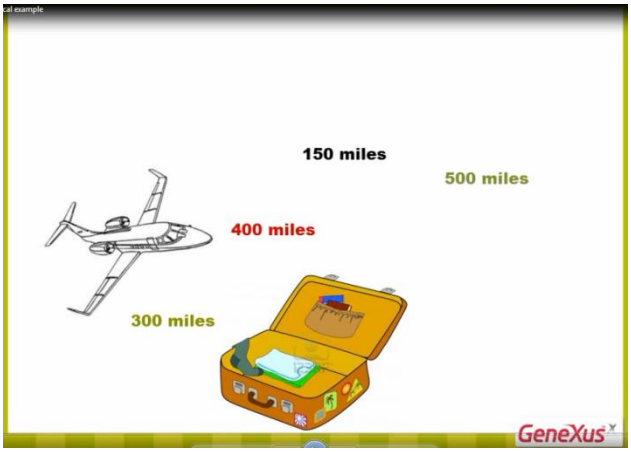
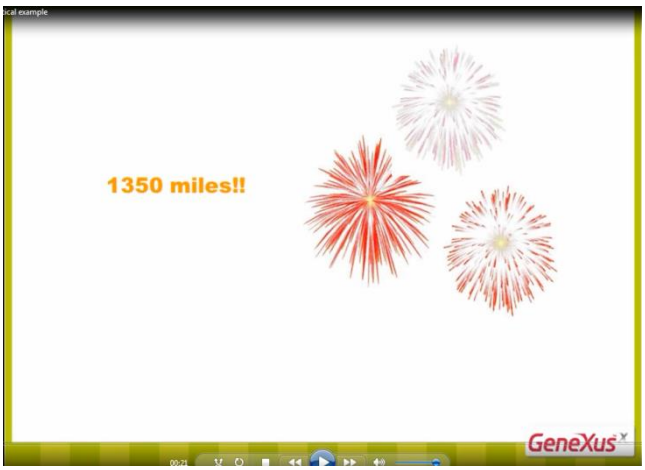
Curso GeneXus - Otro ejemplo de uso de Business components



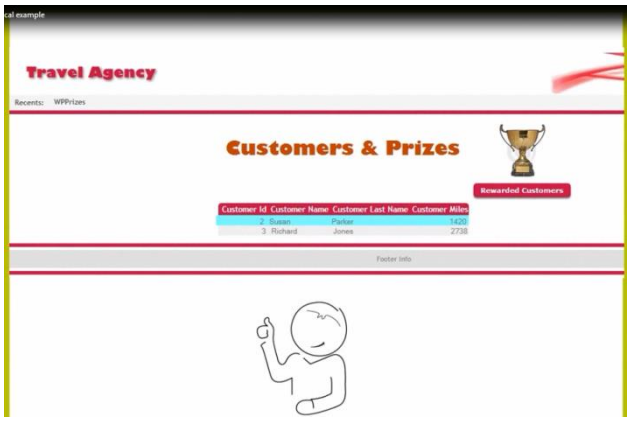
Veremos, a seguir, un exemplo prático do uso de Business Components.

Comecemos postulando o seguinte cenário:



Todo cliente da agência de viagens acumula milhas a cada vez que faz uma excursão.

Como forma de promoção, a agência decide premiar com um jantar certos clientes preferenciais que geraram mais de 1.000 milhas e que não foram premiados anteriormente.

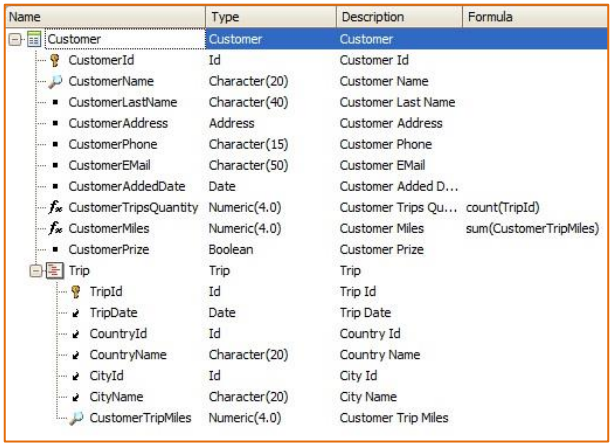


Deseja-se criar uma página através da qual essa premiação possa ser feita.

Empregaremos as ações Drag & Drop para permitir arrastar em execução os clientes escolhidos e soltá-los sobre uma imagem.

Para esses clientes, será criado o prêmio de forma automática, além de adicionadas 100 milhas.

Vejamos o desenho de transação que temos.



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, vamos registrar os seus dados pessoais, bem como sua lista de excursões. Dispomos também da quantidade de excursões que ele fez, da quantidade de prêmios concedidos a ele pela agência e da quantidade de milhas acumuladas.

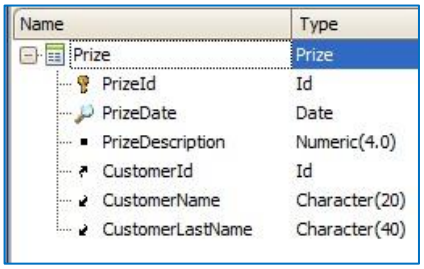
Vamos agora às regras declaradas e, em particular, observemos a regra Add

add(CustomerTripMiles, CustomerTitalMiles);

Ela define que a cada vez que 1 cliente inserir 1 linha com 1 excursão, será adicionada a quantidade de milhas correspondentes ao seu total de milhas.

Caso você exclua uma linha com uma excursão realizada por um cliente, a regra add **irá subtrair** as milhas correspondentes à excursão tirada do total de milhas do cliente.

Observemos agora a transação Prize,

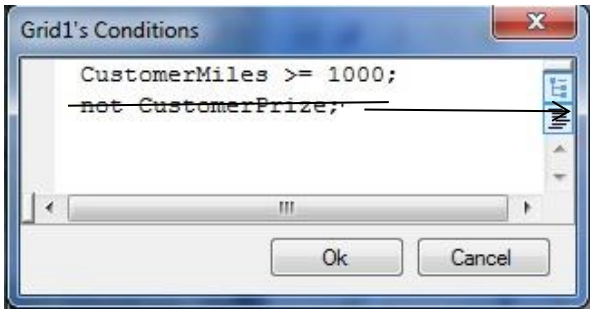


Name	Type
Prize	Prize
PrizeId	Id
PrizeDate	Date
PrizeDescription	Numeric(4.0)
CustomerId	Id
CustomerName	Character(20)
CustomerLastName	Character(40)

que nos permite registrar os prêmios que foram concedidos. Cada prêmio corresponde a um cliente e definimos as seguintes regras para poder carregar a data do prêmio com a data do dia e adicionar 100 milhas ao cliente já que é parte da premiação.

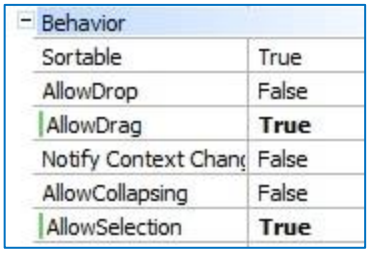
Vejamos agora a web panel WPPrizes, que já havíamos definido e através da qual vamos resolver esse requerimento solicitado:

Essa web panel mostra todos os clientes que tem 1.000 milhas acumuladas ou mais e que não foram premiados.



Como dissemos anteriormente, a ideia é poder arrastar os clientes do grid e soltá-los sobre a imagem e que, fazendo apenas isso, o prêmio seja gerado para eles.

Notemos que especificamos no grid as propriedades Allow Selection e Allow Drag com o valor True. Dessa forma, habilitamos selecionar uma linha do grid e arrastá-la, respectivamente.



Muito bem. O que terá que acontecer no momento de soltar uma linha do grid sobre a imagem?

Se pensarmos em nossa linguagem natural, podemos dizer **que é preciso criar um novo prêmio para o cliente selecionado** com a data do dia, indicar que é um jantar, além de ser preciso acrescentar para o cliente 100 milhas de presente.

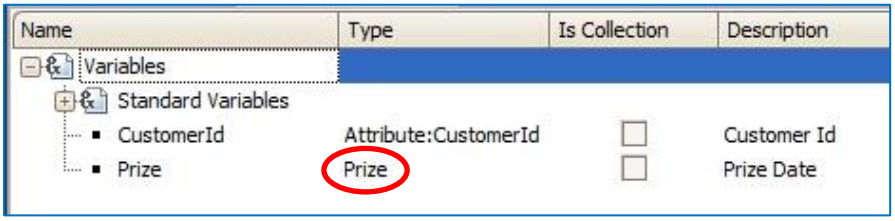
E o que GeneXus nos oferece para resolver isso?

Utilizamos a transação Prize como Buniness Component para inserir o prêmio, e assim **aproveitamos os benefícios que os Business Components nos oferecem.** Para isso, configuramos a propriedade Business Component na transação Prize com o valor True.

BusinessComponent: Prize	
Name	**Prize**
Description	Prize
Folder	Objects
Business Component	**True**

Voltemos à web panel.

Neste objeto, definimos a variável &Prize do tipo de dado Prize.



Em seguida, veremos o uso dessa variável para realizar a inserção do prêmio.

Queremos inserir o prêmio quando o usuário soltar a linha do grid sobre a imagem, assim, a codificação deve estar dentro do evento Drop associado à imagem. Esse evento Drop deve receber como parâmetro o identificador do cliente arrastado, ou seja, CustomerId. Mas esse evento não admite atributos como parâmetros, de modo que devemos definir uma variável &Customer baseada no atributo CustomerId.

Incluímos essa variável como parâmetro do evento e GeneXus entende que se trata do identificador da linha do grid que se arrasta e se solta sobre a imagem.

```
Event Image1.Drop(&CustomerId)

EndEvent
```

Agora, observemos a primeira linha do código desse evento Drop.

À variável &Prize, estamos atribuindo o texto “Dinner” a seu membro PrizeDescription.

Na segunda linha do código, estamos atribuindo ao membro CustomerId dessa variável &Prize o cliente que o usuário arrastou do grid sobre a imagem que temos recebido através do parâmetro &CustomerId.

Agora, revisemos o seguinte:

- O identificador da transação Prize, ou seja, PrizeId, baseia-se no domínio Id que é autonumerado, o que significa que não necessitamos atribuir um valor ao membro correspondente.
- As regras declaradas na transação Prize serão disparadas ao utilizar essa transação como Business Component, de modo que não é preciso atribuir uma data a PrizeDate... e as 100 milhas também serão adicionadas ao cliente.

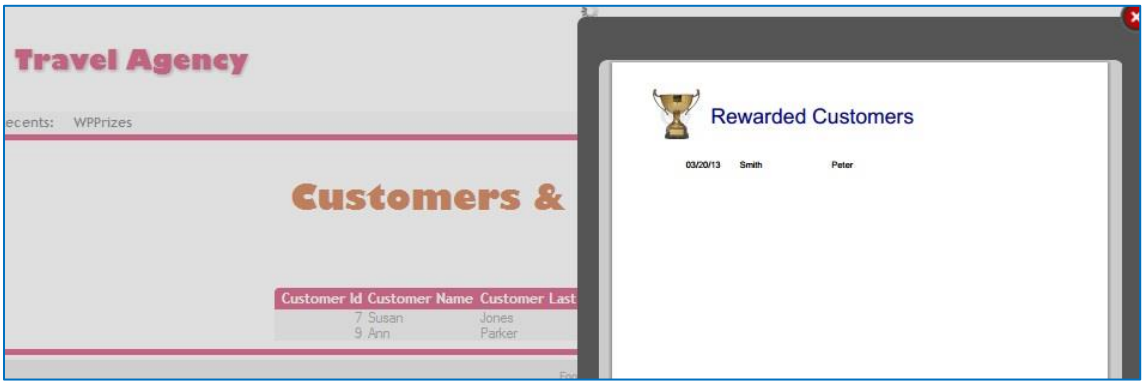
Até agora, trabalhamos na memória. Para salvar fisicamente no banco de dados, declararemos o método Save e o comando Commit, necessários ao trabalhar com Business Component.

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

Pressionemos F5 e vejamos como funciona.

Esse botão nos oferece ver uma lista pdf com todos os prêmios concedidos.

Se observarmos, vemos que um prêmio foi concedido ao cliente Peter Smith.



Vamos premiar Susan Parker. Observemos primeiro que ela já tem 1.250 milhas registradas.

Então selecionamos a linha e arrastamos sobre a imagem. Depois, vemos que a linha desaparece do grid, o que é o comportamento esperado já que este grid mostra apenas os clientes que não foram premiados.

Vejamos agora a lista. Observamos que Susan Parker aparece na lista de prêmios concedidos e

verificamos que tem a data do dia de hoje, e que foram computadas as 100 milhas extras.

Vimos, desta forma, que foi simples definir e usar o conceito de Business Component e que ele, efetivamente, nos fornece tudo que as transações oferecem.

Porque além das regras terem sido disparadas, como já vimos, também foi validada a consistência dos dados que atribuímos e gravamos. Neste caso, o valor do cliente que atribuímos ao prêmio é válido porque foi diretamente arrastado de um dado extraído do banco de dados, mas poderíamos ter atribuído outro valor ou variável carregada de outra forma e a validação de existência seria feita.

Assim, para finalizar e completar a implementação, vejamos o código que definimos no evento Enter associado ao botão Prizes awarded.

Definimos e utilizamos uma variável do tipo de dados Window, já que, entre outras coisas, ela permite mostrar uma lista pdf como uma janela popup.

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

Nesta primeira linha de código, vemos que estamos associando à propriedade Object da variável &Window o objeto que queremos executar.

Vemos também que é possível definir a altura e largura da janela.

E, finalmente, abrimos a janela com a lista.