

Ejercicios prácticos

Actualización de GeneXus 15 a

GeneXusTM 17

Conceptos generales

Marzo 2022

Copyright ©GeneXus S.A. 1988-2020.

All rights reserved. This document may not be reproduced by any means without the express permission of GeneXus S.A. The information contained herein is intended for personal use only.

Registered Trademarks:

GeneXus is trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners.

CONTENIDO

OBJETIVO.....	3
KB DE PARTIDA	3
PROCESOS DE CARGA AUTOMÁTICA DE DATOS.....	3
CLÁUSULA UNIQUE.....	5
TRANSACCIONES DINÁMICAS	5
Caso de uso: Unión de información	5
Transacción dinámica como Transacción base	8
Transacción dinámica actualizable	10
Declaración de reglas en una transacción dinámica actualizable	12
Disparo de las reglas declaradas en las transacciones Ticket y Visit.....	13

OBJETIVO

El objetivo de este práctico es aplicar conceptos y desarrollar funcionalidades importantes como definir procesos automáticos de carga de datos, unificar datos utilizando transacciones dinámicas, Business Components y Data Providers.

KB DE PARTIDA

Se partirá de la versión de la KB que puede crear desde <http://samples.genexusserver.com/v17>, (KB de nombre UpgGX15_GX17) eligiendo la versión de nombre **TravelAgency_GeneralConcepts**.

Observe los objetos presentes en la KB.

Transacciones:

- Country
- Category
- Attraction
- Visit
- Ticket

Dominios

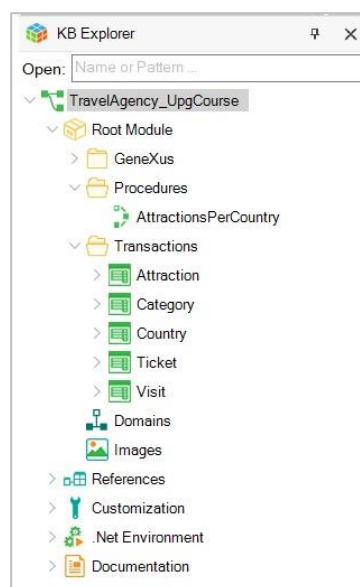
- Id, autonumerado
- Name

Imágenes

- Las imágenes correspondientes a las banderas de los países y a las atracciones turísticas

Procedures

- AttractionsPerCountry



PROCESOS DE CARGA AUTOMÁTICA DE DATOS

Para comenzar a trabajar con la aplicación se necesita contar con un conjunto de datos que se carguen en forma automática, sin depender de la acción del usuario final.

Utilizando la propiedad Data Provider de las transacciones, definir los siguientes procesos de carga automática de datos:

Country	Category	Attraction
• Brazil	• Monument	• Eiffel Tower, Monument, France
• France	• Museum	• Louvre Museum, Museum, France
• China	• Statue	• Great Wall, Landmark, China
• Italy	• Landmark	• Christ the Redeemer, Statue, Brazil

Configure en cada una de las transacciones su propiedad Data Provider con valor True

Data	
Data Provider	True
Used to	Populate data
Update Policy	Updatable

Verifique que en su KB se encuentren las imágenes, tanto de las banderas de los países como de las atracciones turísticas (con su nombre correspondiente).

Recuerde: Los Data Providers asociados a las transacciones con el fin de poblar datos, se disparan en forma automática en diferentes momentos, por lo tanto, asegúrese de que los datos no se multipliquen. Defina los correspondientes índices unique.

Codifique los correspondientes Data Providers. Piense en la carga de las atracciones turísticas. ¿Cómo saber el identificador de la categoría y país correspondientes?

No podemos saberlo, ya que ambos están basados en el dominio Id que es autonumerado, por lo tanto, debemos utilizar la función Find para encontrarlo. A modo de ejemplo:

```
AttractionCollection
{
    Attraction
    {
        AttractionName = "Eiffel Tower"
        AttractionImage = EiffelTower.Link()
        CategoryId = Find(CategoryId, CategoryName = "Monument")
        CountryId = Find(CountryId, CountryName = "France")
    }
}
```

¿Podemos estar seguros de que primero se cargarán los países y las categorías y luego las atracciones? Si, porque CategoryId y CountryId son claves foráneas en Attraction.

Ejecute la aplicación y verifique que los datos se hayan cargado.

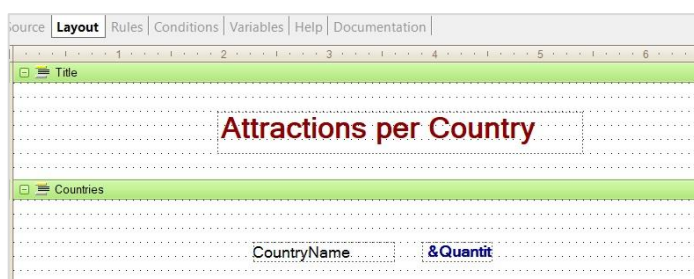
CLÁUSULA UNIQUE

Se requiere listar los países que tengan atracciones turísticas registradas., cada uno con su correspondiente cantidad de atracciones registradas. No se desean ver países repetidos

Abra el procedimiento AttractionsPerCountries que se encuentra dentro de la carpeta Procedures en la KB:

Verifique que el layout ya fue definido. Complete el source:

```
Print Title
For each Attraction
    unique CountryId
    &Quantity = count(AttractionName)
    Print Countries
Endfor
```



Verifique que el procedimiento ya tiene definidas las propiedades necesarias y la regla Output_file. Ejecútelo (click derecho / Run)

TRANSACCIONES DINÁMICAS

CASO DE USO: UNIÓN DE INFORMACIÓN

Las entradas a las diferentes atracciones turísticas se realizan abonando un Ticket, o como parte de una Visita. Observe las transacciones Ticket y Visit, presentes en la carpeta Transactions en su KB.

Ejecute al aplicación e ingrese 3 tickets y 2 visitas.

Se necesita unificar esa información (bajo el concepto de “Pase”) para poder listar, por ejemplo, los pases registrados para una determinada atracción turística en una cierta fecha.

Cree la transacción Pass.

El dominio Type es enumerado y tiene el objetivo de identificar si el pase se trata de una visita o de un ticket. Por lo tanto, debe formar parte de la clave primaria (es posible tener el ticket 1 y la visita 1).

Name	Type
Pass	Pass
PassId	Id
PassType	Type
PassDate	Date
AttractionId	Id
AttractionName	Name
AttractionImage	Image
CountryId	Id
CountryName	Name
PassAmount	Numeric(4.0)

Type Definition	
Based on	(none)
Data Type	Character
Length	10
Enum Values	Visit, Visit, Visit; Ticket, Ticket, Ticket
Collection	False

Defina la transacción Pass como dinámica, y codifique su Data Provider asociado para que unifique la información de Ticket y Visit:

Name	Type
Pass	Pass
PassId	Id
PassType	Type
PassDate	Date
AttractionId	Id
AttractionName	Name
AttractionImage	Image
CountryId	Id
CountryName	Name
PassAmount	Numeric(4.0)

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

```

PassCollection
{
    Pass from Ticket
    {
        PassId = TicketId
        PassType = Type.Ticket
        PassDate = TicketDate
        AttractionId
        PassAmount = TicketAmount
    }

    Pass from Visit
    {
        PassId = VisitId
        PassType = Type.Visit
        PassDate = VisitDate
        AttractionId
        PassAmount = 0
    }
}

```


Ejecute la aplicación. Abra la transacción Pass y navegue entre los registros. La información está unificada.

Application Name by GeneXus

Recent Pass

Pass

« < > » SELECT

Id	3
Type	Ticket
Date	03/15/22
Attraction Id	2
Attraction Name	Christ the Redeemer
Attraction Image	
Country Id	2
Country Name	Brazil
Amount	80

CANCEL

Si bien se trata de una transacción que no tiene una tabla asociada, es un objeto GeneXus de tipo Transaction. ¿Es posible aplicarle el pattern Work With for Web? Aplíquelo, y edite la instancia para que no se visualicen los identificadores (PassId, AttractionId, CountryId).

Work With for Web | **Work With for Smart Devices**

☒ Apply this pattern on save

- Work With Pattern Instance
 - Transaction (Pass)
 - Level (Pass)
 - DescriptionAttribute (PassDate)
 - Selection (Pass)
 - modes (Insert, Update, Delete)
 - Attributes
 - PassId
 - PassType
 - PassDate
 - AttractionId
 - AttractionName
 - AttractionImage
 - CountryId
 - CountryName
 - PassAmount
 - Orders
 - Filter
 - View (Pass Information)






Properties

Filter

attribute: PassId




Attribute	PassId
Description	Id
Link	
Autolink	True
Form	
Visible	False
Class	
Column Class	
Format	<default>

Presione F5.

Application Name						
by GeneXus						
Recents Attraction — Attractions — Ticket — Visit — Pass						
Pass						
<input type="text"/> <input type="button" value="SEARCH"/> <input type="button" value="INSERT"/>						
Type	Date	Attraction Name	Attraction Image	Country Name	Amount	
Ticket	03/08/22	Christ the Redeemer		Brazil	100	UPDATE DELETE
Visit	03/14/22	Christ the Redeemer		Brazil	0	UPDATE DELETE
Ticket	03/15/22	Great Wall		China	150	UPDATE DELETE
Visit	03/15/22	Eiffel Tower		France	0	UPDATE DELETE
Ticket	03/15/22	Eiffel Tower		France	50	UPDATE DELETE

TRANSACCIÓN DINÁMICA COMO TRANSACCIÓN BASE

Una vez unificada la información de Ticket y Visit, se necesita listarla, agrupada por atracción turística, como se muestra.

Passes per Attraction				
 Eiffel Tower				
1	Visit	03/15/22	0	
2	Ticket	03/15/22	50	
 Christ the Redeemer				
1	Ticket	03/08/22	100	
2	Visit	03/14/22	0	
 Great Wall				
3	Ticket	03/15/22	150	

Cree un nuevo procedimiento de nombre PassesPerAttraction.

```
Print Title
```

```
For each Pass order AttractionId
  print Attractions
  For each Pass
    print Passes
  Endfor
Endfor
```

Title			
Passes per Attraction			
Attractions			
AttractionId	AttractionName		
Passes			
PassId	PassType	PassDate	Pass

Antes de ejecutar, recuerde marcar, a nivel del objeto procedimiento, las propiedades Main Program = True, Call protocol = HTTP, y declarar la regla Output_file para generar el listado pdf.





Edite la instancia del pattern Work With aplicado a la transacción Attraction, y defina un Action para acceder al listado PassesPerAttraction.

Work With for Web Work With for Smart Devices

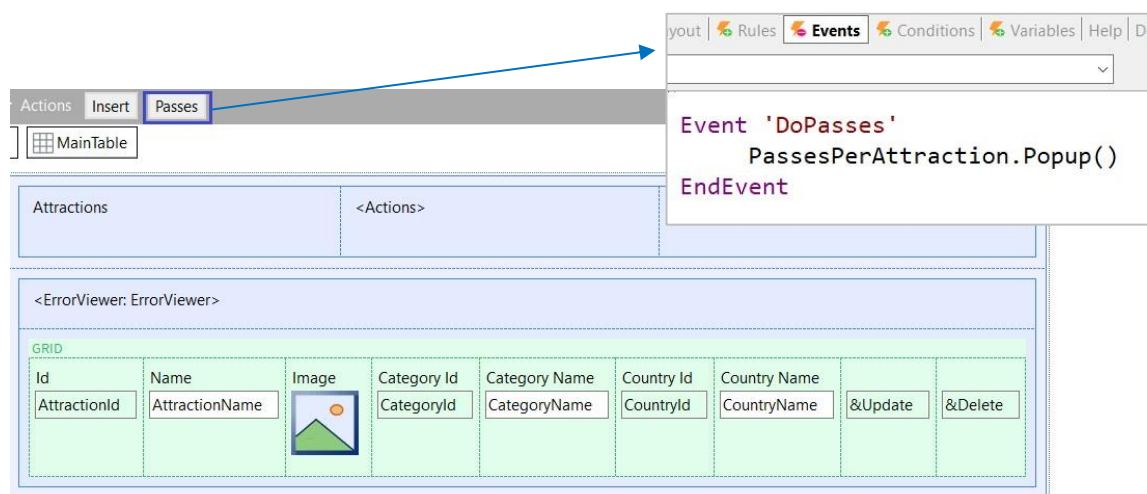
☒ Apply this pattern on save

- Work With Pattern Instance
 - Transaction (Attraction)
 - Level (Attraction)
 - DescriptionAttribute (AttractionName)
 - Selection (Attractions)
 - modes (Insert, Update, Delete)
 - Attributes
 - Orders
 - Filter
 - Attributes
 - AttractionName
 - Conditions
 - AttractionName like &AttractionName when not &AttractionName.IsEmpty()
 - Actions
 - Action (Passes) ← **action: Action (Passes)**
 - View (Attraction Information)

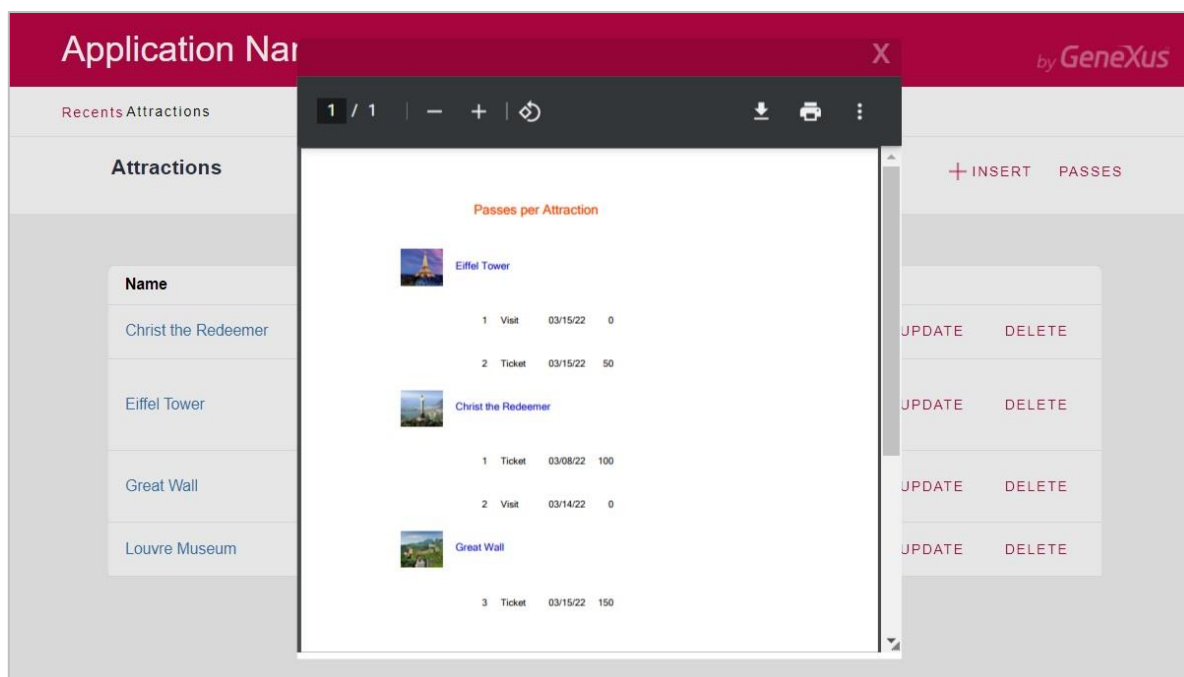
Filter	
action: Action (Passes)	
Name	Passes
Caption	Passes
GXObject	PassesPerAttraction
Condition	

Application Name by GeneXus					
Recent's Attraction — Attractions					
Attractions		Q Name	+ INSERT PASSES		
Name	Image	Category Name	Country Name		
Christ the Redeemer		Statue	Brazil	UPDATE	DELETE
Eiffel Tower		Monument	France	UPDATE	DELETE
Great Wall		Monument	China	UPDATE	DELETE
Louvre Museum		Museum	France	UPDATE	DELETE

Si desea ver el listado como una ventana popup, edite el objeto WWAttractions automáticamente generado por la aplicación del pattern



Presione F5. Entre a WWAttractions y seleccione el botón Passes.



TRANSACCIÓN DINÁMICA ACTUALIZABLE

Se necesita ahora poder ingresar tickets y visitas a través del form de la transacción Pass. Recordemos que dicha transacción es dinámica y, por lo tanto, no tiene una tabla asociada, sino que es una vista sobre los datos devueltos por su Data Provider.

Para permitir trabajar con registros de Passes es necesario indicar que la transacción Pass sea actualizable:

Name	Type
Pass	Pass
PassId	Id
PassType	Type
PassDate	Date
AttractionId	Id
AttractionName	Name
AttractionImage	Image
CountryId	Id
CountryName	Name
PassAmount	Numeric(4.0)

Data	
Data Provider	True
Used to	Retrieve data
Update Policy	Updatable

GeneXus solicita entonces que se codifiquen los correspondientes eventos Insert, Update y Delete para poder interactuar con las tablas correspondientes (según se trate de TICKET o VISIT).

Para eso, declare las transacciones Ticket y Visit como Business Component.

Luego codifique los eventos necesarios en la transacción Pass. Defina las variables &Ticket y &Visit basadas en sus respectivos tipos de datos Business Component;

```

Web Layout | Rules | Events | Variables | Help | Documentation | Patterns

Event Insert
  If PassType = Type.Ticket
    &Ticket = New()
    &Ticket.TicketDate = PassDate
    &Ticket.AttractionId = AttractionId
    &Ticket.TicketAmount = PassAmount
    &Ticket.Insert()
  else
    &Visit = New()
    &Visit.VisitDate = PassDate
    &Visit.AttractionId = AttractionId
    &Visit.Insert()
  endif
Endevent

```

```

Web Layout | Rules | Events | Variables | Help | Documentation | Patterns

Event Update
  If PassType = Type.Ticket
    &Ticket.TicketId = PassId
    &Ticket.TicketDate = PassDate
    &Ticket.AttractionId = AttractionId
    &Ticket.TicketAmount = PassAmount
    &Ticket.Update()
  Else
    &Visit.VisitId = PassId
    &Visit.VisitDate = PassDate
    &Visit.AttractionId = AttractionId
    &Visit.Update()
  Endif
Endevent

```

```

Web Layout | Rules | Events | Variables | Help | Documentation | Patterns

Event Delete
  If PassType = Type.Ticket
    &Ticket.Load(PassId)
    &Ticket.Delete()
  Else
    &Visit.Load(PassId)
    &Visit.Delete()
  Endif
Endevent

```

Importante: Observe que en la programación de los eventos Insert, Update y Delete de la transacción Pass no hemos declarado explícitamente el comando Commit, aunque estamos trabajando con los Business Components de las transacciones Ticket y Visit. ¿Por qué? Porque si bien Pass no tiene tabla asociada, es un objeto de tipo Transaction y por lo tanto tiene por defecto su propiedad Commit on Exit con valor True, y ese Commit se ejecuta automáticamente.

Presione F5 e ingrese, modifique y elimine registros a través del form de la transacción Pass.

DECLARACIÓN DE REGLAS EN UNA TRANSACCIÓN DINÁMICA ACTUALIZABLE

Ahora que es posible trabajar con Tickets y Visits a través del form de la transacción Pass, se requiere agregar comportamiento. ¿Es posible declarar reglas en una transacción dinámica?

- Por defecto, la fecha del pase debe ser la del día actual.
- Una visita no tiene costo.
- El costo de un ticket no puede superar los \$ 500.

```
Default(PassDate, &Today);

Error("A Visit is free")
- if PassType=Type.Visit and PassAmount <> 0;

Error("The Ticket amount can't be greater than 500")
- if PassType=Type.Ticket and PassAmount > 500;
```

Presione F5 y pruebe de ingresar tickets y visitas. Verifique que el comportamiento sea correcto.

The screenshot shows a web application window titled 'Application Name'. Inside, there's a tabbed interface with 'Recents' and 'Pass' tabs. The 'Pass' tab is active, showing a form titled 'Pass'. The form contains the following fields and values:

- Id:** 0
- Type:** Visit (dropdown menu)
- Date:** 03/16/22 (calendar icon)
- Attraction Id:** 1 (dropdown menu with a search icon)
- Attraction Name:** Eiffel Tower
- Attraction Image:**
- Country Id:** 1
- Country Name:** France
- Amount:** 200 (with a yellow tooltip saying 'A Visit is free')

At the bottom of the form are two buttons: 'CONFIRM' and 'CANCEL'.

DISPARO DE LAS REGLAS DECLARADAS EN LAS TRANSACCIONES TICKET Y VISIT

La transacción Pass fue creada con el objetivo de unificar la información de Ticket y Visit. Por lo tanto, es lógico pensar que las transacciones Ticket y Visit ya tenían declaradas sus propias reglas antes de crear la transacción Pass.

Elimine de la transacción Pass, la regla:

```
Error("The Ticket amount can't be greater than 500")
  if PassType=Type.Ticket and PassAmount > 500;
```

Declare la siguiente regla en la transacción Ticket:

```
Error("The Ticket amount can't be greater than 500")
  if TicketAmount > 500;
```

¿Se puede disponer en la transacción Pass de este comportamiento declarado en la transacción Ticket? Si, porque desde Pass se accede a Ticket y Visit utilizando Business Components, y por lo tanto, se dispone de su lógica.

Para poder ver en el form de la transacción Pass la regla declarada en Ticket, se deberá indicar que los eventos Insert, Update y Delete reciban como parámetro la colección de mensajes disparados en la ejecución del Business Component.

Recuerde: Al momento de creación de la KB, GeneXus crea el SDT Messages que tiene como finalidad guardar la colección de mensajes disparados en la ejecución de un Business Component. La variable &Messages debe definirse basada en dicho tipo de dato.

```
Event Insert (&Messages)
  If PassType = Type.Ticket
    &Ticket = New()
    &Ticket.TicketDate = PassDate
    &Ticket.AttractionId = AttractionId
    &Ticket.TicketAmount = PassAmount
    &Ticket.Insert()
    &Messages = &Ticket.GetMessages()
  else
    &Visit = New()
    &Visit.VisitDate = PassDate
    &Visit.AttractionId = AttractionId
    &Visit.Insert()
    &Messages = &Visit.GetMessages()
  endif
Endevent
```

```
Event Update (&Messages)
  If PassType = Type.Ticket
    &Ticket.TicketId = PassId
    &Ticket.TicketDate = PassDate
    &Ticket.AttractionId = AttractionId
    &Ticket.TicketAmount = PassAmount
    &Ticket.Update()
    &Messages = &Ticket.GetMessages()
  Else
    &Visit.VisitId = PassId
    &Visit.VisitDate = PassDate
    &Visit.AttractionId = AttractionId
    &Visit.Update()
    &Messages = &Visit.GetMessages()
  Endif
Endevent
```

Ejecute la aplicación nuevamente y verifique que desde el form de la transacción Pass se accede a la regla declarada en la transacción Ticket.


Application Name

Recents

Pass

Pass

- The Ticket amount can't be greater than 500

Id	0
Type	Ticket
Date	03/16/22
Attraction Id	2
Attraction Name	Christ the Redeemer
Attraction Image	
Country Id	2
Country Name	Brazil
Amount	800

CONFIRM

CANCEL