



Curso GeneXus X Ev2

Ejercicios Prácticos

Última actualización: 12 de junio de 2012

Copyright © Artech Consultores S. R. L. 1988-2010.

Todos los derechos reservados. Este documento no puede ser reproducido en cualquier medio sin el consentimiento explícito de Artech Consultores S.R.L. La información contenida en este documento es para uso personal únicamente.

Marcas Registradas

Artech y GeneXus son marcas o marcas registradas de Artech Consultores S.R.L. Todas las demás marcas mencionadas en este documento son propiedad de sus respectivos dueños.

Notas previas

El presente material corresponde a los ejercicios prácticos que el alumno del curso GeneXus deberá ir desarrollando en forma paralela a las clases teóricas.

Con este práctico el alumno podrá familiarizarse con la herramienta, su ambiente, menús, barra de herramientas, etc., así como podrá poner en práctica de una manera guiada y gradual los conceptos más importantes.

Finalizada esta parte práctica del curso, el alumno estará en condiciones de analizar una realidad simple y modelarla utilizando GeneXus, con el apoyo de un instructor, de forma tal de cerrar un módulo completo de aprendizaje fundamental para poder empezar a desarrollar sus propias aplicaciones.

Dado que la finalidad de este curso es aprender la lógica de GeneXus, su esencia, independientemente de la plataforma en que la aplicación vaya a implementarse, podría ud. prototipar en cualquiera de las plataformas soportadas por GeneXus.

El instructor deberá entregarle al comenzar el práctico los datos de la plataforma elegida para el curso, de manera tal que ud. configure el Environment con esa información.

Encontrará notas a continuación de algunos ejercicios, que invitan a la reflexión sobre lo realizado. Fueron incluidas para que el alumno no pase por alto conceptos teóricos importantes que están siendo aplicados. También las habrá con sugerencias de variaciones al ejercicio, para que las pruebe si así lo desea.

1. CREAR UNA BASE DE CONOCIMIENTO.....	3
2. EJERCICIOS DE TRANSACCIONES.....	3
TRANSACCIÓN ‘COUNTRY’	3
TRANSACCIÓN ‘CUSTOMER’	5
CONSTRUIR Y EJECUTAR LA APLICACIÓN.....	7
<i>Clave candidata</i>	8
<i>Efecto de permitir valores nulos en atributos Foreign Key (FK)</i>	8
MÁS TRANSACCIONES: ‘PRODUCT’ E ‘INVOICE’	8
TRANSACCIÓN ‘INVOICE’	10
EJECUTE LAS TRANSACCIONES ‘PRODUCT’ E ‘INVOICE’	13
<i>Regla de asignación versus fórmula</i>	13
NORMALIZACIÓN	14
AVANZADO: TRANSACCIÓN “CUSTOMER” [OPCIONAL].....	15
AVANZADO: USO DE IMÁGENES [OPCIONAL].....	18
AVANZADO: USO DE DOCUMENTOS Y ARCHIVOS [OPCIONAL]	19
3. CONSULTAS BATCH: PROCEDIMIENTOS.....	20
4. DATA SELECTORS.....	23
5. ACTUALIZACIONES NO INTERACTIVAS A LA BD.....	24
6. SUBTIPOS	26
7. INTERFAZ DE USUARIO: PATTERN WORK WITH Y WEB PANELS	26
8. CONSULTAS DINÁMICAS	27
9. WORKFLOW	27
10. APÉNDICE: SQL SERVER EXPRESS 2005	29

1. Crear una base de conocimiento

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

Nota: Se sugiere no crear la base de conocimiento en la carpeta "Mis Documentos" o cualquier otra carpeta que quede bajo "Documents and Settings", debido a que estas carpetas tienen permisos especiales otorgados por Windows.

Tómese unos minutos para familiarizarse con el IDE. Pruebe mover ventanas, personalizándolas a su gusto, visualizar ventanas (View y View/Other Tool Windows) y observe detenidamente el 'Folder View' dentro de la ventana 'Knowledge Base Navigator'. Verá que aparecen ya inicializados dominios, algunos objetos, imágenes, etc. Sugerencia: mantenga la ventana de propiedades abierta (F4), pues la utilizará continuamente. Dentro del 'Knowledge Base Navigator' eche un vistazo a las 'Preferences' donde se configura el 'Environment'.

2. Ejercicios de transacciones

A continuación se solicita que se definan algunas transacciones en la base de conocimiento creada antes. Para cada una de ellas se muestra una imagen del Web Form y una lista de atributos a incluir en su estructura (sin embargo, podrá ser necesario incluir otros). Se ejercitan diversas características de las transacciones (definición de reglas, fórmulas, etc.).

Recordar que para crear objeto existen varias alternativas:

- Hacerlo desde la 'Start Page'.
- Hacerlo desde el menú: File/New/Object.
- Ctrl+N.

Sugerencias:

- Utilizar la inserción de reglas mediante los snippets de la ToolBox; insertar funciones mediante **Insert/Functions**.
- Utilizar: **View/Work With Object**.
- Utilizar: **View/Work With Attributes**.
- Hacer Diagrama de Tablas y Transacciones creando un objeto tipo Diagram (**File/New object/Diagram**).

Transacción 'Country'

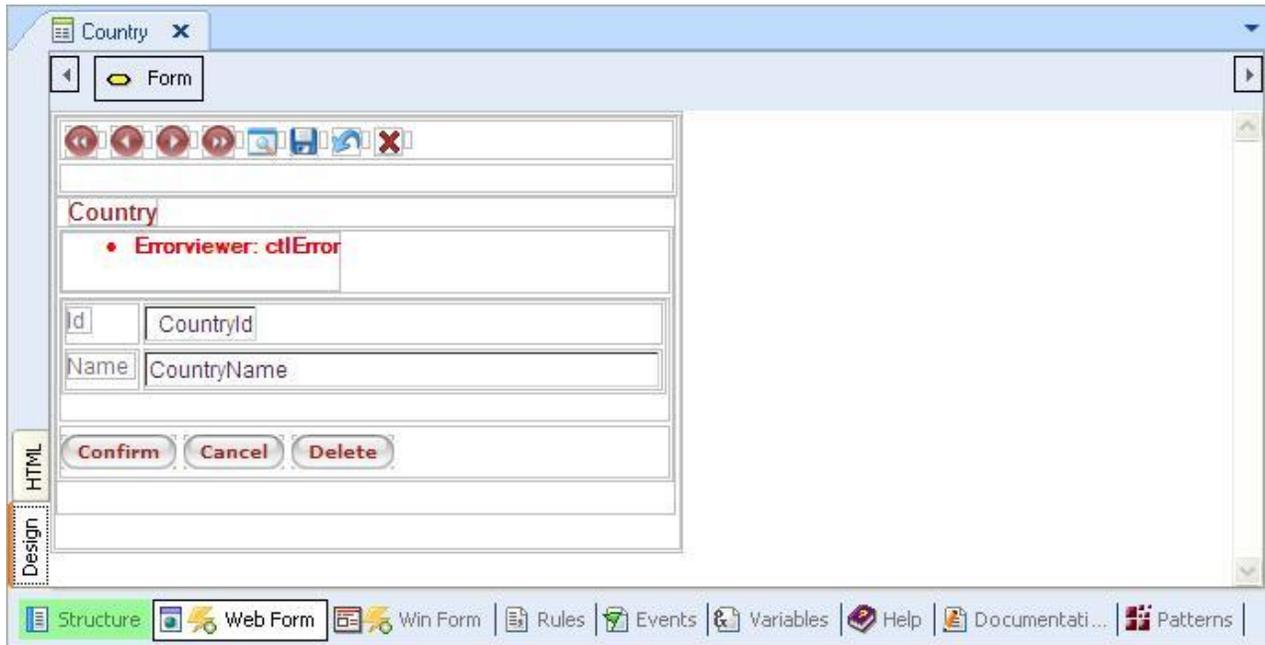
Atributos:

<i>CountryId</i>	Country Id	Numeric(6)
<i>CountryName</i>	Country Name	Character(50)

Siendo *CountryId* atributo clave, **autonumerado**.

- Luego de haber ingresado el *CountryId* en la estructura de la transacción, con toda su información relacionada (Type, Description, Nullable, etc.), y cuando esté pronto para digitar el próximo atributo *CountryName*, en lugar de digitar la primera letra, "C", para escribir Country, digite punto '.'. ¿Qué sucedió?
- En el campo Nullable del atributo *CountryName* en la estructura, se dispone de un combo box con los valores posibles para esa columna ("Yes", "No", por defecto es "No"). ¿Por qué no puede cambiar el valor por defecto del atributo CountryId?

Form Web:



- Para empezar, utilice la inserción de **reglas** mediante los 'Snippets' de la **ToolBox**, (si no tiene la ventana abierta, seleccione: View/Other Tool Windows) y Drag & Drop (D&D); o utilice el menú **Insert**.
- Para utilizar un atributo sin tener que tipearlo, utilice la ventana View/**Work With Attributes** y Drag & Drop; o el menú **Insert/Attribute** (Ctrl+Shift+A).
- Cuando presiona punto '.' luego del nombre de un atributo o variable, se le abrirá una lista de propiedades y métodos asociados a ese atributo o variable, para que seleccione el que desee. Recuerde que los métodos incluyen paréntesis '()'. Si no aparecen deberá agregarlos. Ej '*CountryName.IsEmpty()*'.

Reglas:

- No se permiten países sin nombre.

Observe los **Text Blocks** que aparecen en el form, a la izquierda de los atributos. En la estructura de la transacción, edite las propiedades del atributo *CountryId* y modifique el valor de la propiedad '**ContextualTitle**' por 'Identifier'. Grabe y observe qué sucede con el Text Block de la izquierda de *CountryId* en el form Web. Deduzca el valor por defecto que coloca GeneXus en la propiedad mencionada.

Transacción 'Customer'

Definir 4 dominios siguiendo lo que se indica a continuación.

Existen distintos caminos que conducen a la definición de un dominio:

- Cuando ya antes de comenzar a crear transacciones se identifica que será necesario definir determinado dominio, se puede crear mediante View/Domains o con doble clic sobre 'Domains' en el Folder View. Pruebe definir los dominios 'Name' como Character(50), 'Price' como Numeric(9.2) y 'Total' como 'Price' (dominio basado en dominio) de esta manera centralizada. Luego los utilizará en las transacciones, cuando defina atributos basados en estos dominios.
- Cuando esté definiendo al atributo *CustomerId*, al completar su tipo de datos digite: 'Id = Numeric(6.0)'. Luego de hacerlo, vaya a View/Domains para ver todos los dominios definidos en la KB. ¿Está allí 'Id'? Modifíquelo al dominio la propiedad **Autonumber**, pasando su valor a True.

Cuando esté definiendo al atributo *CustomerName*, observe lo que sugiere GeneXus automáticamente para su tipo de datos. También cuando esté definiendo al atributo *CustomerAddress*, observe lo que sugiere GeneXus para su tipo de datos. ¿Cómo supo GeneXus lo que sugirió en el 1er caso? ¿Y en el 2do?

Atributos:

<i>CustomerId</i>	Customer Id	Id = Numeric(6.0)
<i>CustomerName</i>	Customer Name	Name
<i>CustomerAddress</i>	Customer Address	Address
<i>CustomerGender</i>	Customer Gender (Sexo de Cliente)	Character(1)
<i>CustomerTotalPurchases</i>	Customer Total Purchases (Total de compras de Cliente)	Total
<i>CustomerTotalPayments</i>	Customer Total Payments (Total de pagos de Cliente)	Total
<i>CustomerBalance</i>	Customer Balance (Saldo de Cliente)	Total
<i>CountryId</i>	Country Identifier	Id
<i>CountryName</i>	Country Name	Name

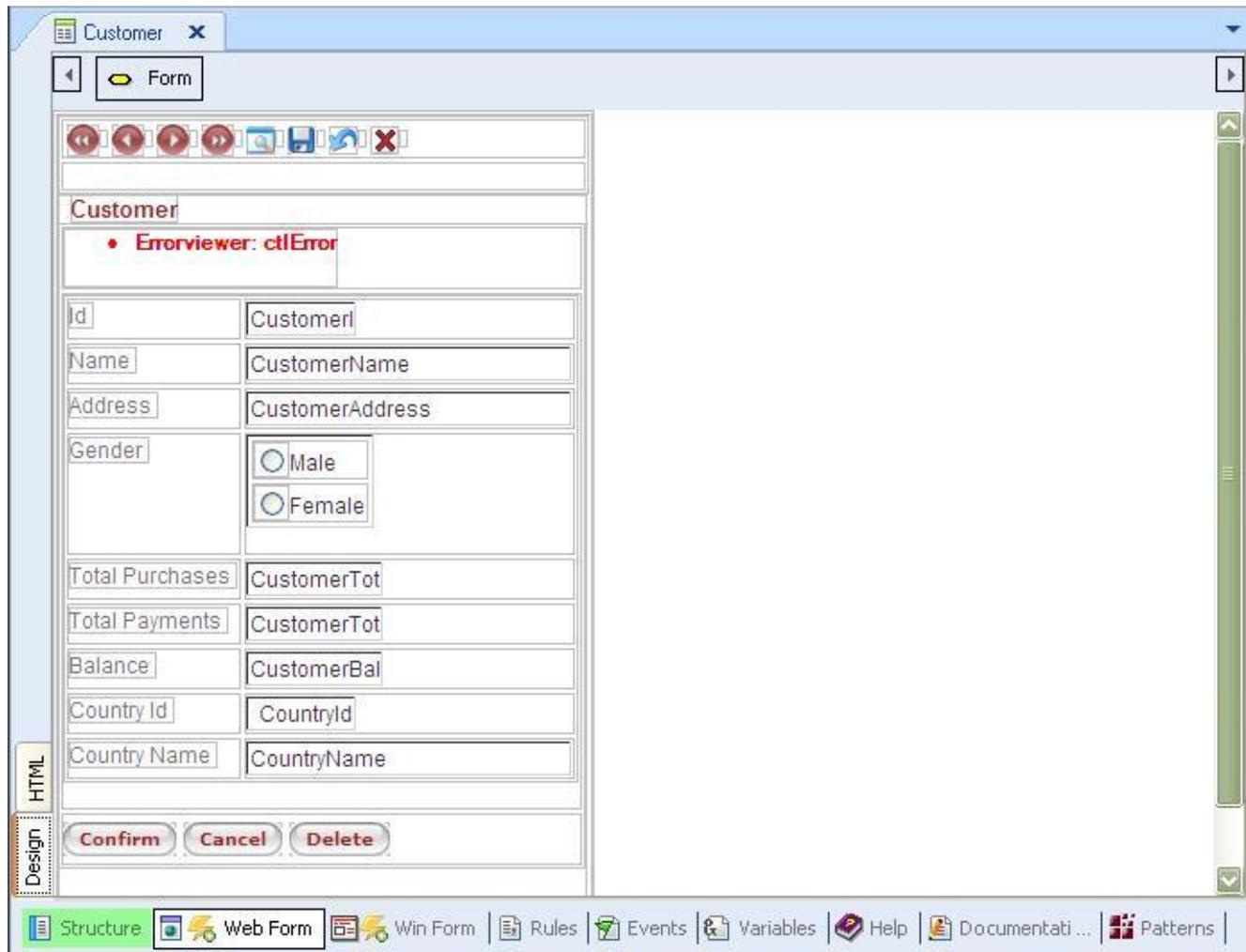
¿Por qué los atributos *CountryId* y *CountryName* no aparecen en la estructura de 'Customer' con los dominios 'Id' y 'Name' respectivamente? Asícieselos directamente en esta transacción. Abra 'Country' (debe refrescar la pantalla si la tenía abierta) y vea que *CountryId* y *CountryName* quedaron con los dominios asignados.

Edite el dominio 'Name' y modifique su tipo de datos para que en vez de Character(50) sea Character(30). Observe la repercusión en los atributos que tienen este dominio asociado (ver **Properties/Type Definition** de los atributos *CountryName* y *CustomerName*).

Observe los íconos que aparecen en la estructura de la transacción 'Customer', a la izquierda de los atributos *CountryId* y *CountryName* respectivamente. ¿Qué le están indicando?

¿Necesita autonumerar el atributo *CustomerId* que es clave? ¿Por qué tiene su propiedad Autonumber activada si usted no la modificó?

Form Web:



Generalidades:

- Definir *CustomerGender* para que aparezca en todo form como un control Radio Button.

Esto se debe hacer modificando las propiedades **del atributo *CustomerGender* en la estructura**. Para la propiedad **Control Type** seleccione el valor 'Radio Button' y para la propiedad **Values** agregue el texto 'Male' (masculino) con el valor M y 'Female' (femenino) con el valor F. Observe que tras grabar, el form se actualiza mostrando el radio button. Por haber modificado estas propiedades **asociadas al atributo**, cada vez que se agregue *CustomerGender* en un form de cualquier objeto GeneXus, por defecto aparecerá como Radio Button. En cambio si se configuran las propiedades **asociadas al control** en el form, solo aplicará a ese control *CustomerGender* en este form en particular, y no en cualquier otro donde se inserte.

Reglas:

- No se permiten clientes sin nombre.
- Avisar en caso de que no se ingrese la dirección.
- No permitir que se ingrese valor en el total de compras (se irá actualizando desde las facturas del cliente).
- El saldo del cliente se calcula como la diferencia entre sus compras y pagos.
- No permitir que se ingrese valor en este atributo (*CustomerBalance*).
- Que ofrezca por defecto el valor 'Female' de *CustomerGender*.

Construir y ejecutar la aplicación

Luego de definidas las dos transacciones anteriores querrá probar en ejecución lo hecho, para lo cual deberá implementar la aplicación en una plataforma determinada.

Recuerde que en el Environment es donde se almacena la información de la implementación del sistema.

Asegúrese de tener editada la ventana 'Output' (View/Other Tool Windows). Para ejecutar la aplicación, **presione F5**.

- Al hacerlo se abrirá una pantalla donde deberá configurar el nombre de la base de datos en la que se crearán las tablas de su aplicación, el servidor del DBMS de esa BD; usuario y password de conexión a la base de datos (debe tener permisos de DBCreator). En el apéndice se le proporcionará la información para realizar esto.
- Una vez ingresada la información, se realizará el **Análisis de Creación de la Base de Datos** (cuya ejecución podrá ir viendo en la ventana de 'Output'), y el resultado será el IAR (reporte de Análisis de Impacto). Observe detenidamente la información que le brinda el reporte, para cada una de las tablas. Luego deberá presionar el botón 'Create', a fin de que sea creada la BD (si no existía) y las tablas.
- Luego de ejecutado el programa de Creación de la base de datos, se comenzarán a **generar los programas** correspondientes a los dos objetos creados, en el lenguaje definido cuando se creó la KB. Al finalizar la construcción de los programas, se abrirá automáticamente una ventana del browser de Internet, con la aplicación en ejecución. Observe todo lo que aparece listado en la ventana de 'Output'. Allí se informa todo lo que se hizo para tener esos programas en ejecución. También observe cómo se abre una solapa 'Navigation View' que le permitirá visualizar los objetos que fue necesario generar para construir la aplicación.

Ingrese algunos países y clientes. ¿De dónde salen el encabezado ('Application Header') y pie de página ('footer') que puede visualizar en ejecución de ambas transacciones? (Observe en GeneXus, en la solapa 'Navigation View' que se abrió en la ventana central, el objeto AppMasterPage generado). Luego:

- Pruebe ingresar un **país** sin dar valor al Identificador. ¿Está funcionando el autonumber?
- Pruebe ingresar un país sin nombre.
- Pruebe los botones de navegación para recorrer los países ingresados.
- Pruebe seleccionar un país de los existentes, utilizando el botón Select. ¿Qué objeto es el que se abre al hacerlo? ¿Lo programó ud.? Vaya a su KB y búsquelo en la lista de objetos.
- Pruebe editar un país existente, digitando su código en el campo *CountryId* y saliendo luego del campo.
- Pruebe vaciar el campo correspondiente al nombre del país y confirmar.
- Pruebe editar un país existente y eliminarlo.
- Pruebe ingresar un **cliente** nuevo, digítele un identificador de país inexistente, o deje vacío el país. ¿Qué índice se utilizó para validar que el país exista?
- Presione la imagen que se ofrece al lado de *CountryId* y seleccione uno de los países existentes. ¿Qué objeto se abrió? Observe que es el mismo que cuando presionó Select en la transacción "Country" para elegir un país.
- Pruebe dejar vacío el nombre de un cliente, y luego la dirección.
- Observe que *CustomerTotalPurchases* está Read Only, al igual que *CustomerBalance*. ¿Por qué?
- Pruebe eliminar un país para el que tenga por lo menos un cliente asociado. ¿Qué índice se utilizó para

- validar que el país tenía al menos un cliente asociado e impedir la eliminación?
- Ingrese 2 clientes con el mismo nombre. A continuación veremos cómo evitar que esto suceda.

Clave candidata

Si en su realidad no pueden existir 2 clientes con el mismo nombre, es decir, si *CustomerName* no puede tener valores duplicados, ¿cómo lo implementa para que sea chequeado automáticamente?

Defina un índice único sobre la tabla CUSTOMER, compuesto por el atributo *CustomerName*, de tipo ascendente.

Para hacerlo deberá abrir la tabla CUSTOMER (Folder View/Tables) y seleccionar la solapa Indexes.

Observe el Warning que se presenta en el 'Impact Analysis' luego del F5: le avisa que de tener clientes con el mismo nombre, la reorganización fallará. Ud. puede proseguir con la reorganización, sin embargo si es su caso el avisado en el warning, la reorganización cancelará. Por lo tanto vaya al browser, ejecute la transacción 'Customer' (<http://localhost/BillingSystem.NetEnvironment/customer.aspx>) y modifique el nombre de uno de los clientes con nombre repetido. Vuelva a GeneXus y presione nuevamente F5. Ahora sí podrá ejecutar la reorganización, se creará el índice único por *CustomerName* en la tabla CUSTOMER, y se validará que no se ingresen clientes con nombre repetido.

Efecto de permitir valores nulos en atributos Foreign Key (FK)

Como habrá visto, los programas GeneXus controlan la integridad referencial entre las tablas. Así es que para cada FK se controla que se le ingrese un valor válido (un valor existente como PK en la tabla referenciada); y también se controla por defecto que las FKs no queden con valor nulo (sin especificarles un valor válido determinado).

Debido a esto, hasta ahora no se le permitió ingresar un cliente sin asociarle país, sino que ha sido obligatorio asociarle a cada cliente un código de país válido.

Sin embargo tal vez para algún cliente en particular o para algunos clientes, no se disponga de la información de cuál es su país de origen. Si desea poder ingresar un cliente **sin** asociarle un país, ¿qué deberá modificar en la estructura de la transacción "Customer"?

Debe en la columna 'Nullable' para el atributo *CountryId*, cambiar el valor de *No* a *Yes*. Luego de hacerlo, salve y presione F5. Observe que se deberá reorganizar la BD. Luego intente dejar un cliente sin país y observe que ahora esto será posible.

Más transacciones: 'Product' e 'Invoice'

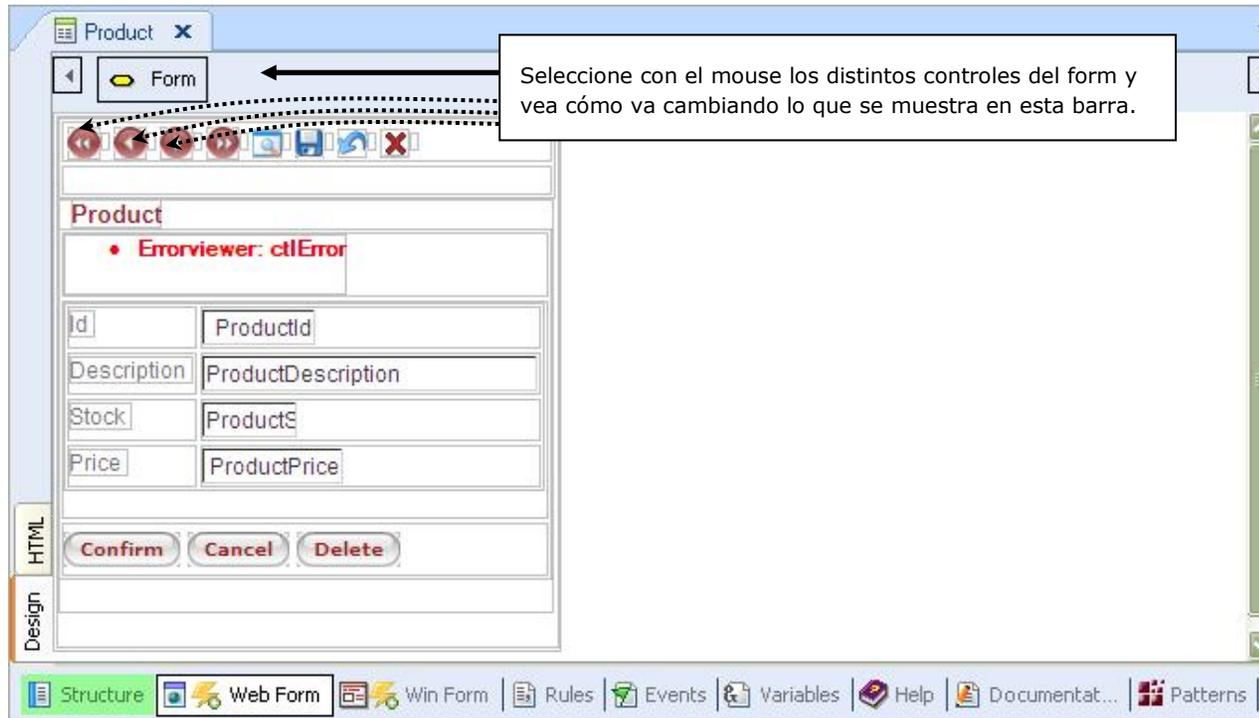
Crear la transacción 'Product' con los siguientes atributos en su estructura:

Atributos:

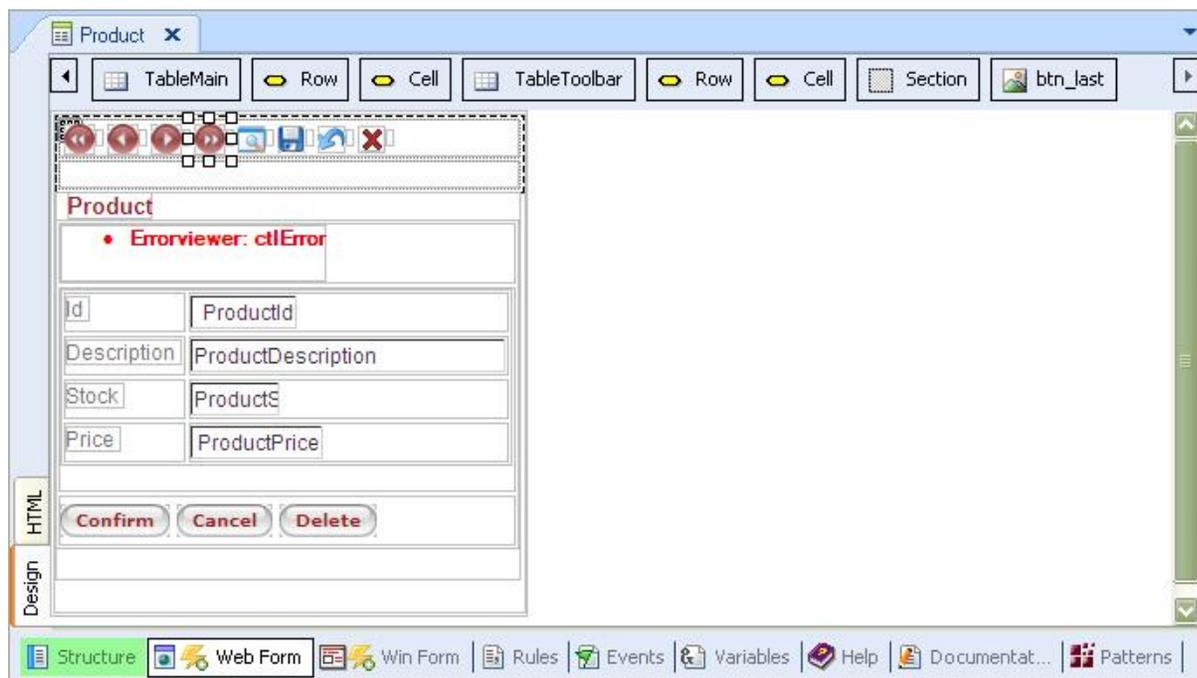
<i>ProductId</i>	Product Id	Id
<i>ProductDescription</i>	Product Description	Description = Character(30)
<i>ProductStock</i>	Product Stock	Stock =Numeric(4)
<i>ProductPrice</i>	Product Price	Price

Defina los dominios inline, es decir, defínalos directamente en la estructura de la transacción, cuando le surja la necesidad de asignarlos a atributos que esté definiendo.

Form Web:



Por ejemplo, al seleccionar el botón btn_last, la barra se ve así:



En su extremo derecho la barra contiene una etiqueta que identifica al control btn_last, y hacia la izquierda contiene etiquetas sucesivas que identifican los controles "contenedores". Vaya posicionando el mouse sobre las etiquetas de la barra que identifican a los controles del form, y vea cómo se remarca cada control dentro del form.

Tip: También cuenta con la ventana 'Document Outliner' para trabajar con los controles del form. Edítela y observe el árbol de jerarquía de los controles.

Reglas:

- No aceptar el stock del producto si se está modificando un artículo ya existente; solo permitir ingresar el stock del producto si se está insertando un producto.
- No permitir que el stock quede con valor vacío cuando se ingresa un producto nuevo.

Puede ejecutar ahora la nueva transacción, o esperar a hacerlo luego de crear la siguiente, 'Invoice'. Sugerimos esto último.

Transacción 'Invoice'

Cree a la transacción 'Invoice' de 2 niveles con los atributos de la siguiente tabla y defina el dominio 'Amount' basado en el dominio 'Total', y los dominios 'ShortId' y 'Quantity' como se indica:

<i>InvoiceId</i>	Invoice Identifier	Id
<i>InvoiceDate</i>	Invoice Date	Date
<i>CustomerId</i>	Customer Id	Id
<i>CustomerName</i>	Customer Name	Name
<i>CustomerTotalPurchases</i>	Customer Total Purchases	Total
{ <i>InvoiceDetailId</i>	Detail identifier	ShortId = Numeric(2)
<i>ProductId</i>	Product Identifier	Id
<i>ProductDescription</i>	Product Description	Description
<i>ProductStock</i>	Product Stock	Stock
<i>ProductPrice</i>	Product Price	Price
<i>InvoiceDetailQuantity</i>	Invoice Detail Quantity (Cantidad llevada de Producto)	Quantity = Numeric(3)
<i>InvoiceDetailAmount }</i>	Invoice Detail Amount (Importe de la línea de detalle)	Amount

<i>InvoiceAmount</i>	Invoice Amount (Importe total de Factura)	Amount
----------------------	--	---------------

- En lugar de insertar uno a uno los atributos *ProductId*, *ProductDescription*, *ProductStock* y *ProductPrice*, puede seleccionarlos del View/**Work With Attributes** y hacer Drag & Drop a la estructura de la transacción.
- Asigne el nombre "Detail" al segundo nivel de la transacción.
- Observe que estando en el segundo nivel, al presionar '.' para nombrar a los atributos *InvoiceDetailQuantity* e *InvoiceDetailAmount*, ya le sugiere 'InvoiceDetail' como primera parte del nombre.
- Recuerde que puede mover un atributo de un nivel hacia el nivel padre, con botón derecho sobre el atributo y **Unindent**. La operación contraria se realiza con **Indent**. Pruebe subir y bajar atributos dentro del mismo nivel, seleccionando al o a los atributos y luego presionando botón derecho y **Move Items up** o **Move Items Down**.
- Salve la transacción definida. Vaya al folder "Tables" del FolderView y observe los nombres que GeneXus asigna por defecto a las tablas correspondientes a cada uno de los niveles. Observe los atributos que aparecen en cada una. La clave primaria de la tabla correspondiente al segundo nivel está conformada por dos atributos: *InvoiceId* e *InvoiceDetailId*. Observe los índices que creará GeneXus sobre esta tabla.

Observe el form web creado por defecto y compárelo con el siguiente que ha sido ligeramente modificado:

Form Web:

- Observe que *CustomerId* y *CustomerName* se han puesto en la misma fila (puede mover controles de lugar con Drag & Drop).
- *CustomerTotalPurchases* no aparece en el form (deberá borrar este control).
- Para eliminar una fila vacía de una tabla, posicione en una celda de la fila a ser eliminada, y seleccione **botón derecho/Table/Delete Row**.
- La forma de alinear controles es colocándolos en tablas. Para insertar una tabla, un camino posible es seleccionar: **Insert/Table**.
- Con el ícono **Show Borders**:  logra que se le muestren (en tiempo de diseño, no en ejecución) los bordes de las tablas donde están contenidos los controles del form. Cliquee este ícono para ver

los bordes de las tablas.

Observe que *ProductStock* no aparece en el grid del form mostrado. Deberá ocultarlo para que no se vea. Recuerde que para trabajar con las columnas de un grid, eliminar atributos o variables de las columnas, reordenar las columnas, etc., cuenta con:

- **Properties (F4):** Estando posicionado sobre cierta columna del grid, edita las propiedades de esa columna. Para la columna *ProductStock* cambie el valor de la propiedad 'Visible' de True a False.
- **Botón derecho sobre el grid/Columns:** Se muestra la lista de columnas incluidas en el grid (por defecto son los atributos correspondientes al segundo nivel de la transacción). Por este camino también es posible seleccionar *ProductStock*, editar sus propiedades y cambiar el valor de la propiedad 'Visible' de True a False.
- **Botón derecho sobre una columna del grid /Insert Column:** para agregar una columna.
- **Botón derecho sobre una columna del grid/Delete Column:** para borrar dicha columna.
- **Arrastrar y soltar (Drag & Drop) una columna:** para cambiarla de lugar en el grid.

Observe que los títulos de las columnas del grid son cortos (no coinciden con las descripciones que le dimos a los atributos). Posiciónese en la columna *ProductDescription* y con F4 edite sus propiedades. Vea que la propiedad **Title** tiene el valor **=ProductDescription.ColumnTitle** (es decir que está tomando dinámicamente el valor de la propiedad **ColumnTitle** del atributo *ProductDescription*). Si ahora se posiciona en la columna *InvoiceDetailQuantity* o en la columna *InvoiceDetailAmount* (que corresponden a atributos de la propia tabla INVOICEDetail), encontrará que sus propiedades **Title** tienen respectivamente los valores **=InvoiceDetailQuantity.ContextualTitle** y **=InvoiceDetailAmount.ContextualTitle** (es decir que están tomando dinámicamente el valor de la propiedad **ContextualTitle** del atributo).

¿Cómo lograr entonces cambiar el valor de los títulos de las columnas?

Opciones:

- Cambiar el valor de la propiedad **Title** de la columna, estáticamente.
- De acuerdo a la columna, cambiar la propiedad **Column Title** o **Contextual Title** del atributo correspondiente en la estructura de la transacción.

También observe en su form el Text Block 'Detail' que da nombre al grid. Vaya a la estructura de la transacción, posiciónese sobre el nombre del segundo nivel y edite sus propiedades. Cambie la propiedad **Description** y observe qué sucede con este text block en el form. ¿Se refrescó el Text Block con la nueva descripción que ingresó? ¿Por qué?

En el form mostrado arriba, el control atributo *InvoiceAmount* aparece en rojo y a la derecha. Para lograrlo:

- Observe que todos los atributos y sus descripciones (tanto los que corresponden al primer nivel, como los que corresponden al segundo nivel y están incluidos en un grid) se encuentran dentro de una tabla de 5 filas x 2 columnas. Inicialmente la tabla tenía 7 filas pero por lo solicitado en este práctico, ud. debe haber eliminado 2 filas. Puede observar esto con mayor claridad valiéndose de la ventana 'Document Outliner'.
- Posiciónese en la fila donde se encuentra el atributo *InvoiceAmount*. En la primera celda de esa fila encontrará el text block de título "Amount", Observe la propiedad **Caption** de este text block (vea que la descripción "Amount" es tomada dinámicamente de la propiedad **ContextualTitle** del atributo *InvoiceAmount*). Ahora posiciónese en la celda en la que se encuentra el atributo *InvoiceAmount*. Haga botón **derecho/Properties (o F4)**, y modifique la propiedad **Align** para lograr una alineación derecha del contenido de la celda.
- Inserte dentro de la celda en la que está el atributo *InvoiceAmount* una tabla de 1 fila x 2 columnas. Para ello puede seleccionar **Insert/Table** o hacer Drag & Drop desde la **Toolbox**.
- Mueva el control correspondiente al atributo *InvoiceAmount* a la celda derecha de la tabla que acaba de crear. Y al text block con texto "Amount" colóquelo en la celda izquierda de la tabla. Para modificar las propiedades de esos controles (texto y atributo) para que aparezcan en negrita y rojo respectivamente, seleccione cada control y modifique la propiedad que corresponda mediante de la ventana de Propiedades.

Reglas:

1. Hacer que la fecha de la factura sea la del día por defecto.
2. No permitir ingresar fecha vacía, ni mayor a la del día.

Ejecute las transacciones 'Product' e 'Invoice'

Luego del F5 no olvide estudiar el IAR para corroborar que se crearán las tablas exactamente como usted pensaba.

Ejecute y observe que se estén disparando las reglas definidas en cada transacción.

Pruebe el manejo de datos dentro del grid: intente insertar, modificar y eliminar líneas del grid de una factura.

Vuelva a GeneXus y agregue las siguientes **reglas** a la transacción 'Invoice':

3. Numerar las líneas de la factura en forma automática (a partir de aquí el atributo *InvoiceDetailId* no necesita ser ingresado por el usuario. Deshabilitarlo.)
4. Actualizar el total de compras del cliente.
5. Actualizar el stock del producto.
6. Controlar que el stock del producto no quede negativo.
7. Evitar que se ingresen más de 5 líneas en la factura (relacionado a fórmula 3, a continuación).

Fórmulas:

1. Calcular el importe de cada línea de la factura (*InvoiceDetailAmount*).
2. Calcular el total de la factura (*InvoiceAmount*).
3. Contar la cantidad de líneas ingresadas.

Construya la aplicación (F5).

¿Por qué hay que reorganizar? Observe detenidamente lo informado por el IAR. Deben eliminarse los atributos *InvoiceAmount* e *InvoiceDetailAmount* de las tablas correspondientes. ¿Por qué?

El listado de navegación de 'Invoice' arrojará 2 Warnings debido a la presencia de las reglas Add y Subtract. Se le informa que en modo update no podrán actualizarse los atributos *CustomerId* y *ProductId*. Explicación: Si se permitiera modificar el *ProductId*, la regla Subtract(*InvoiceDetailQuantity*, *ProductStock*) debería acceder al registro de PRODUCT correspondiente al *ProductId* que la línea tenía inicialmente, deshacer el subtract que se había efectuado previamente, y luego acceder al registro de PRODUCT correspondiente al *ProductId* actual en la línea y realizar el subtract. Es decir, se debería acceder a 2 registros distintos de PRODUCT. Esto no es posible, y por ello se deshabilita la posibilidad de modificar el atributo *ProductId*. Si se desea modificar el producto en una línea, habrá entonces que borrar la línea y volverla a insertar con el Id de producto deseado. Análogo es el caso de la regla Add y el atributo *CustomerId*.

Pruebe en ejecución todo lo que ha agregado.

Regla de asignación versus fórmula

Se desea visualizar en el form de 'Invoice' el saldo del cliente. Recuerde que:

$$CustomerBalance = CustomerTotalPurchases - CustomerTotalPayments.$$

En la transacción 'Customer' se le solicitó definir una regla para calcular el saldo del cliente como la diferencia entre sus compras y pagos. Y en la transacción 'Invoice' se solicitó la regla 4 para actualizar el total de compras del cliente con cada factura: add(*InvoiceAmount*, *CustomerTotalPurchases*).

Pruebe ingresar una factura para un cliente determinado y observe que se dispare la regla 4 definida, pero no se actualiza el saldo, *CustomerBalance*, de acuerdo al cálculo anterior. ¿Por qué?

Modifique lo necesario para que siempre que se consulte el atributo *CustomerBalance* desde cualquier objeto GeneXus, se calcule.

Antes del F5, reflexione sobre si deberá reorganizarse la base de datos. ¿Qué tabla deberá modificarse?

Normalización

Si en la estructura de la transacción 'Invoice' se agregan al primer nivel los atributos *CountryId* y *CountryName*, ¿qué ocurrirá con la tabla INVOICE? ¿Se almacenará el atributo *CountryId* como Foreign Key a la tabla COUNTRY? ¿O GeneXus entenderá que ese atributo deberá inferirse, al igual que *CustomerName*, del cliente *CustomerId*?

Pruébalo en GeneXus (agregue los atributos *CountryId* y *CountryName* al primer nivel de 'Invoice' y vea cómo al salvar, el ícono de la izquierda de *CountryId* en la estructura ya le indican antes del F5 si se tratará de una FK o si será un atributo inferido). Corrobore si el comportamiento es el que esperaba. Reflexione sobre lo realizado.

¿Y si quisiera que la factura tuviera un país propio, que deba ser ingresado asociado a la factura e independiente del país del cliente? Por ahora ud. no cuenta con las herramientas para resolverlo. Lo hará más adelante, cuando estudie el concepto de **subtipos**.

Avanzado: Transacción "Customer" [Opcional]

Compare el form que mostramos antes para la transacción 'Customer' con el siguiente (encuentre las diferencias):

The screenshot shows a web form titled "Customer" with a red error message: "Errorviewer: ctlError". The form contains several fields: "Id" (value: Customer), "Name" (value: CustomerName), "Address" (value: CustomerAddress), "Gender" (radio buttons for Female and Male), "Total Purchases" (value: CustomerTot), "Total Payments" (value: CustomerTot), "Balance" (value: CustomerBal), and "Country Name" (value: CountryId). The "Country Name" field is highlighted with a red border. At the bottom, there are three buttons: "Confirm", "Cancel", and "Delete".

Lo que se hizo aquí fue "disfrazar" el atributo *CountryId* de *CountryName*, para permitirle al usuario en ejecución trabajar con descripciones en lugar de tener que trabajar con códigos. Vea este nuevo form propuesto en ejecución:

The screenshot shows the "Customer" form in execution. The fields are: "Id" (value: 7), "Name" (value: Washington Trelles), "Address" (value: 18 de Julio 1234), "Gender" (radio buttons for Female and Male, with Male selected), "Total Purchases" (value: 0.00), "Total Payments" (value: 0.00), "Balance" (value: 0.00), and "Country Name" (value: u). A dropdown menu is open for the "Country Name" field, showing "United States" and "Uruguay". A "Confirm" button is visible at the bottom left.

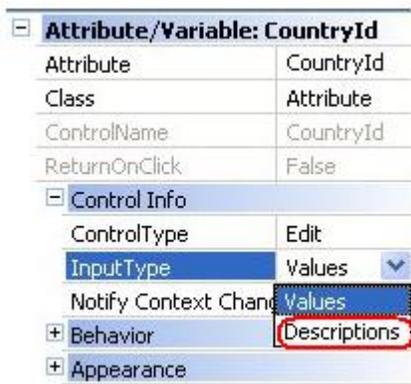
Para el usuario será más fácil recordar el nombre de un país, que su identificador. El sistema internamente sigue trabajando con el identificador.

¿Cómo se logra este comportamiento? Tenemos una forma particular (que aplicará a ese control en ese form) y otra general (que aplicará para todo form en el que se presente *CountryId*).

La forma particular es editando las propiedades **del control atributo *CountryId* en el form**. Hay que seleccionar el control en el form y modificar el valor de la propiedad **InputType asociada al control**. Por defecto esta propiedad tiene el valor 'Values' (que significa que el propio valor del atributo será ingresado) y habrá que seleccionar el valor 'Descriptions' (que significa que el valor entrado será tomado de un atributo descriptivo, pero el valor almacenado será el que corresponda al propio atributo).

Capturas consecutivas de propiedades a configurar para el control atributo *CountryId*:

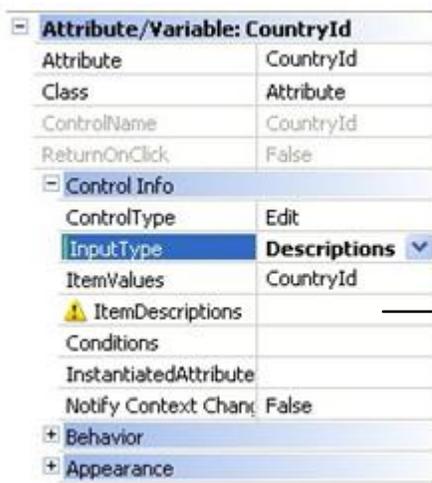
1)



Elegir valor: 'Descriptions'
En consecuencia:

- Se completa automáticamente la propiedad **ItemValues = CountryId**
- Se solicita completar la propiedad **ItemDescriptions**

2)



Completar propiedad **ItemDescriptions = CountryName**

3) Observe en la próxima captura todas las propiedades remarcadas. Una vez configuradas dichas propiedades para el control *CountryId* en el form de 'Invoice', queda implementado el "disfraz" para que el valor que el usuario entre en *CountryId* sea un nombre de país, pero el valor que quede almacenado en el atributo *CountryId* sea el identificador de país correspondiente:

Attribute/Variable: CountryId	
Attribute	CountryId
AutoSize	True
BackColor	<input type="checkbox"/> Window
CaseSensitive	False
Class	Attribute
Conditions	
ControlName	CountryId
ControlType	Edit
Display flat	False
Empty as null	Yes
Fill	True
FilterOperator	Begins with
Font	Microsoft Sans Serif, 10pt
ForeColor	<input type="checkbox"/> WindowText
Format	Text
Height	1row
InputType	Descriptions
InstantiatedAttributes	
IsPassword	False
ItemDescriptions	CountryName
ItemValues	CountryId
Notify Context Change	False
ReadOnly	False
ReturnOnClick	False
SortDescriptions	True
Suggest	Incremental

Observe que se ha configurado también el valor de la propiedad **Suggest**. Esta propiedad permite indicar si se desea sugerir al usuario a medida que éste va digitando, las descripciones almacenadas que coincidan con lo que viene digitando (para que elija uno de los valores ya existentes). En el form en ejecución mostrado arriba, se puede observar este comportamiento.

Observe que luego de definido este "disfraz", el control *CountryId* se visualiza en el form del largo de la definición de *CountryName* (ya que el usuario ingresará nombres de países en dicho control). Quite el control atributo *CountryName* del form, pues ya no se necesita tenerlo aparte.

En lugar de haber configurado estas propiedades a nivel del control en el form, podría haberse optado por la solución general, configurando las mismas propiedades pero asociadas al atributo (editando las propiedades del atributo en la estructura de la transacción). De hacer la definición de esta manera, en todo form en el que se inserte *CountryId* se manejará el "disfraz" (y el control atributo *CountryName* directamente no se incluirá).

Una funcionalidad muy similar a la vista aquí es la de utilizar un Dynamic ComboBox para el atributo *CountryId*.

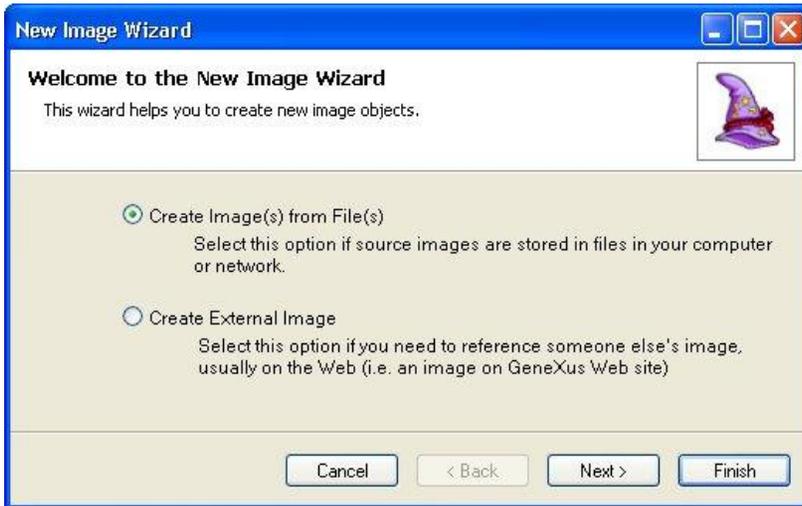
Pruebe el comportamiento.

Avanzado: uso de imágenes [Opcional]

Busque una imagen que identifique al sistema que está construyendo, para agregar al encabezado o pie de página de todas las páginas del sitio.

Las imágenes en GeneXus se agregan a la KB mediante la creación de objetos de tipo Image, y luego pueden insertarse en los forms.

Para realizar esto, abra el nodo Customization/Images del FolderView. Seleccione 'New Image' para insertar una nueva imagen en la KB. Esto le abrirá un Wizard que en el primer paso le preguntará dónde tiene el archivo con la imagen:



El siguiente paso le pedirá ingresar tanto el nombre que tendrá la imagen en la KB, como el path donde se encuentra:



Los check boxes tienen que ver con dos funcionalidades que no veremos en el presente curso. Déjalas desmarcadas. Seleccione Finish en el wizard y grabe la imagen que ha creado en la KB.

Una vez creada la imagen en la KB, podrá insertarla en el form que desee, como control Image. Abra la master page asociada a sus páginas (la master page por defecto) 'AppMasterPage' que se encuentra en el folder 'GeneralWeb'. Inserte un control imagen donde le resulte más conveniente, con la imagen recién creada.

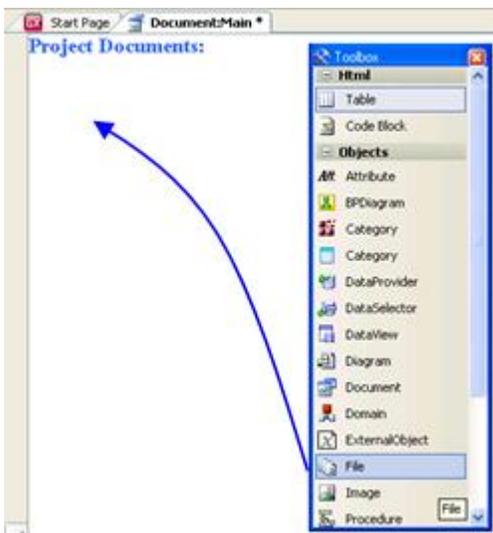
Haga F5 para probar la ejecución. Verá que GeneXus sólo necesita generar el objeto AppMasterPage y ya tiene el cambio en funcionamiento.

Avanzado: uso de documentos y archivos [Opcional]

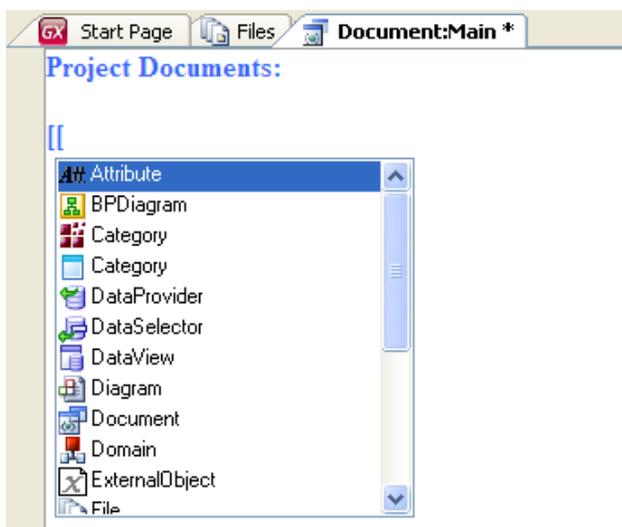
Observe que una KB GeneXus pretende centralizar todo el conocimiento relacionado a una aplicación. En ese entendido, tendrá la posibilidad de incorporar archivos de toda índole a la KB, como el plan director, un archivo word, archivos xsl, etc.. De esta manera no habrá necesidad de tener desperdigada la información relacionada al proyecto en diversos lugares. La KB centralizará todo. Los archivos se agregan en la KB de una forma similar a las imágenes. Vea el nodo 'Files' dentro de 'Documentation' del FolderView.

A su vez se cuenta en la KB con documentación estilo Wiki (una página principal: 'Main Document' que podrá encontrar bajo el nodo 'Documentation' del FolderView). En esta página podrá escribir información y agregar links a otros recursos de la KB (imágenes, páginas correspondientes a la documentación de cada objeto GeneXus, etc.).

Para agregar links en la página 'Main Document' a otros documentos, objetos, imágenes, etc., se utiliza la Toolbox (desde la misma se arrastra y suelta el objeto sobre la página):



Y otra forma es escribir dos paréntesis rectos y se abrirá una lista de objetos a seleccionar, para incluir link a su sección de documentación:



3. Consultas batch: procedimientos

A continuación proponemos realizar una serie de listados.

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

- Main program = 'True'
- Call Protocol = http
- Report output = Only to File

Y la siguiente regla:

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

LISTADOS SOLICITADOS:

1. Listado de clientes cuyos nombres estén incluidos en un rango determinado. Si todavía no vio nada de Web Panels deberá manejar un rango fijo en el Source (hardcoded). Si ya vio cómo implementar un Web Panel para pedirle datos al usuario, impleméntelo e invoque desde el Web Panel al listado solicitado (pase por parámetro al listado, el rango ingresado por el usuario en el Web Panel). Los datos que se desean mostrar en el listado para cada cliente son: Nombre, Dirección y Saldo (*CustomerBalance*).

Antes de ejecutar dedique unos minutos a estudiar el listado de navegación. Recuerde que el mismo le brinda la información de las tablas que serán navegadas, los filtros que serán aplicados, el orden en que serán devueltos los datos, etc. Es fundamental antes de probar en ejecución, prestar especial atención a lo aquí informado.

- Observe el orden que eligió GeneXus para el For each.
- ¿Está optimizado este procedimiento?
- Agregue la cláusula "order *CustomerName*" en el For each y observe qué cambia en el listado de navegación. Observe el índice utilizado en cada caso (si se agrega / saca la cláusula order).
- ¿A qué tablas está accediendo el For each?
- Ahora agregue el nombre del país del cliente a la información que quiere mostrarse, y observe la repercusión en el listado de navegación.

Debug

Active la configuración Debug para controlar el valor que van tomando los atributos durante la ejecución del listado:

- Agregue un breakpoint al comienzo del bloque for each.
- Agregue el atributo *CustomerName* en la ventana de visualización.
- Ejecute la aplicación.
- Observe el valor que va tomando el atributo *CustomerName*.

2. Listado de la factura al finalizar el ingreso de la misma en la transacción 'Invoice'.

- Recuerde detenerse a observar el listado de navegación resultante de la especificación de su reporte.
- Debió incluir un parámetro en el procedimiento para recibir el identificador de la factura que se debe imprimir. ¿Este parámetro lo declaró como el atributo *InvoiceId*, o definió una variable? ¿Qué cambios deberá hacer en su reporte si ahora declarara como parámetro la opción contraria a la que eligió?
- ¿Qué información aparecerá listada si desde la transacción 'Invoice' invoca este reporte con call en las reglas, y condiciona la invocación con el evento de disparo: "On BeforeInsert"?
- ¿Y si condiciona la invocación con el evento de disparo: "On AfterInsert"?
- Estaría bien condicionar el disparo con ¿"On AfterInsert Level *InvoiceDetailQuantity*"?
- Y con ¿"On AfterLevel Level *InvoiceDetailQuantity*? ¿Qué diferencia tiene este caso con "On AfterComplete"?

3. Listado de ventas por cliente, **incluyendo sólo los clientes que tienen facturas** (Corte de Control).

Cliente : CliXXX

Id. Factura	Fecha	Importe
1	01/01/08	\$150
2	01/02/08	\$1000

Total de ventas: \$1150

Cliente : CliYYY

Id. Factura	Fecha	Importe
15	01/01/08	\$700
11	01/02/08	\$300

Total de ventas: \$1000

- Observe cuidadosamente el listado de navegación. ¿Cómo sabe que implementó un corte de control?
- Si además de querer agrupar las facturas por cliente, totalizando el importe, desea que salgan ordenadas las facturas por número, ¿qué deberá hacer? Pruébalo. En el ejemplo la factura 11 debería salir antes que la 15.
- ¿Por qué GeneXus informa en el listado de navegación que el orden será el mismo para los dos For Eachs, incluso cuando ud. especificó un orden para el For Each externo y otro orden para el For Each anidado?

4. [Opcional: Doble Corte de control] Listado de ventas por fecha, agrupadas por cliente. Es decir, para cada fecha en la que hay facturas, se desea ver la información de las mismas (Identificador e Importe), pero agrupada por cliente.

Fecha: 01/01/08

Cliente: CliXXX

Id	Importe
2	\$150
3	\$10

Cliente : CliYYY

Id.	Importe
1	\$700
4	\$300

Total de ventas: \$1160

Fecha: 02/02/08

Cliente: CliYYY

Id	Importe
5	\$200

Total de ventas: \$200

- Este tema no aparece en las diapositivas del curso, pero sí en las notas incluidas al final del tema: Corte de Control. Lea cuidadosamente el ejemplo e implemente el listado que se le pide.
- ¿Cuántos For Eachs son necesarios para implementar un corte de control doble?
- ¿Dónde se especifica cada criterio de corte?

5. Listado de países que tienen asignados más de 5 clientes. Los datos a listar son Identificador y Nombre del país.

Recordar que las fórmulas pueden definirse tanto a nivel global (a nivel de un atributo en una transacción) como a nivel local (inline, en medio de cierto código). Observe con cuidado el listado de navegación. Si la fórmula local se encuentra dentro de un For Each, ¿los atributos mencionados de la fórmula participan en la determinación de la tabla base del For Each?

4. Data Selectors

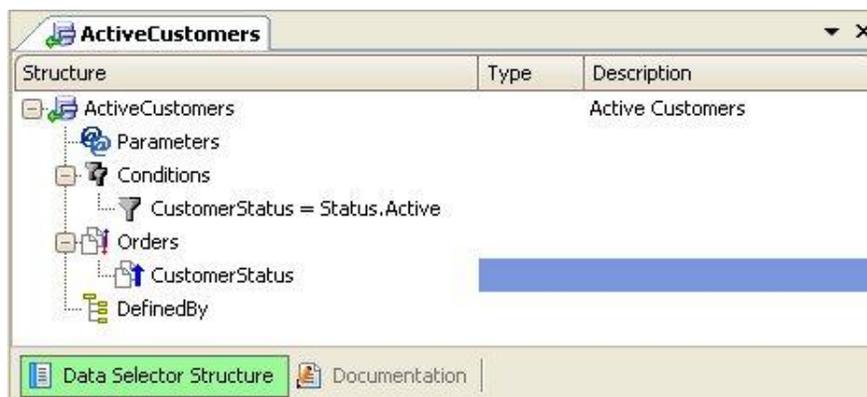
Agregue a la transacción 'Customer' el atributo *CustomerStatus*, que tomará uno de 3 valores: 'Active', 'OnHold' o 'Closed'. Para ello cree el dominio 'Status' enumerado (es decir, ingrese en la propiedad 'Enum values' del dominio los valores: 'Active', 'OnHold' o 'Closed'). Deberá reorganizarse la tabla CUSTOMER para agregar este atributo.

Puede agregar a la transacción 'Customer' la regla: "Default(*CustomerStatus*, Status.Active);" para que los clientes que se inserten mediante la transacción sean inicializados como activos. Pero, ¿qué valor asumirá el atributo *CustomerStatus* en los registros ya existentes de la tabla CUSTOMER (que se insertaron antes de definir esta regla)? Haga F5 y observe el Warning que le muestra el IAR. **Cancele** la reorganización, vuelva al atributo *CustomerStatus* de la transacción, y modifique su propiedad "Initial Value":



Esta propiedad permite especificar un valor para un atributo, **para que puntualmente al momento de efectuar la reorganización que agregue dicho atributo en la tabla, se asigne a los registros que ya estaban almacenados en la tabla, el valor especificado en el atributo creado.** Vuelva a hacer F5. Vea que desapareció la advertencia. Observe con el Scroll en el IAR que en la columna 'Takes value from' aparece el valor Status.Active para este atributo a ser creado.

Suponiendo que repetidamente a nivel de la aplicación se requerirá filtrar por clientes activos, tiene sentido **centralizar esta definición**, de manera tal de reutilizarla en todos los lugares en que se necesite. Para ello defina el siguiente Data Selector:



Ejemplos de utilización de clientes activos:

- Modifique el listado de clientes dentro de un rango, para que muestre sólo los activos, aprovechando la definición del Data Selector.
- Modifique el último listado (el 5 del práctico) que realizó, de manera tal que sólo liste los países con más de 5 clientes **activos**.
- Ahora, en lugar de utilizar el Data Selector en una fórmula inline, defina un atributo fórmula global en la transacción "Country" que cuente la cantidad de clientes activos de cada país. Muestre en el form de la transacción "Country" el atributo fórmula global definido.

Ventajas del uso de Data Selectors: Si la definición de lo que es un cliente activo se modifica, alcanzará con modificarla en un sólo lugar: en el Data Selector, y todos los objetos, atributos, fórmulas que lo utilizan, serán impactadas sin necesidad de tener que ir a buscarlas y modificarlas manualmente.

5. Actualizaciones no interactivas a la BD

1. Realizar la **modificación de diseño necesaria en la transacción "Product"** para que cada producto en lugar de tener un solo precio, tenga un histórico de precios según la fecha.

Es decir, la estructura de la transacción "Product" quedaría:

<i>ProductId</i>	Product identifier	Id
<i>ProductDescription</i>	Product description	Description
<i>ProductStock</i>	Product Stock	Stock
{ <i>ProductPriceListDate*</i>	Product Date	Date
<i>ProductPriceListPrice }</i>	Product Price	Price

Asígnele el nombre "PriceList" al segundo nivel.

El form Web quedará:

Como consecuencia del cambio de diseño anterior en la transacción 'Product', ya no es posible inferir el atributo *ProductPrice* en el segundo nivel de la transacción 'Invoice' (porque *ProductPrice* ya no pertenece a la tabla extendida de la tabla INVOICEDTAIL).

Se debe quitar el atributo *ProductPrice* del segundo nivel de la transacción 'Invoice', definir un nuevo atributo (*InvoiceDetailProductPrice*) y calcular el valor de ese atributo (hay que encontrar el precio del producto de acuerdo a la fecha de la factura). Resuelva esto definiendo el atributo *InvoiceDetailProductPrice* como fórmula global Max.

- Definir un procedimiento para generar una nueva lista de precios para los productos. El procedimiento debería recibir por parámetros: fecha de aumento y porcentaje de aumento. Nuevamente sucede que dependiendo de si ya vio cómo implementar un Web Panel, podrá implementar uno para pedir al usuario estos datos e invocar al procedimiento enviándole lo ingresado por el usuario por parámetro. En caso contrario, coloque valores de fecha de aumento y porcentaje de aumento en el Source de manera fija (hardcoded) para probar. Para cada producto deberá generar un nuevo registro en el histórico de precios, con la fecha de aumento y el precio que resulte de aplicar el porcentaje de aumento al último precio del producto anterior a la fecha de aumento.

¿Qué forma de actualización (Business Component o comando new) eligió y por qué? Recuerde el uso de las fórmulas inline.

Realice también la actualización utilizando la opción contraria a la que eligió.

Ejecute 2 veces el procedimiento, con la misma fecha de aumento. ¿Qué sucederá?

En el caso de actualización con BC realice el manejo de errores y observe los mensajes obtenidos al intentar insertar 2 veces con la misma fecha.

- Definir la transacción 'Bill', para registrar los recibos efectuados a los clientes por concepto de facturas. Su estructura será:

<i>BillId*</i>	Bill Id	Id
<i>BillDate</i>	Bill Date	Date
<i>CustomerId</i>	Customer Id	Id
<i>CustomerName</i>	Customer Name	Name
<i>BillInvoicePeriodStartDate</i>	Bill Invoice Period Start Date	Date
<i>BillInvoicePeriodEndDate</i>	Bill Invoice Period End Date	Date
<i>BillAmount</i>	Bill Amount	Amount

Se desea implementar lo siguiente: para un período de fechas ingresado por el usuario, se desea que un proceso batch sumalice para cada cliente el importe de todas sus facturas y genere un recibo para dicho cliente, con la fecha del día, el rango de fechas considerado y el importe total a cobrarle.

En el material teórico de Business Components se presenta un web panel que solicita un rango de fechas al usuario; defina ese mismo web panel e invoque desde el web panel a un procedimiento que reciba por parámetro los datos ingresados por el usuario en el web panel y resuelva lo solicitado.

Utilice un **Data Provider** para cargar los recibos antes de ingresarlos a la base de datos.

¿Y si sólo se desea se desea ejecutar esto para los clientes activos?

A efectos de puro aprendizaje, puede modificar su Data Provider cargando manualmente un recibo que usted sepa violará la integridad referencial (cliente inexistente). También puede probar agregar alguna regla msg y alguna error en la transacción 'Bill' y en el Data Provider cargar recibos que cumplan las condiciones de esas reglas y ver qué pasa al intentar insertar esos registros.

6. Subtipos

Se desea representar que cada factura pertenece a un país independientemente de que el cliente la factura tiene su país ¿Qué modificaciones debe realizar? Impleméntelas.

Observar que la factura pertenecerá a un país, por ejemplo Uruguay, y el cliente de la factura podrá pertenecer a otro país distinto, por ejemplo United States.

7. Interfaz de usuario: Pattern Work With y Web panels

Se requiere una consulta interactiva vistosa para cada actor: 'Customer', 'Country', 'Product' e 'Invoice' que ofrezca consultar todos los registros, o bien filtrar los registros que cumplan con varias condiciones posibles (a continuación se indicarán cuáles), con varios órdenes posibles para la consulta, y poder seleccionar un registro para modificar sus datos, o bien eliminarlo o insertar un registro nuevo. El objetivo es ofrecer consultas vistosas, uniformes, consistentes y bien funcionales.

Consulta interactiva de clientes:

Al realizar la consulta se desea visualizar de cada cliente:

- su identificador
- su nombre
- su dirección
- su status ('Active', 'OnHold', 'Closed')

Debe ser posible filtrar por: nombre de cliente y/o dirección.

Debe ser posible obtener el resultado de la consulta ordenada por: nombre de cliente o dirección.

La consulta debe ofrecer paginación: Observe en el tab **Work With** del selector **Patterns** de la transacción 'Customer' que en la propiedad 'Page' del nodo **Selection(Work With Customers)** aparece el valor <default>. ¿De dónde se toma ese <default>? Podría fijar un valor e independizarse del default, o modificar lo asignado al default (10).

Recuerde que en la ventana Knowledge Base Navigator, bajo Preferences, se encuentra el nodo Patters/Work With. Aquí están configurados los <defaults> a aplicarse en todos las **Instancias** correspondientes a los **Work With** de todas las transacciones (Work With Pattern Instance). En particular deténgase en las propiedades de los nodos 'Template', 'Grid' y 'Standard Actions'.

Observe que la propiedad 'Page' del nodo 'Grid' tiene como valor un tipo de datos enumerado que se creará automáticamente en la KB una vez aplicado el pattern Work With a alguna transacción.

Se desea poder seleccionar un cliente y modificar sus datos o eliminarlo. También debe ser posible insertar un cliente nuevo. Observe que estos son los modos ofrecidos por defecto (acciones estándar). Puede probar variar cuáles modos ofrecer (modificando los <default> o directamente desde la instancia de la transacción).

Observe cómo al grabar la transacción se generará en el FolderView, bajo la transacción un nodo WorkWithCustomer que corresponde a la propia definición de la instancia del Pattern para esa transacción y subordinados todos los objetos que el Pattern creará para construir el work with.

Ejecute y pruebe lo realizado hasta el momento.

Agregue al grid de selección la cantidad de facturas del cliente. Pruebe su solución.

Observe en ejecución que el atributo *CustomerName* es el que tiene el link para visualización de los datos de ese cliente, con la posibilidad de actualizar o eliminar el cliente, así como ver sus facturas. ¿Por qué el pattern eligió ese atributo? Observe la instancia del pattern y deduzca la razón.

Modifique los atributos que se visualizan al seguir ese link: elimine de la visualización los atributos correspondientes a *Total Purchases*, *Total Payments* y *Country Id* y reordénelos para que salgan en el orden que ud. prefiera. Deshabilite la posibilidad de insertar una nueva factura mediante el Tab (Invoice) y pruébelo.

Se requiere además que la consulta interactiva de clientes, contenga un botón bajo el grid, que llame a otra página que muestre lo siguiente:

- una lista (grid) con todos los países, siendo posible seleccionar uno y a partir de él que se muestre otra lista (grid) con todas los clientes activos de ese país, y luego eligiendo uno de estos clientes, que se muestren todas las facturas del cliente elegido.

Defina **consultas interactivas de países, productos y facturas**, para sentir lo fácil que es obtener una aplicación vistosa en segundos.

8. Consultas dinámicas

Queremos consultar por país y cliente, los totales por concepto de facturas, junto con la cantidad de facturas.

Luego, queremos agregar como eje, las fechas de factura y poder pivotear la tabla, de manera de poder ver los totales por país, cliente y fecha, y luego por país, fecha y cliente.

Luego de probar lo anterior en un objeto Query, integre la consulta a la aplicación web, creando un web panel que muestre el resultado de la consulta en una pivot table.

9. Workflow

Modelaremos y ejecutaremos un workflow pequeño para ver en la práctica esta funcionalidad.

Como paso previo, defina la transacción 'PaymentMethod':

Name	Type	Description	Formula	Nullable
PaymentMethod	PaymentMethod	Payment Method		
PaymentMethodId	Id	Payment Method Id		No
PaymentMethodDescription	Description	Payment Method Descri...		No

Agregue a la transacción 'Invoice' los atributos *PaymentMethodId* y *PaymentMethodDescription*.

Cree un diagrama de proceso de nombre 'SaleProcess' y arrastre a la transacción 'Invoice' como primer tarea interactiva. Observe que se ha definido automáticamente el dato relevante InvoiceId.

Cuando una vendedora ingrese una venta (a través de la transacción 'Invoice'), ingresará el cliente, los productos solicitados, las cantidades, la forma de pago solicitada por el cliente, grabará la venta y ahí terminará la primer actividad. Luego esa venta deberá ser evaluada por un supervisor (quien evaluará de qué cliente se trata, el monto, la forma de pago), y el supervisor deberá aceptar o denegar la venta. Para realizar esta tarea el supervisor ejecutará un web panel de nombre 'Authorization'.

Defina un web panel de nombre 'Authorization' que reciba por parámetro al atributo *InvoiceId*. Incluya en el form del web panel los atributos con la información de la invoice y su customer. Este web panel ofrecerá al supervisor 2 botones: "Authorize" y "Refuse".

Este web panel servirá al supervisor para tomar la decisión de autorizar o denegar la venta. Si el supervisor presiona "Authorize", cargue una variable con valor 1 y si en cambio el supervisor presiona "Refuse" cargue a la misma variable con 0. Resuelva que el valor asignado a la variable "se vea" en el diagrama de proceso, para que a continuación de esta tarea interactiva (web panel 'Authorization') pueda insertar una condición que evalúe si el supervisor autorizó la venta o no.

En caso de haber sido autorizada la venta, que el flujo siga hacia un web panel que diga solamente "venta autorizada" y tenga un botón para cerrarlo. En caso de haber sido rechazada la venta, que el flujo siga hacia un web panel que diga solamente "venta rechazada" y tenga un botón para cerrarlo.

Proceda a ejecutar este workflow (si lo desea previamente cree roles y asígnelos a las tareas).

10. Apéndice: Sql Server Express 2005

Si utiliza Sql Server Express 2005 como DBMS que manejará la base de datos de su aplicación, aquí indicamos cómo crear un usuario y password con permisos para crear base de datos. De esta manera cuando desee configurar el acceso a la base de datos que tendrá la aplicación generada por GeneXus, podrá dejar que GeneXus cree la base de datos sin tener que crearla usted.

1. Ejecute el Microsoft SQL Server Management Studio Express que tiene instalado en su máquina.
2. Observe que allí aparece el valor Server name que deberá utilizar en GeneXus cuando se le pida esta información.
3. Conéctese con Windows Authentication
4. Acceda a Security/Logins
5. Botón derecho/New Login...
6. Ingrese el nombre del login (y recuérdelo)
7. Seleccione SQL Server authentication e ingrese la Password correspondiente (también recuérdela).
8. Deshabilite el check box 'Enforce password policy' (en nuestro caso estamos creando usuario y password de prueba, no necesitamos protección especial de nuestra password).
9. En la ventana de la izquierda 'Select a page' posicione sobre 'Server Roles' y allí marque 'dbcreator'.
10. Grabe.
11. Listo. Ahora podrá configurar en GeneXus la creación de la base de datos con el nombre que ud. desee, indicando el Server name que obtuvo del paso 2 y el usuario y clave de los pasos 6 y 7.