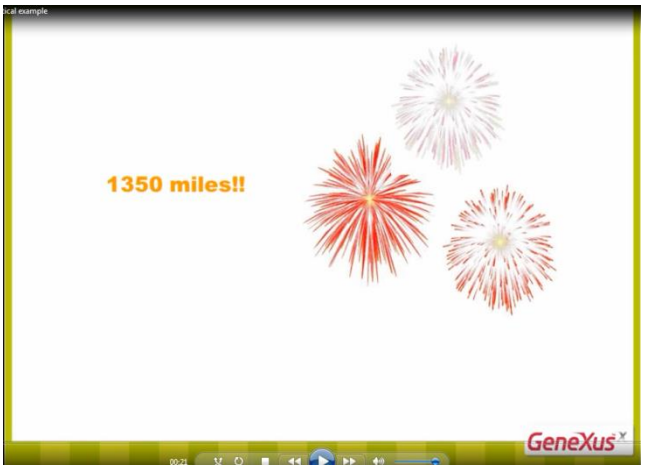
Otro ejemplo de uso de Business Components

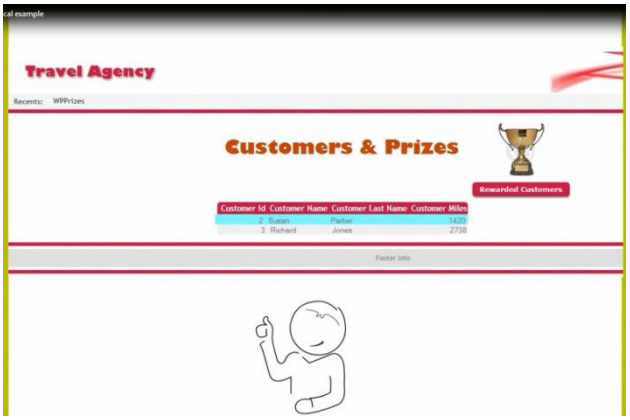
Vamos a ver a continuación un ejemplo práctico de uso de Business Components.

Comencemos planteando el siguiente escenario:



Todo cliente de la agencia de viajes acumula millas cada vez que realiza excursiones

Como forma de promoción, la agencia decide premiar con una cena, a ciertos clientes preferenciales que generaron más de 1000 millas y no han sido previamente premiados.

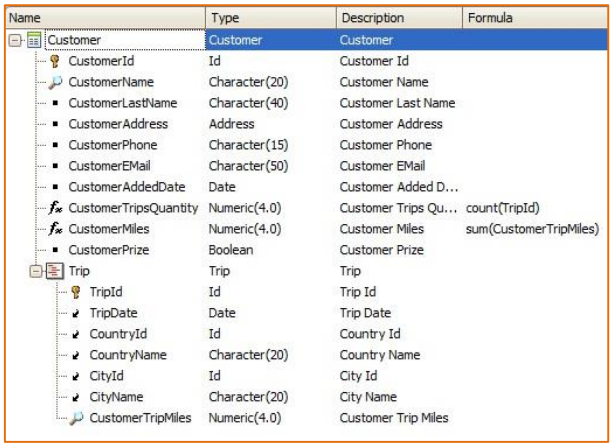


Se desea disponer de una página, a través de la cual se pueda proceder a esta premiación.

Emplearemos las acciones Drag & Drop para permitir arrastrar en ejecución los clientes elegidos y soltarlos sobre una imagen.

Para estos clientes entonces, se creará el premio en forma automática, y también se les adicionará 100 millas.

Veamos el diseño de transacciones que tenemos:



Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Id	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(40)	Customer Last Name	
CustomerAddress	Address	Customer Address	
CustomerPhone	Character(15)	Customer Phone	
CustomerEMail	Character(50)	Customer EMail	
CustomerAddedDate	Date	Customer Added D...	
CustomerTripsQuantity	Numeric(4.0)	Customer Trips Qu...	count(TripId)
CustomerMiles	Numeric(4.0)	Customer Miles	sum(CustomerTripMiles)
CustomerPrize	Boolean	Customer Prize	
Trip	Trip	Trip	
TripId	Id	Trip Id	
TripDate	Date	Trip Date	
CountryId	Id	Country Id	
CountryName	Character(20)	Country Name	
CityId	Id	City Id	
CityName	Character(20)	City Name	
CustomerTripMiles	Numeric(4.0)	Customer Trip Miles	

De cada cliente, registramos sus datos personales así como su lista de excursiones. Disponemos tambien de la cantidad de excursiones que realizó, de la cantidad de premios que la agencia le ha otorgado y de la cantidad total de millas acumuladas.

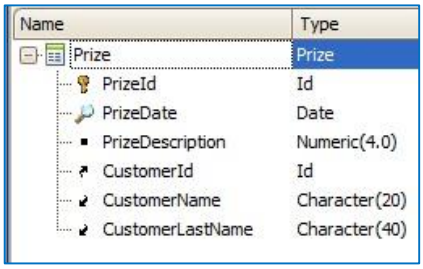
Vayamos ahora a las reglas declaradas, y en particular observemos esta regla Add

add(CustomerTripMiles,CustomerTotalMiles);

Esta definiendo que cada vez que a 1 cliente se le inserta 1 línea con 1 excursión, se le adicionarán la cantidad de millas correspondientes a su total de millas a su total de millas

En caso de eliminar una línea con una excursión realizada por el cliente, la regla add **le restará** las millas correspondientes a dicha excursión quitada al total de millas del cliente.

Observemos ahora la transacción Prize

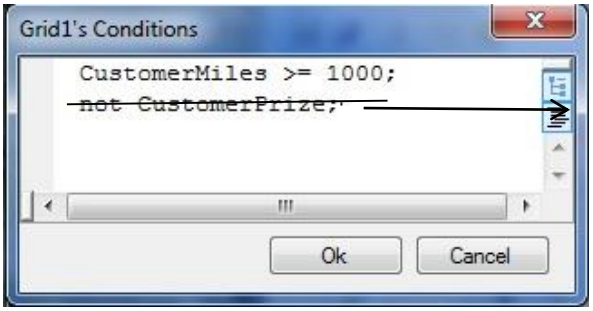


Name	Type
Prize	Prize
PrizeId	Id
PrizeDate	Date
PrizeDescription	Numeric(4.0)
CustomerId	Id
CustomerName	Character(20)
CustomerLastName	Character(40)

que nos permite registrar los premios que se otorgan. Cada premio corresponde a un cliente y hemos definido las siguientes reglas para poder cargar la fecha del premio con la fecha del día y adicionarle 100 millas al cliente ya que es parte de la premiación

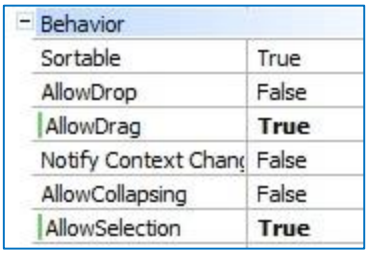
Bien. Pasemos ahora a ver el web panel WPPrizes, que ya hemos definido, y a través del cual vamos a resolver este requerimiento solicitado:

Este web panel muestra todos los clientes que tienen acumuladas 1000 millas o más y que no fueron premiados



Como dijimos anteriormente, la idea es poder arrastrar clientes desde la grilla y soltarlos sobre la imagen, y que simplemente de esa manera, se les genere el premio.

Observemos ahora que hemos especificado en el grid las propiedades Allow Selection y Allow Drag con el valor True. De esta forma estamos habilitando a seleccionar una línea del grid y arrastrarla, respectivamente.

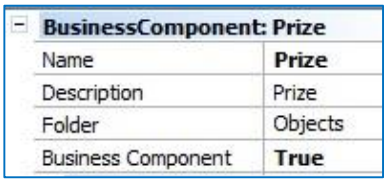


Ahora bien. ¿Qué tendrá que suceder al momento de soltar una línea del grid sobre la imagen?

Si lo pensamos en nuestro lenguaje natural podemos decir **que hay que crear un nuevo premio para el cliente seleccionado**, con la fecha del día, indicar que es una cena, a la vez que hay que sumarle al cliente 100 millas de regalo.

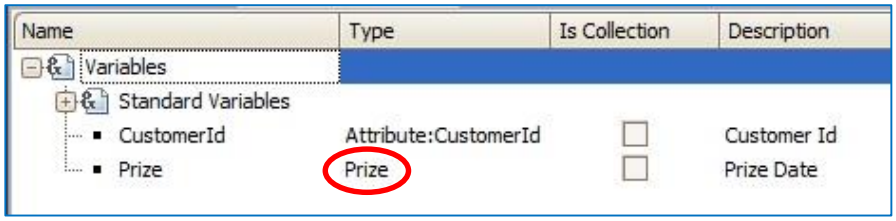
¿ Y qué nos ofrece GeneXus para resolver esto?

Utilizaremos la transacción Prize como Business Component para insertar el premio, y **así aprovechamos los beneficios que nos brindan los Business Components** Para ello, configuramos la propiedad Business Component en la transacción Prize con el valor True.



Volvamos al web panel.

En este objeto hemos definido la variable &Prize del tipo de dato Prize.



Enseguida veremos el uso de esta variable para realizar la inserción del premio.

Queremos insertar el premio cuando el usuario suelte la línea del grid sobre la imagen, entonces la codificación debe estar dentro del evento Drop asociado a la imagen. Este evento Drop tiene que recibir como parámetro el identificador del cliente arrastrado, o sea CustomerId. Pero este evento no admite atributos como parámetros, así que debemos definir una variable &CustomerId basada en el atributo CustomerId .

Incluimos entonces esta variable como parámetro del evento y GeneXus entiende que se trata del identificador de la línea del grid que se arrastra y se suelta sobre la imagen.

```
Event Image1.Drop(&CustomerId)

EndEvent
```

Ahora observemos la primera línea de código de este evento Drop.

A la variable &Prize le estamos asignando el texto “Dinner” a su miembro PrizeDescription.

En la segunda línea del código, al miembro CustomerId de esta variable &Prize, le estamos asignando el cliente que el usuario arrastro del grid sobre la imagen, el cual tenemos recibido a través del parámetro &CustomerId.

Ahora repasemos lo siguiente:

- El identificador de la transacción Prize, o sea PrizeId, está basado en el dominio Id que es auto numerado, así que no necesitamos asignarle un valor al miembro correspondiente
- Y las reglas declaradas en la transacción Prize se disparan al utilizar esta transacción como Business Component, así que no es necesario asignar una fecha a PrizeDate.. y las 100 millas serán también adicionadas al cliente.

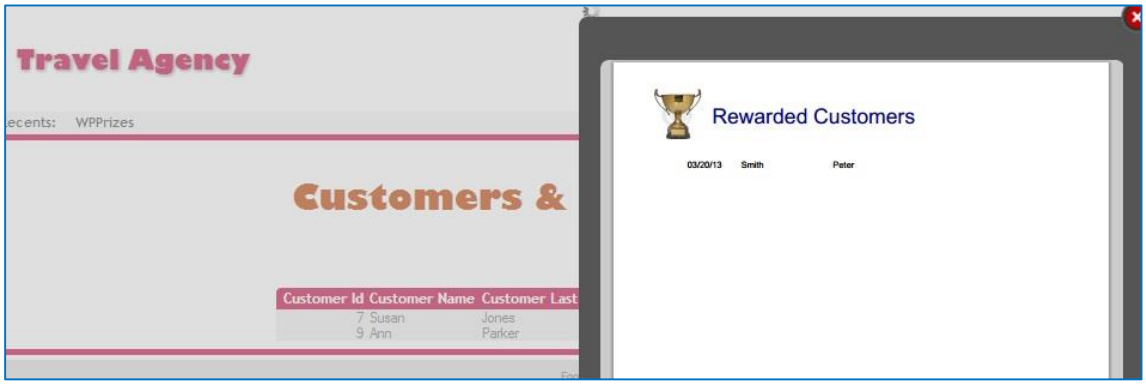
Bien. Hasta ahora hemos trabajado en memoria. Para realizar la grabación física en la base de datos, declaremos el método Save y el comando Commit, necesarios al trabajar con Business Component.

```
Event Image1.Drop(&CustomerId)
    &Prize.CustomerId = &CustomerId
    &Prize.PrizeDescription = "Dinner"
    &Prize.Save()
    commit
EndEvent
```

Presionemos F5 y veamos cómo funciona.

Este botón nos ofrece ver un listado pdf con todos los premios otorgados.

Si observamos vemos que ya hay un premio otorgado al cliente Peter Smith.

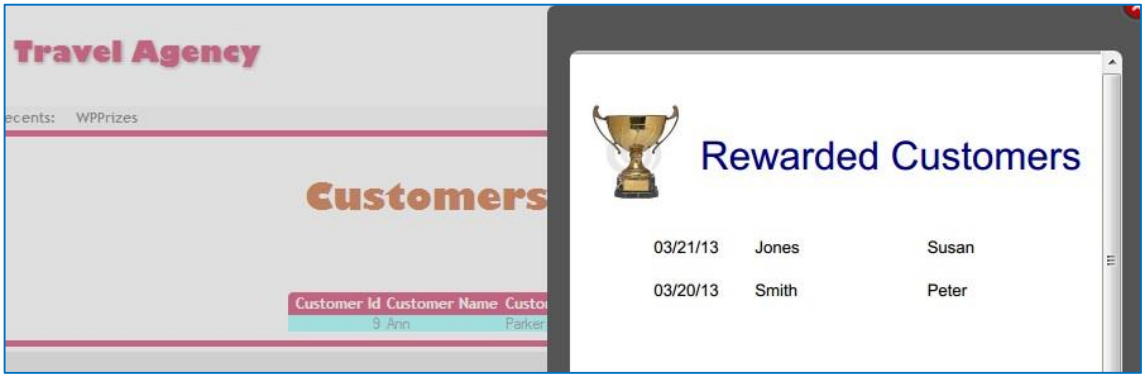


Vamos ahora a premiar a Susan Parker. Observemos primero que ya tiene 1250 millas registradas.

Así que seleccionamos la línea y la arrastramos sobre la imagen. Enseguida vemos que la línea desaparece del grid, lo cual es el comportamiento esperado ya que este grid solamente muestra los clientes que no han sido premiados.

Veamos ahora el listado. Vemos que aparece Susan Parker en la lista de premios otorgados y verificamos tambien que tiene la fecha del día de hoy, y que se le han acreditado las 100 millas extras.

Hemos visto de esta forma lo simple que fue definir y usar el concepto de Business Component. Y que efectivamente nos provee todo lo que ofrecen las transacciones.



Porque además de haberse disparado las reglas, como ya hemos visto, también se validó la consistencia de los datos que asignamos y grabamos. En este caso, el valor de cliente que asignamos al premio es válido porque fue directamente arrastrado de un dato extraído de la base de datos, pero podríamos haber asignado otro valor o variable cargada de otra manera y la validación de existencia se realiza.

Para finalizar y completar asi la implementación, veamos el código que definimos en el evento Enter asociado al botón Prizes awarded.

Hemos definido y utilizado una variable del tipo de datos Window, ya que entre otras cosas permite mostrar un listado pdf como una ventana popup.

```
Event Enter
    &window.Object = ViewCustomers.Create()
    &window.Width = 500
    &window.Height = 700
    &window.Open()
Endevent
```

En esta primera línea de código vemos que le estamos asociando a la propiedad Object de la variable &Window el objeto que queremos ejecutar.

Vemos también que es posible definir el alto y ancho de la ventana

Y finalmente abrimos la ventana con el listado.