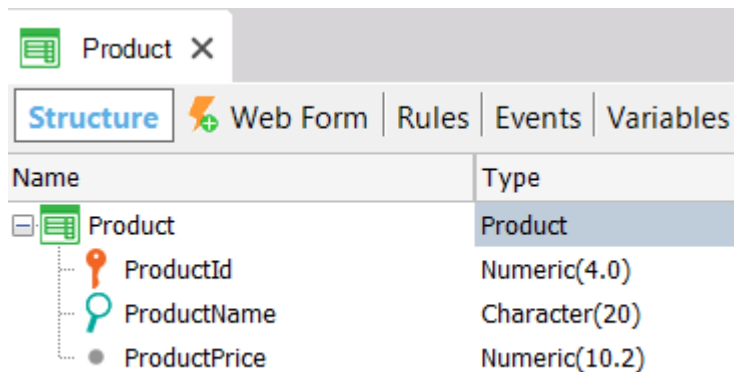
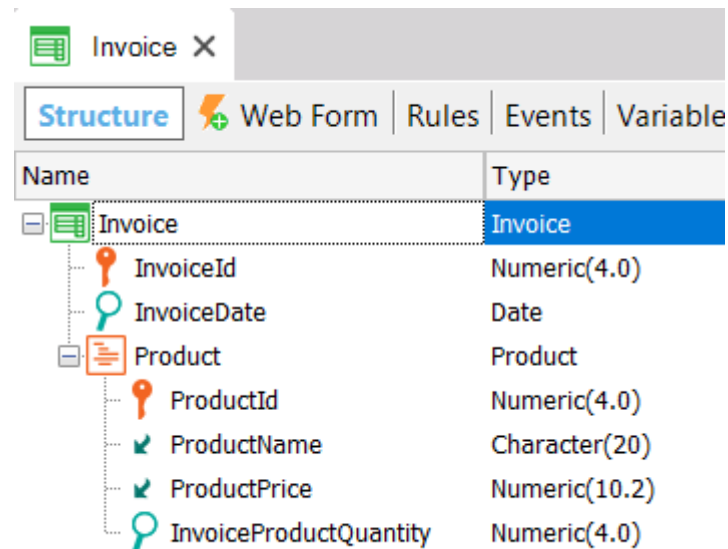


Transacciones

¿Qué nombres 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

Entidad + Categoría [+ Calificador]

Product

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

Invoice

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

Diseño de Transacciones

1 - N fuerte

Cada cliente pertenece a un país y un país tiene muchos clientes



| COUNTRY |
|-------------|
| CountryId* |
| CountryName |



Country
{
CountryId* (PK)
CountryName
}

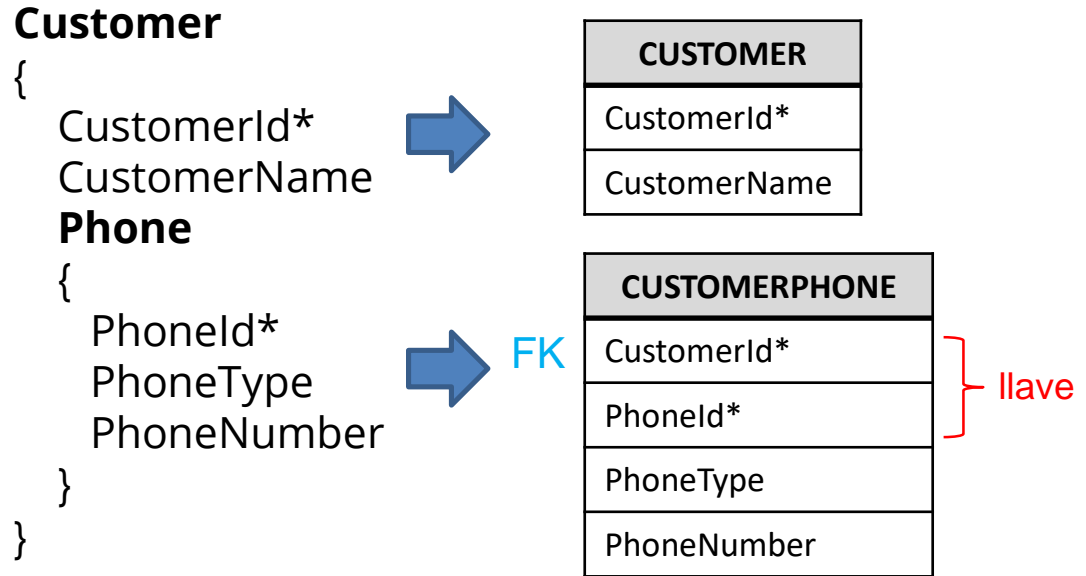
Customer
{
CustomerId*
CustomerName
CountryId (FK)
CountryName
}



| CUSTOMER |
|--------------|
| CustomerId* |
| CustomerName |
| CountryId |

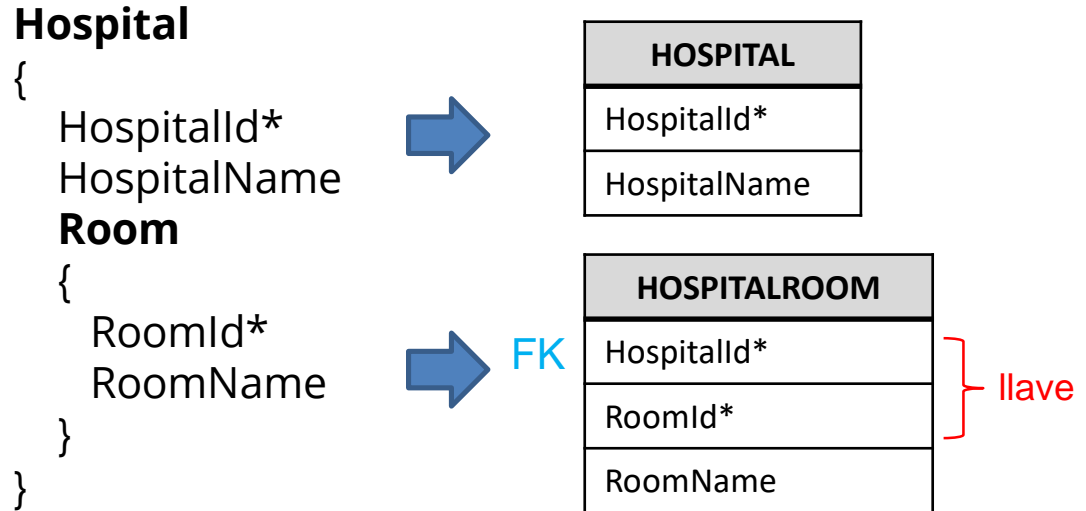
1 - N débil

Cada cliente tiene muchos teléfonos y cada teléfono pertenece a un único cliente



1 - N débil

Cada hospital tiene muchas salas y cada sala pertenece a un único hospital



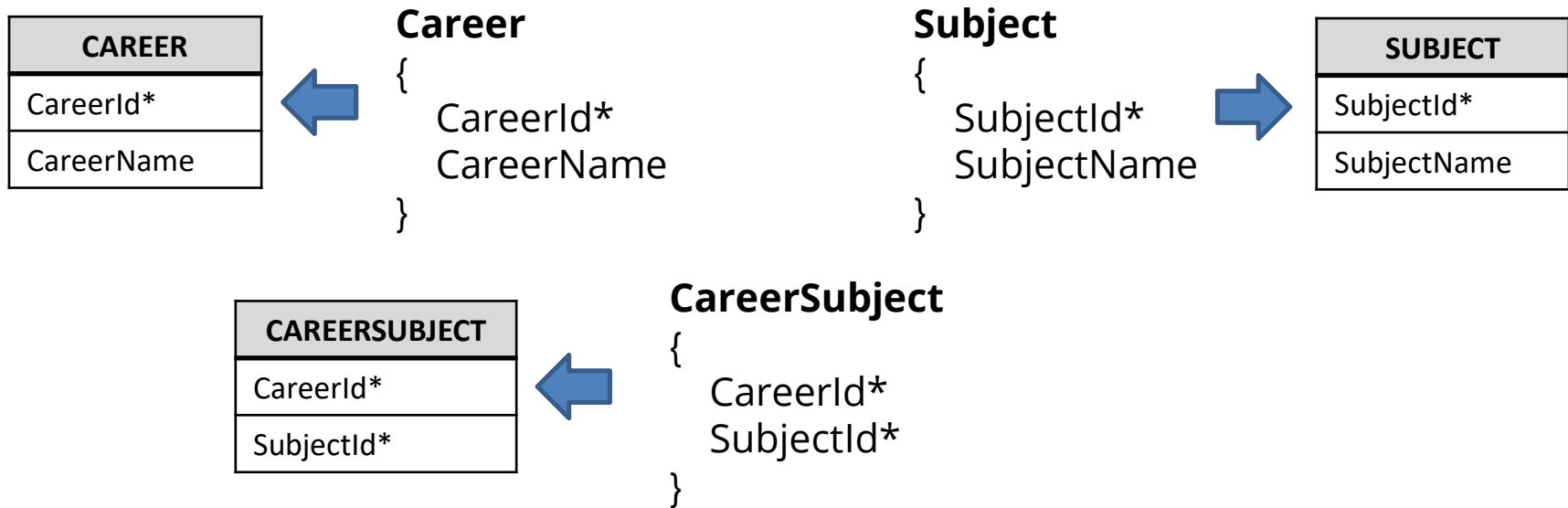
N - N (M)

Cada carrera tiene muchas materias y cada materia puede estar en muchas carreras



N - N: Opción 1 de 4

Cada carrera tiene muchas materias y cada materia puede estar en muchas carreras



N - N: Opción 1 - Tablas generadas

CAREER

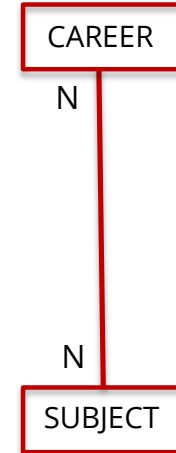
| CareerId | CareerName |
|----------|-------------------------|
| 1 | Computer Science |
| 2 | Data Science for Health |

SUBJECT

| SubjectId | SubjectName |
|-----------|--------------------------|
| 1 | Computer Logic |
| 2 | Programming Fundamentals |

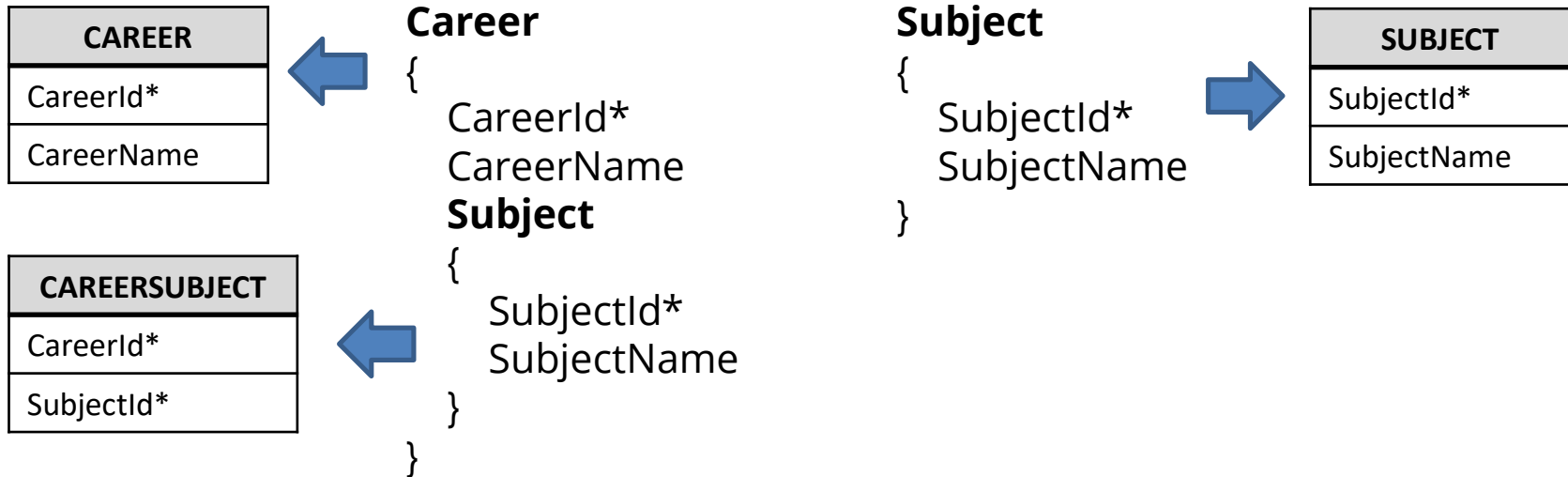
CAREERSUBJECT

| CareerId | SubjectId |
|----------|-----------|
| 1 | 1 |
| 1 | 2 |
| 2 | 2 |



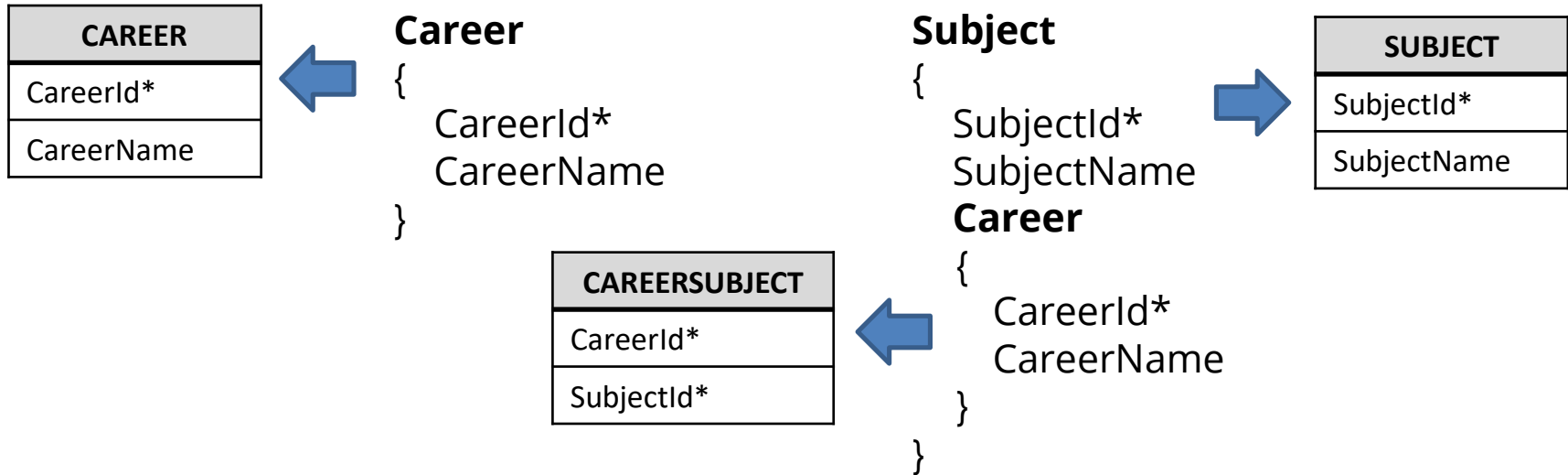
N - N: Opción 2 de 4

Cada carrera tiene muchas materias y cada materia puede estar en muchas carreras



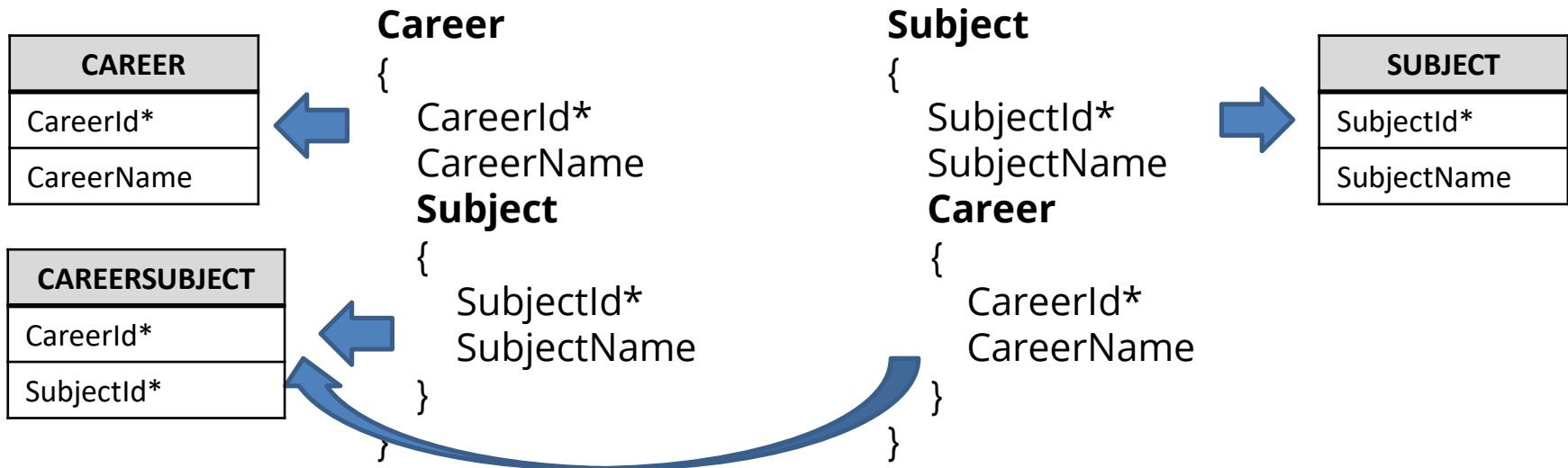
N - N: Opción 3 de 4

Cada carrera tiene muchas materias y cada materia puede estar en muchas carreras



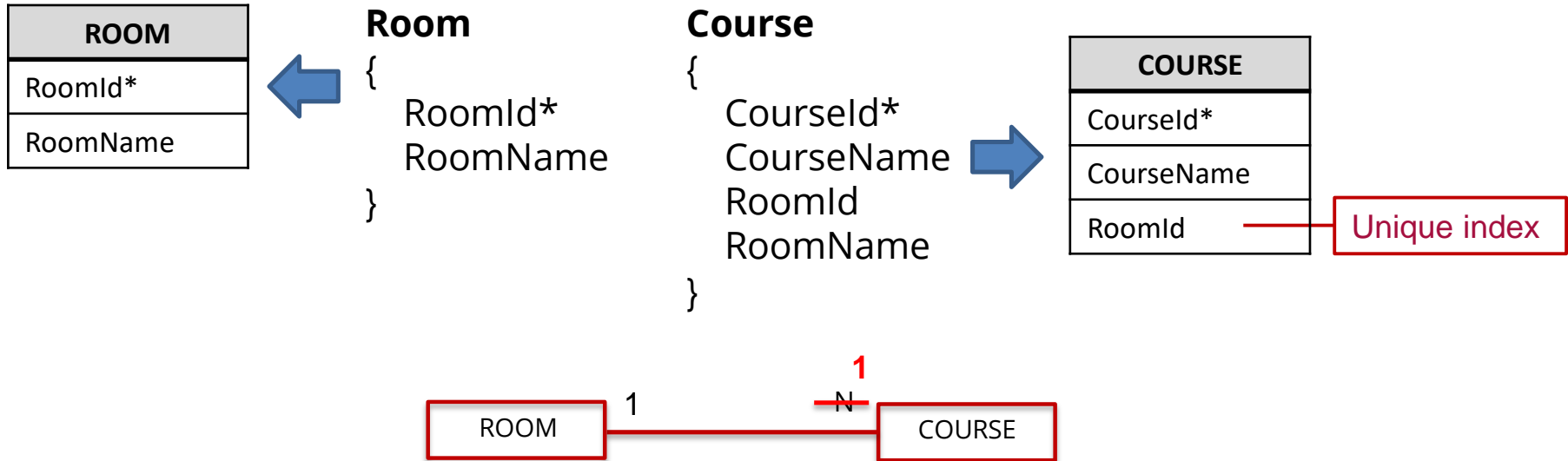
N - N: Opción 4 de 4

Cada carrera tiene muchas materias y cada materia puede estar en muchas carreras



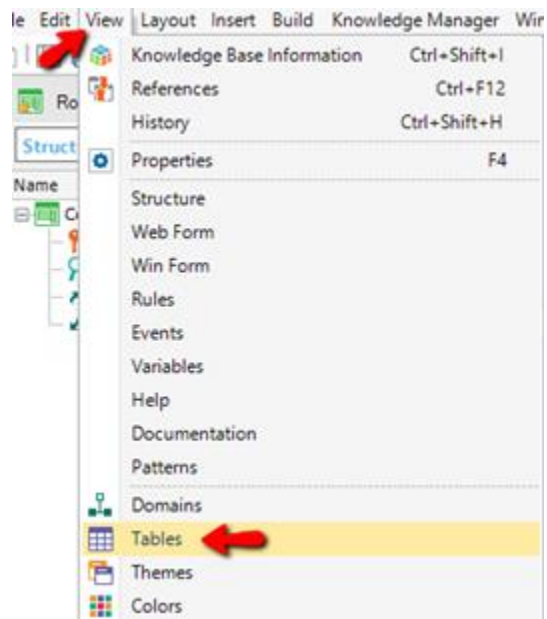
1 - 1

Cada curso se dicta en una sala y dicha sala solo podrá ser utilizada para el dictado de ese curso

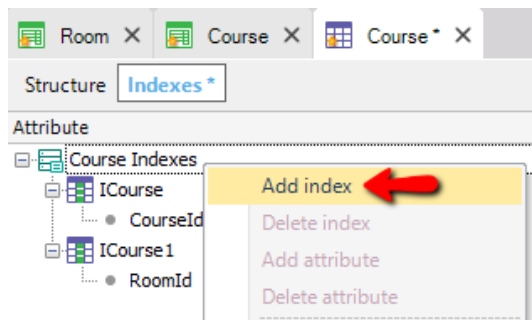


Creación de un índice

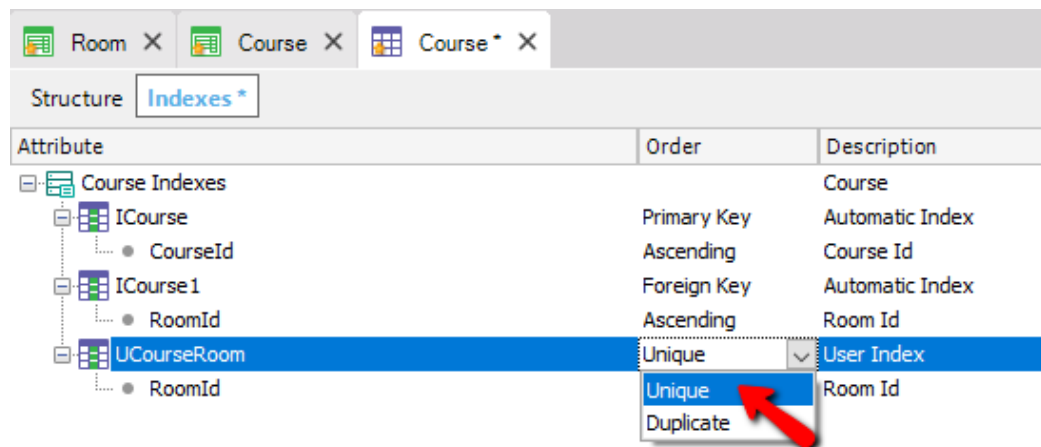
1



2



3



Normalización

GeneXus normaliza las tablas en Tercera Forma Normal (3NF)

- Los atributos inferidos en una transacción, no son incluidos en la tabla generada

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 |

Integridad referencial

Integridad referencial

Country

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

| CountryId | CountryName |
|-----------|-------------|
| 1 | URUGUAY |
| 2 | ARGENTINA |

Customer

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

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



No se inserta
el registro



Integridad referencial

Country

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

| CountryId | CountryName |
|-----------|---|
| 1 | URUGUAY  |
| 2 | ARGENTINA |

No se elimina
el registro



Customer

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

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

Tabla Base y Extendida

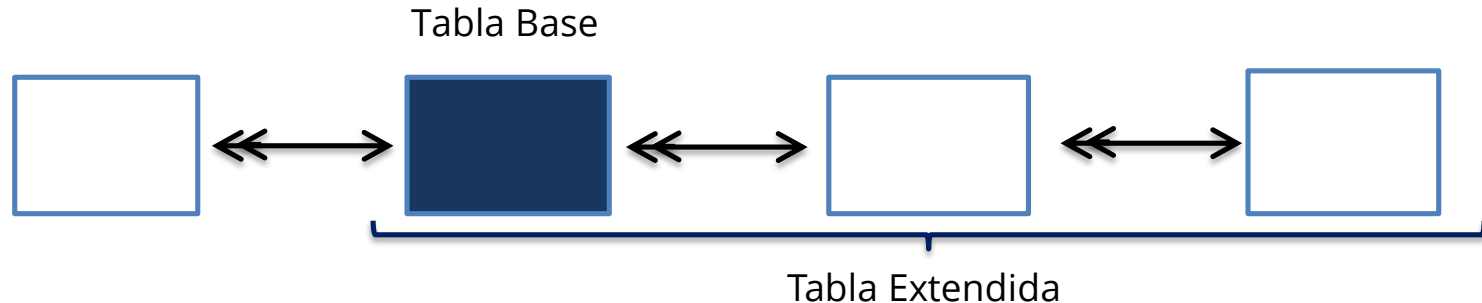
- **Tabla Base**

Es cualquier tabla de la base de datos en la cual estemos posicionados trabajando en determinado momento.

- **Tabla Extendida**

Dada una tabla, su tabla extendida es un concepto que nos permite pensar toda la información a la que tenemos acceso desde ella misma, a partir de sus claves foráneas.

Es el conjunto de atributos de la propia tabla + todos los atributos de las tablas con las que ella tenga una relación N a 1 directa o indirectamente.



Ejemplo

Customer

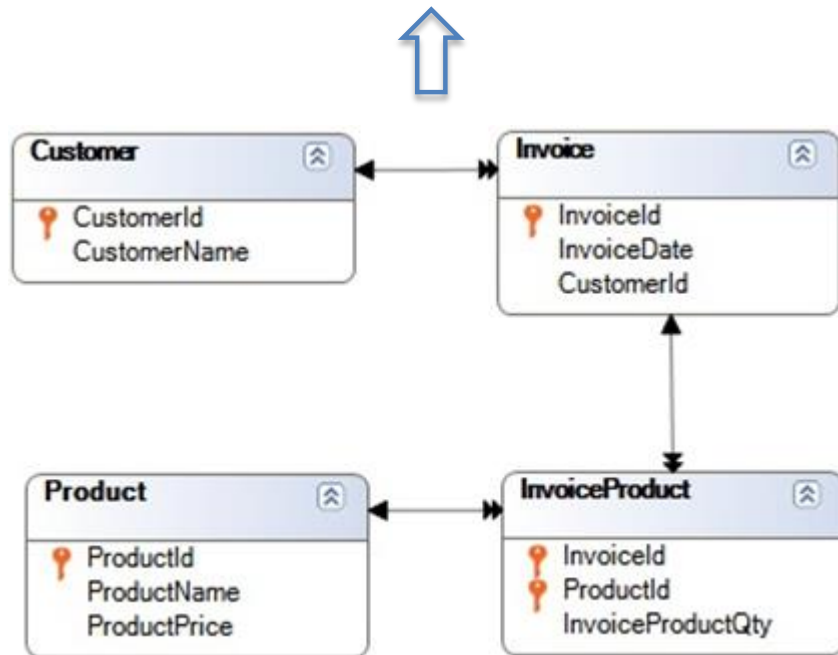
```
{
  CustomerId*
  CustomerName
}
```

Product

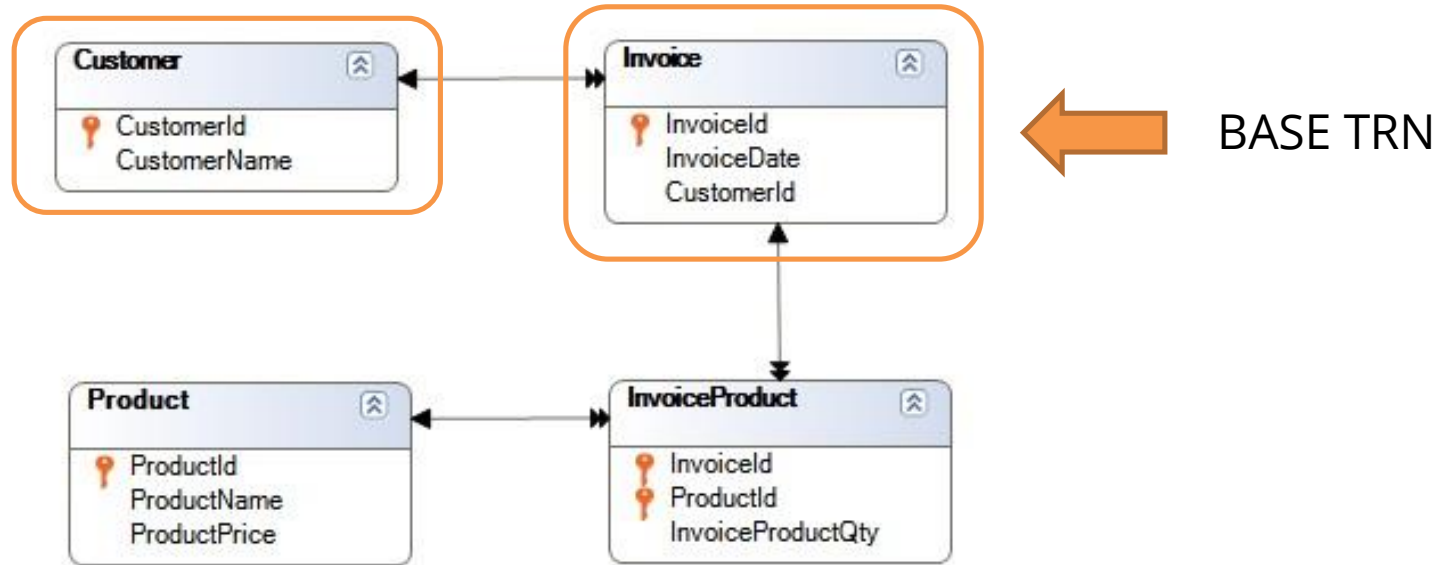
```
{
  ProductId*
  ProductName
  ProductPrice
}
```

Invoice

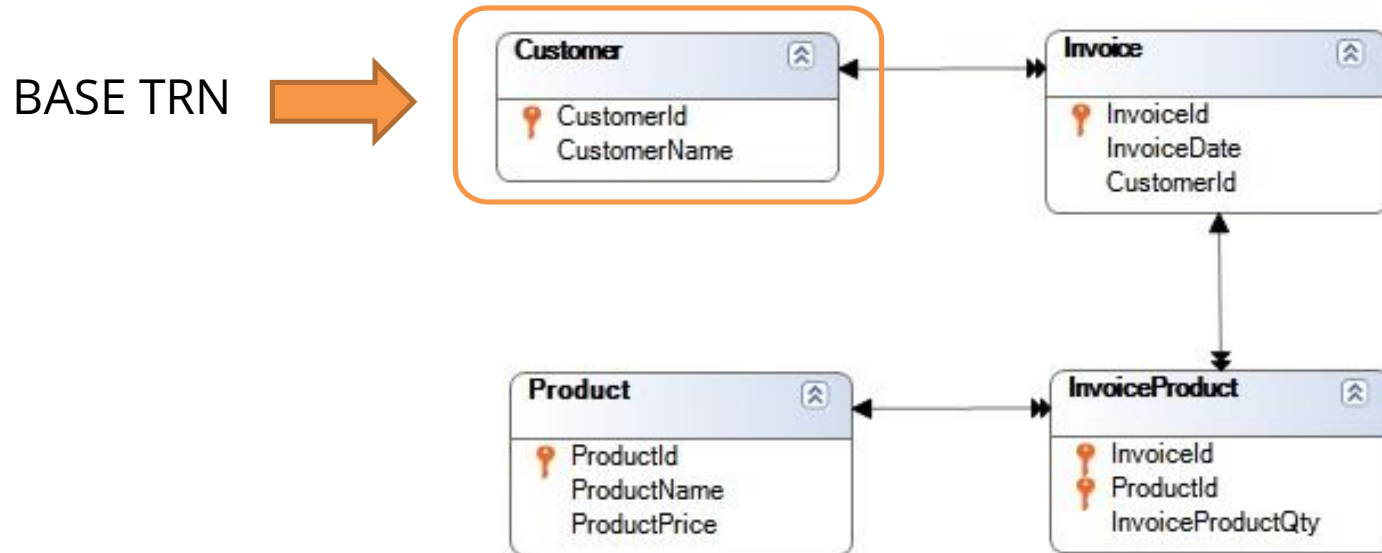
```
{
  Invoiceld*
  InvoiceDate
  CustomerId
  CustomerName
Product
  {
    ProductId*
    ProductName
    ProductPrice
    InvoiceProductQty
    -----
  }
}
```

Diagrama de Tablas
(diagrama de Bachman)

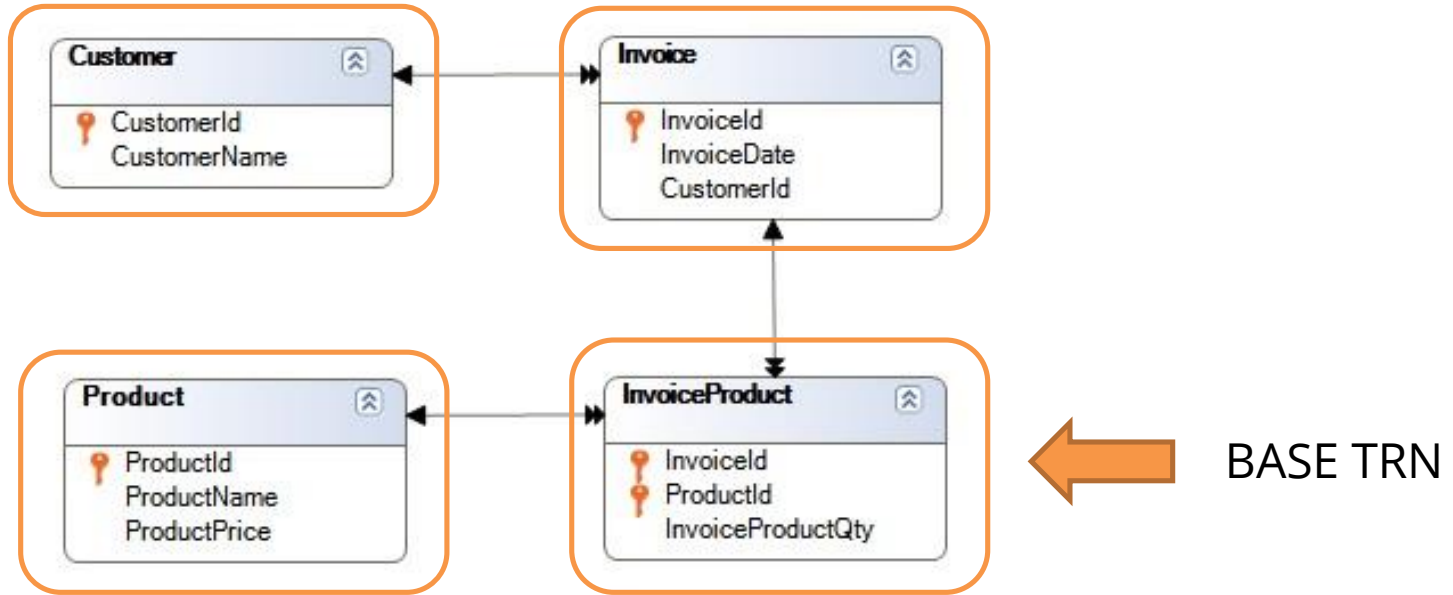
Ejemplo: Tabla Extendida de Invoice



Ejemplo: Tabla Extendida de Customer

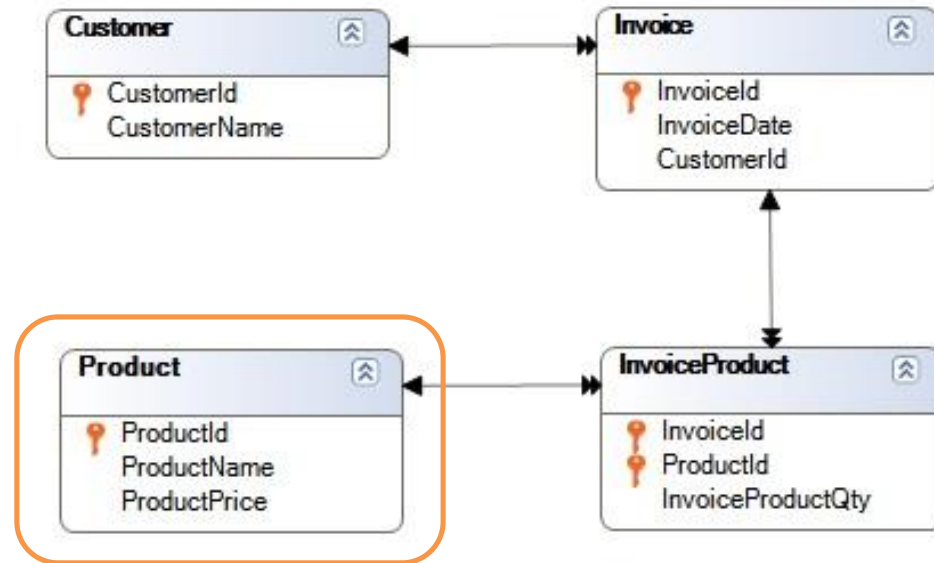


Ejemplo: Tabla Extendida de InvoiceProduct



Ejemplo: Tabla Extendida de Product

BASE TRN



Subtipos

Referencias múltiples:

Se requiere guardar para cada vuelo el aeropuerto de partida y el aeropuerto de llegada

Airport

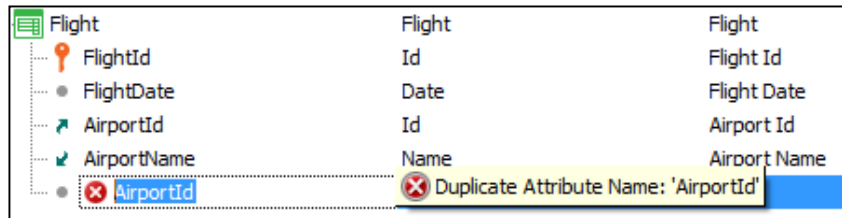
```
{  
  AirportId*  
  AirportName  
}
```

Flight

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



Error por nombre de atributos duplicado



Solución 1 de 3: crear dos grupos de subtipos, uno para el aeropuerto de partida y otro para el aeropuerto de llegada

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 |

Solución 2 de 3: crear un grupo de subtipos, solamente para el aeropuerto de partida




Airport

```
{
  AirportId*
  AirportName
}
```

Flight

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

} Subtype group: DepartureAirport

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




Solución 3 de 3: crear un grupo de subtipos, solamente para el aeropuerto de llegada

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 |

Referencias múltiples:

Se requiere guardar además del país del cliente, el país donde fue emitida la factura

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"> InvoiceCountryId | Invoice Country Id | CountryId |
| <ul style="list-style-type: none"> InvoiceCountryName | Invoice Country Name | CountryName |

Reglas

Reglas

| 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 |

Reglas

| 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,);



Variable: Espacio en memoria que posee un nombre y tipo de dato que puede guardar. Se referencian utilizando “&”.

Momentos de disparo

Browser

Transaction

http://trialapps3.gonexus.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 ✓

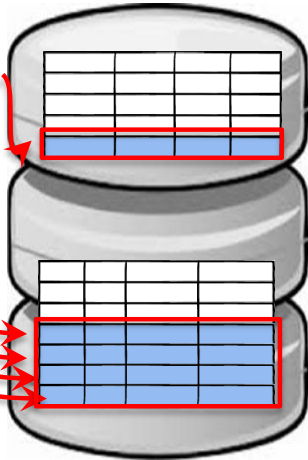
...

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 | ... | ☑ |

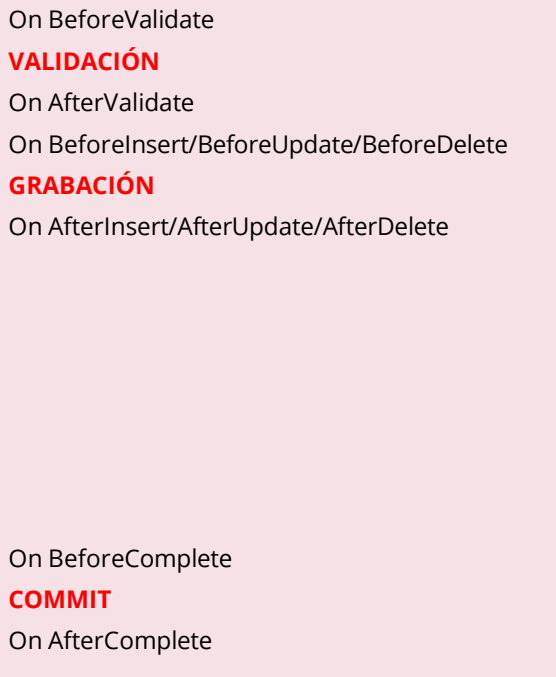
Database



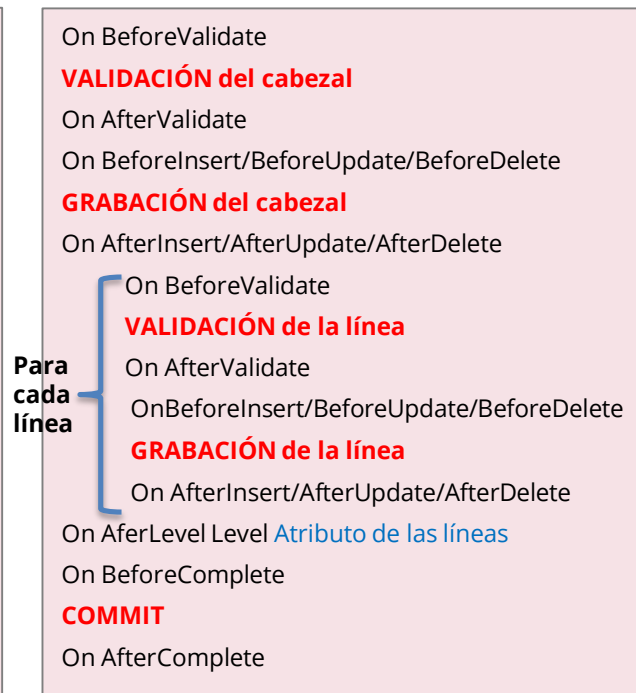
Commit

Momentos de disparo de reglas

En transacciones de un sólo nivel:

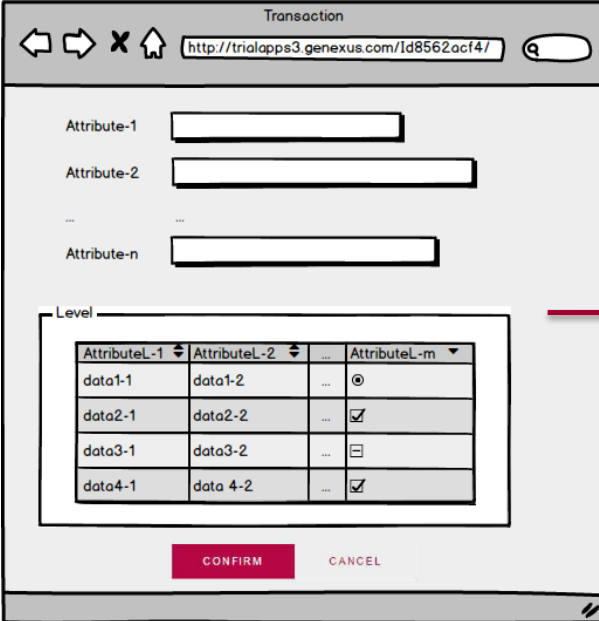


En transacciones de dos niveles:



Momentos de disparo de reglas

PrintCustomer(CustomerId) on AfterInsert, AfterUpdate; ¿Es correcto o no?  ¡Es correcto!



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

VALIDACIÓN

On Aftervalidate

GRABACIÓN

On AfterInsert / On AfterUpdate / On AfterDelete

Momentos de disparo de reglas

PrintCustomer(CustomerId) on AfterDelete;

¿Es correcto o no?



No es correcto porque se invoca DESPUÉS de la eliminación y no se encontrará al cliente con ese Id en la tabla

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

VALIDACIÓN

On Aftervalidate

GRABACIÓN

On AfterInsert / On AfterUpdate / On AfterDelete

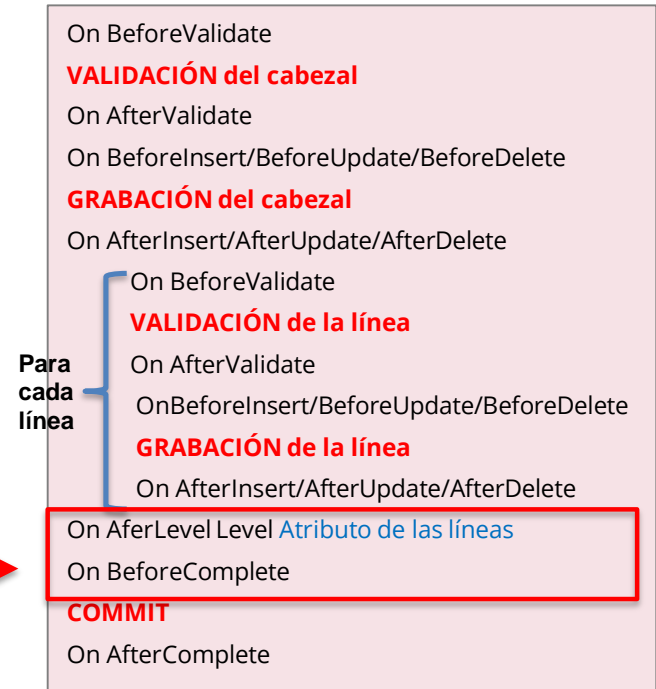
Momentos de disparo de reglas

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

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

< 8

En transacciones de dos niveles:

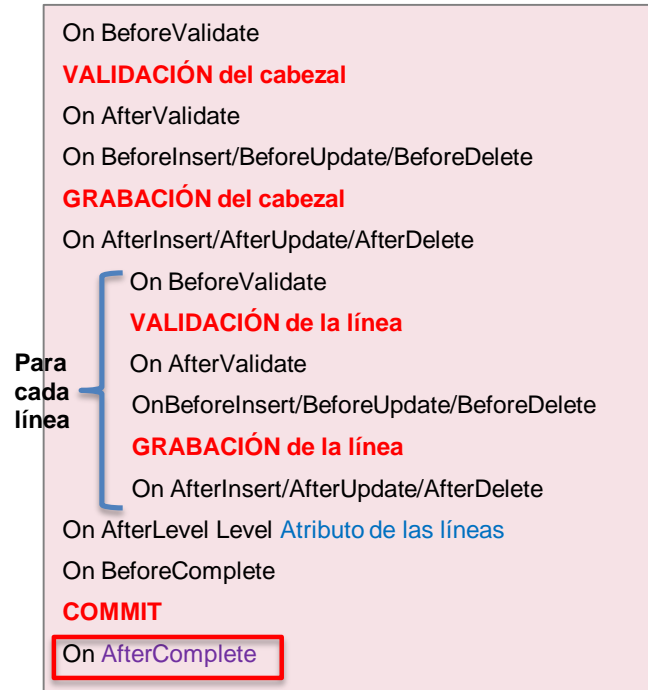


Momentos de disparo de reglas

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

✓on **AfterComplete**:
Inmediatamente después
de que se realice el
Commit en la base de
datos

En transacciones de dos niveles:



Ejemplos

Invoice

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

Determine si es correcto o no:

PrintInvoiceDetail(Invoiceld) on **AfterComplete**;



ProductControl(ProductId) on **BeforeInsert**;

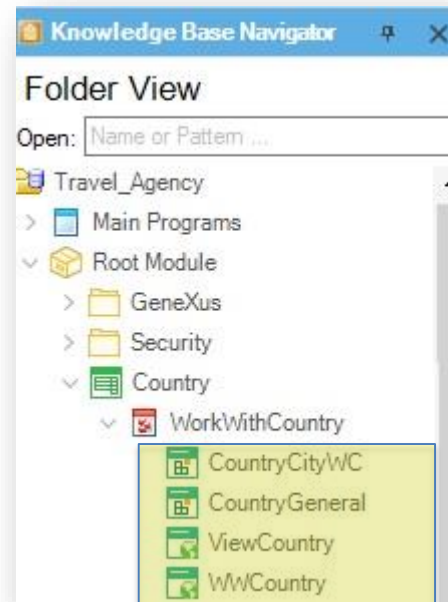
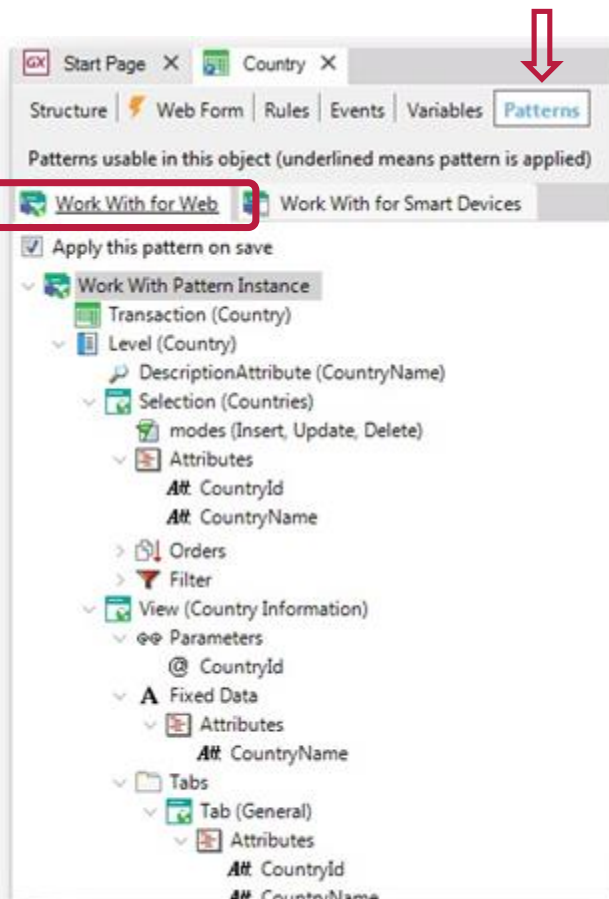


ProductControl(ProductId) on **AfterComplete**;



¿Puedo asignar un valor a un atributo on
AfterInsert?  **NO**

Patterns



Generados automáticamente por GeneXus

Dinamismo



Example: New attribute CountryFlag

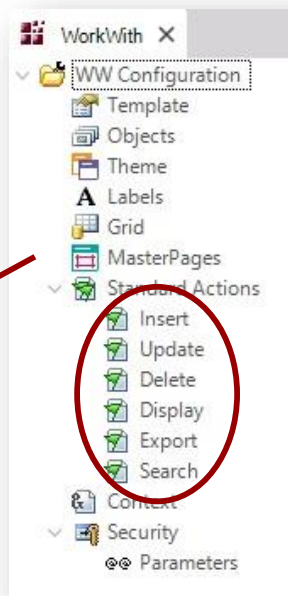
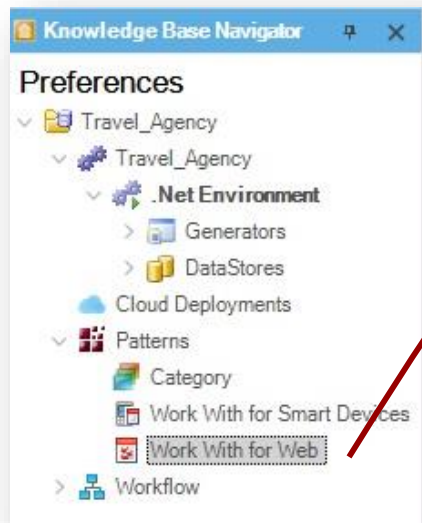
The image illustrates the dynamic generation process in GeneXus. It shows three main components:

- 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' 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 'CountryFlag' attribute in the configuration window. A dashed arrow points from the 'CountryFlag' attribute in the configuration window to the 'Flag' column in the web application table.

Configuraciones Generales

¿Cómo se inicializan todas las instancias?



En Country...

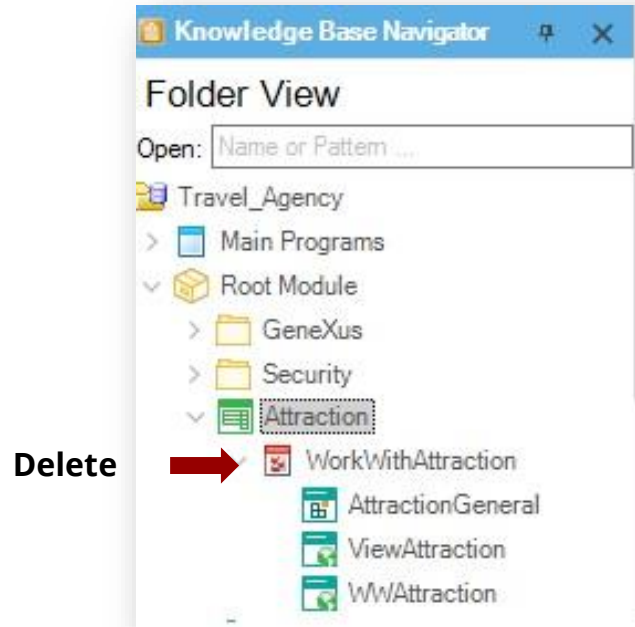
Properties

Filter

modes: Ins: default, Upd: default, ...

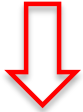
| | |
|---------|---------|
| Insert | default |
| Update | default |
| Delete | default |
| Display | default |
| Export | default |

Eliminación de la aplicación del pattern



Fórmulas

Fórmulas Globales



- Es un cálculo que definimos asociado a un atributo en una transacción
- Se conocen en toda la KB

```

Product
{
  ProductId*
  ProductName
  ProductPrice
}
  
```

```

Invoice
{
  Invoiceld*
  InvoiceDate
  InvoiceAmount → Sum(InvoiceProductAmount)
  Product
  {
    ProductId*
    ProductName
    ProductPrice
    InvoiceProductQuantity
    InvoiceProductAmount
  }
}
  
```

Atributos "virtuales" (no guardados en la BD)

```

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 los clientes por país o la cantidad total de clientes?

Contará los clientes por cada país ya que GeneXus realiza un filtro automático por el atributo en común (*CountryId*).

Customer

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

Invoice

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

Dominio InvType
que ofrece
dos Enum Values


Credit
Cash

Sum(InvoiceAmount, InvoiceType = InvType.Credit)

If CustomerId = 3

condición de
cálculo

condición de
disparo

- Fórmulas Inline**  - Son fórmulas definidas en el código de un objeto
- Se conocen solamente en el objeto en que se han definido

Requerimiento: Nos solicitan un listado de países con la cantidad de atracciones de cada uno

| Country | Quantity |
|----------------|----------|
| Argentina | 2 |
| Uruguay | 3 |
| Paraguay | 1 |
| Estados Unidos | 5 |



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

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

Formula inline en el código de un objeto Procedimiento

```
Print Header
```

```
For each Country
```

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

```
    Print Countries
```

```
Endfor
```

Tabla base del For Each: COUNTRY

Count(AttractionName)

Tabla navegada por la fórmula: ATTRACTION

¿Esta fórmula contará las atracciones por país o la cantidad total de atracciones?

Contará las atracciones por cada país ya que se realiza un filtro automático por el atributo en común *CountryId* (ambas tablas están relacionadas).

For each

Transacción Base

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

For each **Flight**

Endfor

For each **Flight.Seat**

Endfor

Transacción Base

Nombre de la transacción
cuya tabla física asociada
se desea recorrer

Orden


```
Customer  
{  
  CustomerId*  
  CustomerName  
  -----  
}
```

Requerimiento: nos solicitan un listado de todos los clientes ordenados alfabéticamente por su nombre.

```
For each Customer order CustomerName  
  -----  
Endfor
```

Requerimiento: nos solicitan un listado de todos los clientes ordenados de forma descendente por su nombre.

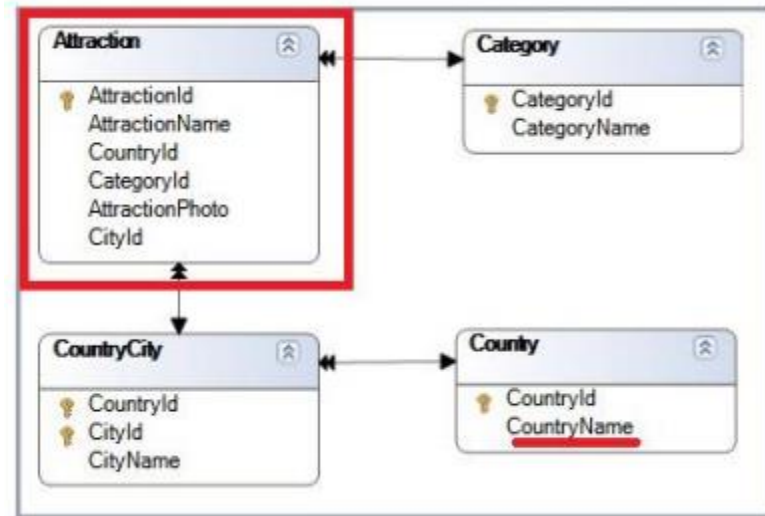
```
For each Customer order (CustomerName)  
  -----  
Endfor
```



Orden

GeneXus permite ordenar por el valor de un atributo que no está en la tabla que estamos recorriendo, sino en su tabla extendida.

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



Filtros con Where

Flight

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

For each Flight

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

Endfor

For each Flight.Seat

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

Endfor

Filtros + Orden

Customer

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

Index



For each Customer order CustomerName
Where CustomerName >= &NameFrom

Endfor

Warnings

 [spsc0038](#) 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 Anidados + Tabla base diferente + Tablas NO relacionadas = PRODUCTO CARTESIANO

Country

```
{  
  CountryId*  
  CountryName  
}
```

Room

```
{  
  RoomId*  
  RoomName  
}
```

Producto 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 Anidados + Tabla base diferente + Tablas 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 Anidados + Misma Tabla base + Tablas relacionadas = CORTE DE CONTROL

Country

```
{  
  CountryId*  
  CountryName  
}
```

Customer

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

Corte de control

For each Customer order CountryId

Print Country

For each Customer

Print Customer

Endfor

Endfor

1 - Brasil

LUIS
JORGE

4 - Estados Unidos

ANA

Resumen

Producto
Cartesiano

For each Country
Print Country

For each Room
Print Room
Endfor

Endfor

Tablas
diferentes, sin
relación

Join

For each Country
Print Country

For each Customer
Print Customer
Endfor

Endfor

Tablas
diferentes, con
relación

Corte de
control

For each Customer order CountryId
Print Country

For each Customer
Print Customer
Endfor

Endfor

Misma tabla,
Agrupando con order

Resumen

Producto Cartesiano

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

“No existe relación entre ambas entidades, mostrar todas las posibilidades para cada país”

Join

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

“Todos los países y sus clientes, independientemente que tenga clientes o no”

Corte de control

| | |
|--------------------|-------|
| 1 - Brasil | LUIS |
| | JORGE |
| 4 - Estados Unidos | ANA |

“Solo aquellos países que tengan clientes”

Comunicación entre objetos

Ejemplo: pasaje de parámetros

Country

```
{  
  CountryId*  
  CountryName  
}
```

Customer

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

Por ejemplo, en las Reglas de la trn Country:

CustomerList(CountryId) on AfterComplete;



CustomerList

Parm(in: &CountryId);

For each Customer

Where CountryId = &CountryId

Endfor



Variable



Filtro explícito

Ejemplo: pasaje de parámetros

Country

```
{
  CountryId*
  CountryName
}
```

Customer

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

Por ejemplo, en las Reglas de la trn Country:

CustomerList(CountryId) on AfterComplete;



CustomerList

Parm(in: CountryId);

For each Customer

~~Where CountryId = &CountryId~~

Endfor

Atributo

Filtro implícito



Ejemplo: retorno de un valor

Country

```
{  
  CountryId*  
  CountryName  
}
```

Customer

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

Por ejemplo, en las Reglas de la trn Customer:

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



CustomerControl

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

```
For each Customer  
  Where CustomerId = &CustomerId
```

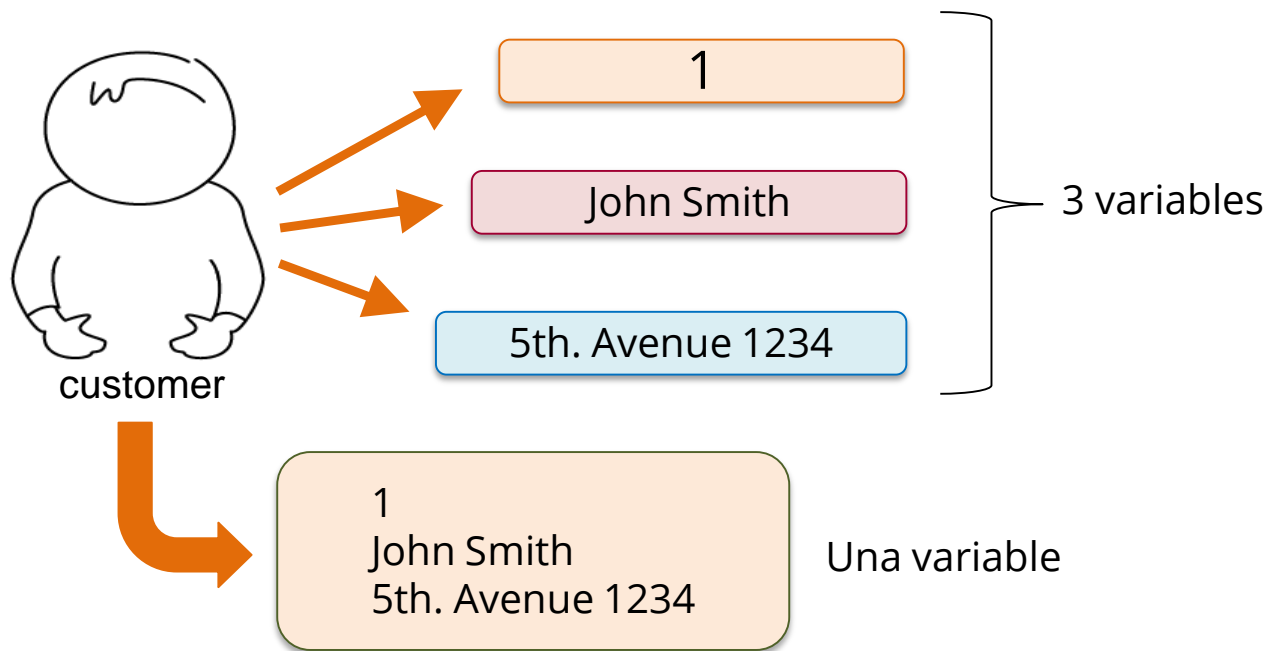
```
-----
```

```
  &Control = True
```

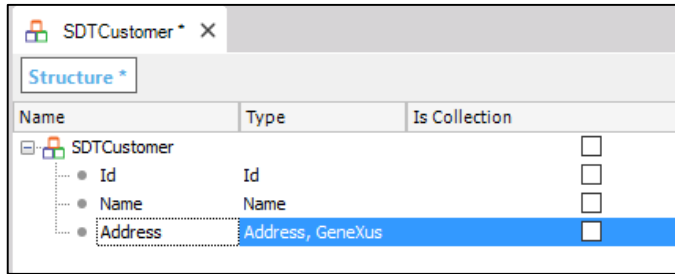
```
Endfor
```

Tipos de Datos Compuestos

Concepto



Definición



| 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



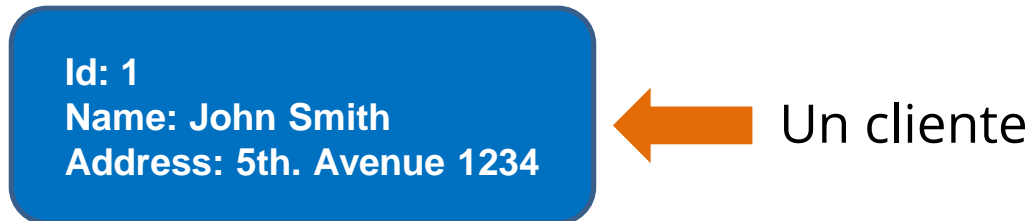
customer



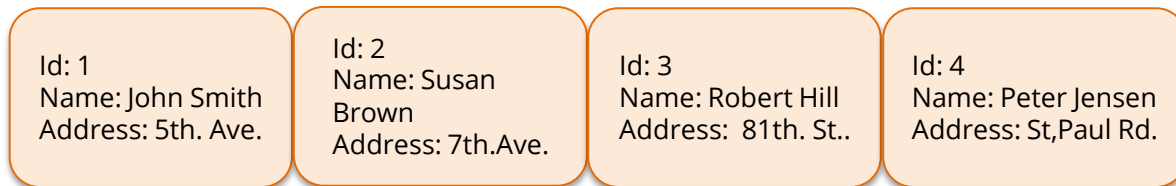
&OneCustomer: SDTCustomer

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

Data Providers



SDTCustomer



↑
Colección de Clientes

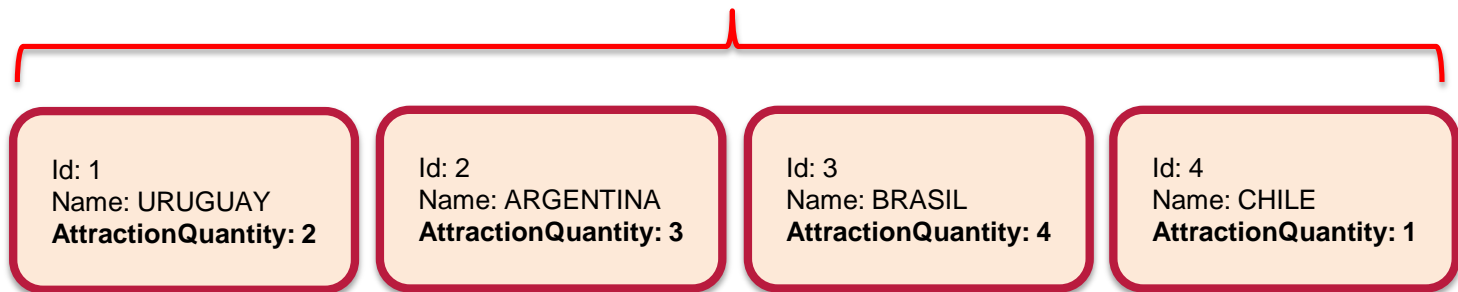
Ejemplo: Ranking de cantidad de atracciones por país

| Pais atracciones | Cantidad |
|-----------------------------|-----------------|
| BRASIL | 4 |
| ARGENTINA | 3 |
| URUGUAY | 2 |
| CHILE | 1 |
| | |
| ... | |

Ejemplo: Ranking de cantidad de atracciones por país

SDTCountries

Una colección de países



Ejemplo: Ranking de cantidad de atracciones 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"/> |

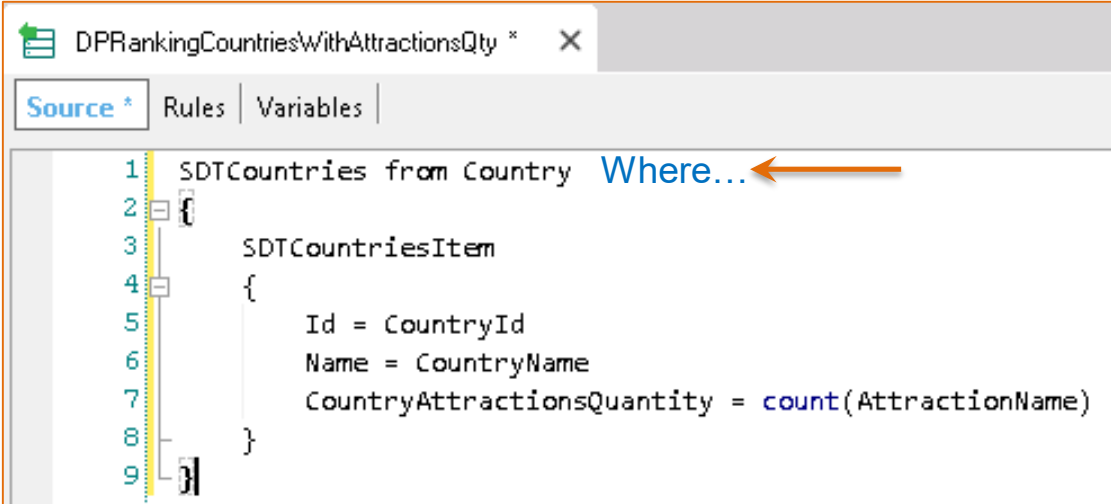
Attraction

```
{
  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 |

Ejemplo: Ranking de cantidad de atracciones 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.

Ejemplo: Ranking de atracciones por país

Country

```
{
  CountryId*
  CountryName
}
```

Attraction

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

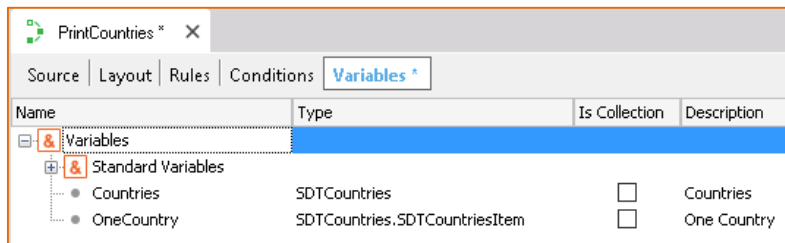
País

Cant. Atracc.

| | |
|-----------|---|
| BRASIL | 4 |
| ARGENTINA | 3 |
| URUGUAY | 2 |
| CHILE | 1 |



```
PrintCountries * X
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
```



| 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 |

Variables Colección

&numbers

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

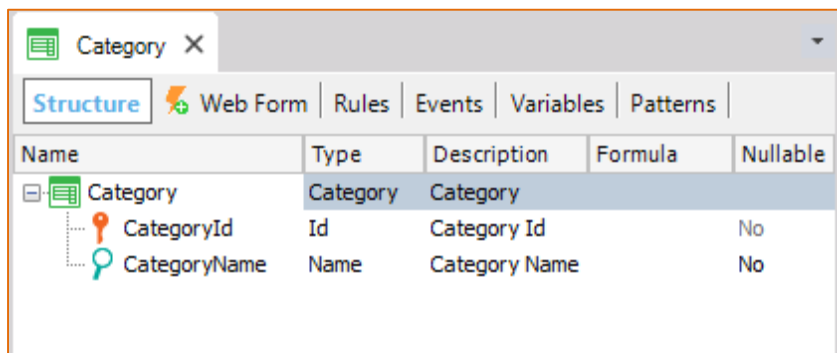
| Name | Type | Description | Is Collection |
|------------|---------------|-------------|--------------------------|
| SDTCountry | | SDTCountry | <input type="checkbox"/> |
| • Id | Numeric(4.0) | Id | <input type="checkbox"/> |
| • Name | Character(20) | Name | <input type="checkbox"/> |

&Countries

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

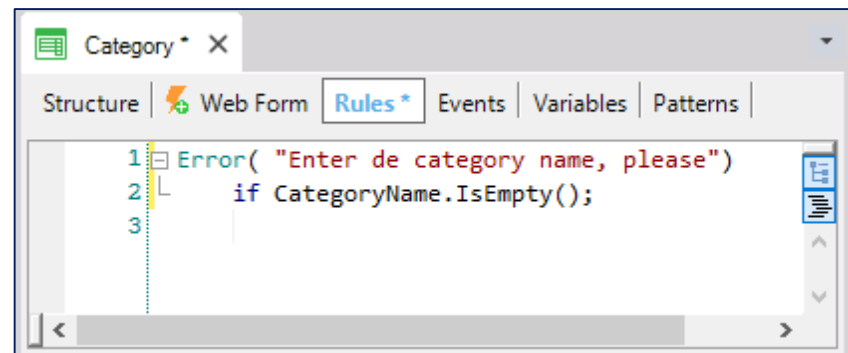
Business Components

Concepto: tipo de datos especial basado en una transacción



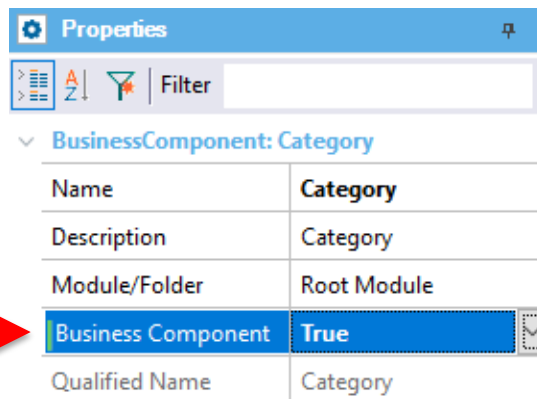
Structure | Web Form | Rules | Events | Variables | Patterns

| 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 |

Concepto: tipo de datos especial basado en una transacción

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

| Name | Type |
|------------------------|----------|
| & Variables | |
| + & Standard Variables | |
| • Category | Category |

Variable: &Category

| | |
|--------------|-----------------|
| Name | Category |
| Description | Category |
| Column title | Category |
| Class | Attribute |

[Help](#)

Type Definition

| | |
|---------------------|-----------------|
| Based on | (none) |
| Data Type | Category |
| Collection | |
| Initial value | |
| Validation | |
| Value range | |
| Validation Failed M | |
| Control Info | |
| Control Type | |
| Input Type | |

- Image
- LongVarChar
- Numeric
- VarChar
- Video
- Extended Types
- Structured Data Types
- Business Components**
- Airline
- Attraction
- Category**
- Country
- Country.City
- External Objects

Properties

Filter

BusinessComponent: Category

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

Ejemplos: inserción y modificación

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
```

Ejemplo: eliminación

Category

```
{  
  CategoryId*  
  CategoryName  
}
```

Delete



The screenshot shows a code editor window with a tab labeled 'Source *'. Below the tab are several menu items: 'Layout', 'Rules', 'Conditions', and 'Variables *'. A dropdown menu labeled 'Subroutines' is open, showing a list of subroutines. The first three subroutines are highlighted in yellow:

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


Métodos Insert y 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

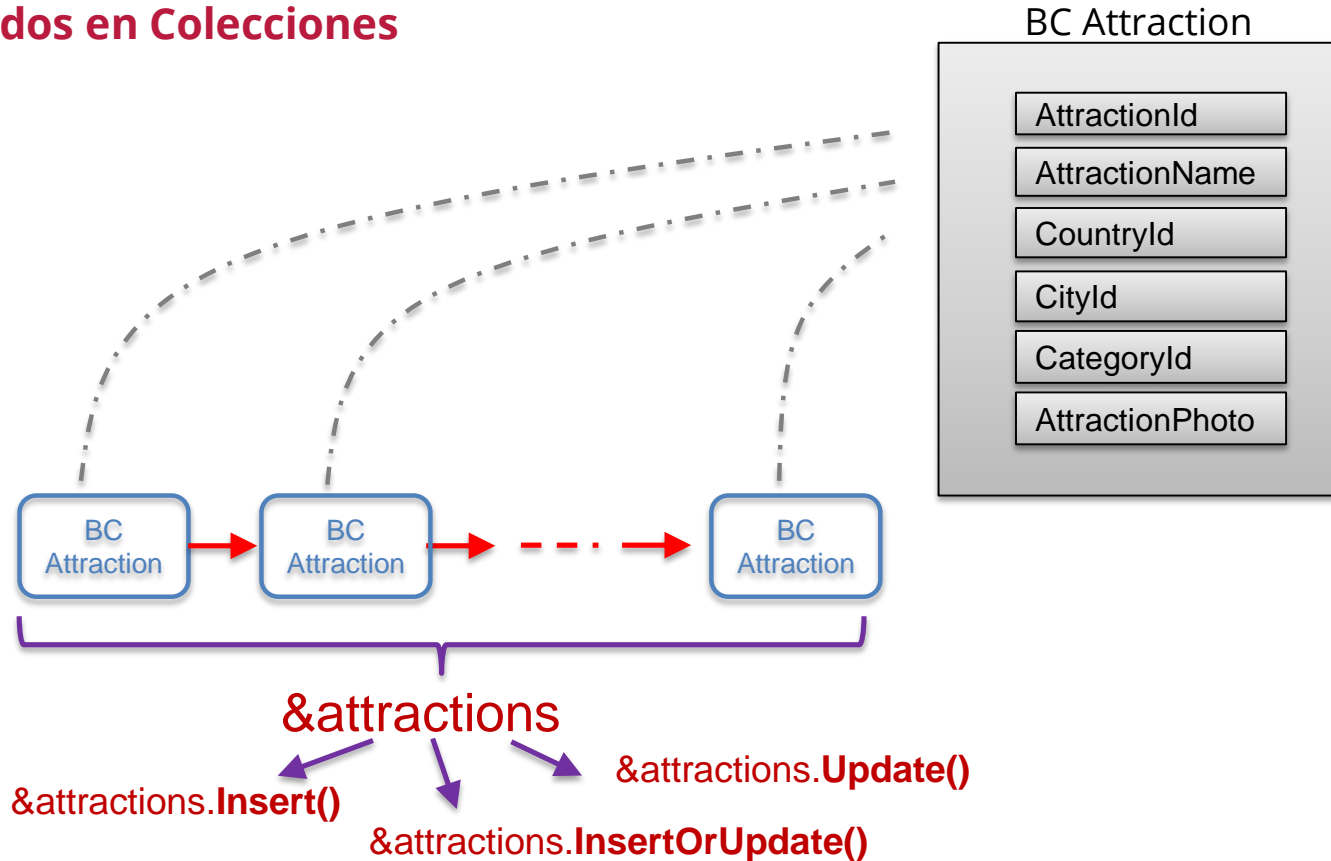
```
{  
  CategoryId*  
  CategoryName  
}
```

InsertOrUpdate



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

Métodos en Colecciones



Insert / Update / InsertOrUpdate

Aconsejamos la utilización de los métodos ***Insert***, ***Update*** e ***InsertOrUpdate*** ya que:

- Usando los métodos *Load* y *Save* para modificar se accede dos veces a la base de datos -lo que disminuye la performance- mientras que con el *Update* o *InsertOrUpdate* se accede solamente una vez
- Los nombres de estos nuevos métodos ya dan la pauta del objetivo claramente

Manejo de errores trabajando con BC

Por cada variable Business Component se carga en memoria una colección con todos los mensajes de advertencia o error que se produzcan como consecuencia de las operaciones.

```

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"/> |

Poblado de Datos

Transacción

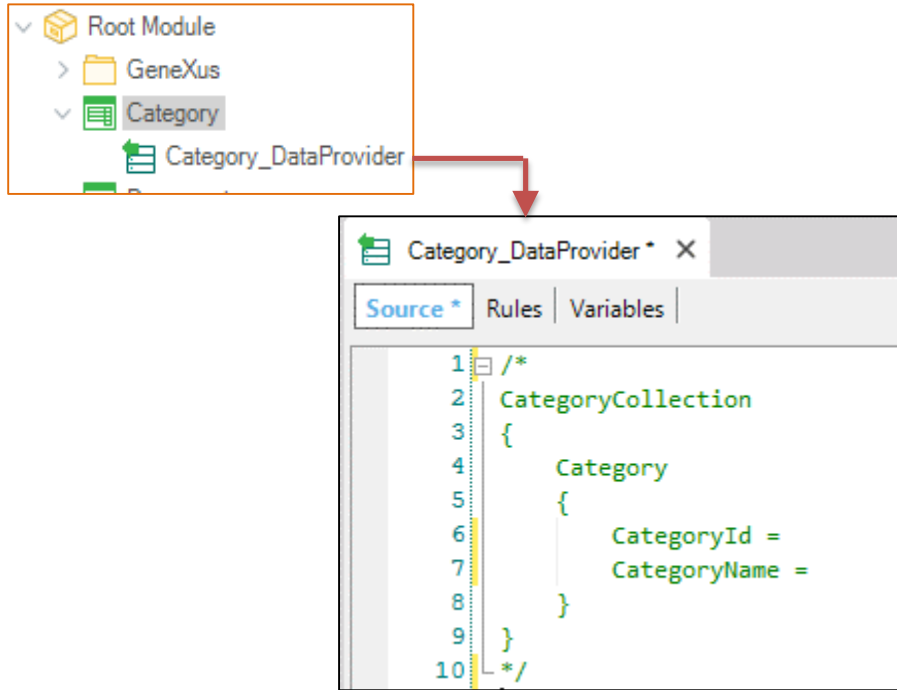
Inicializar datos de forma automática

GeneXus 15 ofrece una alternativa muy fácil para definir los datos con los cuales se desean poblar las tablas físicas que se crean asociadas a las transacciones, lo que evita tener que resolver las cargas de datos de otras maneras.

| 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 |

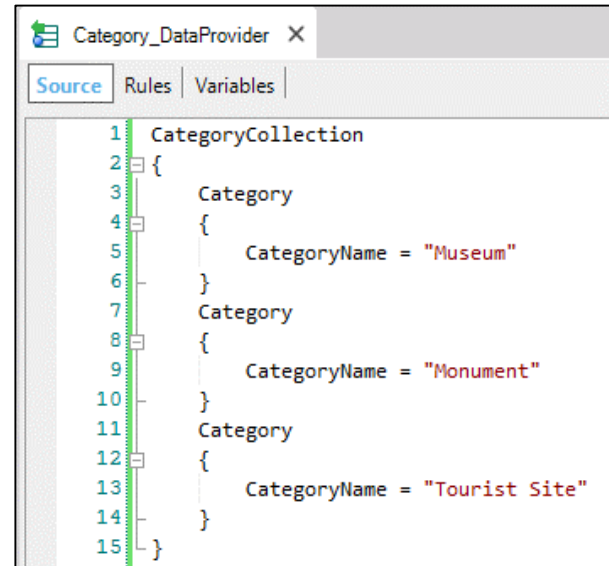
Inicializar datos



The screenshot shows the GeneXus project explorer on the left, with a red box highlighting the 'Category' folder and its 'Category_DataProvider' sub-item. A red arrow points from this sub-item to the 'Category_DataProvider' editor window on the right. The editor window shows the 'Source' tab with the following code:

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

No cargamos el CategoryId
ya que fue definido como autonumerado



The screenshot shows the 'Category_DataProvider' editor window with the 'Source' tab selected. The code defines a 'CategoryCollection' containing three 'Category' objects, each with a 'CategoryName' property. A red arrow points from the text above to the 'Category' objects in the code.

























```
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 datos: Solo lectura

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 |

Poblado de Datos

Business Components y Data Providers

Ejemplo

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 | Country |
| Collection | True |
| Collection Name | Countries |

Web Form | Rules | Events | Conditions | Variables

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

Poblado de Datos con Procedimientos Comandos New / For each / Delete

Inserción – Comando NEW

Category

```
{  
  CategoryId*  
  CategoryName  
}
```

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

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



Si el atributo es autonumerado
no lo inserto

Modificación / 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
}
```

Eliminación – Comando DELETE

Attraction

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

```
For each Attraction  
  Delete  
Endfor
```

Consideraciones

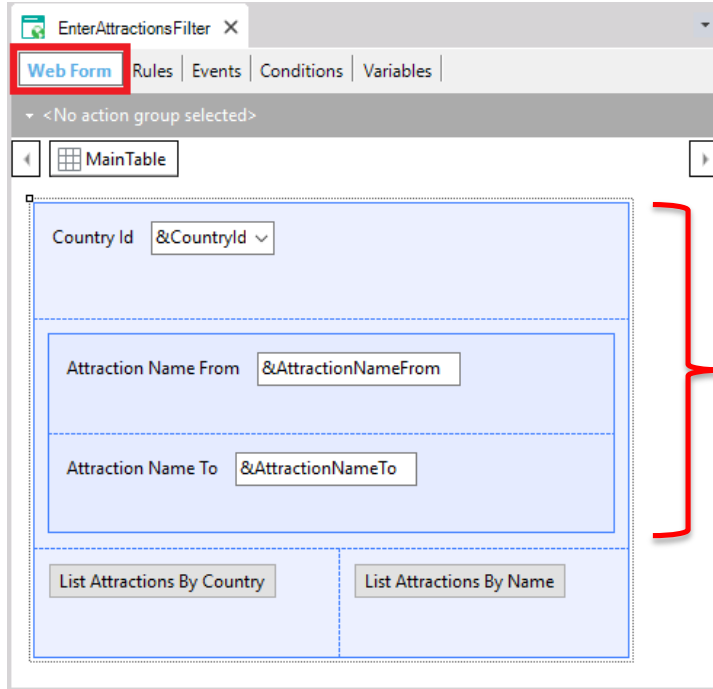
Los comandos New / For Each / Delete sirven para insertar, actualizar y eliminar información de la base de datos, pero solo pueden ser usados en Procedimientos.

Si bien los siguientes comandos sirven para insertar, actualizar y eliminar información de la base de datos, recomendamos la utilización de Business Component, ya que los mismos:

- Controlan la integridad referencial
- Disparan las reglas definidas en la transacción

Web Panels

Web Panel sin grid, con variables en el 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

Variables: **de entrada**
(no readonly)

Web Panel sin grid pero con atributos en el form

Parm(in: **AttractionId**);

ViewAttractionFromScratch X

Web Form | Rules | Events | Conditions | Variables

<No action group selected>

Id

Name

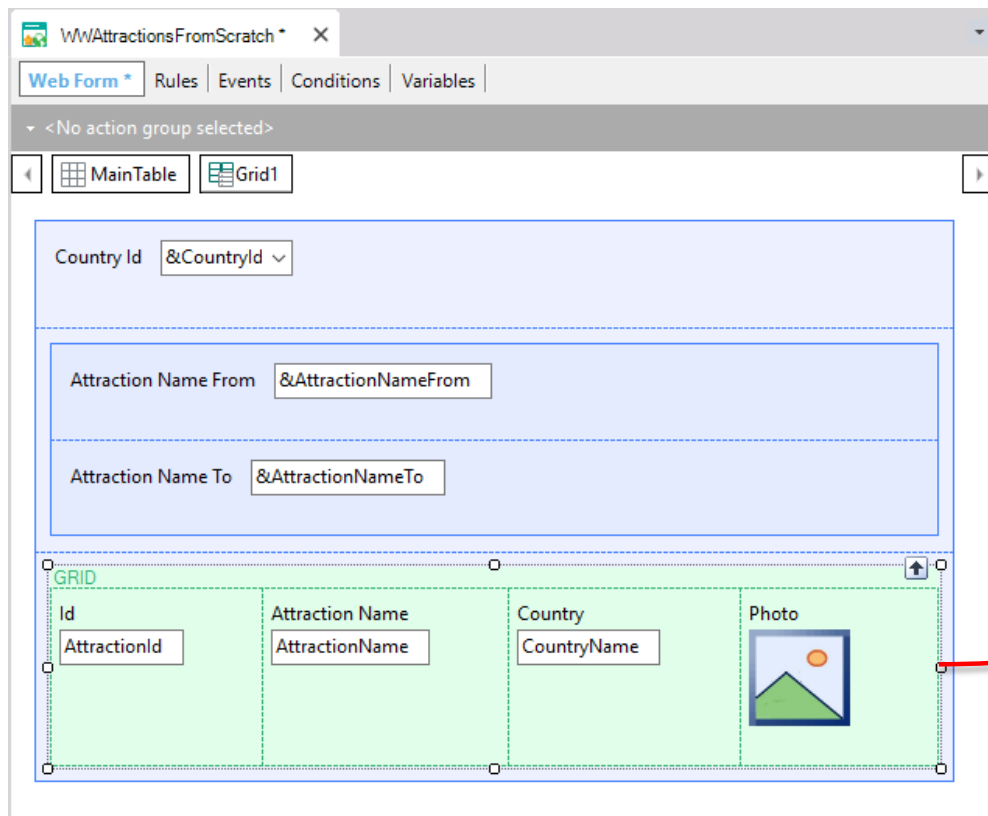
Country Name

Category Name

City Name

Se carga solo **un** registro

Grid: CON TABLA BASE



WWAttractionsFromScratch * x

Web Form * Rules Events Conditions Variables

<No action group selected>


MainTable Grid1

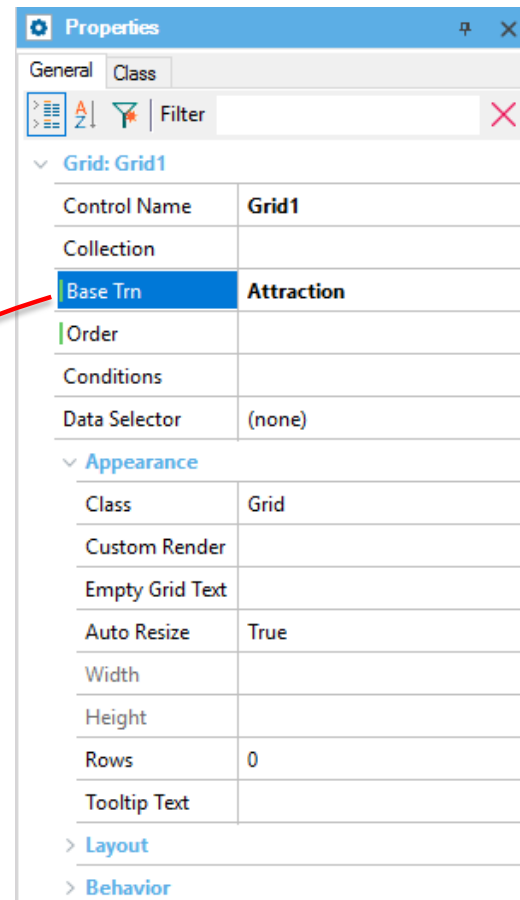
Country Id &CountryId v

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

Transacción 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 = &AttractionNameFrom
5 where AttractionName = &AttractionNameTo
6 print Attractions
7 endfor
```

A red box highlights lines 3 through 6, and a red arrow points from this box to the grid control in the web form.
- Properties Window (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. Other properties like 'Control Name', 'Order', 'Conditions', 'Data Selector', 'Appearance', and 'Behavior' are also visible.

The overall layout is titled 'AttractionsByName' and includes a 'Navigation View' and 'Web Form' tabs. The 'Properties' window is open on the 'Class' tab.

Orden

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

- Web Form:** Shows a form with three input fields: "Country Id" (dropdown), "Attraction Name From", and "Attraction Name To". Below them is a 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 dialog box "Grid1's Order" is open, showing "CountryName" as the selected value. A red arrow points from the "Order" property field in the Properties window to the dialog box.

The "Grid1's Order" dialog box has a text input field containing "CountryName" and "OK" and "Cancel" buttons.

Grid: CON TABLA 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


Condiciones de filtro

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

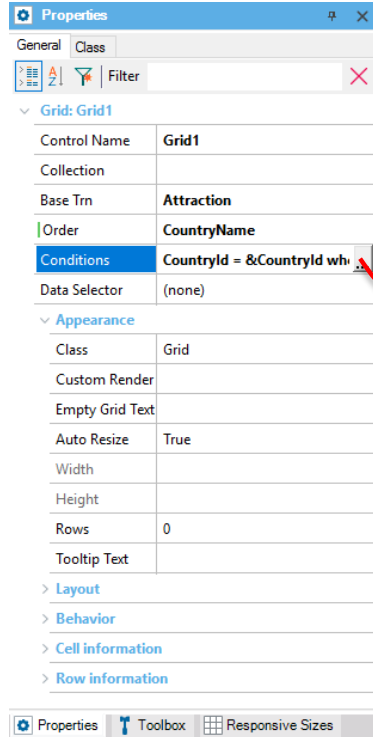
- Web Form:** Shows a form with three input fields: "Country Id" (with a dropdown menu), "Attraction Name From", and "Attraction Name To". Below these is a grid control labeled "GRID".
- Source:** Contains the following code:

```
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:

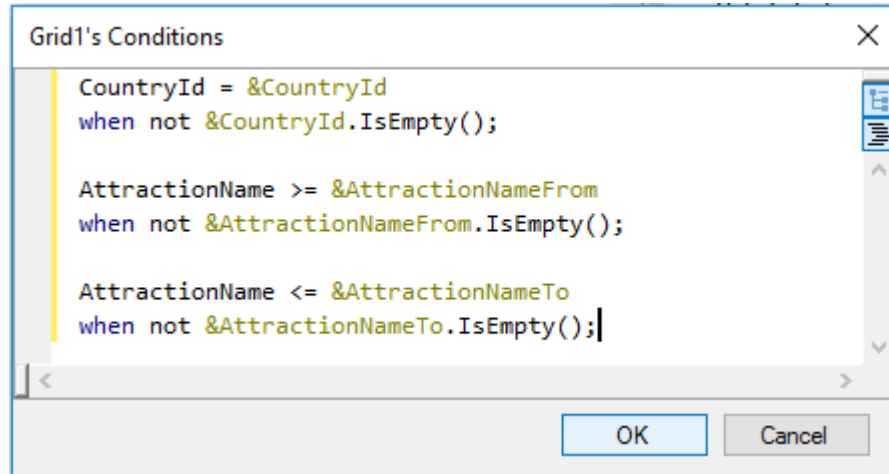
| Id | Attraction Name | Country | Photo |
|--------------|-----------------|-------------|---|
| AttractionId | AttractionName | CountryName |  |

Muchas condiciones



The Properties window shows the configuration for a Grid control named Grid1. The 'Conditions' property is highlighted, showing the expression 'CountryId = &CountryId wh...'. A red arrow points from this property to the 'Grid1's Conditions' dialog box.

| 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 | |



The 'Grid1's Conditions' dialog box displays the following conditions:



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

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

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

Buttons: OK, Cancel

Eventos

| | | | | | | |
|----------------------|---------------------|-------------|---|--------|---|----------|
| Country Id | &CountryId ▾ | | | | | |
| Attraction Name From | &AttractionNameFrom | | | | | |
| Attraction Name To | &AttractionNameTo | | | | | |
| GRID | | | | | | |
| Id | Attraction Name | Country | Photo | Trips | | |
| AttractionId | AttractionName | CountryName |  | &trips |  | &newTrip |
| Total Trips | &totalTrips | | | | | |

First time

Start

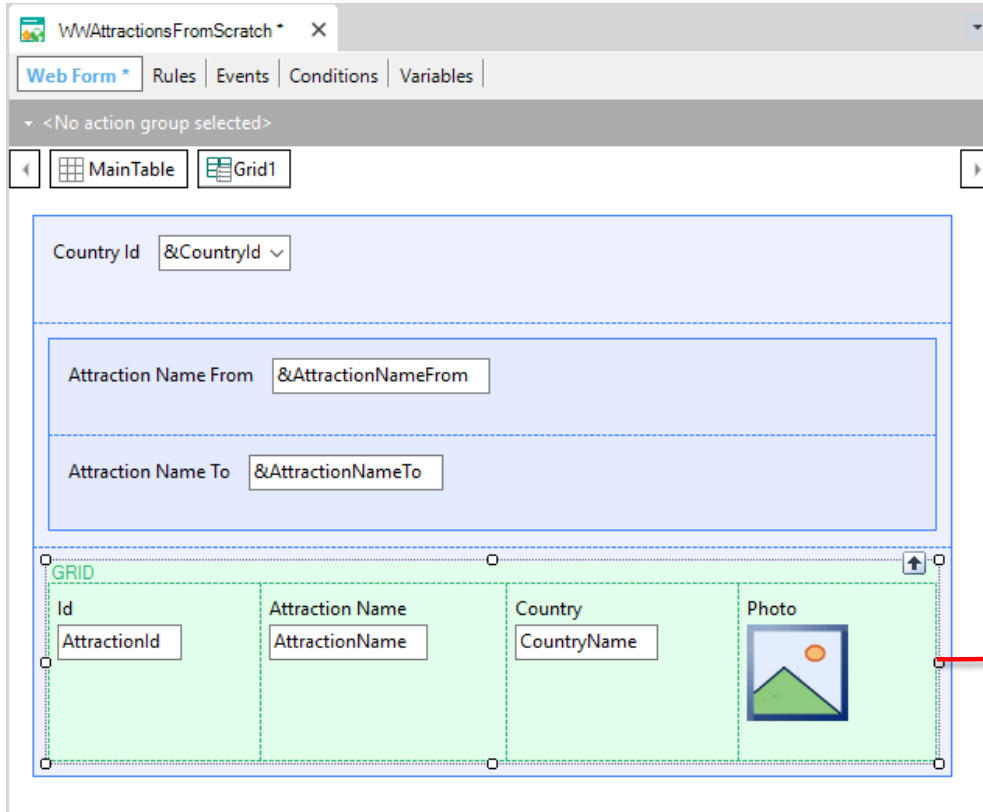
Refresh

Load

User / Control

Event

Evento Load en Web Panel CON tabla base



The screenshot shows the GeneXus IDE interface for a web panel named "WWAttractionsFromScratch". The "Web Form" tab is active, and the "Events" menu is open. The main workspace contains a form with three input fields: "Country Id" with a dropdown menu, "Attraction Name From", and "Attraction Name To". Below the form is a grid component labeled "GRID" with four columns: "Id" (AttractionId), "Attraction Name" (AttractionName), "Country" (CountryName), and "Photo" (represented by a photo icon). A red arrow points from the "Photo" column of the grid to a database icon, indicating a data source connection.

Evento LOAD

“N veces,
tantas como
registros
existan en la
tabla
recorrida”

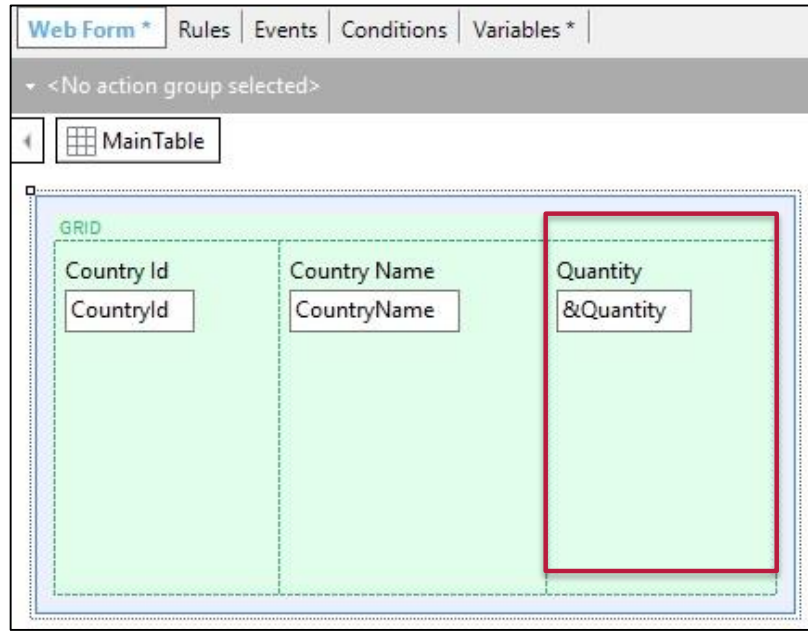
Otro ejemplo

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 |
| Total Trips | | | 4 |

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 |
| Total Trips | | | 7 |

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 |
| Total Trips | | | 3 |



Atributos en el Grid

The screenshot displays the GeneXus IDE interface. The main workspace shows a web form with several input fields and a grid. The grid is highlighted in green and contains columns for 'Id', 'Attraction Name', 'Country', 'Photo', and 'Trips'. A red arrow points from the 'AttractionId' column header in the grid to the 'Visible' property in the Properties panel, which is currently set to 'False'.

Properties Panel:

| 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
sin tabla base



Web Panels SIN TABLA BASE

Country Id

Attraction Name From

Attraction Name To

GRID

| Attraction Id | Attraction Name | Country | Photo | Trips |  |
|--|--|---|---|---|---|
| <input type="text" value="&AttractionId"/> | <input type="text" value="&AttractionName"/> | <input type="text" value="&CountryName"/> |  | <input type="text" value="&trips"/> | <input type="text" value="&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
Múltiples grids

Múltiples grids

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

PARALELO

Navegación independiente

FreeStyleGrid

The diagram shows a nested grid structure. The outer grid is labeled 'GRID' and contains two rows: 'Category Id' with a text input field 'CategoryId' and 'Category Name' with a text input field 'CategoryName'. The inner grid is also labeled 'GRID' and contains four columns: 'Attraction Id' with a text input field 'AttractionId', 'Attraction Name' with a text input field 'AttractionName', 'Country Name' with a text input field 'CountryName', and 'Attraction Photo' with a placeholder image of a mountain and sun. A red arrow points from the text 'FreeStyleGrid' to the top-right corner of the inner grid, indicating its styling.

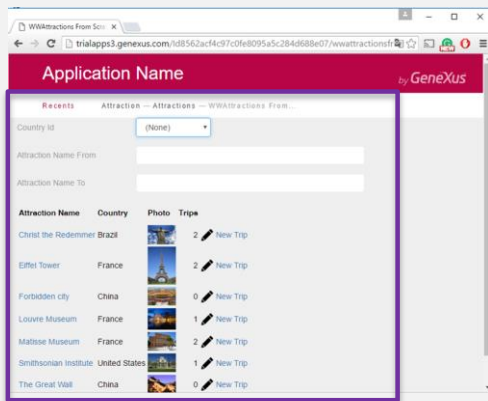
ANIDADO

Navegación tablas relacionadas

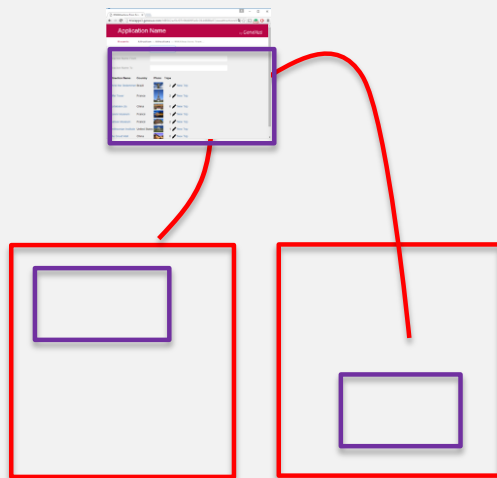
Tipos de Web Panels

Tipos de Web Panels

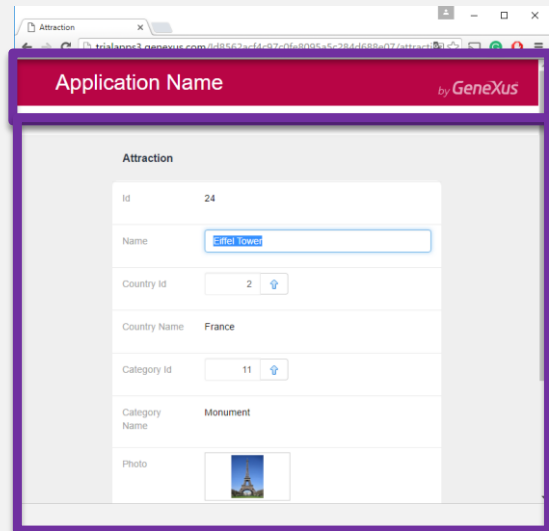
Web page (default)



Component



Master page



Design Systems

Design Systems

Elementos que dan
uniformidad y
coherencia a la UX

Master Page

e.g. Header / footer

Theme

Classes

Responsive sizes

Responsive tables

Controls

User control objects

Base styles

CSS libraries

Stencils

Design components

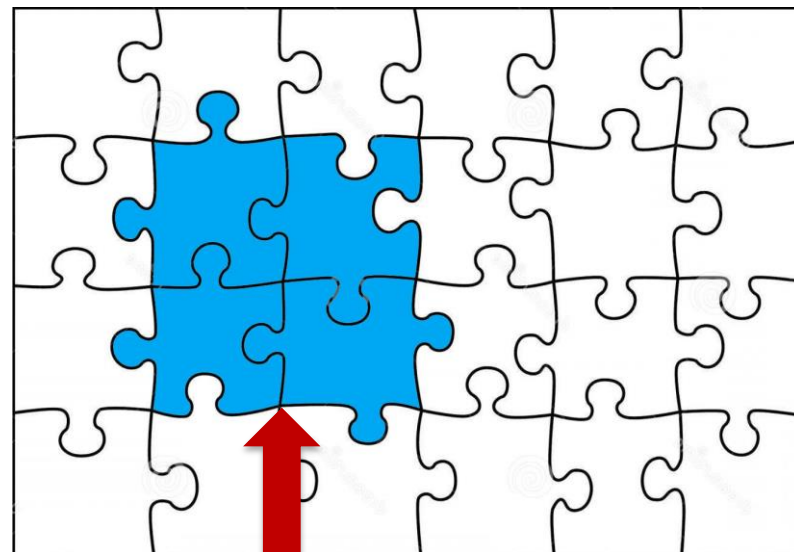
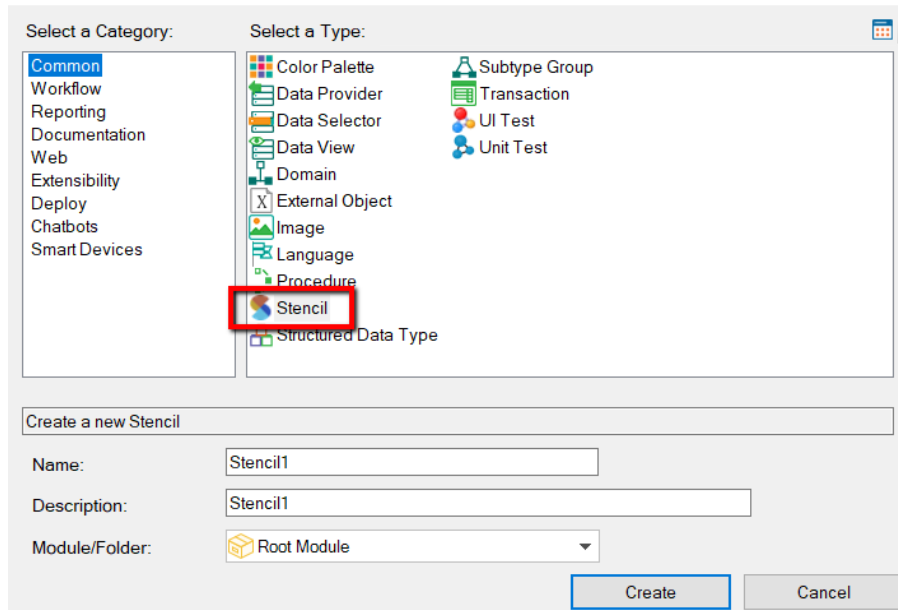
Patterns

Design / behaviour



RWD

Stencils



Design component

Objeto que permite repetir el diseño de una misma porción de pantalla (un conjunto de controles), en muchas pantallas.

Diseño responsivo

Diseño web responsive

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

| 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 |

| 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 |

Diseño web responsive

The image shows the GeneXus web form designer interface for a project named "WAttractionsFromScratch". The main workspace displays a form with several fields: "Country Id" with a dropdown menu, "Attraction Name From", "Attraction Name To", a "GRID" section with columns for "Id", "Attraction Name", "Country", "Photo", "Trips", and a "Total Trips" field at the bottom.

On the right side, the "Responsive Sizes" panel is open, showing configuration for the "Table- MainTable". The "Size" dropdown is set to "Medium (Desktop >= 992 px) inherits from Small". Below this, a list of responsive sizes is shown, with the first one highlighted in purple:

| |
|-----------------|
| 1,1 &CountryId |
| 2,1 Table1 |
| 3,1 Grid1 |
| 4,1 &totalTrips |

Below the list, the "Values" section is visible, with "Width" set to 100%, "Label Width" to 25%, "Offset" to 0%, "Visible" to True, and "Move" buttons. A "Default" checkbox is checked. At the bottom of the panel, there is a button labeled "Use default values for all screen sizes".

Red arrows point from the responsive size list to the corresponding fields in the form: from "1,1 &CountryId" to the "Country Id" dropdown, from "2,1 Table1" to the "Attraction Name From" field, from "3,1 Grid1" to the "GRID" section, and from "4,1 &totalTrips" 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

Pattern:

Category: *ALL Folder: *ALL

Pending Commits (8/8) Ignored Objects

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...

Commit

Knowledge Manager / Team Development

Partial commit

Team Development X

Commit to: http://sandbox.genexusserver.com/salto/home.aspx?Travel_Agency_0

Pattern:

Category: Folder:

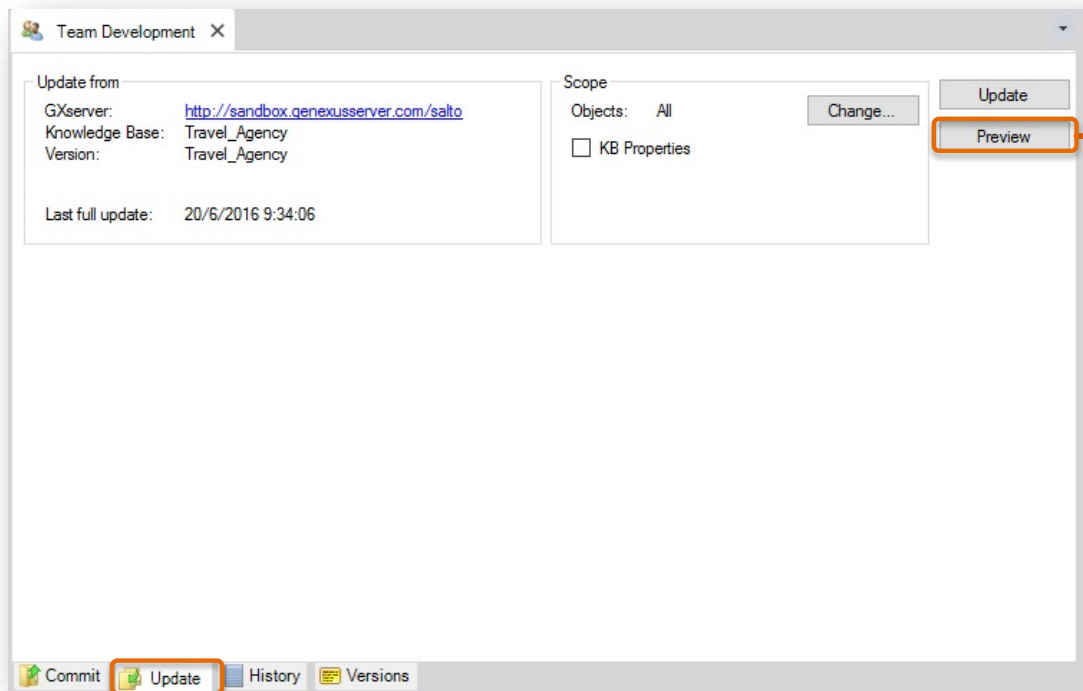
Pending Commits (8/8) Ignored Objects

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 the selected change.

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. |

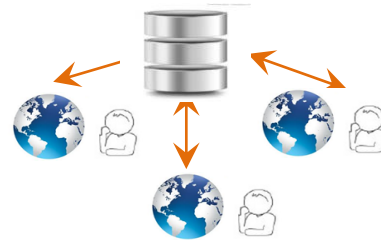
#3 - 20/6/2016 10:10:22 - GXTechnical\acaggiano

WorkWithAirline and AttractionsAndCategories listing

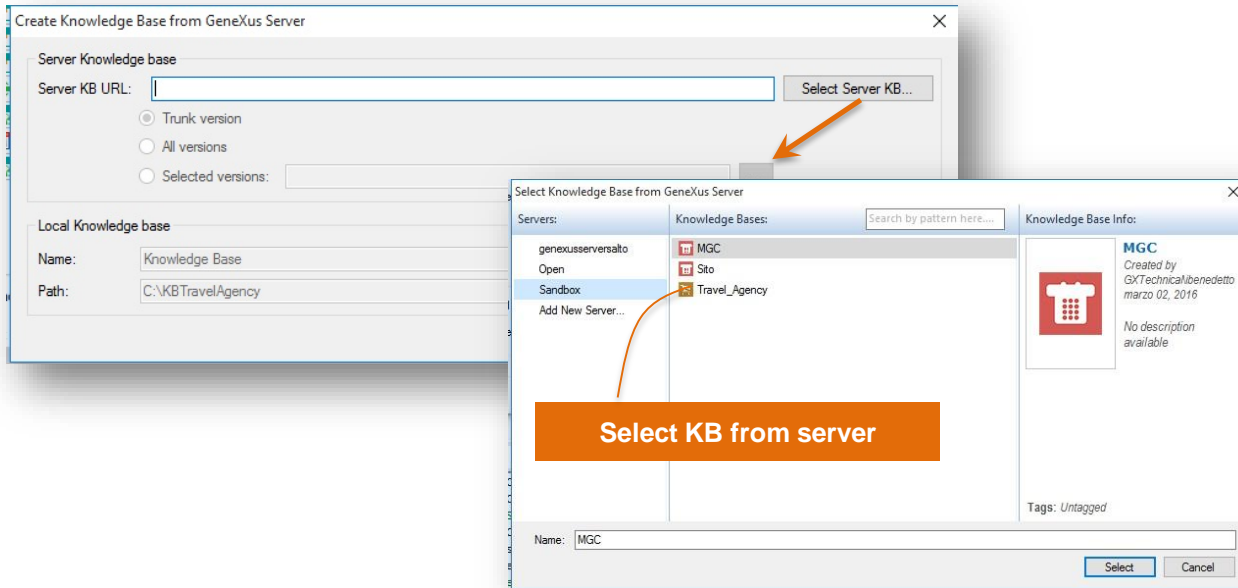
| Name | Type | Description | Action |
|--------------------------|-------------------|----------------------------|----------|
| Airline | Transaction | Airline | Modified |
| AirlineFlight1WC | Web Panel | Airline Flight1 WC | Inserted |
| AirlineFlightWC | Web Panel | Airline Flight WC | Inserted |
| 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



Seguridad con GAM



AUTHENTICATION



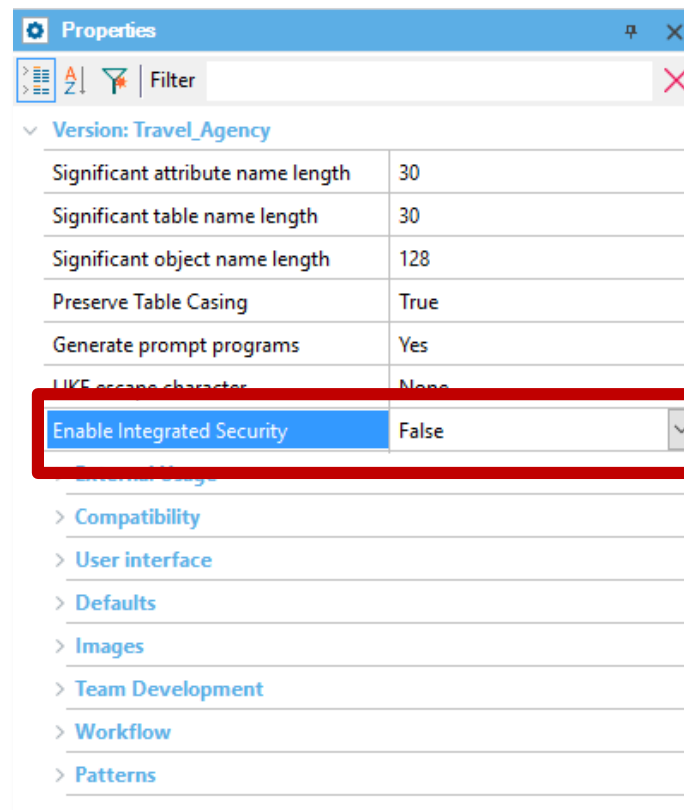
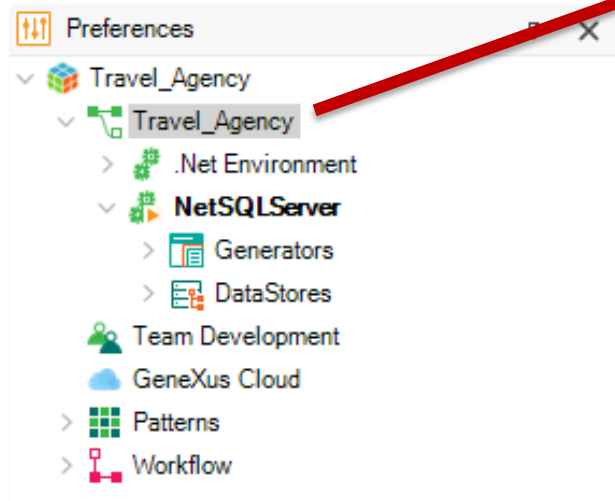
AUTHORIZATION

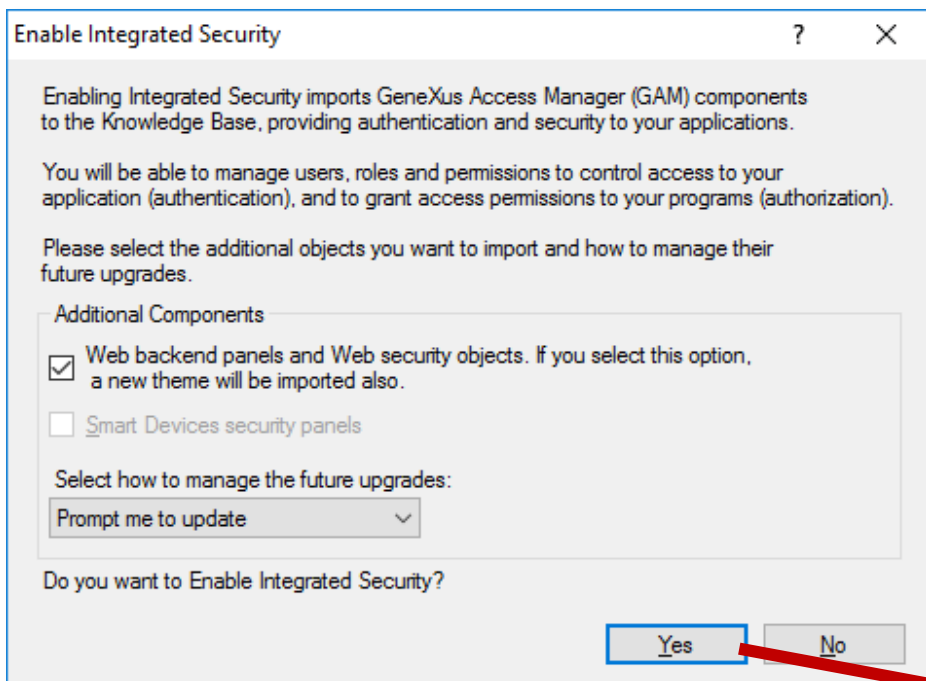
GeneXus™

ACCESS MANAGER

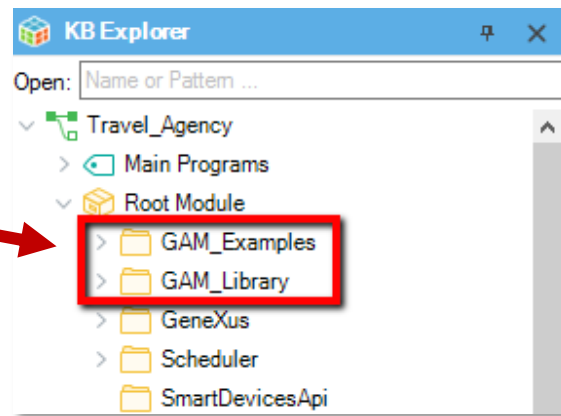


Habilitando el GeneXus Access Manager





Importando los objetos del GAM



Properties

> A Z | Filter

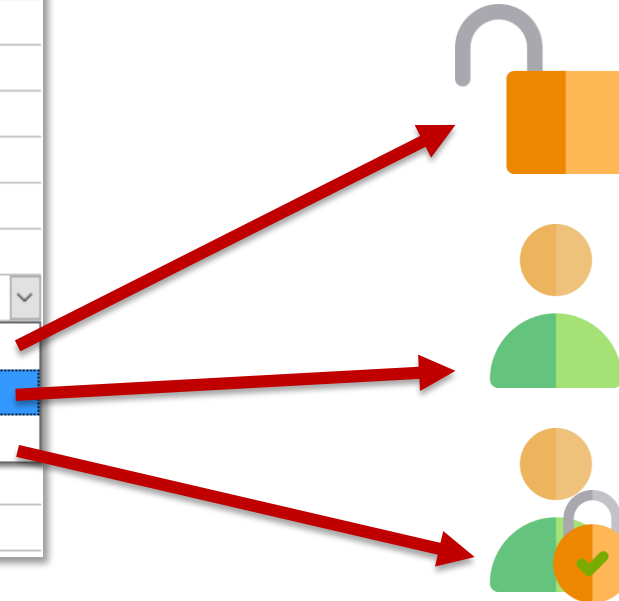
Version: Travel_Agency

| | |
|-----------------------------------|-------------|
| Significant attribute name length | 30 |
| Significant table name length | 30 |
| Significant object name length | 128 |
| Preserve Table Casing | True |
| Generate prompt programs | Yes |
| LIKE escape character | None |
| Enable Integrated Security | True |

Integrated Security

| | |
|---------------------------|----------------|
| Integrated Security Level | Authentication |
| Application ID | None |
| > Web specific | Authentication |
| > SmartDevices specific | Authorization |
| > External Usage | |
| > Compatibility | |

Seleccionando el nivel de seguridad



Acceso al backend del GAM (objeto GAM Home)

The screenshot displays the GeneXus Access Manager (GAM) backend interface. The top navigation bar is dark red and contains the text "GENEXUS ACCESS MANAGER" on the left, and "USERS", "ROLES", and "SETTINGS" with a dropdown arrow in the center. On the right side of the navigation bar, there is a circular profile icon with the letter "A" and the text "Administrator".

Below the navigation bar, the main content area has a white header with "SHOW FILTERS" (with a checkmark), the title "Users", a search bar with the placeholder "Try a Search", and a "+ ADD" button.

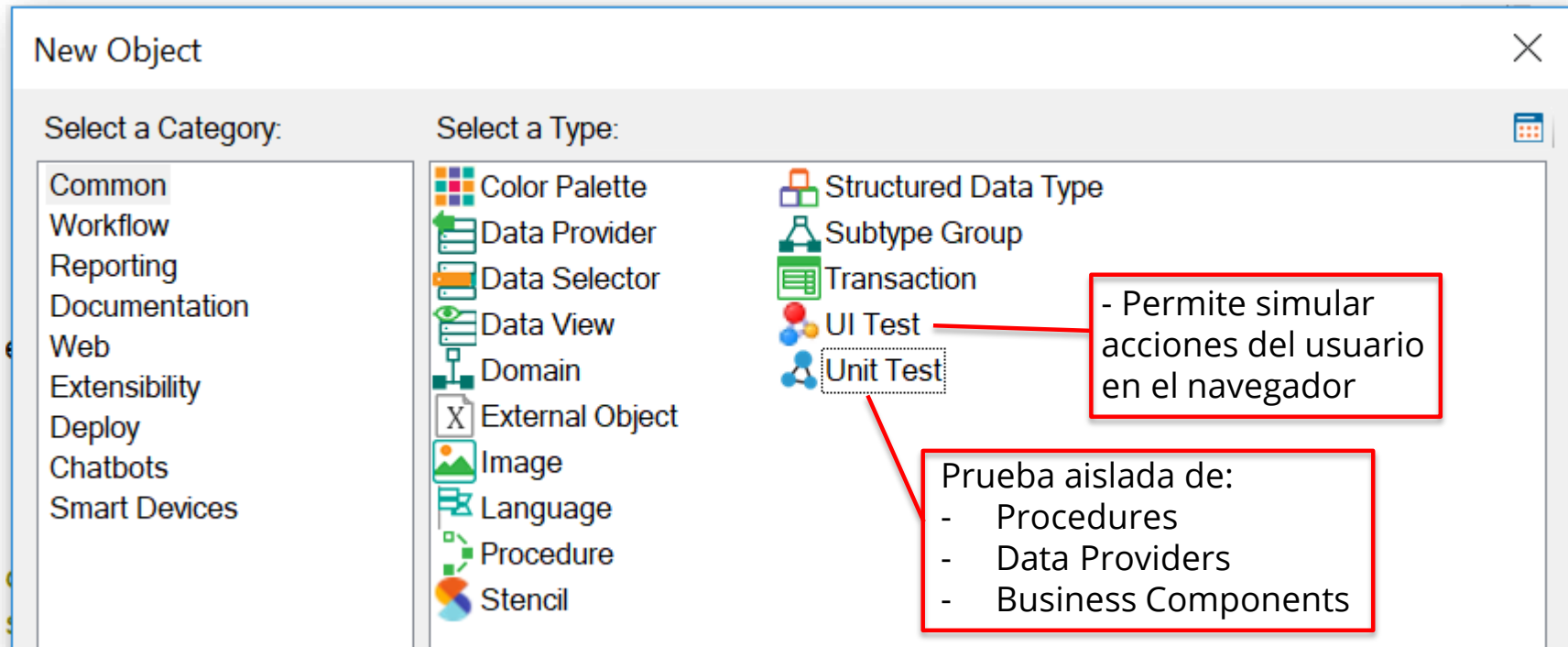
The main content area features a table with the following columns: "User Name", "First Name", "Last Name", "Authentication", and "EDIT". The table contains one row of data:

| User Name | First Name | Last Name | Authentication | EDIT |
|-----------------------|---------------|-----------|----------------|------|
| admin | Administrator | User | local | EDIT |

Below the table, there are navigation links: "FIRST /", "PREV /", and "NEXT".

Testing

Objetos para generar testing unitarios y testing de interfaz



Test Unitario

Se crean los objetos:

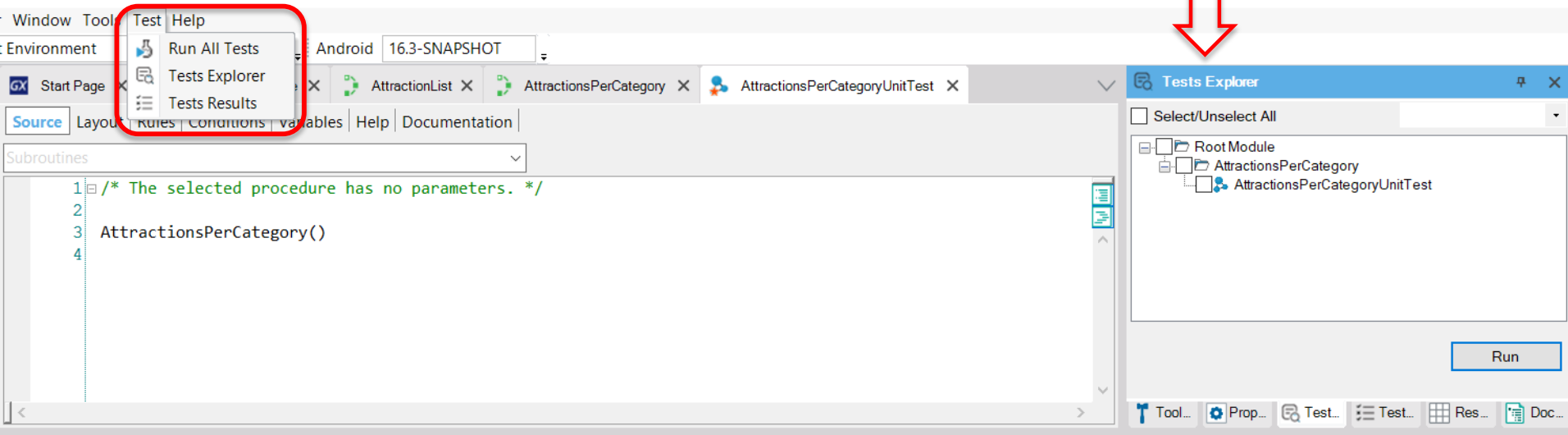
- <ObjectName>UnitTest
- <ObjectName>UnitTestData
- <ObjectName>UnitTestSDT

The screenshot shows the GeneXus IDE interface. The top menu bar includes 'Source', 'Layout', 'Rules', 'Conditions', 'Variables', 'Help', and 'Documentation'. The 'Subroutines' dropdown is open, showing a list of subroutines. The code editor displays 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
```

The context menu is open, showing various options. The 'Create Unit Test' option is highlighted at the bottom.

Test Explorer



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

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

The **Test** menu is open, with the following options:

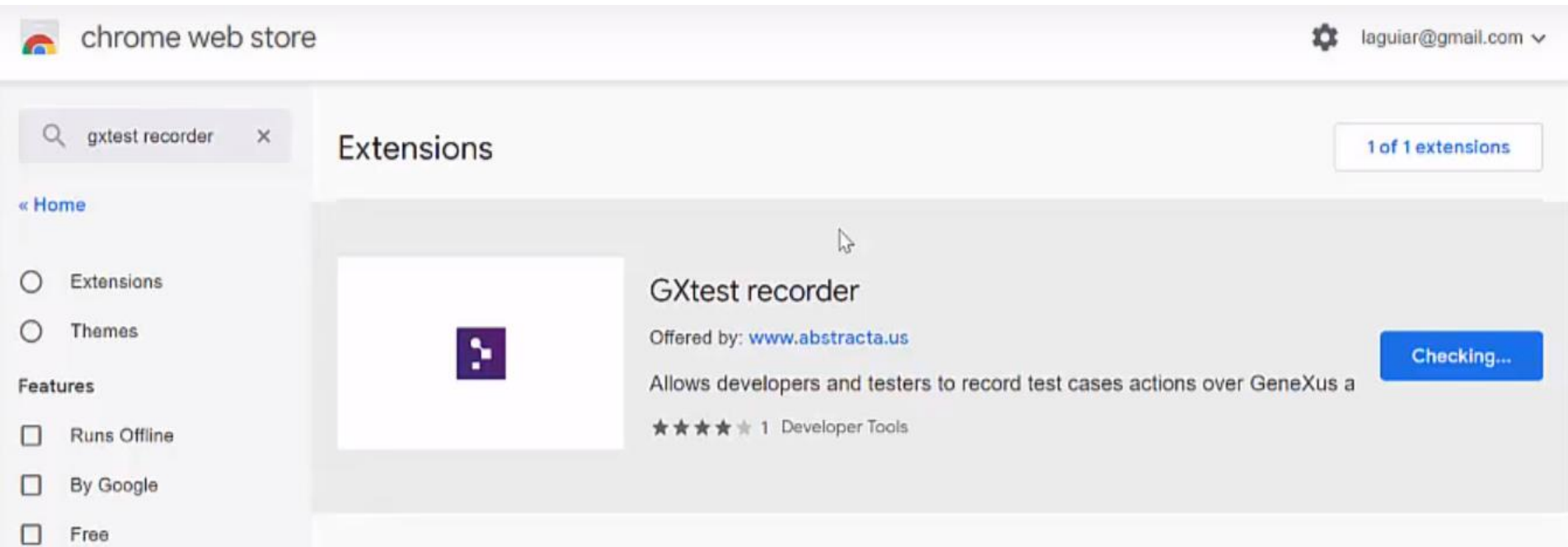
- Run All Tests
- Tests Explorer
- Tests Results

The **Tests Explorer** window is open on the right side of the IDE. It displays a tree view of the test structure:

- Root Module
 - AttractionsPerCategory
 - AttractionsPerCategoryUnitTest

A red arrow points to the **Tests Explorer** window. The **Run** button is visible at the bottom right of the Tests Explorer window.

Test de Interfaz: 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 navigation menu with 'Home', 'Extensions', and 'Themes'. 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

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

★★★★★ 1 Developer Tools

Checking...

Test de Interfaz

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: []
 TARGET: [] [Select] [Find]
 VALUE: []

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: []
 TARGET: [] [Select] [Find]
 VALUE: []

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

Uruguay

Rio de Janeiro

Monument

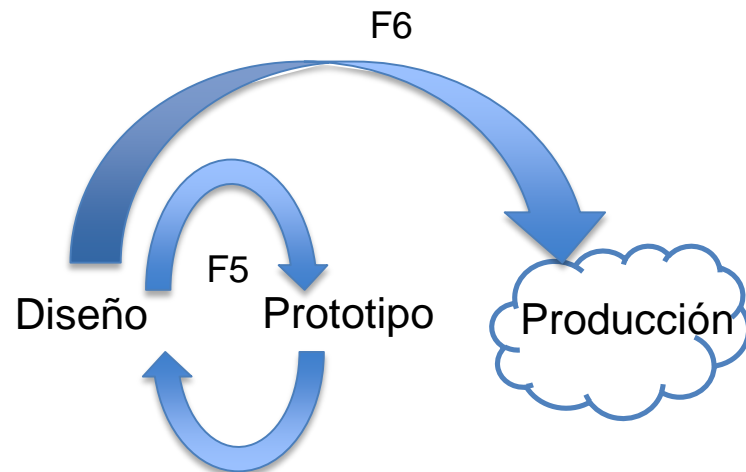
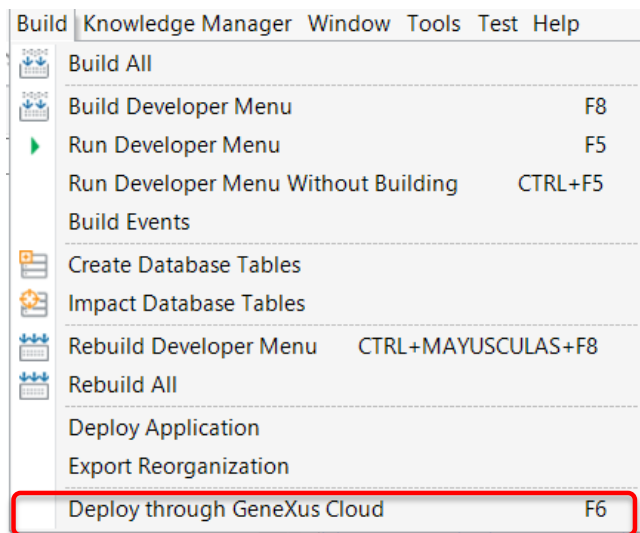
Attraction,Country already exists

Command Recorded
 command: ClickBy
 target: id=BTNINSERT

UI Test (Requiere licencia GXTest)

Deployment

Despliegue automático con 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

Integración

Módulo de Inteligencia 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

New Object

Select a Category: Select a Type:

Common
Workflow
Reporting
Documentation
Web
Deploy
Chatbots
Smart Devices

Conversational Flows

Create a pattern instance of Conversational

Name: ConversationalFlows1

Description: Conversational Flows1

Module/Folder: Root Module

Create Cancel

ConversationalFlows1* X

Instance Data Try Live Documentation

- Conversational Flows Instance
 - Flow: Greetings
 - Response
 - Message (text message)
 - Flow: Hello World
 - User Input
 - UserName
 - Response
 - Message (text message)

Citizen Service

9/14/18 8:33 PM

Please describe the problem:

9/14/18 8:33 PM

There's a missing traffic poster

9/14/18 8:33 PM

Where did it happen?

9/14/18 8:33 PM

Milan 99

9/14/18 8:34 PM

Please select the traffic signal: [See again](#)

9/14/18 8:34 PM

Type to talk...

Smart Devices

Aplicar Pattern en Transacción

The screenshot shows the 'Patterns' configuration window for a 'Country' object. The window title is 'Country X'. The navigation menu includes 'Structure', 'Web Form', 'Win Form', 'Rules', 'Events', 'Variables', 'Help', 'Documentation', and 'Patterns'. Below the menu, it states 'Patterns usable in this object (underlined means pattern is applied)'. The 'Work With for Smart Devices' pattern is selected and underlined. A checkbox labeled 'Apply this pattern on save' is checked. A tree view on the left shows the object structure: 'Level (Country)' with sub-items 'List', 'Detail', 'Section (General)', 'Section (Place)', and 'Section (Sale)'. A red arrow points to the 'List' item. A large empty box on the right contains the text 'Select List, Detail or Section on the tree to edit it'.

Country X

Structure | Web Form | Win Form | Rules | Events | Variables | Help | Documentation | **Patterns**

Patterns usable in this object (underlined means pattern is applied)

Category | Work With for Web | Work With for Smart Devices

Apply this pattern on save

↑ Level (Country)

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

Select List, Detail or Section on the tree to edit it

Aplicar Pattern en Transacción

The screenshot shows the GeneXus KB Explorer interface. On the left, the 'Country' object is expanded, and 'WorkWithDevicesCountry' is selected. A red arrow points to this object. The main window displays the 'Patterns' tab for 'Country X', showing that the 'WorkWithDevice' pattern is applied to the 'List' section. 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 'Run' selected.

Properties window:

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

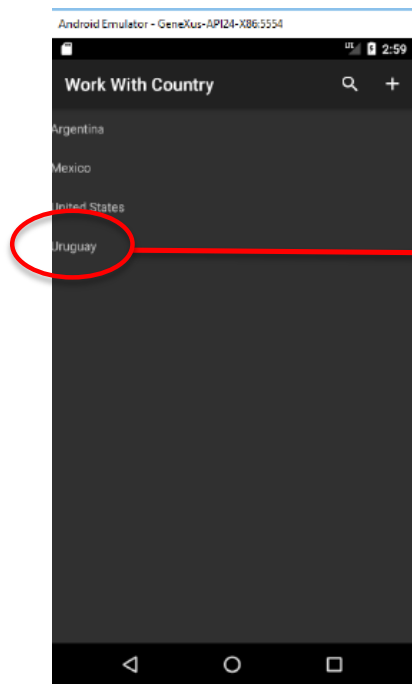
Context menu:

- 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

Hay que configurar su propiedad **Main program = True** y sobre el objeto presionar **click derecho y Run:**

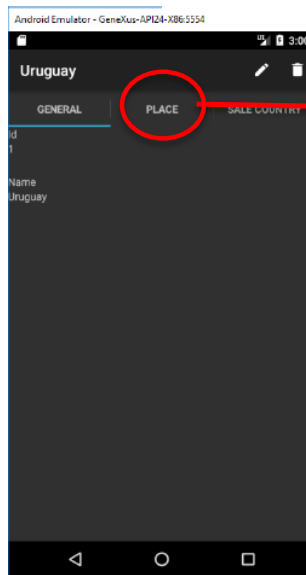
Aplicar Pattern en Transacción

botón derecho y Run:

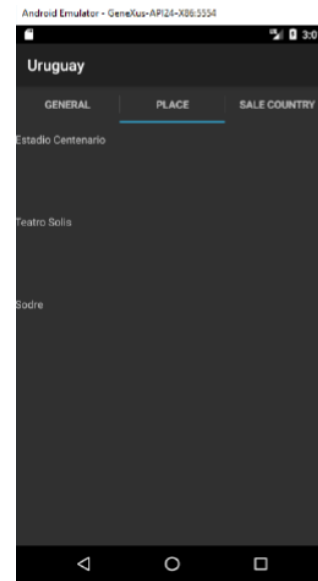
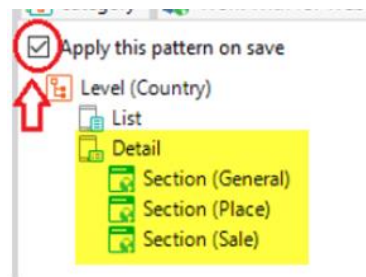


GENERAL

Presiono 1 country

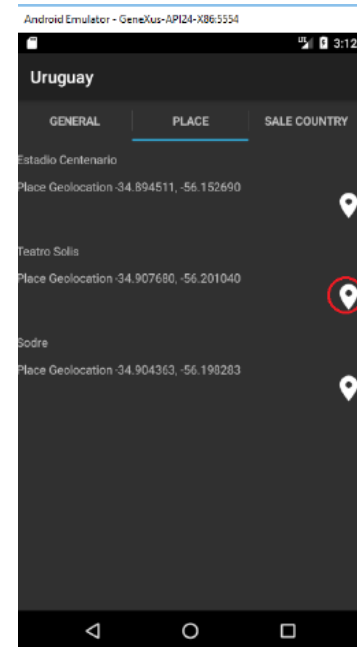
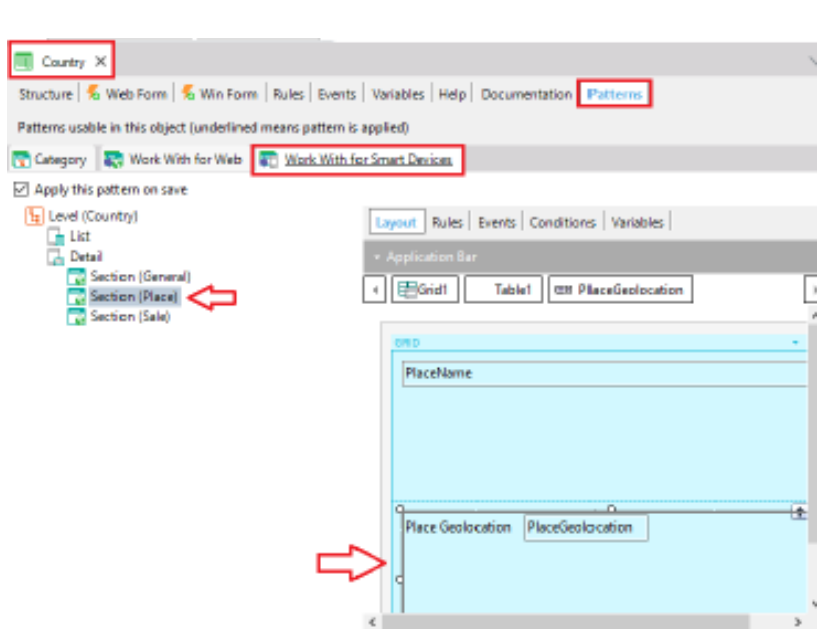


PLACE



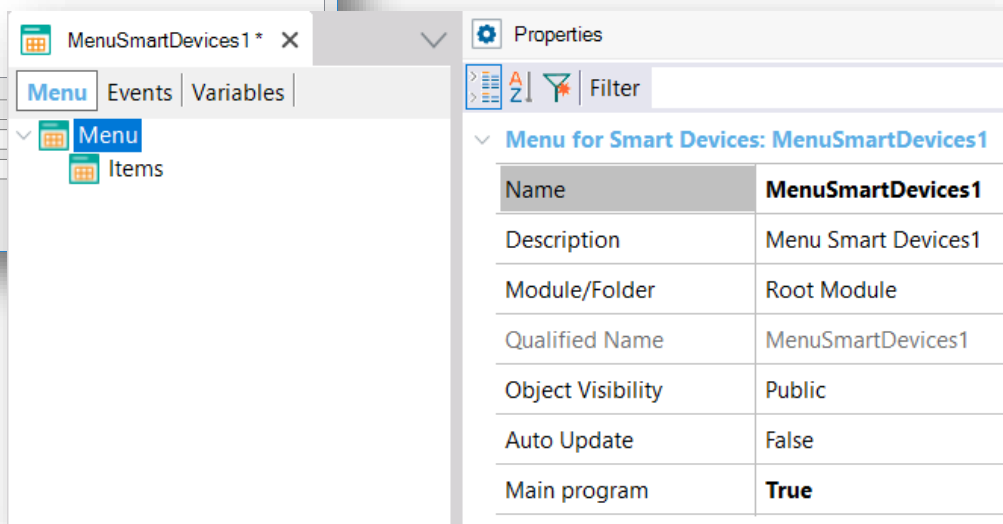
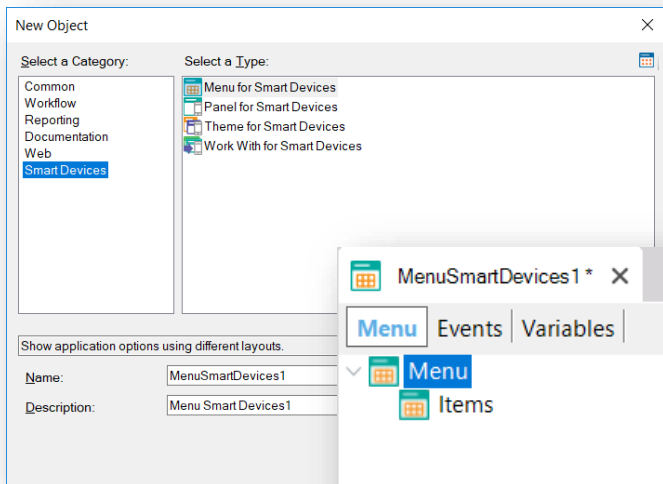
Aplicar Pattern en Transacción

Queremos para cada lugar, además de ver su nombre, ver su geolocalización.

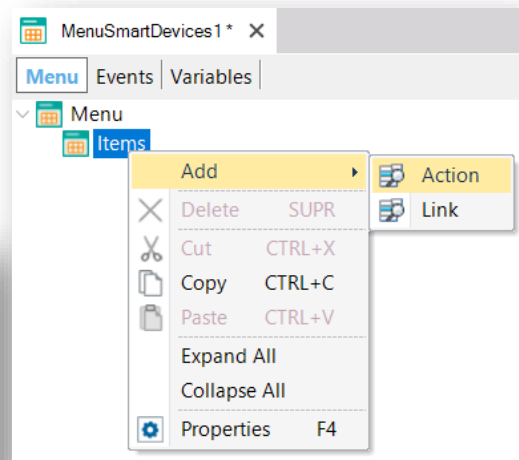


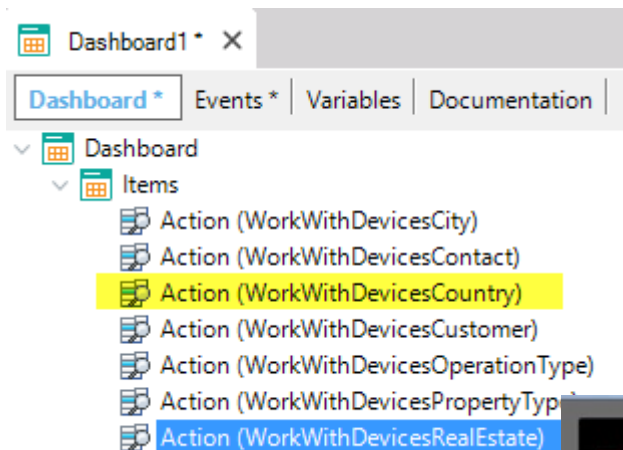
Menú de acceso : Menu for Smart Devices

Creación:



Add action...





Al presionar F5...



GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications