmpty()
  where AttractionName <= &AttractionNameTo when not &AttractionNameTo.IsEmpty()
  &AttractionId = AttractionId
  &AttractionName = AttractionName
  &CountryName = CountryName
  &AttractionPhoto = AttractionPhoto
  &trips = count( TripDate )
  Load
  &totalTrips = &totalTrips + &trips
endfor
Endevent
  
```



Web panels  
Múltiplas grids

## Múltiplas grids

The diagram shows two separate grids stacked vertically. The top grid is labeled 'GRID' and contains two input fields: 'Category Id' with value 'CategoryId' and 'Category Name' with value 'CategoryName'. The bottom grid is also labeled 'GRID' and contains four input fields: 'Attraction Id' with value 'AttractionId', 'Attraction Name' with value 'AttractionName', 'Country Name' with value 'CountryName', and 'Attraction Photo' with a placeholder image of a landscape with a sun and a mountain.

PARALELO

Navegação independente

FreeStyleGrid

The diagram shows a nested grid structure. The outermost container is labeled 'GRID' and contains two input fields: 'Category Id' with value 'CategoryId' and 'Category Name' with value 'CategoryName'. A red arrow points from the text 'FreeStyleGrid' to this container. Inside this container is another grid labeled 'GRID' which contains four input fields: 'Attraction Id' with value 'AttractionId', 'Attraction Name' with value 'AttractionName', 'Country Name' with value 'CountryName', and 'Attraction Photo' with a placeholder image of a landscape with a sun and a mountain.

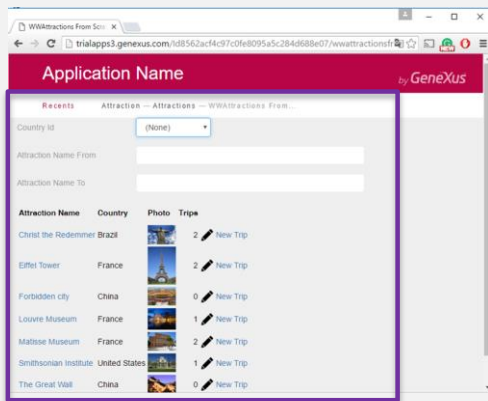
ANINHADO

Navegação com tabelas relacionadas

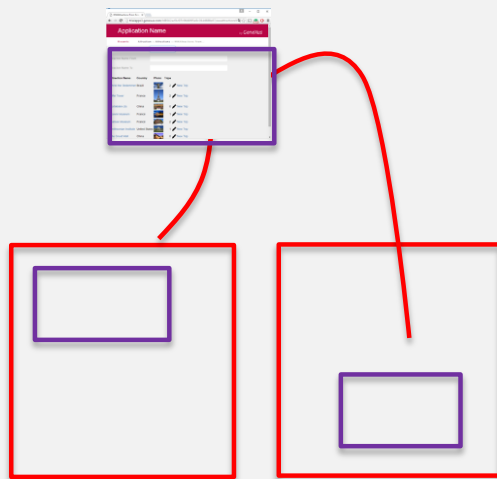
# Tipos de Web Panels

## Tipos de Web Panels

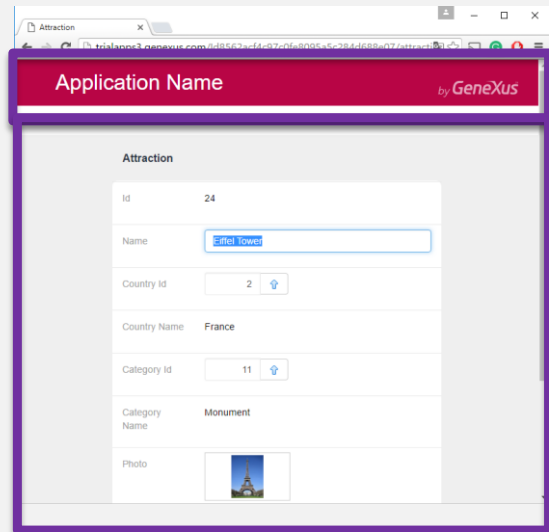
### Web page (default)



### Component



### Master page



# Design Systems

## Design Systems

Elementos que dão  
uniformidade e  
coerência ao UX

**Master Page**

e.g. Header / footer

**Theme**

Classes

**Responsive sizes**

Responsive tables

**Controls**

User control objects

**Base styles**

CSS libraries

**Stencils**

Design components

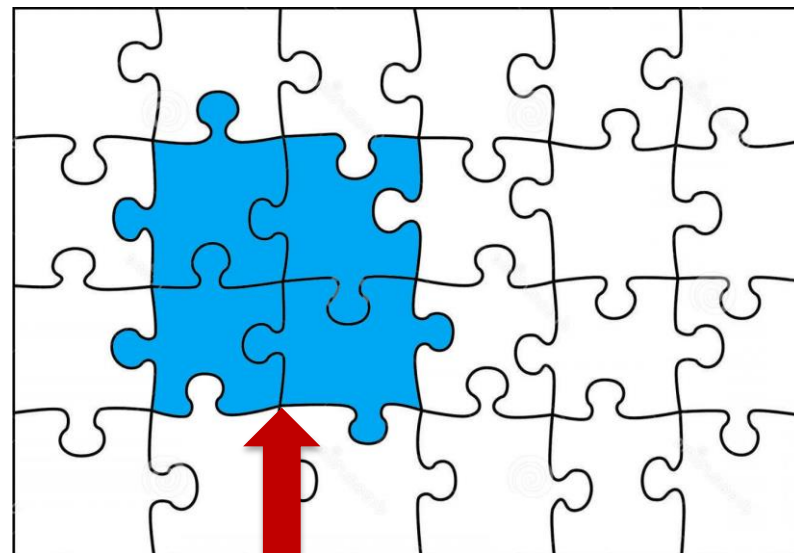
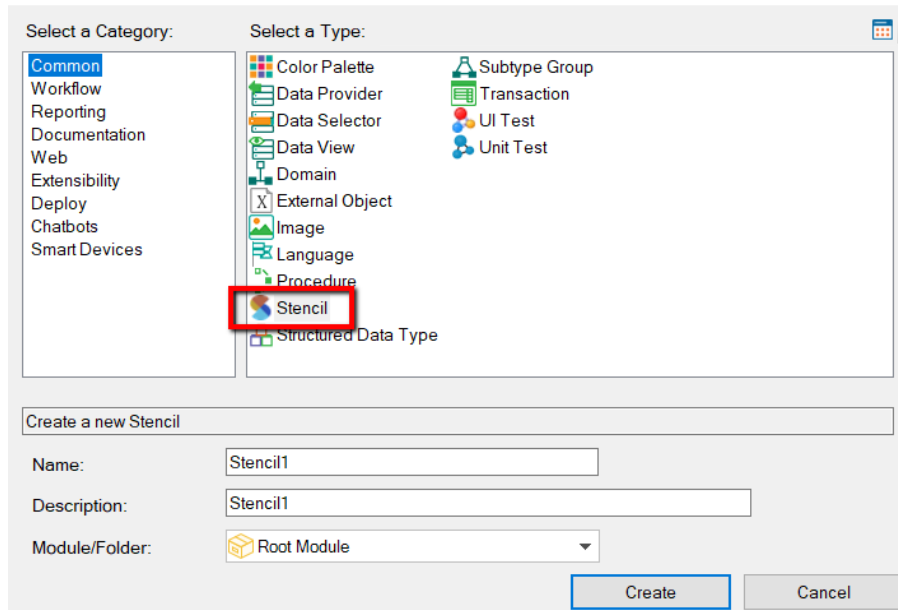
**Patterns**

Design / behaviour



**RWD**

## Stencils



Design component

Objeto que permite repetir o desenho da mesma parte da tela (um conjunto de controles), em várias telas.

# Responsive Web Design



# Responsive Web Design

Application Name by GeneXus

Recents Attraction — Attractions

**Attractions**  + INSERT

Ordered By: **Name**

COUNTRY NAME

Id	Name	Country Name	Category Name	Photo	City Name
25	Christ the Redemmer	Brazil	Monument		Rio de Janeiro <span>UPDATE</span> <span>DELETE</span>
24	Eiffel Tower	France	Monument		Paris <span>UPDATE</span> <span>DELETE</span>
28	Forbidden city	China	Tourist Site		Beijing <span>UPDATE</span> <span>DELETE</span>
22	Louvre Museum	France	Museum		Paris <span>UPDATE</span> <span>DELETE</span>
27	Matisse Museum	France	Museum		Nice <span>UPDATE</span> <span>DELETE</span>
26	Smithsonian Institute	United States	Museum		Washington <span>UPDATE</span> <span>DELETE</span>
23	The Great Wall	China	Tourist Site		Beijing <span>UPDATE</span> <span>DELETE</span>

Application Name by GeneXus

Recents Attraction — Attractions

**Attractions**  + INSERT

Ordered By: **Name**

COUNTRY NAME

Name	UPDATE	DELETE
Christ the Redemmer	UPDATE	DELETE
Eiffel Tower	UPDATE	DELETE
Forbidden city	UPDATE	DELETE
Louvre Museum	UPDATE	DELETE
Matisse Museum	UPDATE	DELETE
Smithsonian Institute	UPDATE	DELETE

Application Name by GeneXus

Recents

**Attractions** + INSERT

Christ the Redemmer	UPDATE	DELETE
Eiffel Tower	UPDATE	DELETE
Forbidden city	UPDATE	DELETE
Louvre Museum	UPDATE	DELETE
Matisse Museum	UPDATE	DELETE
Smithsonian Institute	UPDATE	DELETE

## Responsive Web Design

The screenshot displays the GeneXus IDE interface for a web form titled "WAttractionsFromScratch". The main workspace shows a form layout with several fields: "Country Id" (with a dropdown), "Attraction Name From", "Attraction Name To", a "GRID" section with columns for "Id", "Attraction Name", "Country", "Photo", "Trips", and a new trip button, and "Total Trips".

The "Responsive Sizes" panel on the right is used to configure the form's behavior across different screen sizes. The "Size" dropdown is set to "Medium (Desktop >= 992 px) inherits from Small". The "Table Default" section shows "Label Width" set to 25%.

The "Values" section shows the following settings:

- Width: 100%
- Label Width: 25%
- Offset: 0%
- Visible: True
- Move: < >
- Default

At the bottom, there is a checkbox for "Use default values for all screen sizes".

Red arrows indicate the mapping between the responsive size configuration and the form elements:

- The "Size" dropdown is highlighted in red, with an arrow pointing to the "Country Id" field.
- The "1,1 &CountryId" row in the responsive size table is highlighted in purple, with an arrow pointing to the "Country Id" field.
- The "2,1 Table1" row is highlighted in light blue, with an arrow pointing to the "Attraction Name From" field.
- The "3,1 Grid1" row is highlighted in light blue, with an arrow pointing to the "GRID" section.
- The "4,1 &totalTrips" row is highlighted in light blue, with an arrow pointing to the "Total Trips" field.

# GeneXus Server

## Send Knowledge Base to GeneXus Server

Send Knowledge Base

**Send the Knowledge Base to GeneXus Server**

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

Travel\_Agency

Security

Authentication Type: GXTechnical

Username:

Password:

Save Password

Alias:

Work with Lock Model

All versions in Knowledge Base

## Team Development

Team Development X

Commit to: [http://sandbox.genexusserver.com/salto/home.aspx?Travel\\_Agency\\_0](http://sandbox.genexusserver.com/salto/home.aspx?Travel_Agency_0)

Pattern:

Category:  Folder:

Pending Commits (8/8) Ignored Objects Refresh

Drag a column here to group by that column

<input checked="" type="checkbox"/>		Name	Type	Description	Modified On	Module	Action	Last Synchroniz	User
<input checked="" type="checkbox"/>		Airline	Transaction	Airline	20/6/2016 9:33	Root Module	Modified	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		AirlineFlight1WC	Web Component	Airline Flight1 WC	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		AirlineFlightWC	Web Component	Airline Flight WC	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		AirlineGeneral	Web Component	Airline General	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		AttractionsAndCategor...	Procedure	Attractions And Ca...	20/6/2016 9:33	Root Module	Modified	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		ViewAirline	Web Panel	View Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		WorkWithAirline	Work With for W...	Work With Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...
<input checked="" type="checkbox"/>		WwAirline	Web Panel	Work With Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECHlacaggia...

Add Knowledge Base properties to list

Remind me to move changes to... Cancel Commit

Commit Update History Versions

# Commit

## Knowledge Manager / Team Development

Partial commit

Team Development X

Commit to: [http://sandbox.genexusserver.com/salto/home.aspx?Travel\\_Agency\\_0](http://sandbox.genexusserver.com/salto/home.aspx?Travel_Agency_0)

Pattern:

Category:  Folder:

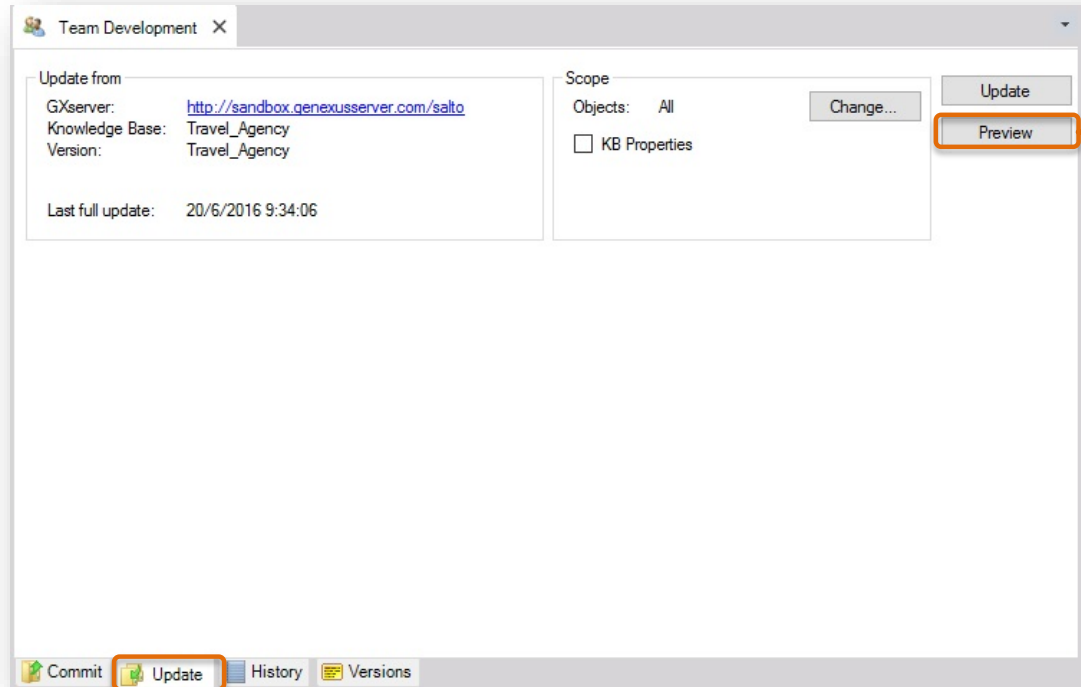
Pending Commits (8/8) Ignored Objects

Drag a column here to group by that column

<input checked="" type="checkbox"/>		Name	Type	Description	Modified On
<input checked="" type="checkbox"/>		Airline	Transaction	Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>		AirlineFlight1WC	Web Component	Airline Flight1 WC	20/6/2016 9:33
<input checked="" type="checkbox"/>		AirlineFlightWC	Web Component	Airline Flight WC	20/6/2016 9:33
<input checked="" type="checkbox"/>		AirlineGeneral	Web Component	Airline General	20/6/2016 9:33
<input checked="" type="checkbox"/>		AttractionsAndCategor...	Procedure	Attractions And Ca...	20/6/2016 9:33
<input checked="" type="checkbox"/>		ViewAirline	Web Panel	View Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>		WorkWithAirline	Work With for W...	Work With Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>		WWAirline	Web Panel	Work With Airline	20/6/2016 9:33

Add Knowledge Base properties to list

# Update



Partial selection

# History

The screenshot shows the 'Team Development' window with the 'History' tab selected. The window displays a list of changes and a detailed view of a specific commit.

Version:  Refresh

From:  To:  Search:  Apply

Drag a column here to group by that column

#	Comment	User	Commit Date
3	WorkWithAirline and AttractionsAndCategories listing	GXTechnical\acaggiano	20/06/2016 10:10 a. m.
1	Travel_Agency is now hosted by GeneXus Server	GXTechnical\acaggiano	20/06/2016 09:33 a. m.

#3 - 20/6/2016 10:10:22 - GXTechnical\acaggiano

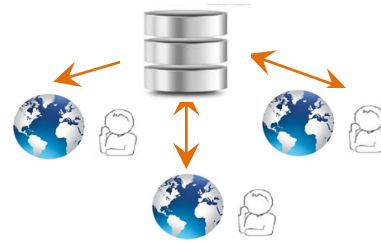
WorkWithAirline and AttractionsAndCategories listing

Name	Type	Description	Action
Airline	Transaction	Airline	Modified
AirlineFlight1WC	Web Panel	Airline Flight1 WC	Inserted
AirlineFlightWC	Web Panel	Airline Flight WC	Inserted
AirlineGeneral	Web Panel	Airline General	Inserted
AttractionsAndCategories	Procedure	Attractions And Categories	Modified
ViewAirline	Web Panel	View Airline	Inserted
WorkWithAirline	Work With for Web	Work With Airline	Inserted

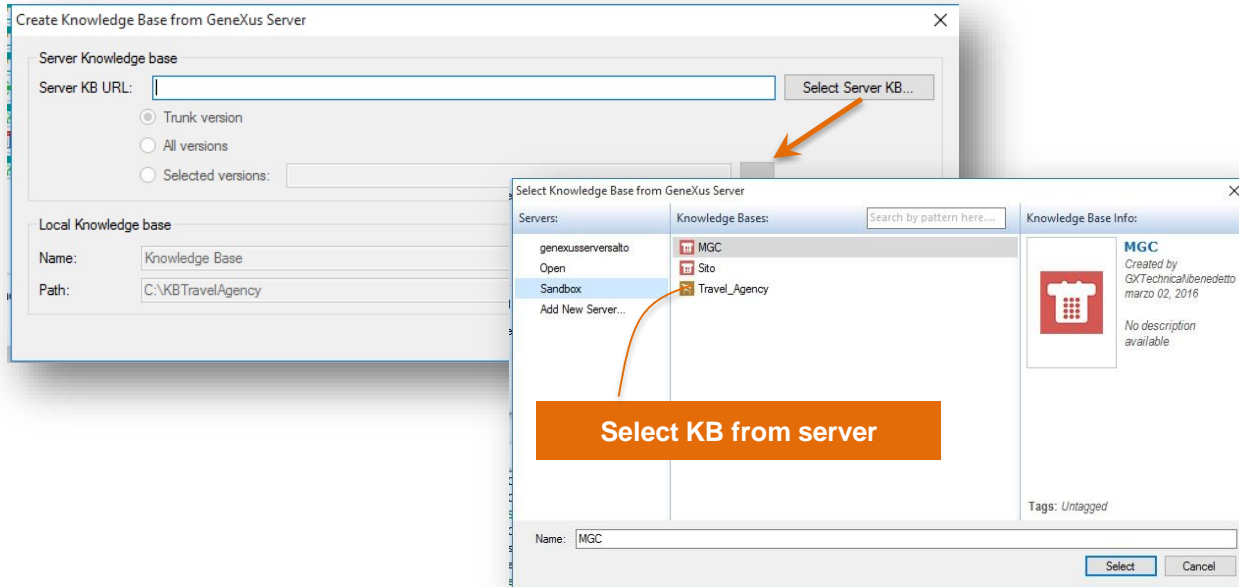
Commit Update **History** Versions



## Create KB from GeneXus Server



File / New / Knowledge Base from Server



# Segurança com GAM



**AUTHENTICATION**



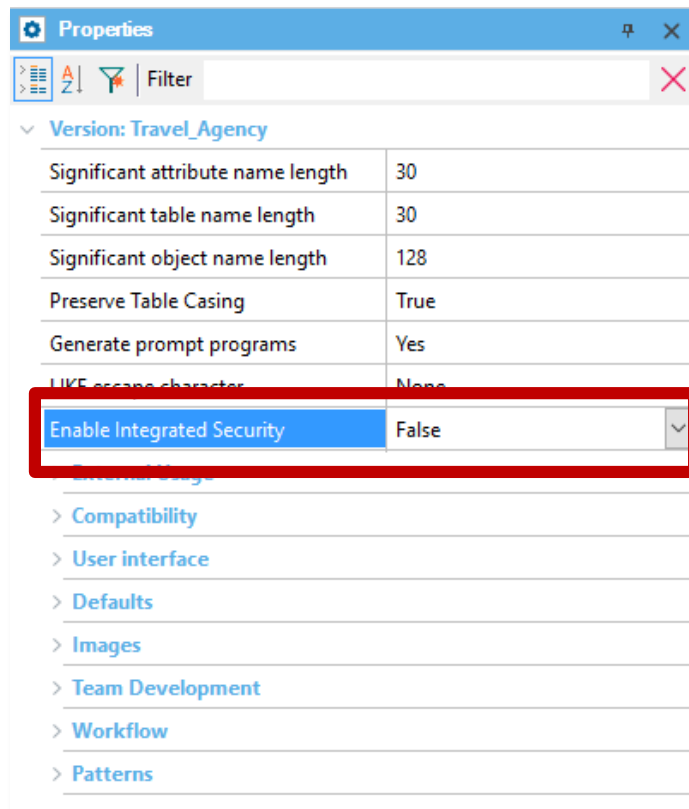
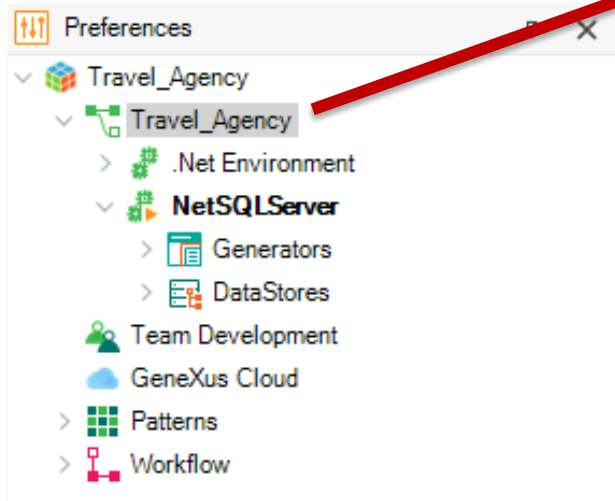
**AUTHORIZATION**

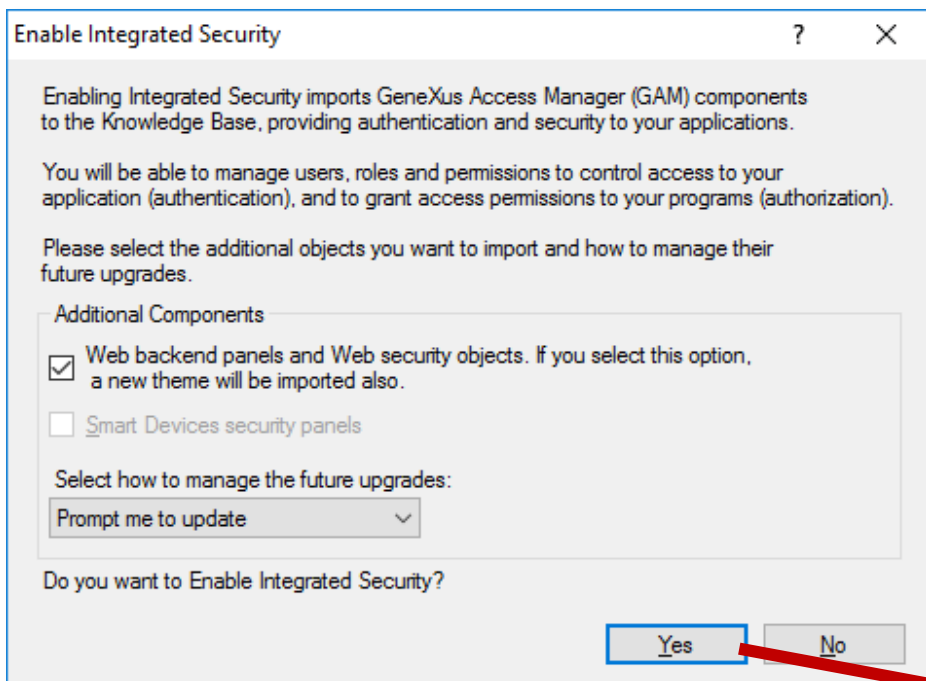
# GeneXus™

ACCESS MANAGER

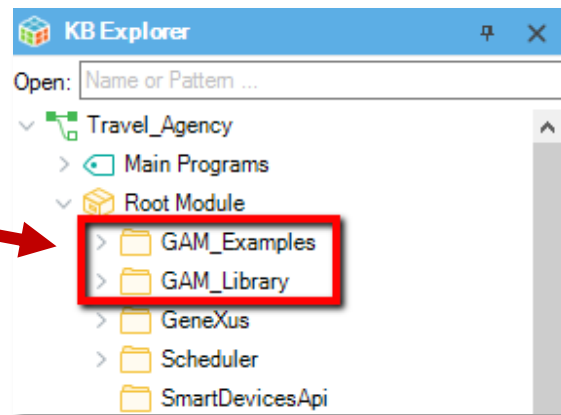


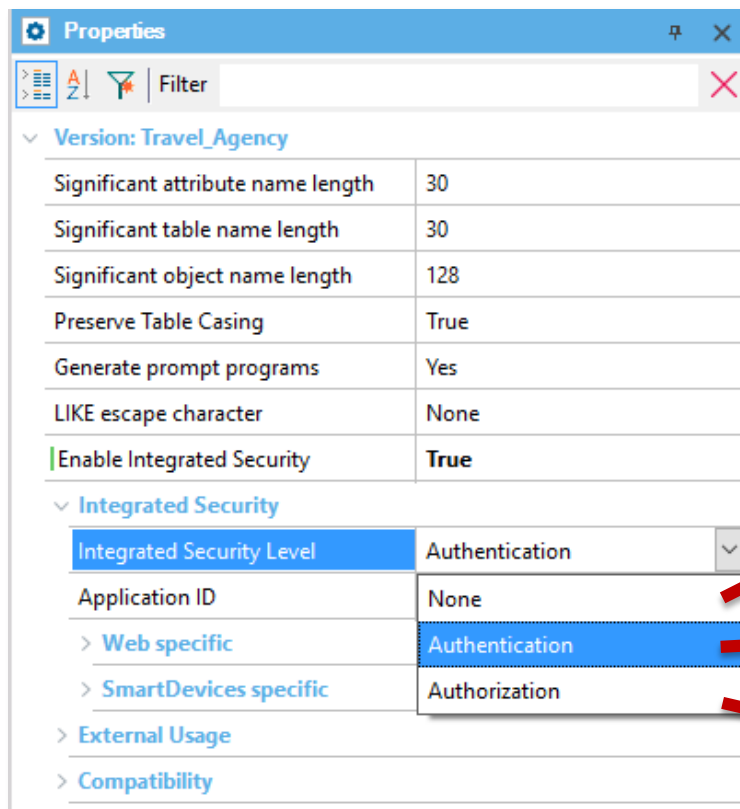
## Habilitando o GeneXus Access Manager



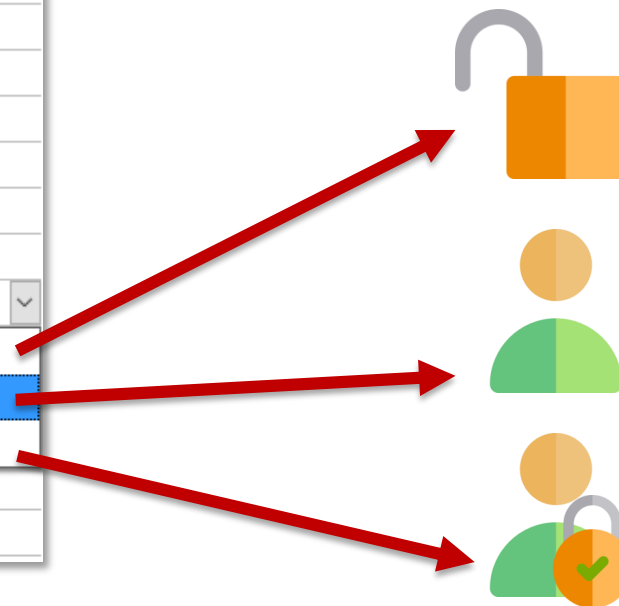


## Importando os objetos do GAM





## Selecionando o nível de segurança



## Acesso ao backend do GAM (objeto GAM Home)

The screenshot displays the 'Users' management interface in the GeneXus Access Manager. The top navigation bar includes 'GENEXUS ACCESS MANAGER', 'USERS', 'ROLES', and 'SETTINGS'. The user 'Administrator' is logged in. The main content area shows a table of users with the following data:

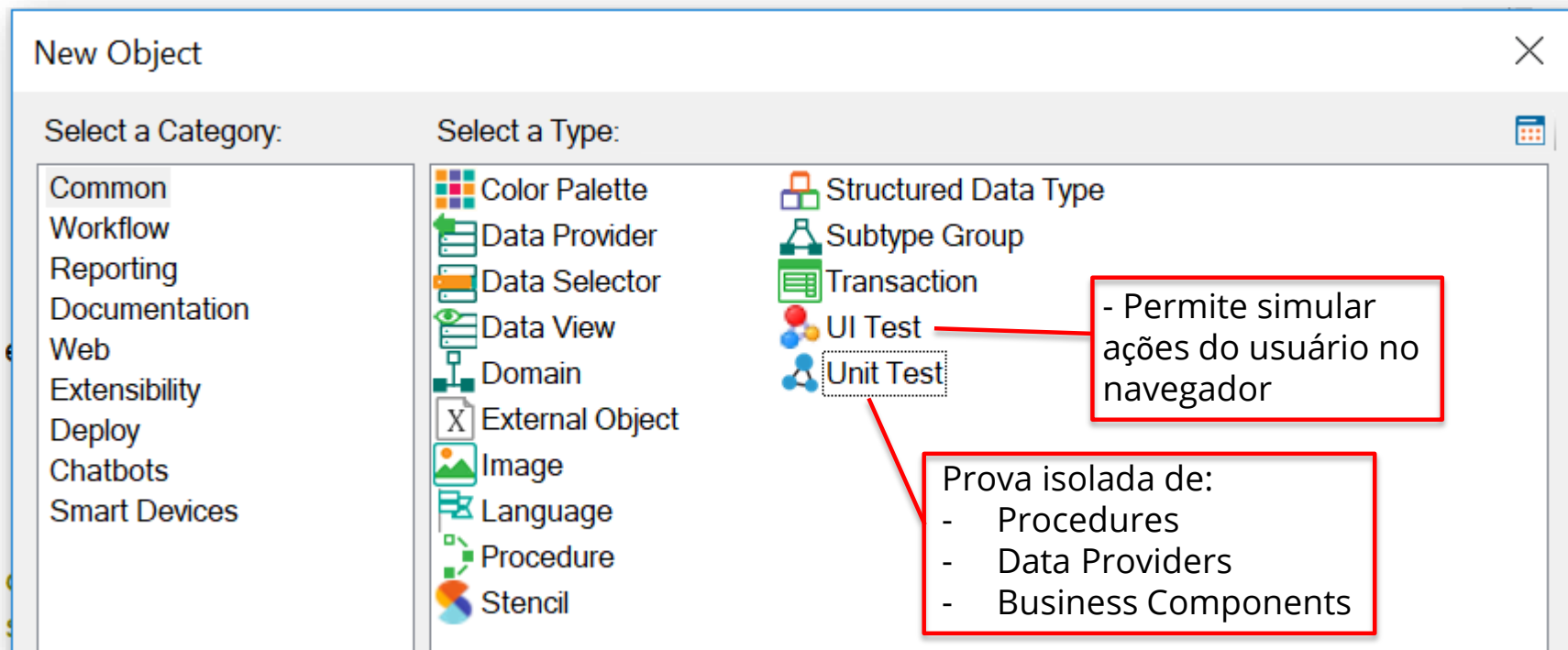
User Name	First Name	Last Name	Authentication	
<a href="#">admin</a>	Administrator	User	local	<a href="#">EDIT</a>

Navigation controls include 'SHOW FILTERS', a search bar with 'Try a Search', '+ ADD', and pagination links 'FIRST / PREV / NEXT'.

# Testing



## Objetos para gerar testes unitários e testes de interface



## Teste Unitário

São criados os objetos:

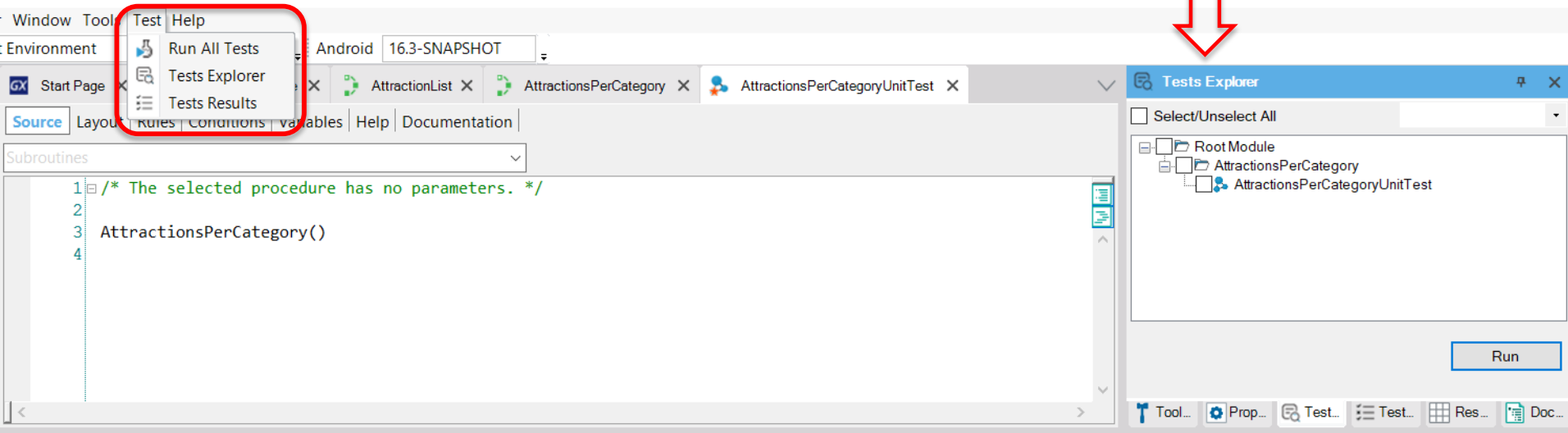
- <ObjectName>UnitTest
- <ObjectName>UnitTestData
- <ObjectName>UnitTestData

The screenshot shows the GeneXus IDE interface. The top bar contains several tabs: 'Start Page', 'AttractionsByName', 'AttractionList', and 'AttractionsPerCategory'. Below the tabs is a menu bar with 'Source', 'Layout', 'Rules', 'Conditions', 'Variables', 'Help', and 'Documentation'. A dropdown menu is open, showing options like 'Close', 'New Horizontal Tab Group', 'New Vertical Tab Group', 'Close All But This', 'Full Screen', 'Open', 'Open Part', 'Delete', 'Save As...', 'References', 'History', 'Team Development', 'Properties', 'Locate in KB Explorer', 'Build', 'Rebuild', 'Run', 'Run Without Building', 'Run With This Only', 'Build With This Only', 'Set As Startup Object', 'View Last Navigation', 'View Navigation', 'Export', 'Select Left Side To Compare', and 'Create Unit Test' (highlighted in yellow).

The main editor area shows a subroutine named 'Subroutines' with the following code:

```
1 Print Title
2
3 For each Attraction order CategoryId
4     Print Categories
5     Print ColumnTitles
6     For each Attraction
7         Print Attractions
8     Endfor
9 Endfor
10
```

## Test Explorer



The screenshot displays the GeneXus IDE interface. The main window shows a source code editor with the following content:

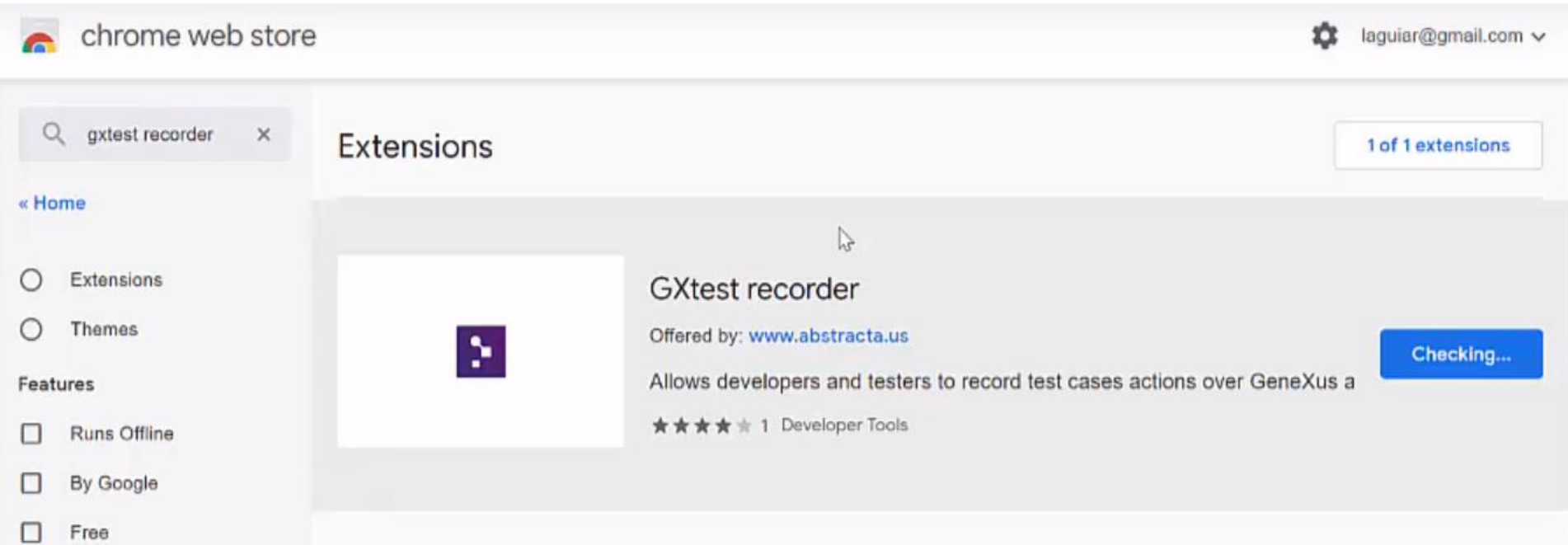
```
1 /* The selected procedure has no parameters. */  
2  
3 AttractionsPerCategory()  
4
```

The **Test** menu is open, and the **Test Explorer** option is highlighted. A red box highlights the **Test** menu and the **Test Explorer** option. A red arrow points to the **Test Explorer** window on the right side of the IDE. The **Test Explorer** window shows a tree view with the following structure:

- Root Module
  - AttractionsPerCategory
    - AttractionsPerCategoryUnitTest

The **Run** button is visible at the bottom right of the **Test Explorer** window.

## Teste de Interface: GXTest



The screenshot shows the Chrome Web Store interface. At the top left is the 'chrome web store' logo. At the top right is the user profile 'laguiar@gmail.com' with a gear icon. A search bar contains 'gxtest recorder' with a magnifying glass icon and a close button. Below the search bar is a left sidebar with navigation options: 'Home', 'Extensions', and 'Themes'. Under 'Features', there are three checkboxes: 'Runs Offline', 'By Google', and 'Free'. The main content area is titled 'Extensions' and shows a search result for 'GXtest recorder'. The extension card includes a purple icon, the title 'GXtest recorder', the developer 'www.abstracta.us', a description 'Allows developers and testers to record test cases actions over GeneXus a', a rating of 1 star, and a 'Checking...' button.

chrome web store

laguiar@gmail.com

gxtest recorder

Extensions

1 of 1 extensions

« Home

Extensions


Themes

Features

Runs Offline

By Google

Free



**GXtest recorder**

Offered by: [www.abstracta.us](http://www.abstracta.us)

Allows developers and testers to record test cases actions over GeneXus a

★ ★ ★ ★ ☆ 1 Developer Tools

Checking...

# Teste de Interface

Command Target Value

Command	Target	Value
ClickBy	id=COUNTRYID	
SelectBy	id=COUNTRYID	label=Uruguay
ClickBy	id=COUNTRYID	
ClickBy	id=ATTRACTIONNAME_Balloor	
AssertTextBy	id=ATTRACTIONNAME_Balloor	Attraction,Country already exists
ClickBy	id=CITYID	
ClickBy	id=ATTRACTIONNAME	
TypeBy	id=ATTRACTIONNAME	Estadio Centenario 2

COMMAND: [input]  
 TARGET: [dropdown] [Select] [Find]  
 VALUE: [input]

Log Reference Debug Save Clear

Command Target Value

ClickBy	id=COUNTRYID	
SelectBy	id=COUNTRYID	label=Uruguay
ClickBy	id=COUNTRYID	
ClickBy	id=ATTRACTIONNAME_Balloor	
AssertTextBy	id=ATTRACTIONNAME_Balloor	Attraction,Country already exists
ClickBy	id=CITYID	
ClickBy	id=ATTRACTIONNAME	
TypeBy	id=ATTRACTIONNAME	Estadio Centenario 2

COMMAND: [dropdown]  
 TARGET: [input] [Select] [Find]  
 VALUE: [input]

Log Reference Debug Save Clear

```
[info] Executing: | ClickBy | id=ATTRACTIONNAME_Balloor | |
[info] Executing: | AssertTextBy | id=ATTRACTIONNAME_Balloor |
Attraction,Country already exists |
[info] Executing: | ClickBy | id=CITYID | |
[info] Executing: | ClickBy | id=ATTRACTIONNAME | |
```

Estadio Centenario

Attraction,Country already exists

Uruguay

Rio de Janeiro

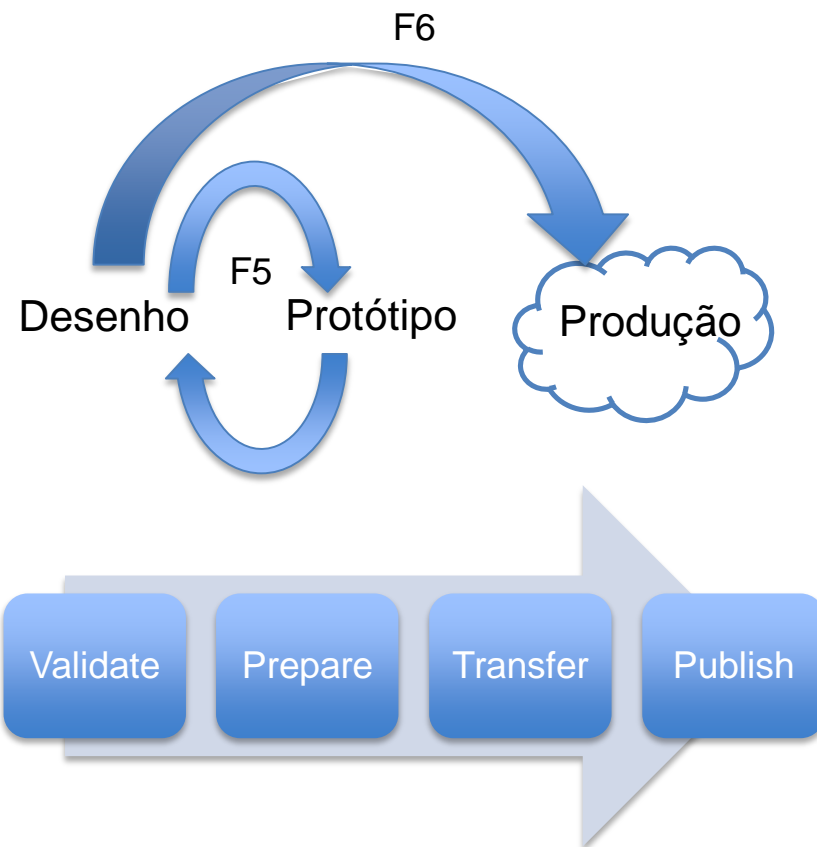
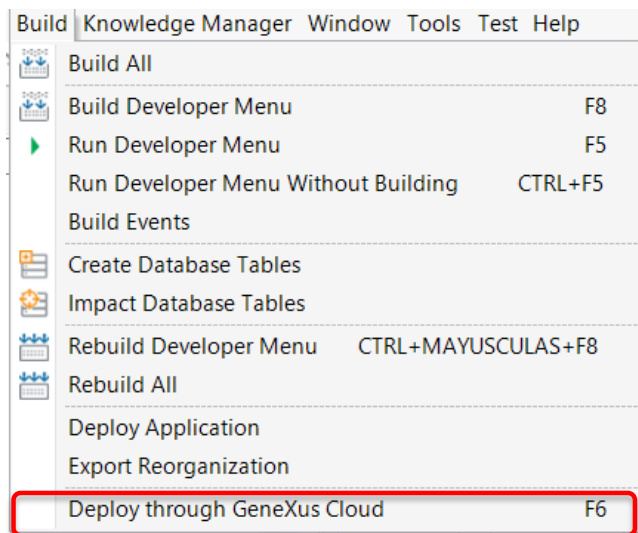
Monument

Command Recorded  
 command: ClickBy  
 target: id=BTNINSERT

UI Test (Requer licença GXTest)

# Deployment

## Deploy automático com F6



Build Knowledge Manager Window Tools Test Help

- Build All
- Build MenuTravelAgency F8
- Run MenuTravelAgency F5
- Run MenuTravelAgency Without Building CTRL+F5
- Build Events
- Create Database Tables
- Impact Database Tables
- Rebuild MenuTravelAgency CTRL+MAYUSCULAS+F8
- Rebuild All
- Deploy Application**
- Export Reorganization
- Deploy through GeneXus Cloud

## Application Deployment Tool

Deploy Application X

Select objects to deploy

New Deployment Unit ... Add...

- Web Panel
  - Home

Objetos main

Target: Local Options

Application Server: Microsoft IIS 7

Deploy



# Integração

# Módulo de Inteligência Artificial

Manage Module References

Modules: Search by pattern here...

**Chatbot (2.1.10.129299)**  
GeneXus Chatbot module is a basic set of interfaces and implementations of data structures and algorithms needed to implement a Chatbot solution.

**GeneXusAI (1.1.21.129329)** Install  
GeneXusAI contains a common set of Artificial Intelligence tasks, including audio, text and image processing, all of

**GeneXus (2.1.7.129290)**  
GeneXus Core Module is a basic set of interfaces and

**GXtest (0.4.2)**  
GXtest Module provides core functionality for creating, running and reporting tests on genexus and over ci/cd pipelines. <https://wiki.genexus.com/commwiki/servlet/wiki?>

MyKB

- Main Programs
- Root Module
- References
  - GeneXus
    - GeneXusAI**
  - Customization
  - Documentation

Module Information:

**GeneXusAI**

**Module is not installed**

Available Versions: 1.1.21.129329

**Author:** GeneXus S.A.  
**Owner:** GeneXus S.A.

**Description:**

GeneXusAI contains a common set of Artificial Intelligence tasks, including audio, text and image processing, all of them provided by several Cloud Platforms (e.g. IBM Watson, Microsoft Azure Cognitive Services, SAP Leonardo)

**Platforms:**

- C# Web

**Dependencies:**

- GeneXus 1.12.13.125610

**Id:** 733e9734-9f0a-4620-b397-d84fbc2cef10

- GeneXusAI
  - Audio
    - SpeechToText
    - TextToSpeech
    - Domains
  - Configuration
    - Provider
    - Domains
  - Image
    - Classify
    - DetectFaces
    - DetectObjects
    - DetectScene
    - OCR
  - Text
    - DetectLanguage
    - ExtractEntities
    - KeyPhrases
    - SentimentAnalysis
    - Translate
  - Video
    - Analyze
    - OutputAnalysis
    - Process
    - Domains

IBM Watson

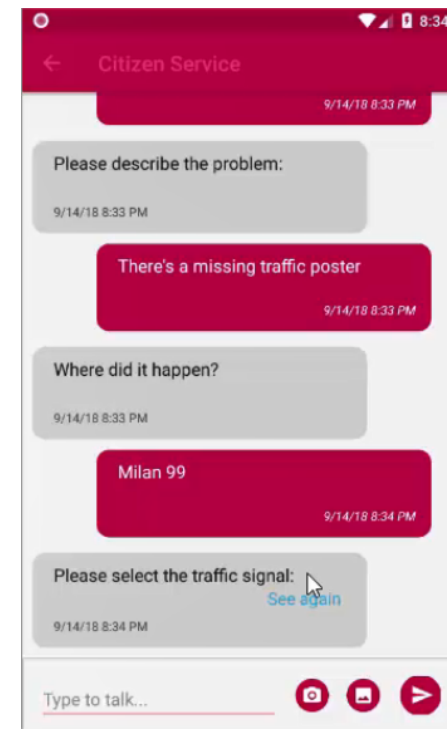
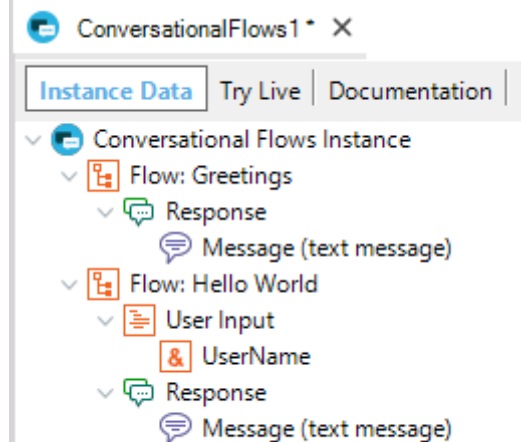
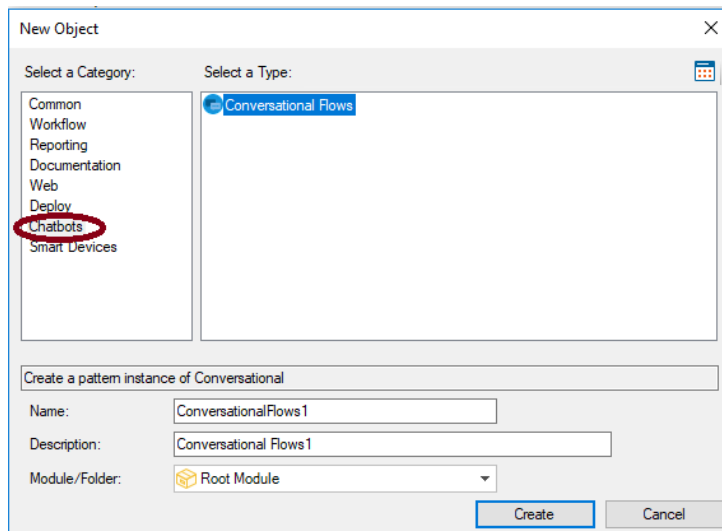
Microsoft Cognitive Services

SAP Leonardo

Google Cloud Services

Amazon Web Services

## Chatbots: objeto Conversational Flow



# Smart Devices

## Aplicar Pattern em Transação

The screenshot shows the 'Patterns' dialog box for a 'Country' object. The 'Country' title bar is highlighted with a red box. The 'Patterns' tab is also highlighted with a red box. Below the tabs, the text 'Patterns usable in this object (underlined means pattern is applied)' is displayed. The 'Work With for Smart Devices' pattern is selected and underlined, and its tab is highlighted with a red box. A red circle highlights the checked checkbox for 'Apply this pattern on save'. A red arrow points to the 'Level (Country)' tree item. The tree structure is as follows:

- Level (Country)
  - List
  - Detail
    - Section (General)
    - Section (Place)
    - Section (Sale)

A large empty box on the right contains the text: 'Select List, Detail or Section on the tree to edit it'.

## Aplicar Pattern em Transação

The screenshot shows the GeneXus KB Explorer interface. On the left, the 'Country' project tree is expanded to 'Country\_DataProvider', where 'WorkWithDevicesCountry' is selected. A red arrow points to this object. The main workspace shows the 'Patterns' tab with 'WorkWithDevice' applied to the selected object. The 'Apply this pattern on save' checkbox is checked. The 'Properties' window on the right shows the 'Main program' property set to 'True', which is highlighted with a red box. A context menu is open over the 'WorkWithDevice' object, with the 'Run' option highlighted.

Properties window:

Location When In Use Use	
Log Maximum Size in KByte	5000
Main program	<b>True</b>
Microphone Usage Descrip	
Motion Usage Description	
Multitex Build	False
Name	<b>WorkWithDe</b>
Obfuscate Application	False
Object Visibility	Public
Offline Data Base Access L	Off
Offline Database	(none)
Offline Synchronization Lo	Off

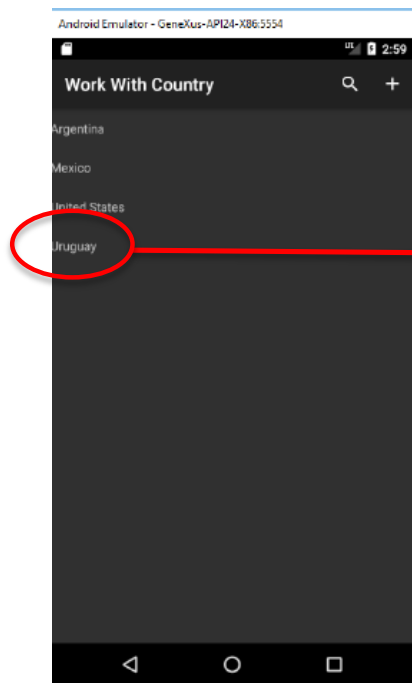
Context menu:

- New
- Open F12
- Open Part
- Copy CTRL+C
- Cut CTRL+X
- Paste CTRL+V
- Delete SUPR
- Save As...
- References CTRL+F12
- History CTRL+MAYUSCULAS+H
- Properties F4
- Build
- Rebuild
- Run**
- Run Without Building
- Run With This Only
- Build With This Only

Tem que configurar sua propriedade **Main program = True** e, sobre o objeto, clicar com o **botão direito e Run**

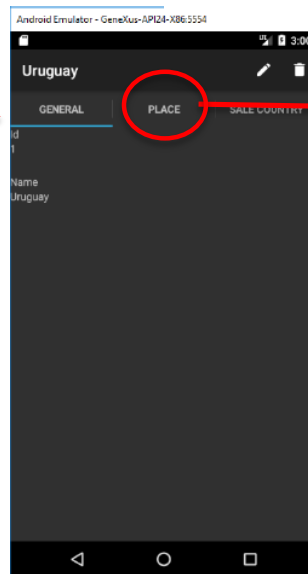
## Aplicar Pattern em Transação

botão direito e Run:

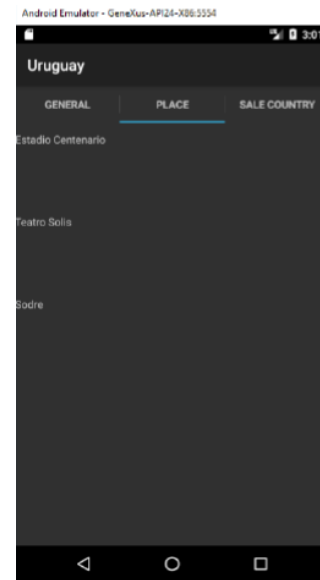
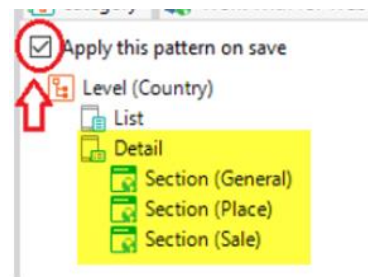


GENERAL

Escolho 1 country

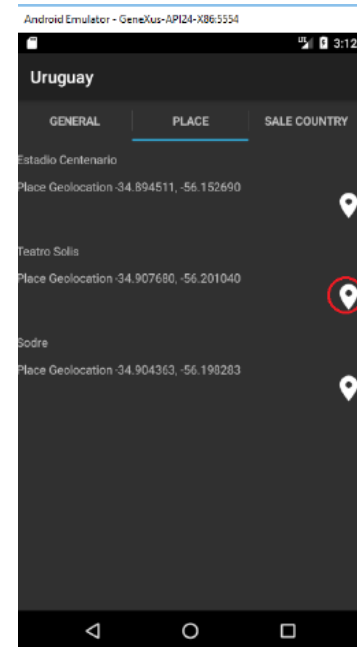
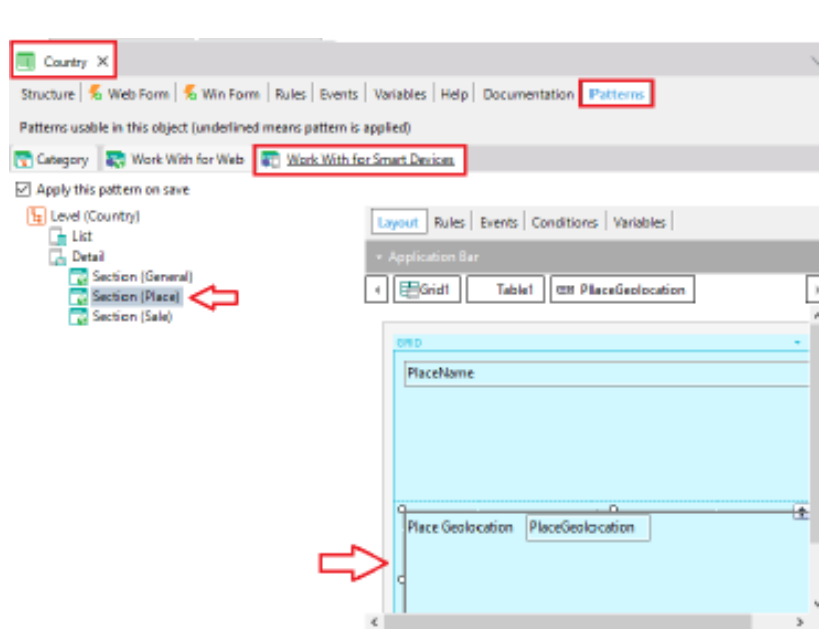


PLACE



## Aplicar Pattern em Transação

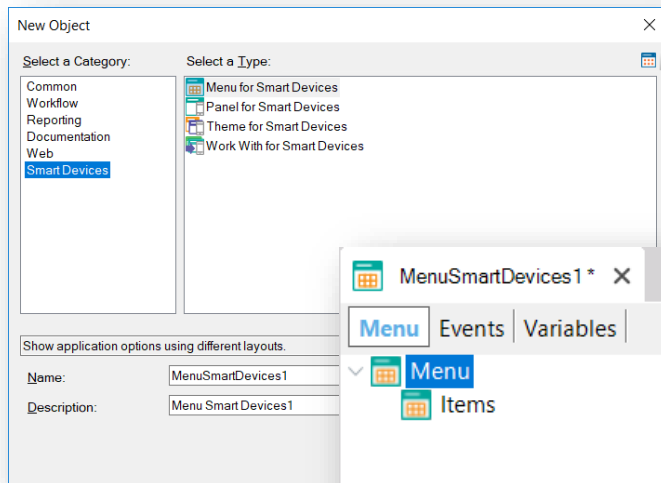
Queremos, para cada lugar, além de ver seu nome, ver sua geolocalização.





## Menu de acesso: Menu for Smart Devices

Criação:



MenuSmartDevices1 \* X

Menu Events Variables

Menu

Items

Properties

Name	MenuSmartDevices1
Description	Menu Smart Devices1
Module/Folder	Root Module
Qualified Name	MenuSmartDevices1
Object Visibility	Public
Auto Update	False
Main program	<b>True</b>

Add action...

MenuSmartDevices1 \* X

Menu Events Variables

Menu

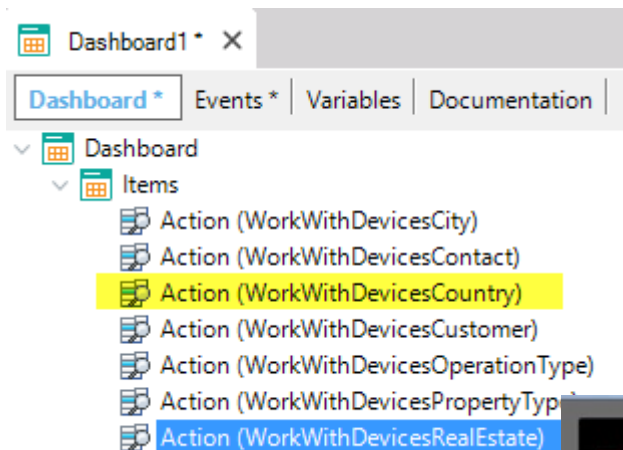
Items

Add

- Delete SUPR
- Cut CTRL+X
- Copy CTRL+C
- Paste CTRL+V
- Expand All
- Collapse All
- Properties F4

Action

Link



Ao pressionar F5...



# *GeneXus*<sup>™</sup>

Videos

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

Documentation

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

Certifications

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