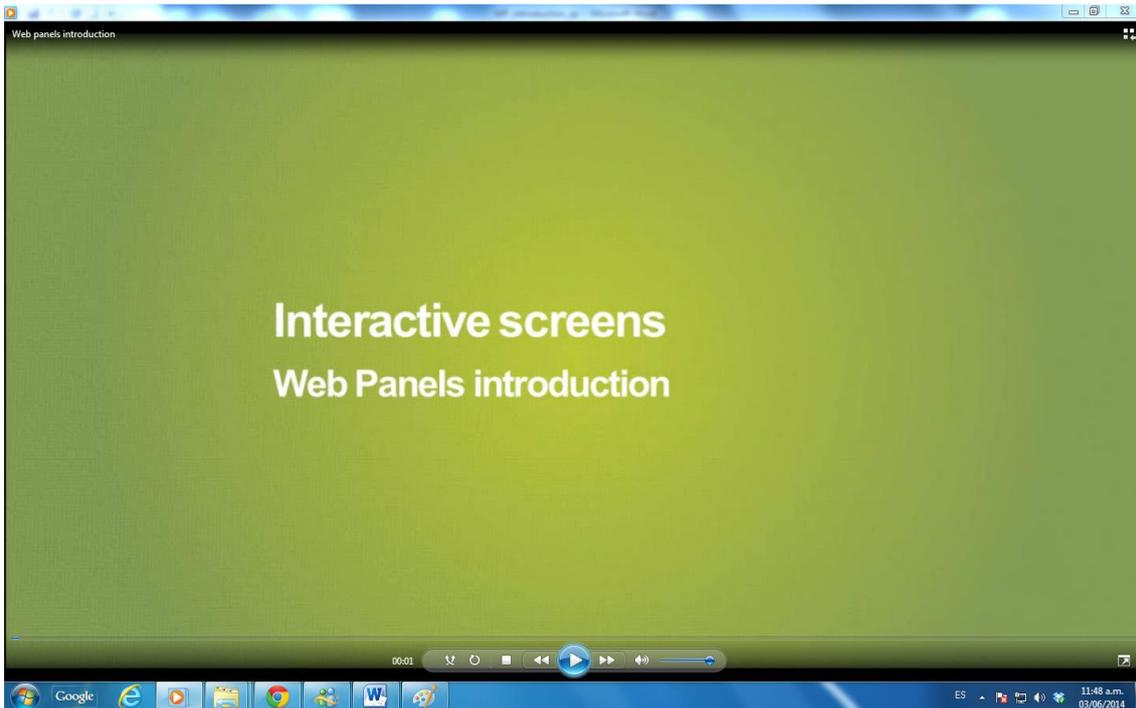
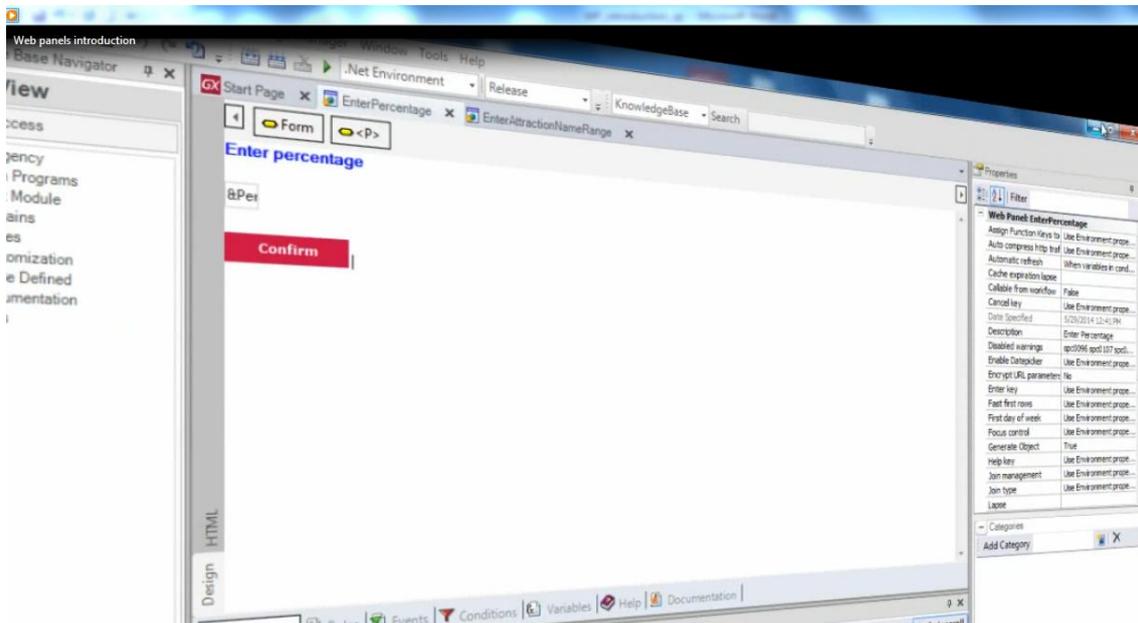


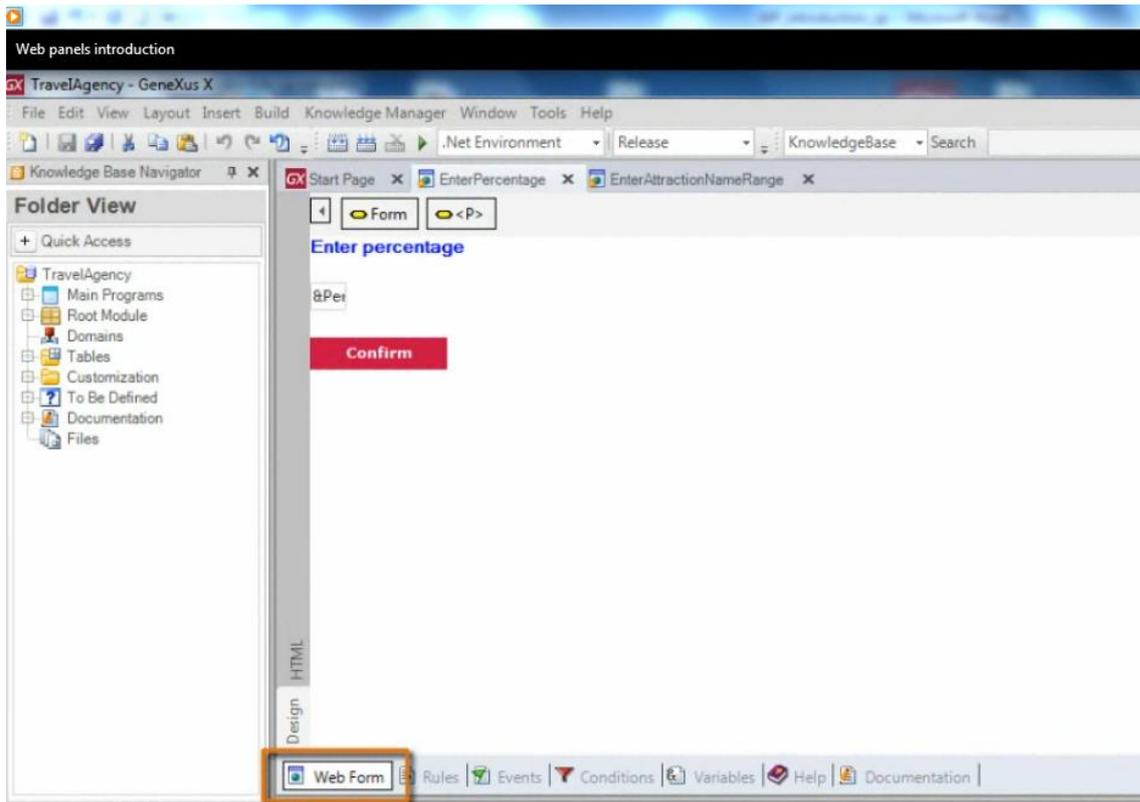
Pantallas interactivas (Objeto Web Panel)



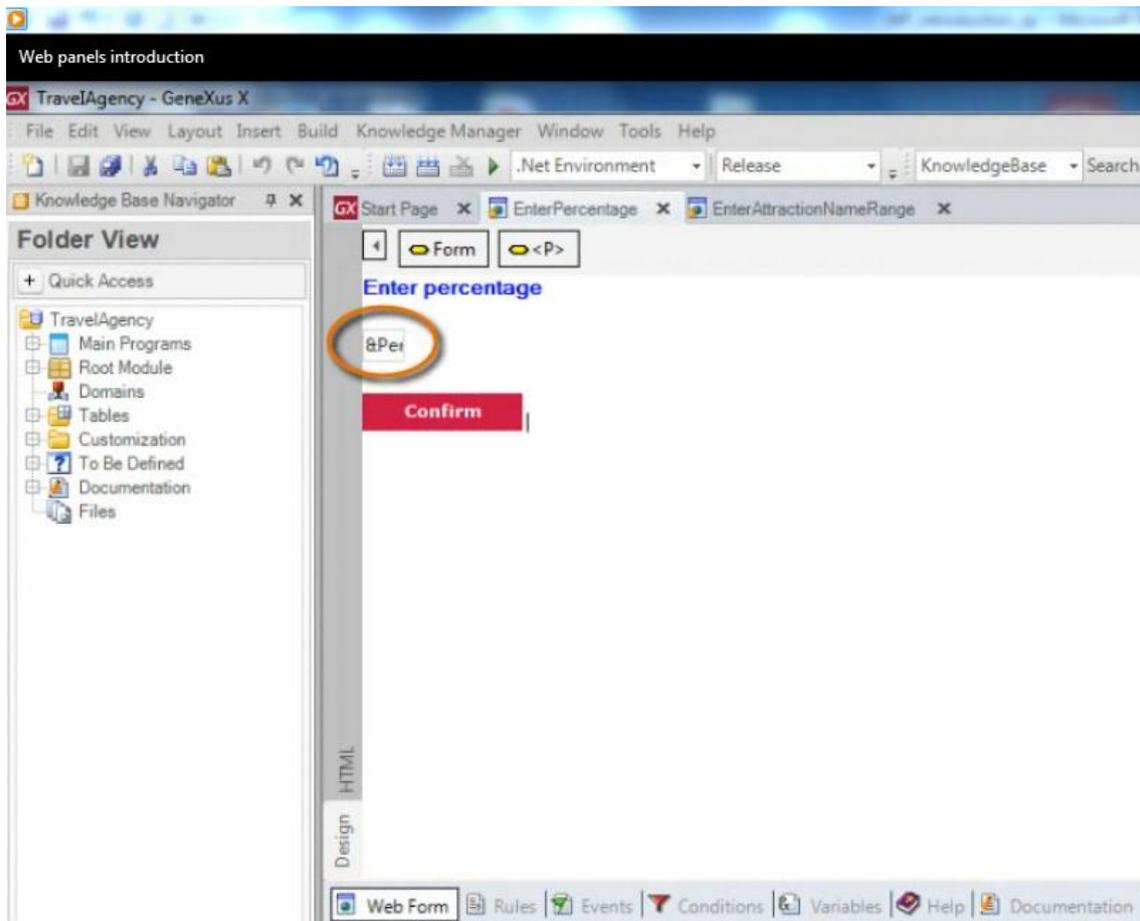
El web panel es el objeto más flexible que provee GeneXus.



Como ya hemos visto en algunos ejemplos que hemos mostrado, todo web panel ofrece un web form, que es una página web que nos permite diseñar y ofrecer variadas funcionalidades.

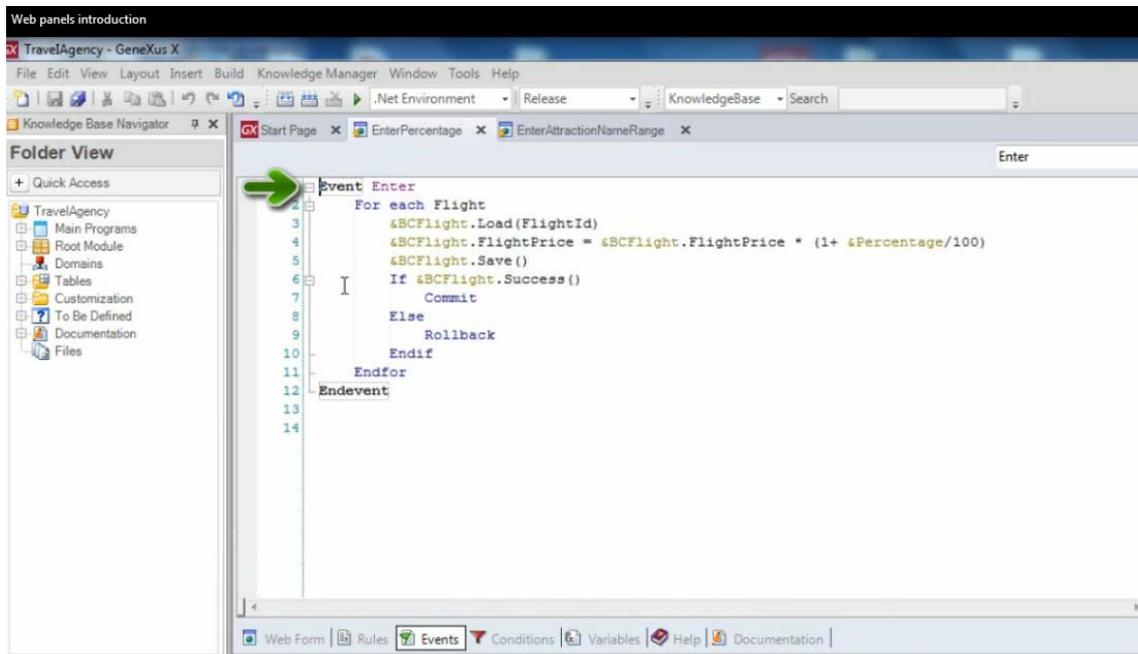


En este ejemplo habíamos visto que por el hecho de incluir una variable en el web form

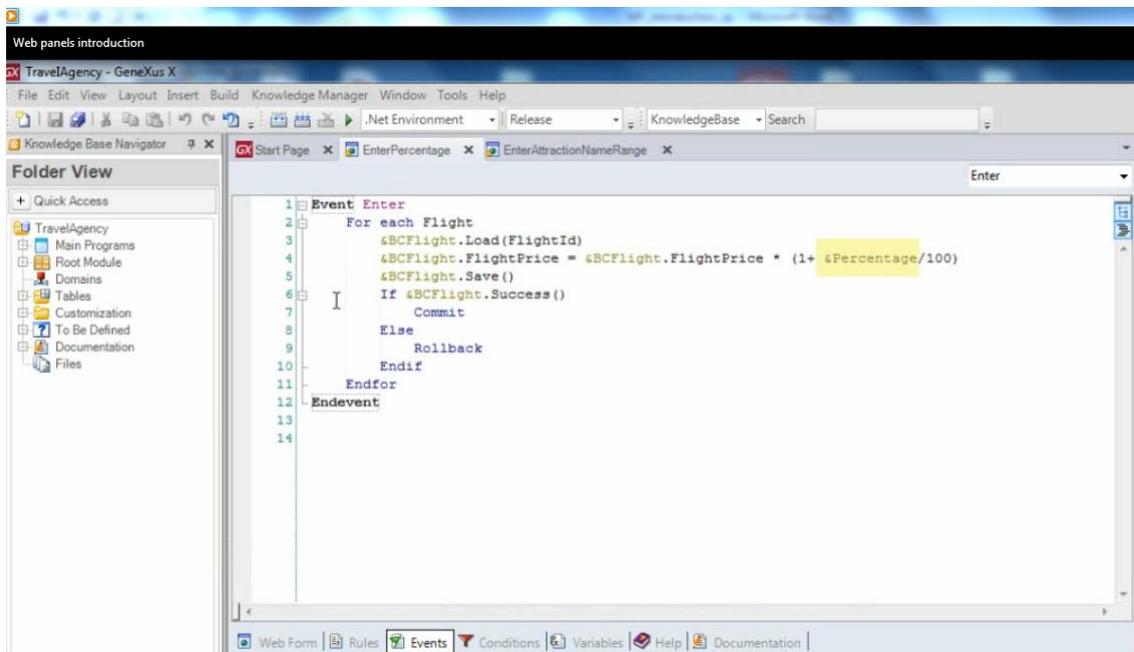


la misma quedaba habilitada para que el usuario le ingrese algún valor.

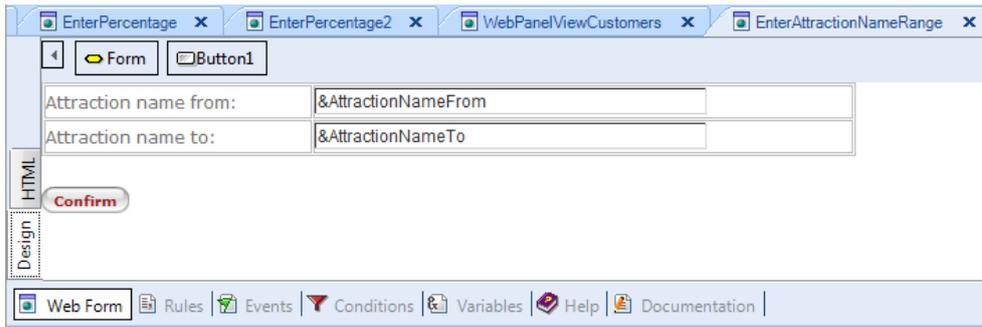
En particular esta variable esperaba un porcentaje de aumento y al presionar el usuario el botón Confirm, se ejecutaba el evento asociado al mismo



procesando el aumento de precios de todos los vuelos, aplicándoles el porcentaje de aumento digitado por el usuario en la variable

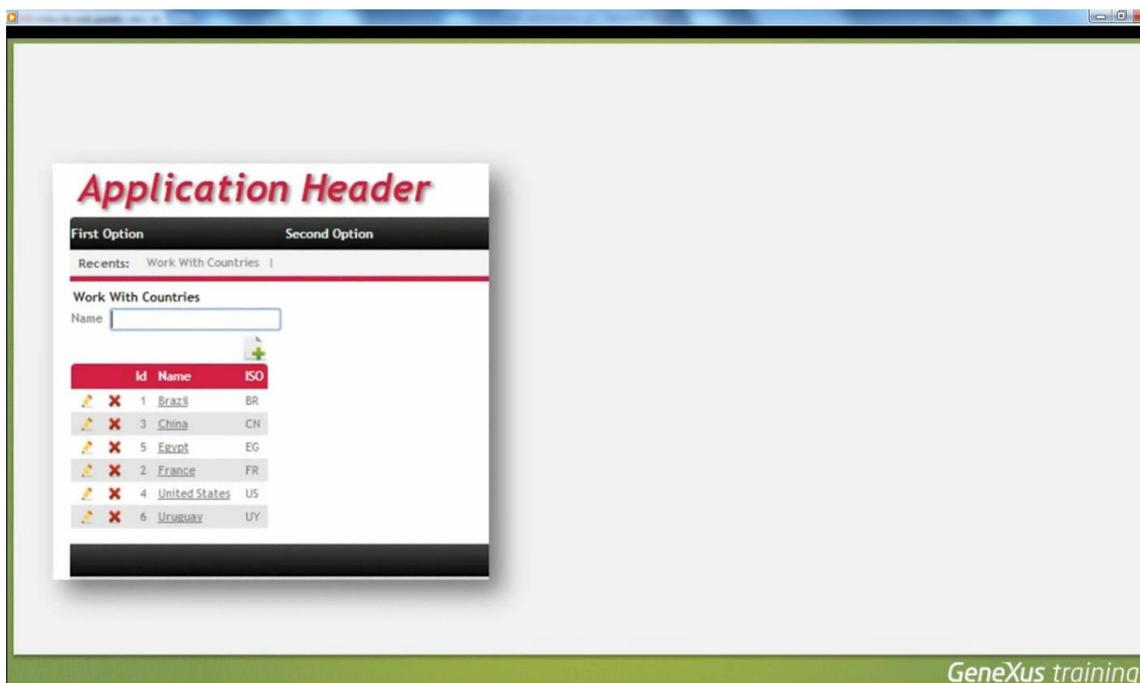


Este otro web panel que hemos resuelto, es muy similar al anterior, en el sentido que contiene también variables -en este caso 2- y un botón:



Ambas variables, se presentan en ejecución, habilitadas para el ingreso de datos por parte del usuario... y al presionar el usuario el botón Confirm , invocamos en el evento que tiene asociado , 1 procedimiento al cual le enviamos por parámetro las variables, y el mismo las recibe e imprime en formato pdf, las atracciones cuyos nombres están incluidos en el rango recibido.

Pero además de definir web panels para aceptar valores en variables y luego utilizar esos valores para ejecutar alguna acción determinada, los web panels nos permiten como objetivo primordial, implementar y ofrecer a los usuarios consultas **interactivas** a la base de datos.



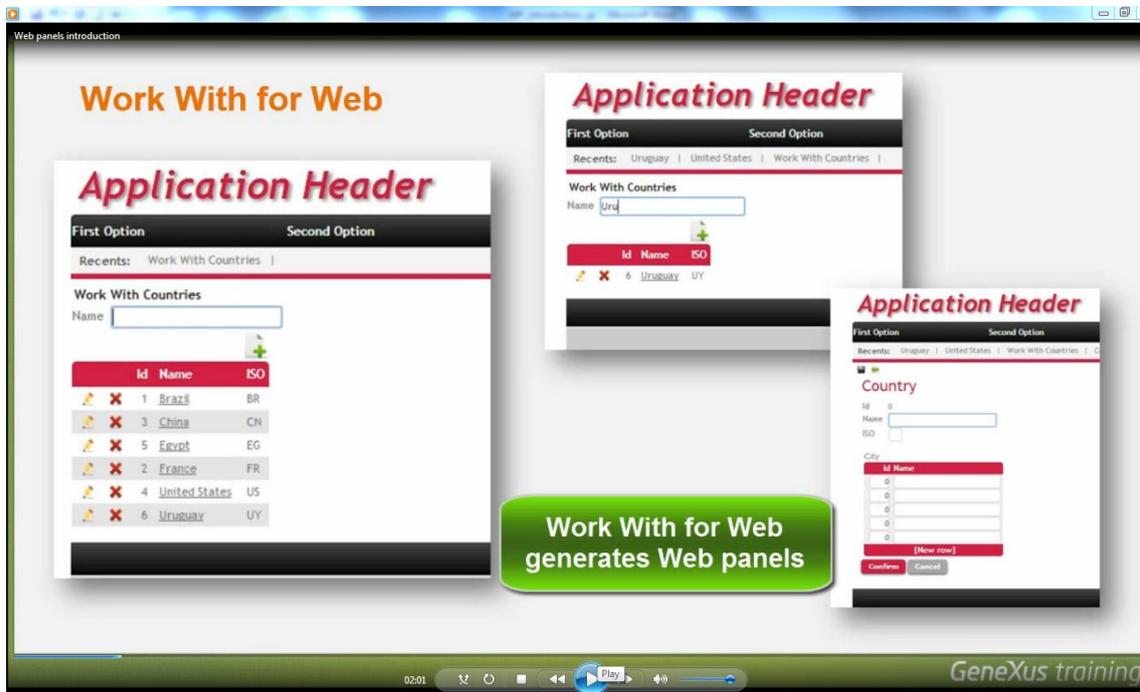
El término “**interactivas**” se refiere a que el usuario puede ingresar en la página del web panel una y otra vez distintos valores –**en variables**- y consultar a continuación datos de la BD que concuerden con esos valores ingresados, utilizándolos como filtros.

También luego, sobre los datos consultados que se muestran en una grilla, el usuario puede realizar distintas acciones!!

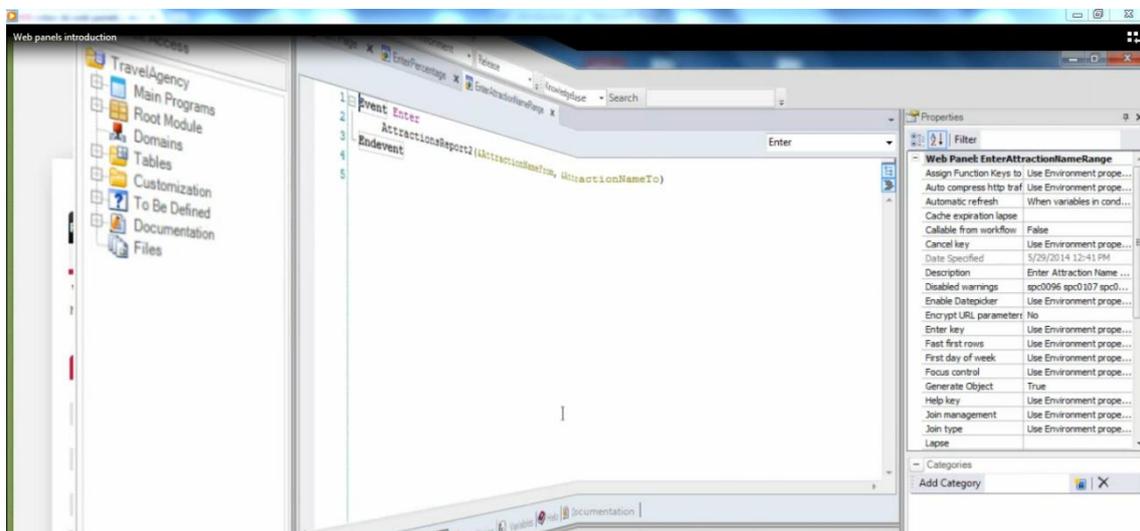
¿Hemos visto ya algo así?

¡¡ Sí !! Lo generado por el patrón “work with for web” cuando lo hemos aplicado a transacciones !!

Porque el patrón “work with for web”, genera web panels !!!



Vamos ahora a definir paso a paso un web panel que implemente 1 consulta interactiva... y luego lo compararemos con un web panel Work With for Web.



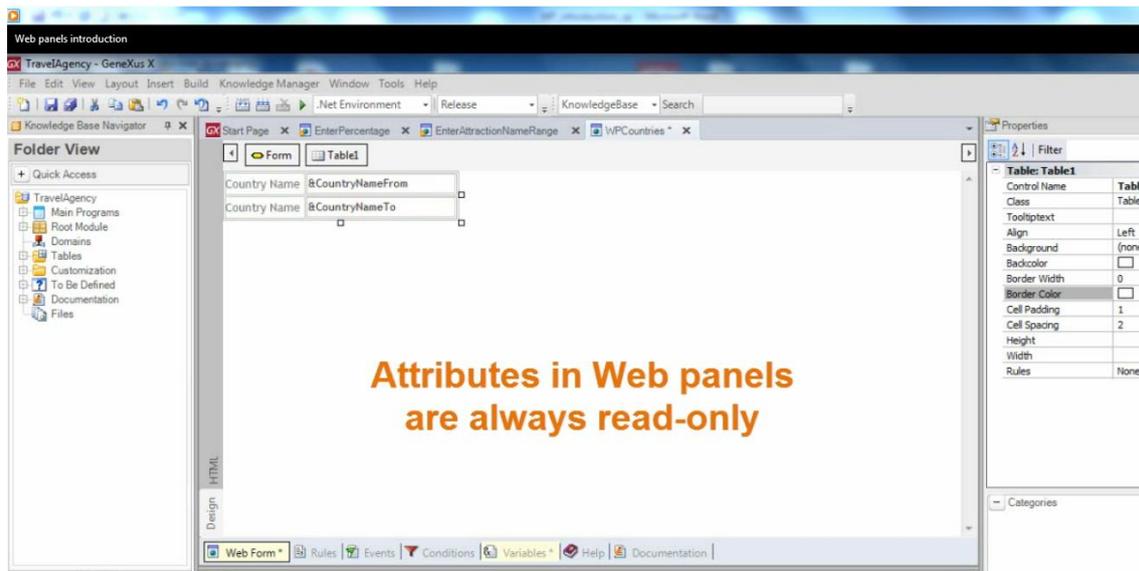
Vamos a crear 1 objeto de tipo web panel , le damos el nombre: WPCountries, y queda abierto en su sección: web form.

Vamos a la sección de variables y definimos 2 variables: Una de nombre CountryNameFrom y otra de nombre CountryNameTo, ambas del mismo tipo que el atributo CountryName .

Ahora volvemos al web form e insertaremos ambas variables. Seleccionamos: Insert / Variable, las elegimos, y vemos que se agregaron automáticamente en una tabla de 2 filas y 2 columnas, para que queden alineadas las variables con sus títulos.

Bien. Sabemos que las variables estarán habilitadas para el ingreso de valores en ellas.

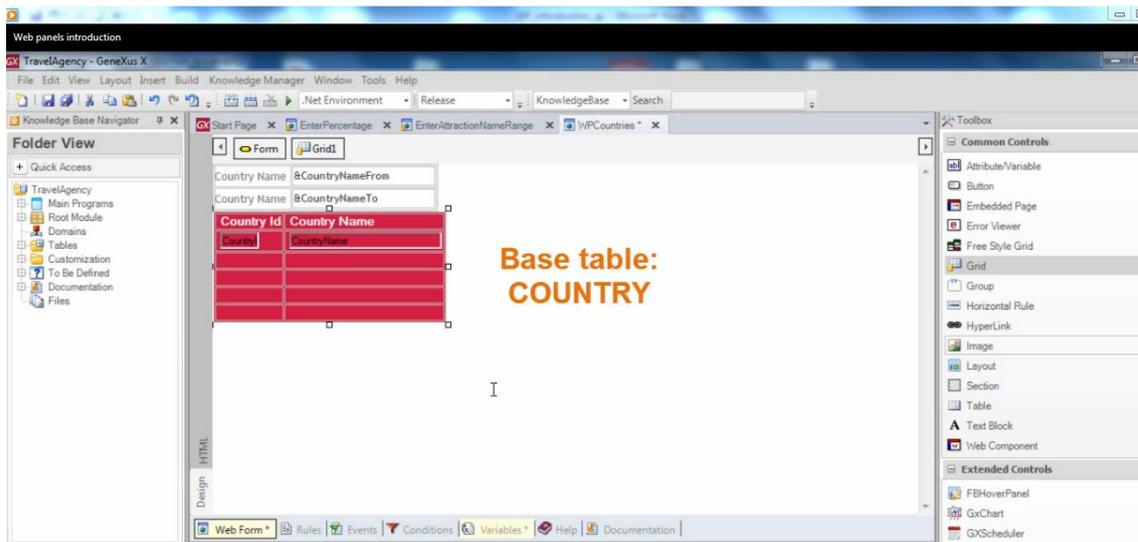
Y así como sabemos eso, debemos saber que **los atributos en los web panels, siempre serán read-only.**



Es decir, serán indefectiblemente de salida, para mostrar su contenido almacenado.

Bien. Ahora vamos a insertar en el web form, una grilla con los atributos cuyos datos deseamos consultar. Seleccionamos Insert/grid y elegimos incluir en la misma a: CountryId y CountryName.

GeneXus va a determinar para esta grilla, navegar cierta tabla física, por el hecho de contener atributos. Decimos que esta grilla va a tener una tabla base que lógicamente por los atributos involucrados es: COUNTRY.



Además, si observamos las propiedades del grid, vemos que contamos con esta propiedad del grid llamada “base transaction” que ofrece lo mismo que en el For each... o sea, que el objetivo de esta propiedad es que indiquemos en ella, el nombre de la transacción cuya tabla física asociada queremos navegar.

Debemos saber que en un web panel, una grilla que contiene atributos es como un For each. No escribimos el For each, pero está implícito. Por eso decimos que la grilla tiene tabla base.

Lo que configuremos en la property “**base transaction**” le indica a GX la tabla base, y los atributos que se pueden incluir en la grilla, son los de la tabla base deseada y su tabla extendida también.

Vamos a ejecutar la aplicación para ver el funcionamiento de este web panel. F5, ejecutamos el web panel...

Y vemos que las variables están habilitadas para que ingresemos algo en ellas. Vemos también que en la grilla se muestran todos los países almacenados en la tabla física COUNTRY.

Bien. Hasta el momento no hemos hecho nada para relacionar lo que el usuario ingrese en las variables con respecto la navegación que se hace a la tabla física COUNTRY, por eso se muestran todos los países almacenados.

Vamos a definir los filtros ahora.

En las properties de la grilla, podemos definir **Conditions**, que es exactamente lo mismo que escribir cláusulas where en For each. Así que en las conditions ponemos:

```
CountryName>=& CountryNameFrom;
CountryName<=& CountryNameTo;
```

Vamos a ejecutar nuevamente la aplicación. Ahora pongamos desde la A hasta la Z... y vemos que se cargan en la grilla todos los países que tenemos almacenados.

Probemos poner ahora hasta la E... y vemos que se cargan menos países. France ya no se carga, Uruguay tampoco ni United States, etc.

Volvamos a GeneXus y observemos el listado de navegación que se generó para el web panel.

Vemos que hay una navegación a la tabla COUNTRY y se están aplicando los filtros que definimos en las conditions del grid. También vemos que la consulta se está realizando, ordenando los registros por CountryId, que es la clave primaria de la tabla COUNTRY.

Si quiséramos especificar otro orden, por ejemplo por: CountryName para que salgan en la grilla los países ordenados alfabéticamente, contamos con la property del grid: order. Vamos a configurar para la propiedad order el atributo: CountryName.

Ejecutamos nuevamente. Ponemos filtros.. y vemos que se muestran los países incluidos en el rango especificado, ahora ordenados en la grilla por nombre de país.

Volvamos a observar el listado de navegación correspondiente a este web panel. Ahora vemos el orden por CountryName... y además, vemos que contamos con una mayor optimización al ejecutar la consulta, porque ahora no se recorren todos los registros verificando si cada uno sirve o no, por estar incluidos en el rango solicitado.... sino que ahora, como se consulta la tabla ordenada alfabéticamente , es posible posicionarse en el primer registro de país cuyo nombre es mayor o igual al valor solicitado, e ir listando los registros consecutivos que se tienen ordenados, hasta que se llegue al primer registro que ya no cumpla con el rango.

Siempre es interesante observar los listados de navegación que GeneXus esboza para los objetos que definimos, ya que los mismos nos permiten corroborar la lógica que GeneXus determinó y verificar si coincide con lo que pretendíamos.

O sea, no es sólo con la ejecución que podemos ver el resultado, sino que estos listados de navegación, nos permiten ver la lógica generada, que es lo más importante, es decir, un esbozo del código que GeneXus generará... y analizando eso podemos detectar tal vez que GX determinó hacer algo que no pretendíamos.. y analizar si definimos algo errónamente..

Bien. Vamos a ejecutar ahora el “Work With Countries”, que se había generado automáticamente .Ahora sabemos que se trata de un objeto web panel 😊

Vemos que en este caso no hay 2 variables para ingresar el rango inicial y final, sino una variable sola, pero eso es un detalle que se puede hacer a gusto.. lo importante es ir asociando los conocimientos de web panels que estamos adquiriendo, con lo que generó el patrón.

Work With Countries

Name



	Id	Name
 	1	<u>Brazil</u>
 	2	<u>China</u>
 	3	<u>France</u>

Vemos que este web panel, además de ser vistoso, y de haberse generado rápidamente con tan solo hacer un click al aplicar el patrón.... es bien completo, en el sentido que además de ofrecer filtrar y consultar ciertos países, nos ofrece también la posibilidad de editar un país, o eliminarlo.. así como también insertar físicamente un país nuevo.

Nosotros todo eso lo podemos definir también y veremos cómo se hace... **pero nuestra sugerencia es aprovechar el uso del patrón "work with"** -en este caso estamos hablando del que es "for web", pero vale también para el que es "for smart devices" - **para ganar productividad, cuando lo que necesitemos sea obtener este tipo de consultas interactivas, con formato standard!**

Entonces: Consideramos importante **conocer lo que genera el patrón work with.. y conocer bien el objeto web panel de GeneXus**, por varios motivos:

- 1) Porque nos pueden solicitar consultas o usos que puedan resolverse con web panels, pero no exactamente lo que brinda el patrón work with for web... así que especialmente en esos casos deberemos tener buenos conocimientos acerca de este objeto.

y..

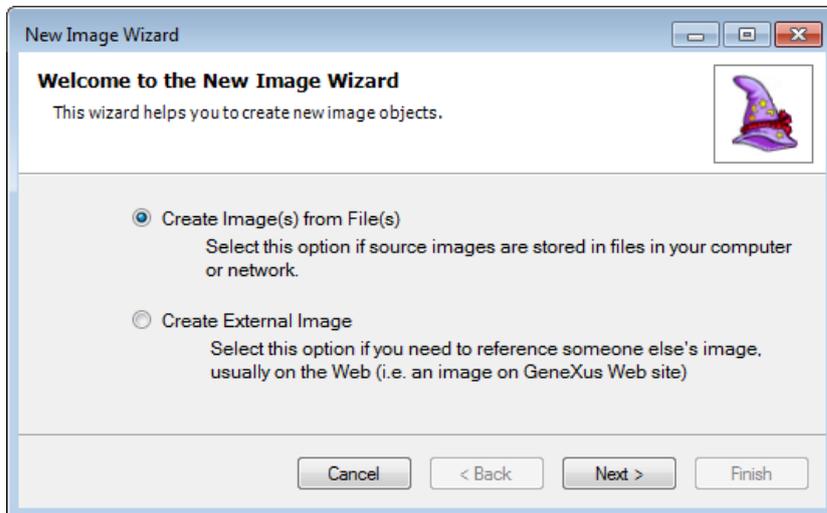
- 2) Para poder abrir un objeto web panel generado automáticamente por el patrón y entender y conocer sus partes.

Bien. Continuemos con la definición del web panel que habíamos comenzado a implementar. Procederemos a agregar dentro de la grilla, una imagen que aparezca para cada línea, con el objetivo de que el usuario pueda presionar dicha imagen para cierta línea, cuando desee editar los datos de dicho país.

Vamos a definir una nueva variable y la agregamos como columna del grid . Movemos la variable para que quede al final y le quitamos el título a la columna .

La imagen que queremos cargar en la variable, debemos insertarla en la base de conocimiento. Una forma de hacerlo es yendo a customization , elegimos Images, que es donde se listan todas las imágenes actualmente almacenadas en la KB, y elegimos insertar una nueva

La queremos cargar de un archivo...



La elegimos..

Le dejamos como nombre Update...

Y grabamos...

Y ahora tenemos que asignarle la imagen a la variable... el lugar más apropiado para realizar esta asignación es el evento Start, ya que es el primer evento que se ejecuta al abrir el web panel en tiempo de ejecución y es el evento que justamente se utiliza para definir inicializaciones.

Dentro del evento start entonces, digitamos ampersond, elegimos la vble &UpdateImage... punto... elegimos el método: FromImage... y le pasamos como parámetro el nombre de la imagen en la base de conocimiento, o sea: Update.

Presionemos F5 para ver el resultado de lo que hemos definido

Ejecutamos nuestro web panel... y vemos la imagen en cada línea.

Así hemos comprobado que a la variable se le asignó la imagen de inmediato al ejecutarse el web panel, y por estar incluida como columna en el grid, la imagen salió para cada una de las líneas.

Nos está faltando ahora asociarle 1 acción a la imagen. Lo que queremos es que cuando el usuario haga clic sobre ella, se abra la transacción Country, mostrando los datos del país de esa línea, y en particular el form esté en modo de edición, para que el usuario pueda modificar los datos del registro mostrado y grabar los cambios.

Bien, vamos a programar el evento click asociado a la variable &UpdateImage.

En la sección de eventos del web panel , posicionamos el cursor debajo del evento start y seleccionamos **Insert/Event**.

Este diálogo muestra todos los controles incluidos en el form y para cada control, se nos van mostrando los eventos posibles de ser ejecutados para ellos. Así que ubicamos la variable &UpdateImage y de todos los eventos posibles, elegimos click.

Se nos inserta la sintaxis que define al evento y vamos a codificar dentro, lo que necesitamos efectuar..

Seleccionamos: Insert / Object.... e insertamos la transacción Country, que es la que deseamos invocar:

```
Event &UpdateImage.Click
      Country()
Endevent
```

Pero habíamos dicho que queríamos que se abra la transacción en modo de edición, con el país en cuestión, instanciado..

Así que debemos pasarle por parámetro a la transacción: el texto fijo "UPD", que significa UPDATE... y GeneXus lo reconoce y entiende que tiene que abrir la transacción en modo de edición, y el segundo parámetro que tenemos que enviarle a la transacción es el identificador del país de la línea: CountryId .

```
Event &UpdateImage.Click
      Country("UPD", ContryId)
Endevent
```



Ahora vayamos a la transacción Country, ya que estos parámetros los tenemos que recibir, declarando una regla parm en la sección Rules de la misma.

Y observemos que ya hay una regla declarada parm, que es precisamente la que íbamos a declarar nosotros ... ¿por qué ya está definida?

Porque el patrón work with for web lo hizo cuando lo habíamos aplicado a esta transacción i!

```
/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
  Parm(in:&Mode, in:&CountryId);

  CountryId = &CountryId if not &CountryId.IsEmpty();
  NoAccept (CountryId);
  NoPrompt (CountryId);
}
```

```
/* Generated by Work With Pattern [End] - Do not change */
```

Observemos que el 1er parámetro es una variable de nombre &Mode. Lo antecede el operador “in”, indicando que es un parámetro de entrada... y algo bien importante de destacar es que **en todas las transacciones, la variable &Mode ya está predefinida... y la misma se usa solamente para ser recibida como parámetro, ya que su función es reconocer automáticamente y exclusivamente los valores de tipo carácter:**

“INS” = INSERT mode

“UPD” = UPDATE mode

“DLT” = DELETE mode

“DSP” = DISPLAY mode

Observemos ahora el 2do parámetro. Es una variable : &CountryId, también de entrada...

Y esta regla

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

Define que el valor recibido en la variable se le asigne al atributo .



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

Es decir, es como un filtro que se establece... siempre y cuando no tenga la variable recibida valor vacío . Y en el único caso que mandamos sin valor este parámetro es cuando queremos llamar a la transacción, para insertar un nuevo registro (ya que si queremos editar o eliminar, debemos mandar la clave primaria del registro que queremos editar o eliminar).

Bien, hemos visto entonces, que la transacción Country está siendo invocada desde nuestro web panel WPCountries, y también desde el web panel Work With Countries... y lógicamente la regla parm definida por el patrón nos sirve, porque estamos conociendo más sobre los web panels y ejercitando prácticas típicas, que el patrón las genera rápidamente, y estamos viendo en detalle cómo se definen.

Presionemos F5, ejecutemos nuestro web panel WPCountries y ahora hagamos click en esta imagen para este país y vemos que se abre la transacción en modo edición con el país instanciado. Modificamos su nombre y ponemos: USA. Grabamos. Y vemos que se vuelve a cargar el grid, con la consulta mostrando la modificación que hicimos.

Vamos a nombrarlo nuevamente: United States .

Esta misma implementación que hemos definido para ofrecer editar la información de los países, la podemos repetir para implementar la eliminación de los mismos. Definiríamos otra variable, la agregaríamos al grid, le asignaríamos una imagen representativa, y en el evento click de la variable, pondríamos esta misma invocación a la transacción Country

```
Event &UpdateImage.Click  
Country("UPD", ContryId)  
Endevent
```



Botón derecho + Open...

con la única diferencia, de que en vez de enviarle a la transacción el parámetro "UPD", mandaríamos "DLT" ya que en la transacción, el parámetro recibido en la variable: &Mode entendería y abriría la transacción ofreciendo eliminar el registro cuya clave primaria fue recibida.

Ahora vamos a agregar un botón en el web form, que ofrezca insertar un país nuevo, tal como lo ofrece el Work With Countries generado automáticamente .

Arrastramos de la toolbox un botón aquí .. y podríamos asociarle una imagen o en este caso cambiaremos el caption del botón. Le ponemos: "Insert Country".

Hacemos doble click sobre el botón y vemos que por defecto este botón tiene asociado el evento enter. Todo botón que se agrega tiene por defecto asociado el evento enter... pero vamos a crear un evento nuevo, de usuario, asociado a este botón ..y le damos el nombre: Insert Country.

Bien.. ¿Cuándo presionen este botón, y se ejecute el evento asociado.. qué debe suceder?

Se debe abrir la transacción Country, ofreciendo el form listo para un nuevo ingreso..

Así que debemos invocar a la transacción Country.

Seleccionamos: Insert / Object... seleccionamos a la transacción Country .. y abrimos y cerramos un par de paréntesis. El primer parámetro que enviaremos a la transacción es: "INS" y el 2do parámetro será un cero .. ya que en este caso queremos completar el parámetro esperado con: vacío. No tenemos un valor para enviar sino que se autonumerará cuando ingresen los datos del país y confirmen.

Presionemos F5 para probar esta nueva funcionalidad que hemos agregado. Ejecutamos nuestro web panel, presionamos el nuevo botón y vemos que se abre la transacción Country, lista para que hagamos un ingreso .

Bien. Ahora supongamos que nos piden ver para cada país, la cantidad de ciudades que tiene registradas. Este dato no lo tenemos en un atributo, entonces vamos a definir una nueva variable y la agregaremos en el grid como columna .

¿Cómo podremos resolver que para cada línea que se carga con un país de la tabla COUNTRY, se calcule y muestre su cantidad de ciudades?

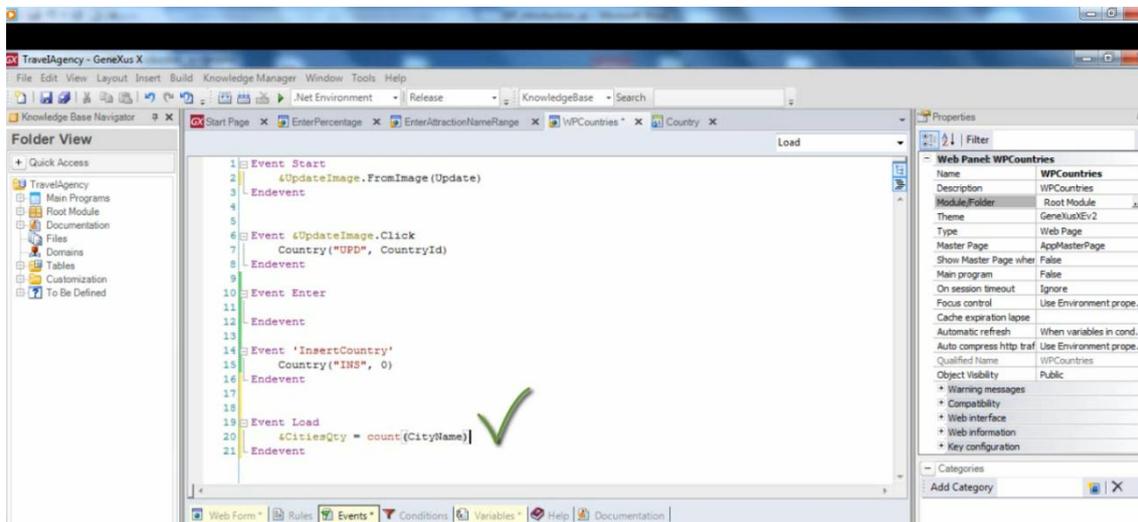
Vayamos a la sección de eventos... y observemos este evento propio de los web panels: el evento Load .

Load significa “cargar” y a continuación explicaremos cuántas veces se ejecuta este evento según en el caso y en qué momento exacto.

Como este web panel tiene 1 solo grid, el cual tiene tabla base y ya hemos explicado que es COUNTRY, el evento Load, se ejecutará tantas veces como líneas se carguen en el grid con países.

Para ser exactos, debemos saber que se ejecutará **1 vez por cada registro de la tabla navegada, justo antes de cargar cada registro como línea en el grid.**

O sea que como este evento se ejecutará para cada línea del grid, justo antes de mostrar los valores de los atributos en la línea, podemos en este mismo momento, asignarle a la variable &CitiesQty el resultado de una fórmula inline que cuente las ciudades del país que está a punto de cargarse y llegamos a tiempo para que se muestre dicha cuenta



Ya hemos aprendido en otros videos, que esta fórmula determinará la tabla física a navegar por los atributos que figuren dentro del paréntesis

```
1 Event Start
2     &UpdateImage.FromImage (Update)
3 Endevent
4
5
6 Event &UpdateImage.Click
7     Country("UPD", CountryId)
8 Endevent
9
10 Event Enter
11
12 Endevent
13
14 Event 'InsertCountry'
15     Country("INS", 0)
16 Endevent
17
18
19 Event Load
20     &CitiesQty = count (CityName)
21 Endevent
```

así que la fórmula contará en la tabla COUNTRYCITY

```
1 Event Start
2     &UpdateImage.FromImage (Update)
3 Endevent
4
5
6 Event &UpdateImage.Click
7     Country("UPD", CountryId)
8 Endevent
9
10 Event Enter
11
12 Endevent
13
14 Event 'InsertCountry'
15     Country("INS", 0)
16 Endevent
17
18
19 Event Load
20     &CitiesQty = count (CityName)
21 Endevent
```

COUNTRYCITY

pero no todas las ciudades que hayan registradas, cualquiera sea su país...porque el evento Load se está ejecutando N veces mientras se va navegando la tabla COUNTRY. O sea que hay una navegación que se está haciendo en el contexto en el cual estamos.

Entre COUNTRY y COUNTRYCITY está en común el atributo CountryId, por lo tanto en el evento Load tendremos cada vez, 1 registro navegado de país, y teniendo su CountryId, aplicará 1 filtro automático al contar las ciudades. Este filtro hará que se cuenten las ciudades del CountryId que se está navegando en ese momento y a punto de cargar como línea.

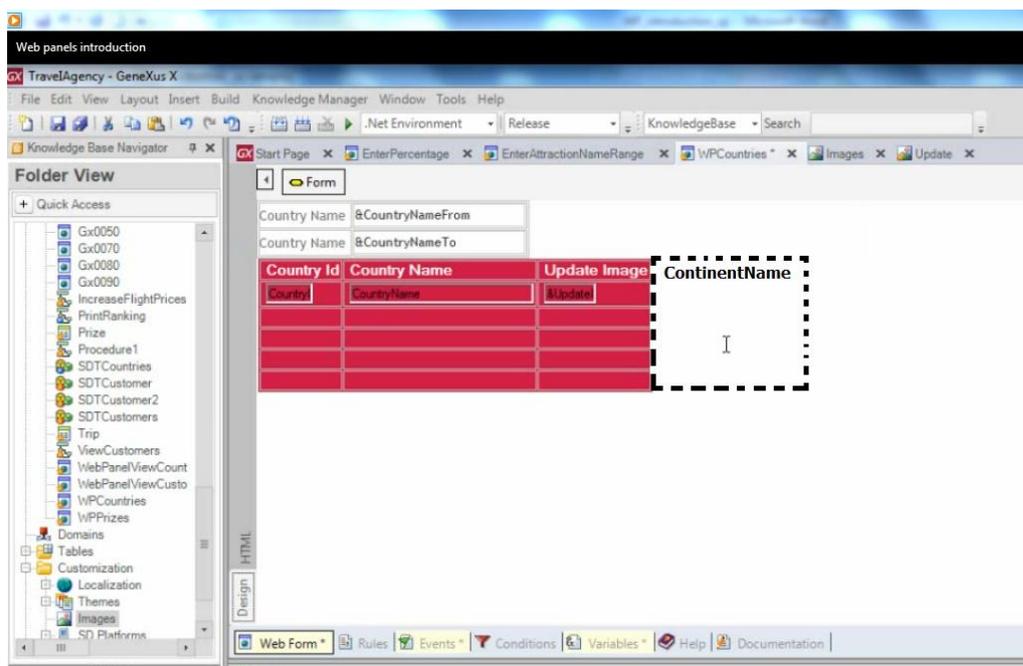
Veamos entonces este resultado en ejecución. Presionemos F5.

Indicamos el rango, mostramos todos los países y vemos efectivamente la cantidad de ciudades para cada país.

Bien. Volvamos a observar el form de nuestro web panel. Habíamos dicho que en la grilla podíamos mostrar todos los atributos que quisiéramos de la tabla extendida de la tabla base navegada. En nuestra aplicación, por el momento, en la tabla extendida de COUNTRY no hay otras tablas incluídas más que ella misma. Pero si nos pidieran registrar para cada país, el continente en el cual se encuentra:



crearíamos una transacción Continent, e incluiríamos en la transacción Country los atributos ContinentId y ContinentName .Y en nuestro web panel en la grilla podríamos incluir a ContinentName por ejemplo:



También en las **conditions del grid** , podemos filtrar por atributos que se encuentren en la tabla extendida de la tabla base del grid... **y en la propiedad order** podemos indicar un atributo de la tabla extendida también.

Ahora cuestionémonos lo siguiente:

¿Cuántas veces se ejecutará el evento Load en un web panel que no contiene atributos, sino solamente variables?

Si el web panel no contiene atributos, ni grids con la propiedad Base Trn configurada, como por ejemplo los web panels que hicimos que piden 1 porcentaje o un rango de nombres de atracciones, GeneXus no tiene cómo elegir una tabla física a navegar. Y debemos saber que en esos casos, el evento Load se ejecuta 1 sólo vez!

Ya que no hay una navegación determinada de forma automática y carga de los registros navegados.

Los web panels ofrecen mucho más! Pero lo enseñaremos en otros videos.

Por ejemplo un web panel puede tener varios grids... tanto paralelos, cada cual con su navegación independiente... **como anidados**, ya que existe un tipo de grid diferente del standard, llamado grid free style, que permite diseñar 1 línea con un diseño más libre y vistoso que no contiene columnas estructuradas y esa línea se repetiría mostrando ese diseño para cada registro navegado ... e incluso 1 grid free-style puede contener dentro otro grid free-style dentro o standard.

COMMIT