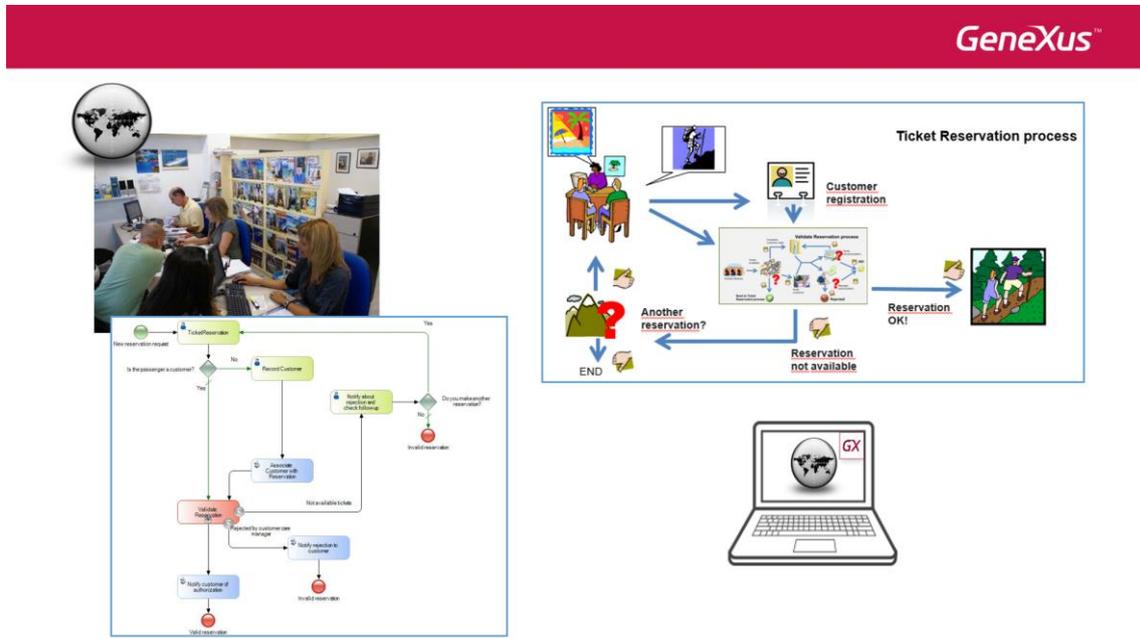


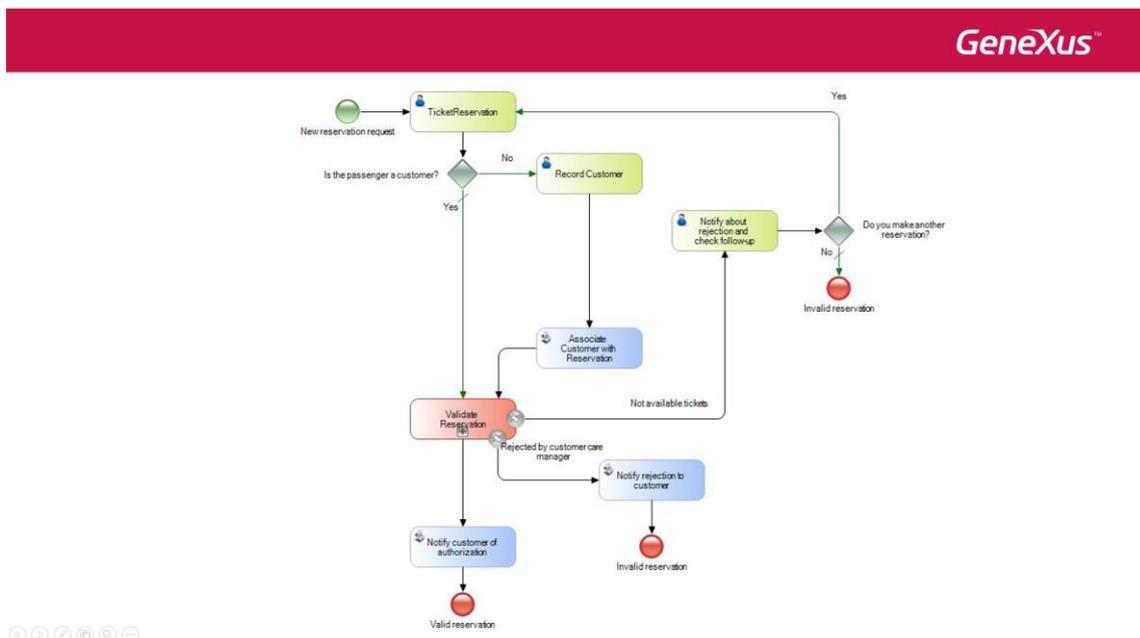
Convirtiendo el diagrama en una aplicación funcional

En videos anteriores hemos modelado el proceso de reserva de pasajes de la Agencia de Viajes mediante un diagrama de procesos de negocios, hemos visto una breve introducción al estándar BPMN y las facilidades que GeneXus nos proporciona para construir diagramas que cumplan con dicho estándar.



Ahora haremos que el modelo se convierta en una aplicación funcional, utilizando objetos GeneXus asociados a los símbolos de nuestro diagrama.

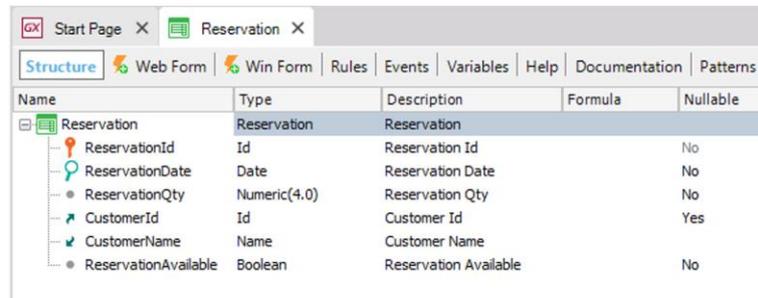
Vamos a comenzar con el diagrama de procesos de reservas de pasajes, que teníamos definido.



La primera actividad TicketReservation es del tipo interactiva y su función es registrar la reserva del ticket.

Para cumplir con esto, podríamos utilizar un objeto transacción con la siguiente estructura:

Reservation transaction object

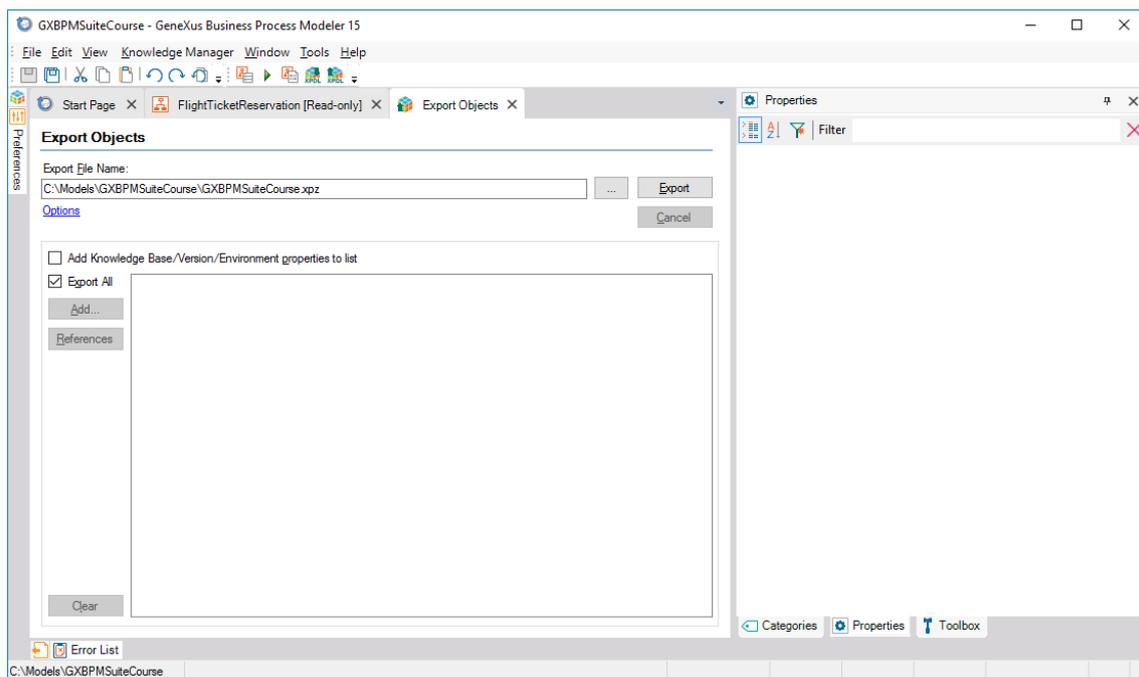


Name	Type	Description	Formula	Nullable
Reservation	Reservation	Reservation		
ReservationId	Id	Reservation Id		No
ReservationDate	Date	Reservation Date		No
ReservationQty	Numeric(4,0)	Reservation Qty		No
CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
ReservationAvailable	Boolean	Reservation Available		No

Aquí vemos que el atributo ReservationId es el identificador de la transacción y tenemos también un atributo ReservationAvailable, para indicar si la reserva está disponible o no.

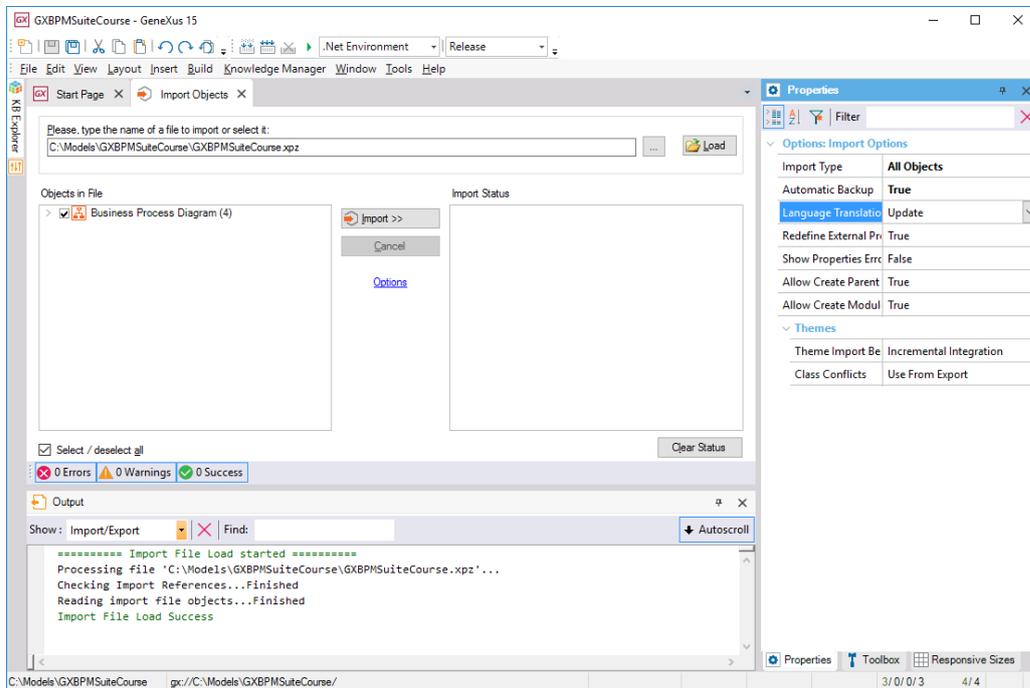
Para hacer que cuando se ejecute la tarea TicketReservation, se abra la pantalla de la transacción Reservation, vamos a asociar la tarea al objeto transacción. Estas asignaciones las vamos a hacer desde una KB de GeneXus, por lo que precisamos importar los diagramas que creamos con el Business Process Modeler.

Ejecutamos el Business Process Modeler, abrimos la KB GXBPMsuiteCourse, seleccionamos Knowledge Manager y elegimos Exportar. Marcamos Export All y luego Export.



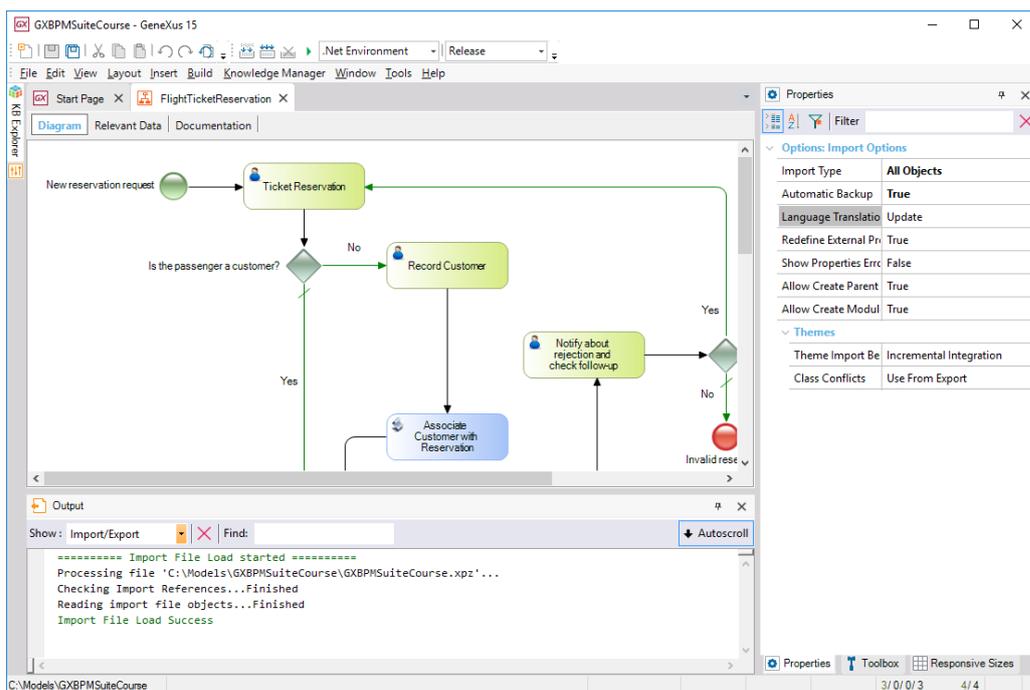
Ahora ejecutamos GeneXus y abrimos la KB GXBPMsuiteCourse que habíamos creado previamente, en la que teníamos creados varios objetos.

Seleccionamos Knowledge Manager y elegimos Import. Buscamos el archivo que habíamos creado desde el Business Process Modeler y presionamos Load.



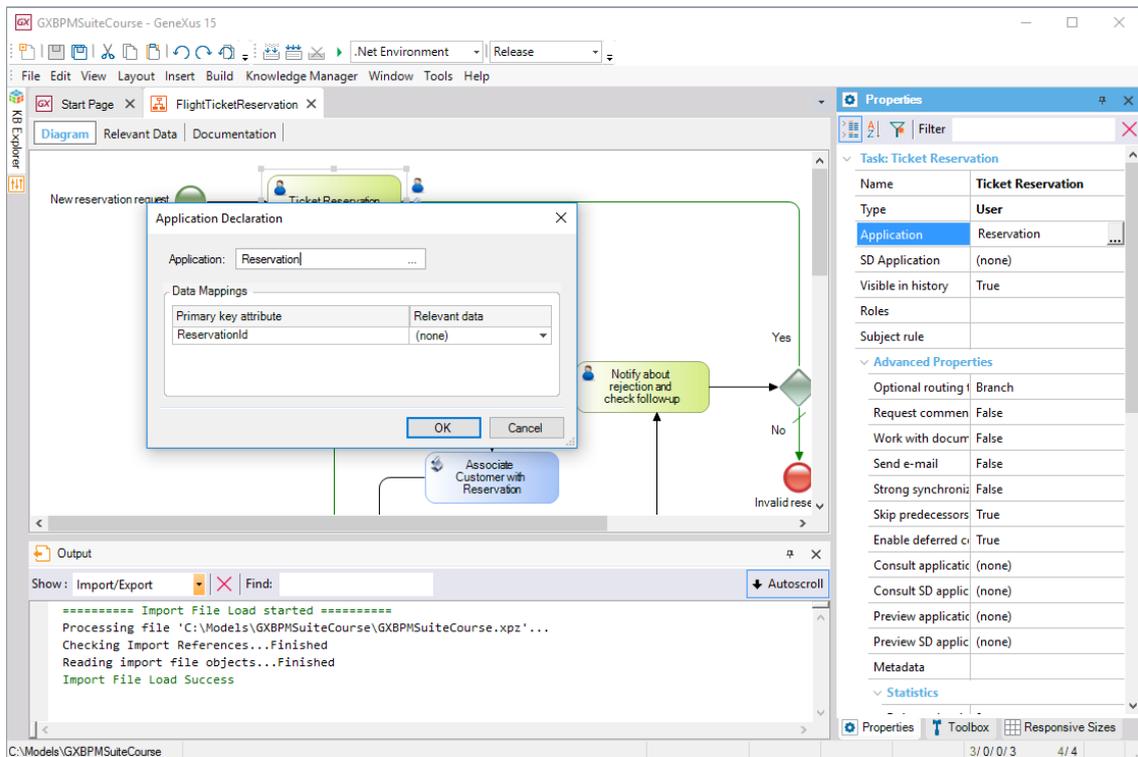
En este caso estamos utilizando un archivo de exportación con extensión .xpz para pasar diagramas desde el GXBPM a GeneXus. Otra posibilidad podría haber sido enviar desde el GXBPM la KB a un servidor GXServer y luego desde GeneXus hacer un Create KB from Server.

Ahora con Ctrl-O, abrimos el objeto FlightTicketReservation.



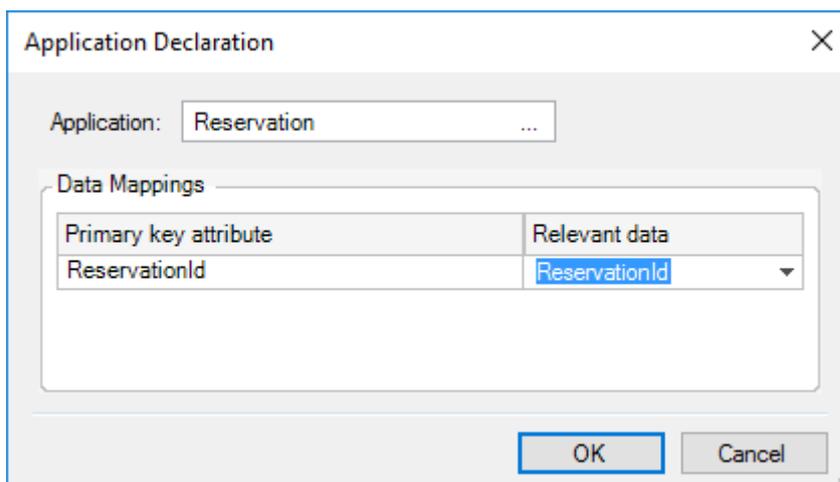
Seleccionamos la tarea TicketReservation y vamos a sus propiedades. Hacemos clic en la propiedad Application y presionamos el botón con tres puntos.

En el cuadro de diálogo, hacemos clic en el botón donde dice Application, seleccionamos la transacción Reservation y presionamos Ok.



Vemos que en el sector de Mapeo de Datos, nos muestra que la clave primaria de la transacción es el atributo ReservationId. Movemos el mouse sobre la columna RelevantData y vemos que aparece un botón y una flecha.

Presionamos el botón a la izquierda de la flecha y aparece seleccionado ReservationId.

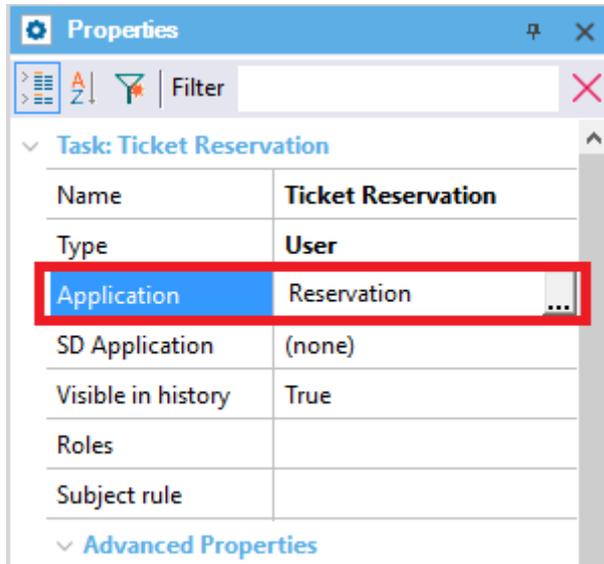


Quiere decir que ReservationId será el nombre de un dato relevante de nuestro proceso.

Un dato relevante es un dato global conocido a lo largo de un proceso. Podemos asimilarlo a una variable global y es el mecanismo utilizado por Workflow para el pasaje de información entre las tareas.

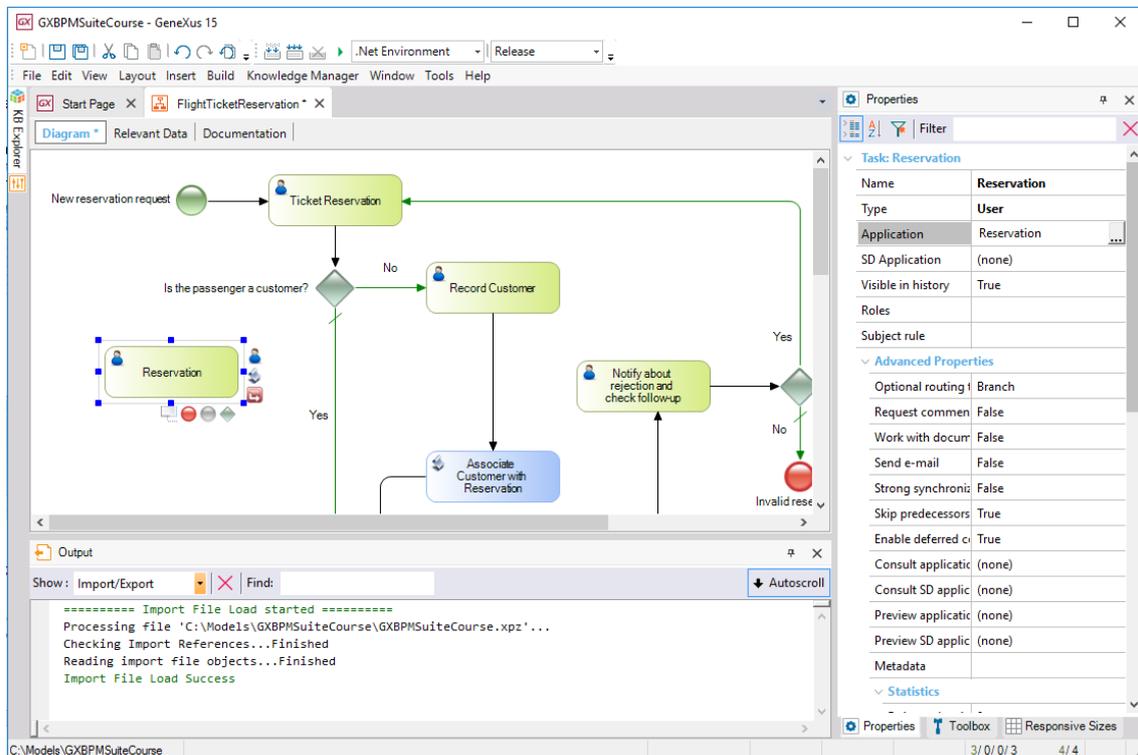
Al asociar una transacción a una tarea interactiva, GeneXus crea automáticamente un dato relevante con el mismo nombre que el atributo identificador de la transacción.

Presionamos OK y con esto, hemos establecido una relación entre la tarea y el objeto GeneXus y además un mapeo entre el atributo ReservationId y el dato relevante &ReservationId.



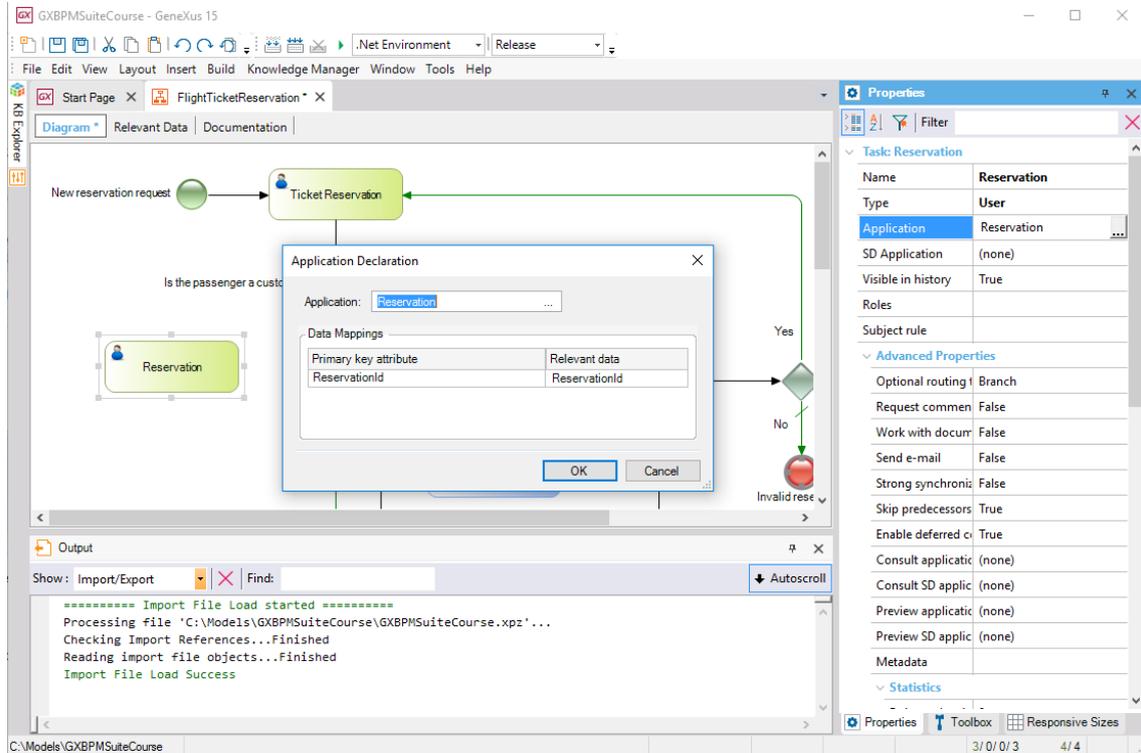
Esto implica que cada vez que se instancie la tarea TicketReservation, se ejecutará la transacción Reservation, ya que es la aplicación asociada a la tarea.

En este proceso de modelado, primero hemos creado la tarea interactiva y luego la asociamos a la aplicación. En lugar de crear la tarea, podríamos haber arrastrado directamente a la transacción Reservation desde el Folder View al diagrama.



Al hacerlo, vemos que se crea automáticamente una tarea interactiva con el mismo nombre que la transacción y que en su propiedad Application dice: Reservation, ya que se asoció automáticamente el objeto GeneXus a la tarea.

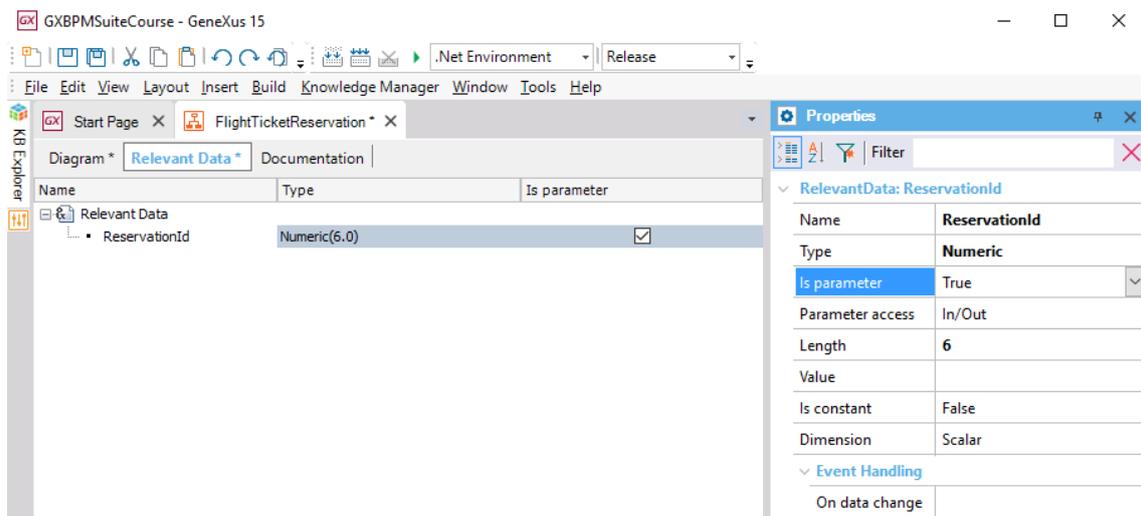
Si presionamos el botón de los tres puntos, vemos que también se creó el dato relevante y se asoció en forma automática al mismo, el atributo clave primaria de la transacción.



Este dato relevante tendrá igual nombre y tipo que la clave primaria de la transacción.

Si la clave primaria es compuesta por más de un atributo, se creará un dato relevante por cada atributo integrante de la clave primaria.

En la solapa Relevant Data del proceso FlightTicketReservation, podemos ver los datos relevantes definidos.

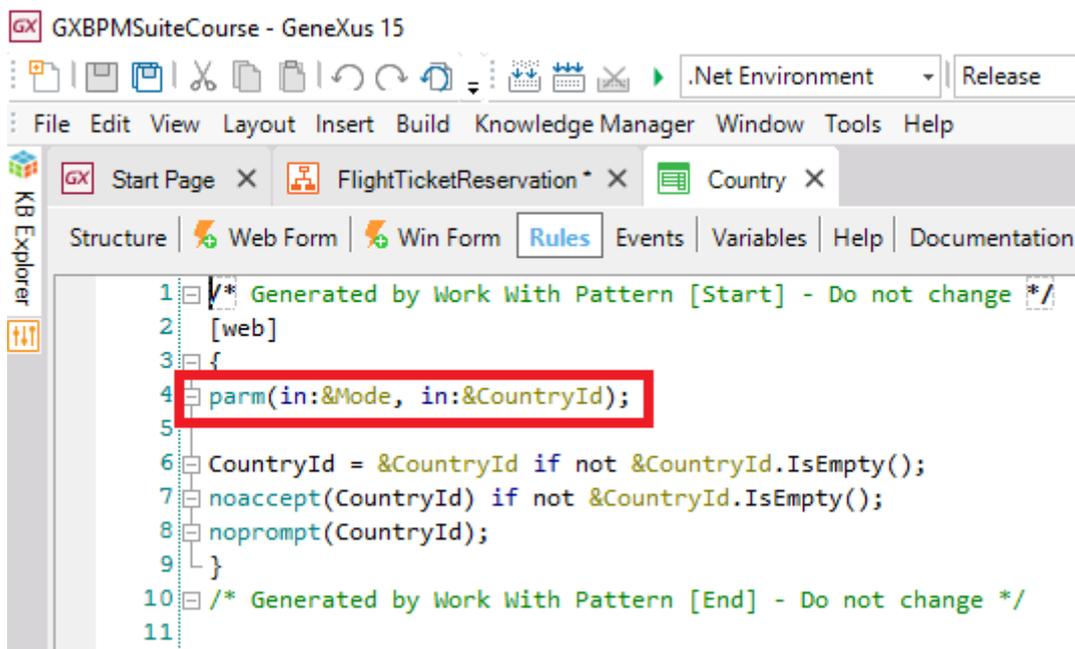


Vemos que aparecen bajo el símbolo de "ampersand" para indicar que se tratan variables.

Cuando hacemos referencia al dato relevante ReservationId lo haremos siempre utilizando el ampersand como prefijo, para diferenciarnos del atributo ReservationId, al igual que con las variables en los objetos GeneXus.

Otra cosa a tener en cuenta cuando arrastramos un objeto transacción al diagrama, es si la misma tiene definida una regla Parm y en dicha regla se recibe el valor de la clave primaria en una variable.

Aquí vemos a la transacción Country que se le aplicó el patrón Work With y como consecuencia se agregó una regla Parm que recibe la variable &CountryId como parámetro.



```
1 /* Generated by Work With Pattern [Start] - Do not change */
2 [web]
3 {
4 parm(in:&Mode, in:&CountryId);
5
6 CountryId = &CountryId if not &CountryId.IsEmpty();
7 noaccept(CountryId) if not &CountryId.IsEmpty();
8 noprompt(CountryId);
9 }
10 /* Generated by Work With Pattern [End] - Do not change */
11
```

El hecho de arrastrar la transacción al diagrama crea los datos relevantes, pero si los nombres de las variables en la regla Parm, coinciden exactamente con los nombres de los atributos de la clave primaria, habrá un mapeo entre los datos relevantes y estas variables, de modo que **cuando se ejecuta la transacción, las variables tomarán los valores almacenados en los datos relevantes en ese momento.**

En el siguiente video, Automatización Parte 2, continuaremos asociando objetos GeneXus a las tareas de nuestro diagrama de reserva de pasajes.