

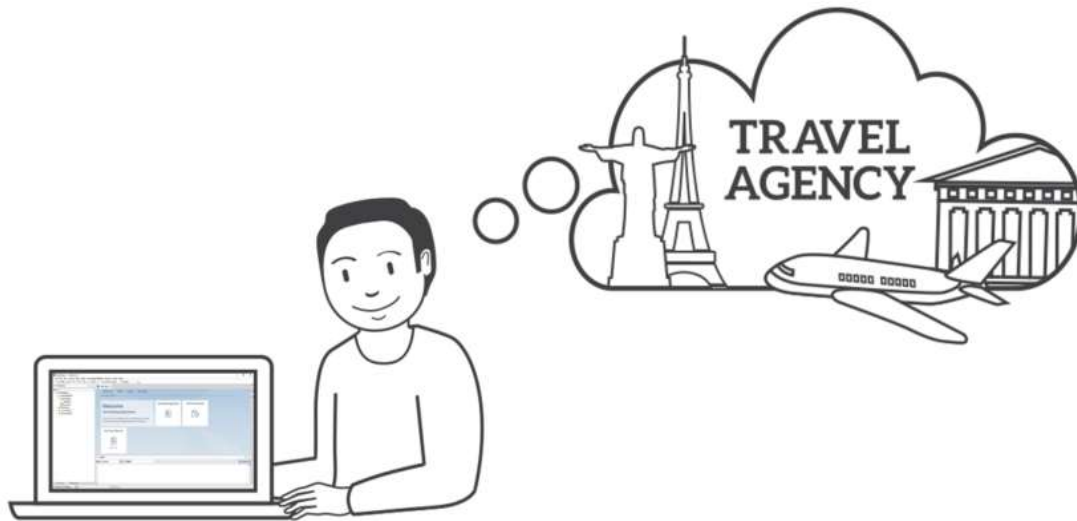
## Diseñando la primera transacción

GeneXus



Luego de creada la base de conocimiento, el siguiente paso

GeneXus



consiste en describir los objetos de la realidad mediante objetos GeneXus .....



Para identificar los objetos de la realidad, recomendamos prestar atención a los sustantivos que mencionan los usuarios.

En la agencia de viajes que nos solicitó la aplicación,



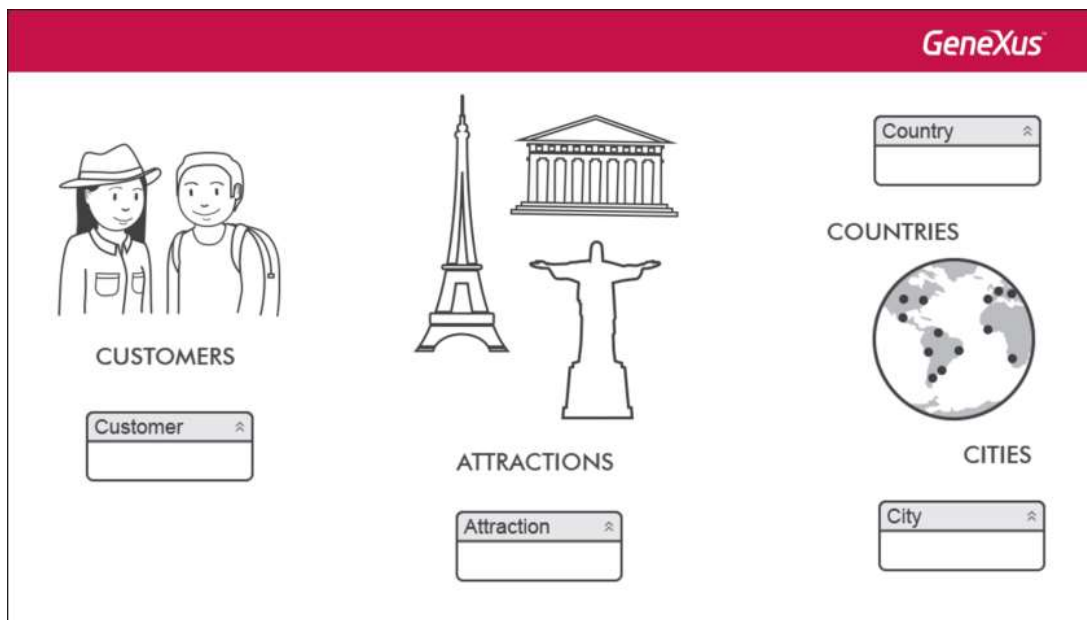
nos contaron que requieren registrar sus **clientes**, las **atracciones turísticas** que suelen sugerirles, así como los **países** y **ciudades** que ofrecen para visitar.

A partir de esto identificamos cuatro objetos de la realidad a describir en la base de conocimiento:



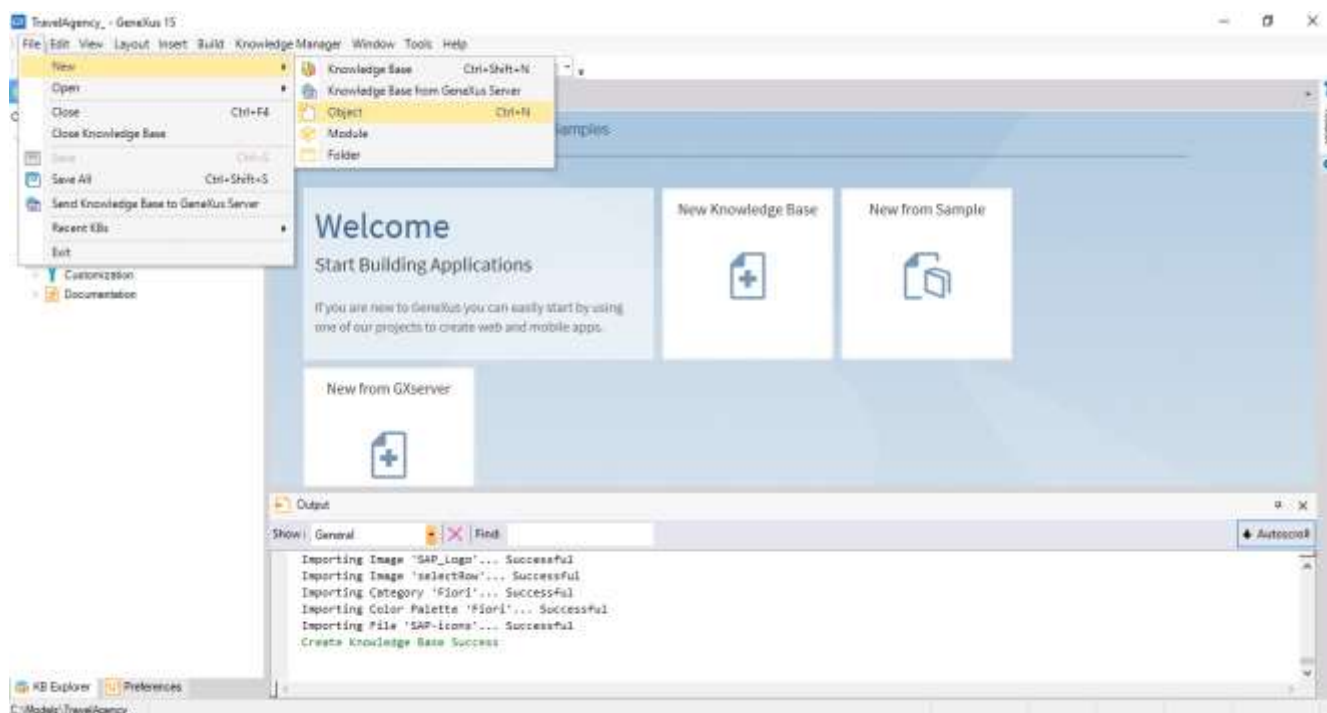
- Los **C**lientes
- Las **A**tracciones turísticas
- Los **P**aíses
- y las **C**iudades

Por cada objeto de la realidad identificado, **crearemos un objeto GeneXus de tipo Transacción**.

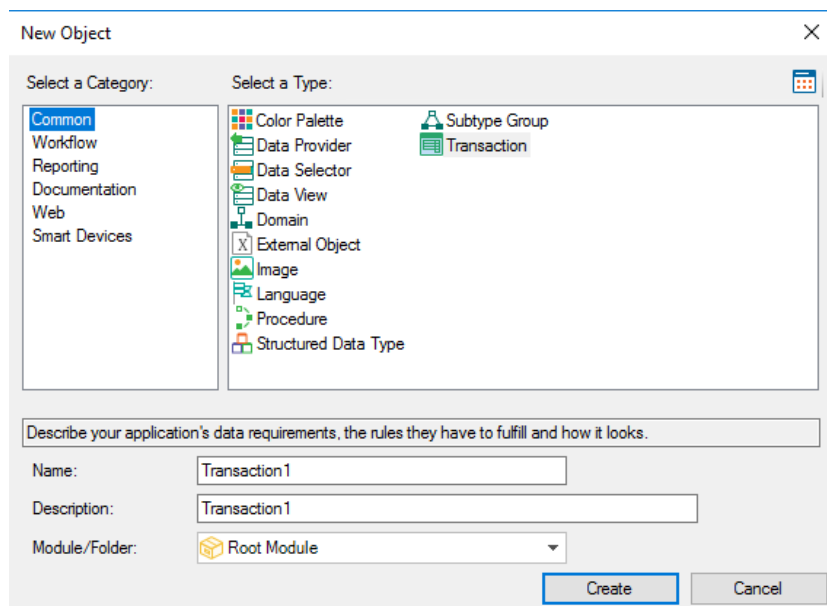


Los primeros objetos GeneXus que se definen en una base de conocimiento son transacciones, ya que las mismas permiten describir los objetos u actores de la realidad. Vamos a GeneXus para hacerlo.

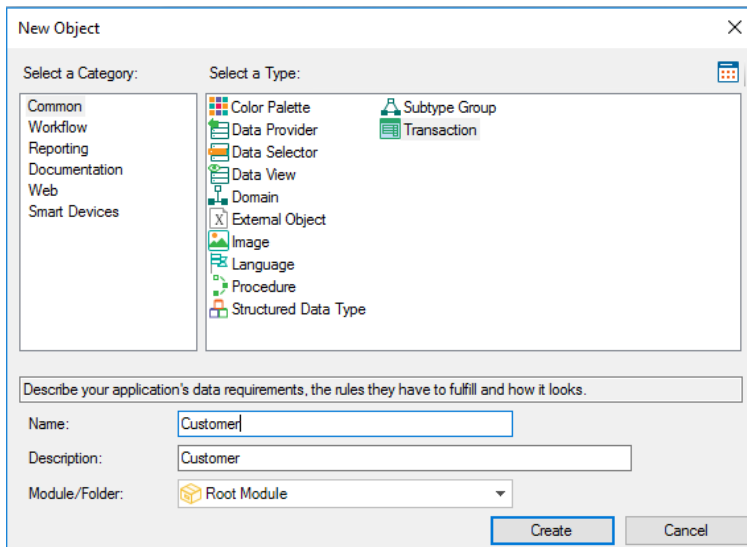
Para crear un objeto GeneXus, elegimos File / **New** / **Object**:



Seleccionamos esta opción y vemos que se abre el siguiente diálogo para crear un objeto GeneXus, donde podemos elegir el tipo de objeto a crear:

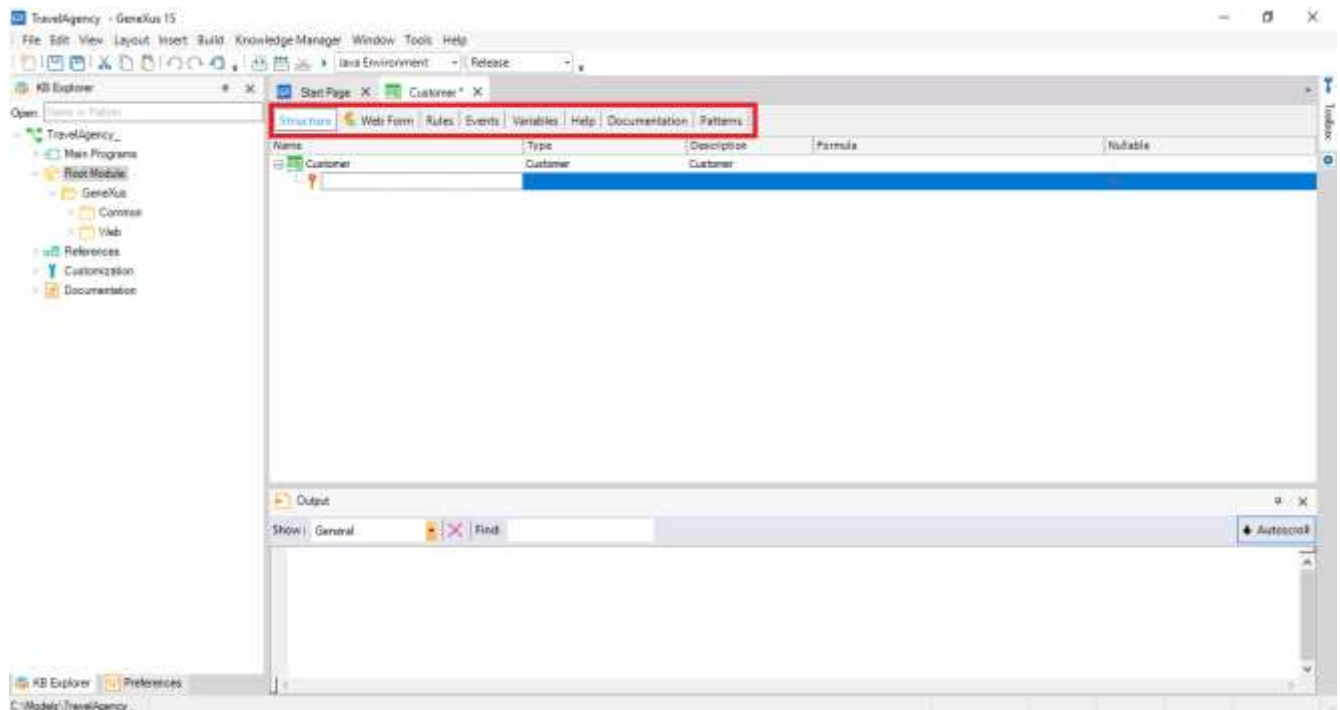


Seleccionamos el tipo de objeto **transacción**... (no se trata del concepto de “transacción” de base de datos, sino que será un objeto GeneXus específico para trabajar con las entidades de la realidad de la app) y le daremos a la nueva transacción que estamos creando el nombre: “Customer” (en español: “Cliente”)



Presionamos el **botón** “Create”....

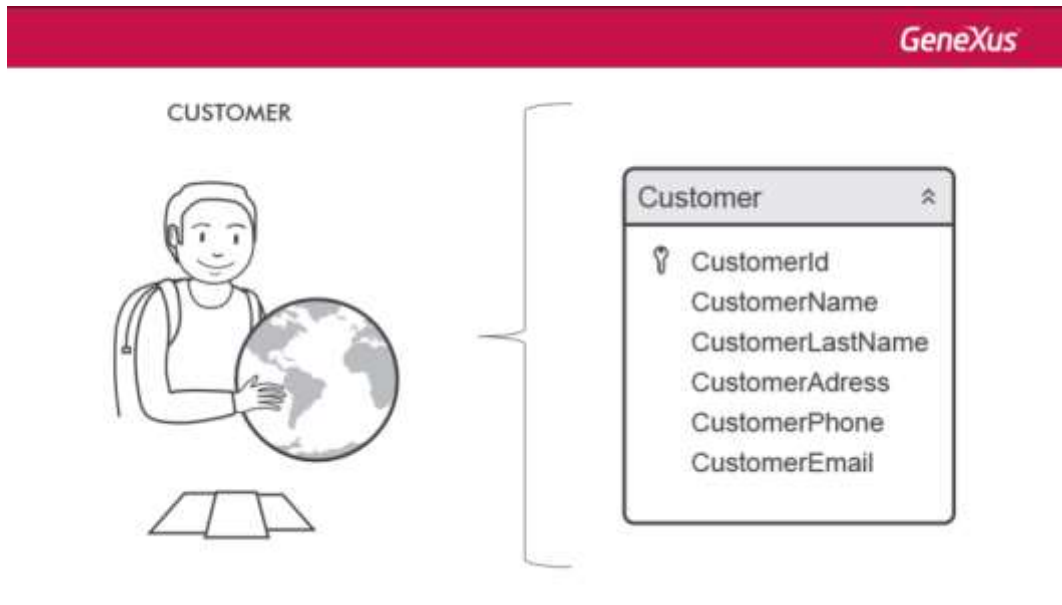
Y aquí podemos ver la transacción creada, lista para que comencemos a definir en primer lugar su estructura:



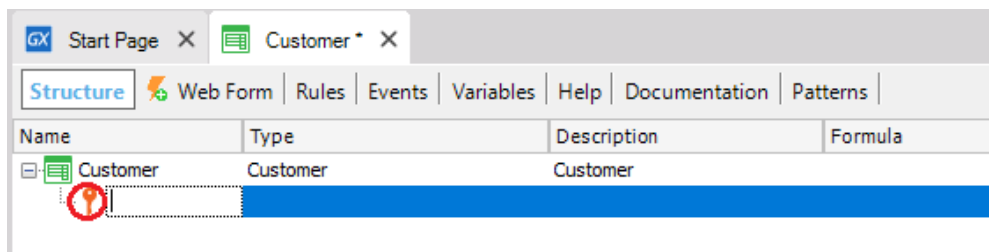
Toda transacción que se cree contendrá estas secciones que iremos explicando oportunamente.

**La estructura de una transacción permite definir los atributos o campos que describen al objeto de la realidad.**

En la agencia de viajes nos transmitieron que de cada cliente interesa registrar su **nombre, apellido, dirección, teléfono y correo electrónico**. De modo que estos datos que se deben registrar para cada cliente **corresponden a los atributos a ser definidos para esta transacción**.



Comencemos a definir entonces **los atributos de la transacción Customer**.



Veamos que se nos crea una primera línea para definir el primer atributo...

**Y observemos que hay un ícono de llave asociado a esta línea.**

Esto se debe a que **en toda transacción es requisito definir un atributo -o conjunto de atributos- con rol de identificador o llave**...

El concepto de identificador o llave tiene por objetivo permitir identificar de forma única a cada cliente que se registre, o al objeto de la realidad del que se trate.

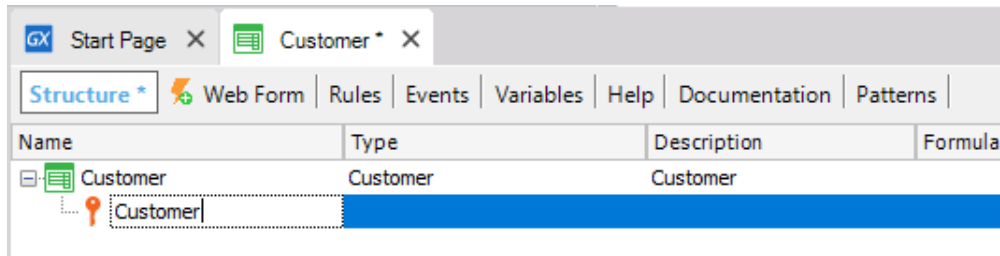
En otras palabras, no será posible ingresar dos clientes con el mismo valor de identificador.

Pasemos ahora a definir al atributo llave de la transacción Customer...

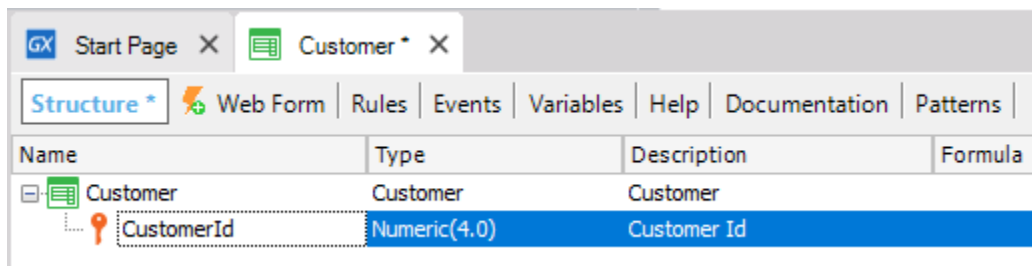
Dado que no nos solicitaron que almacenemos el pasaporte, ni el documento de identidad —que podrían ser

datos candidatos a ser elegidos como identificadores—, crearemos un atributo con el nombre “CustomerId” (es una abreviación de Customer Identifier, o sea Identificador de Cliente)... y en breve veremos cómo hacer para que se autonumere correlativamente.

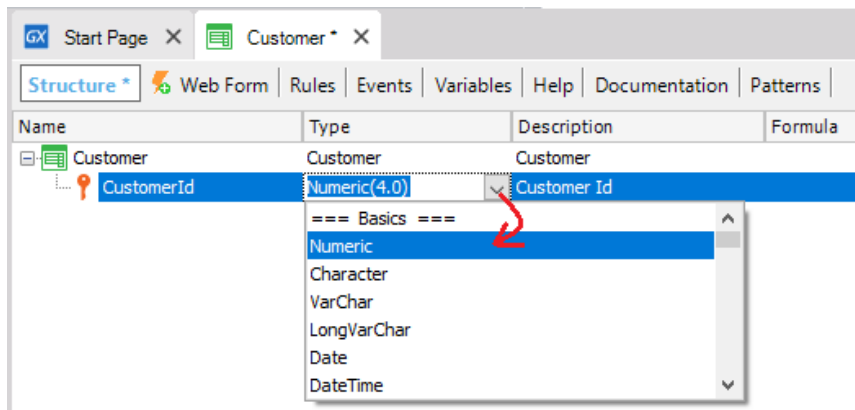
Observemos que digitando la tecla “punto” del teclado, GeneXus nos escribe al nombre de la transacción como prefijo en el nombre del atributo...



Solamente nos resta digitar a continuación del prefijo “Customer”, la palabra “Id”:



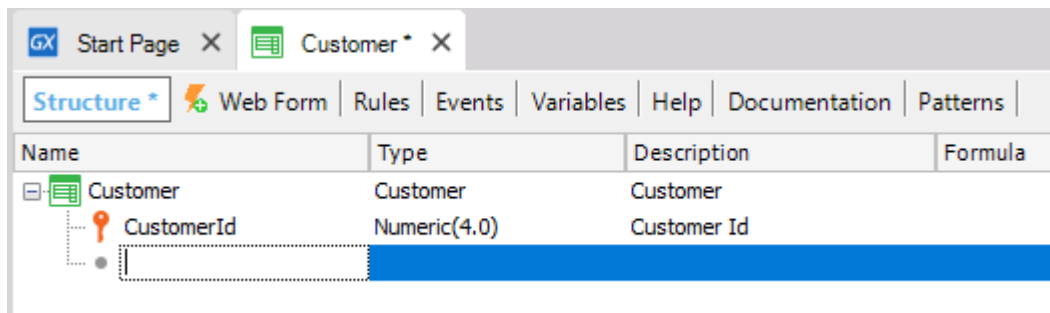
Pulsamos la tecla tabulador y debemos elegir el **tipo de datos** que este atributo va a almacenar.



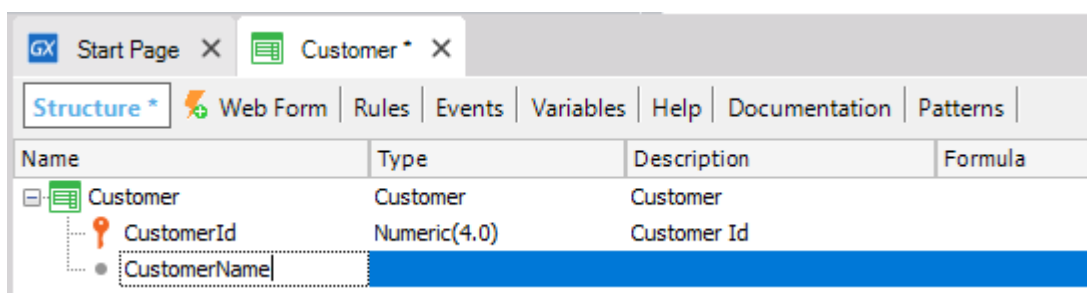
Si presionamos en la flecha, vemos los tipos de datos que GeneXus nos ofrece... y vamos a dejar para este atributo el tipo de datos sugerido, es decir: **numérico de 4 dígitos (y sin decimales)**.

Presionamos ENTER y vamos a pasar a definir el segundo atributo.

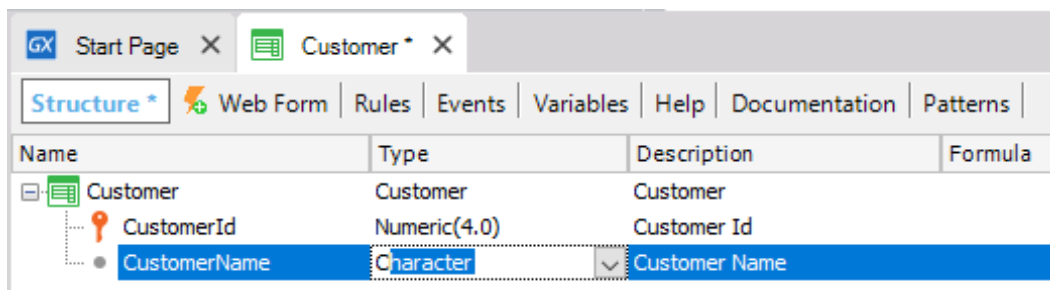
Se abre una nueva línea,



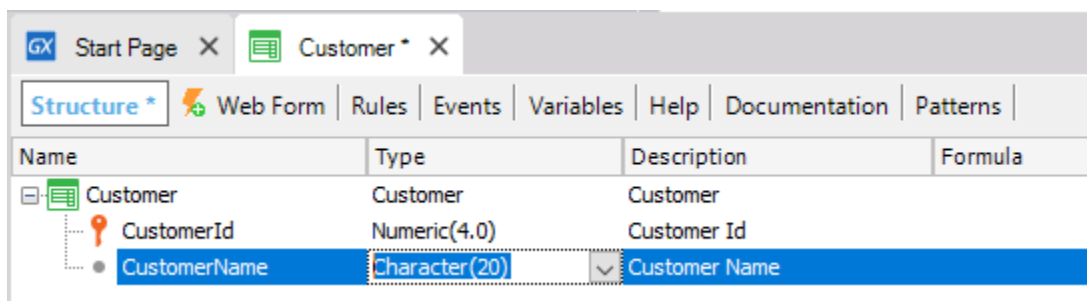
nuevamente digitamos "." y completamos el nombre del atributo con "Name", es decir: "CustomerName" (o sea, nombre de cliente).



Para definir el tipo de datos que almacenará el atributo CustomerName en este caso seleccionaremos el tipo de datos Character.



Observemos que si pulsamos paréntesis de apertura...

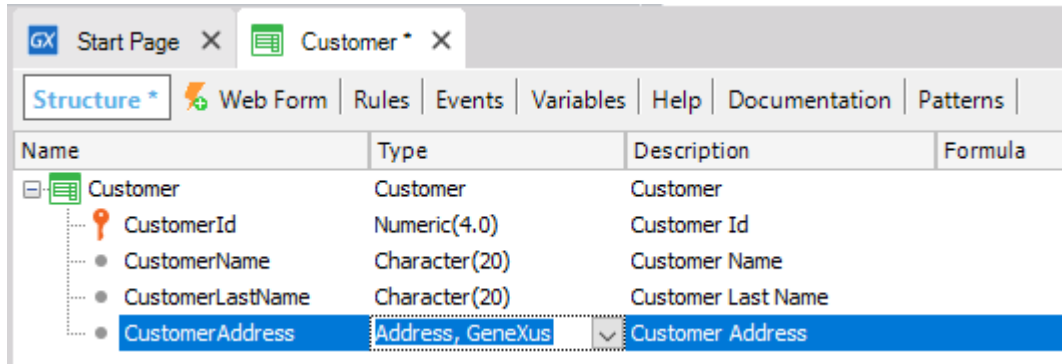




el largo por defecto que se nos ofrece es de 20 caracteres.... y lo dejaremos así.

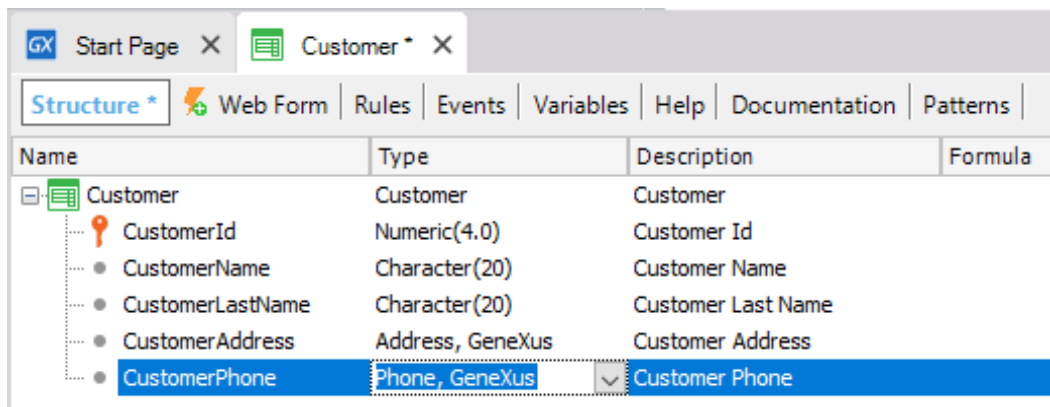
Con el mismo procedimiento ingresamos el atributo CustomerLastName, o sea apellido del cliente, que también será del tipo Character de largo 20.

Ahora definimos al atributo CustomerAddress (dirección del cliente), y vemos que en este caso el tipo de datos se asignó automáticamente. GeneXus se dio cuenta de que queremos crear un atributo cuyo nombre coincide en parte con el de un tipo de datos existente.



Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerAddress	Address, GeneXus	Customer Address	

Seguimos... con CustomerPhone (teléfono del cliente), y vemos que GeneXus le asignó el tipo de datos Phone.



Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerAddress	Address, GeneXus	Customer Address	
CustomerPhone	Phone, GeneXus	Customer Phone	

..... Y por último ingresamos al atributo CustomerEMail, al que GeneXus le asigna el tipo de datos EMail. En particular, los tipos de datos Address, Phone e Email son tipos de datos especiales llamados **dominios semánticos** y más adelante veremos que cuando trabajamos con ellos, incluyen funcionalidades que son propias de una dirección, un teléfono o un correo electrónico respectivamente.

Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerAddress	Address, GeneXus	Customer Address	
CustomerPhone	Phone, GeneXus	Customer Phone	
CustomerEmail	Email, GeneXus	Customer Email	

Ahora grabaremos esta transacción.

Observemos antes **que hasta ahora se nos está mostrando un asterisco en esta solapa con el nombre de la transacción Customer...**

Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerAddress	Address, GeneXus	Customer Address	
CustomerPhone	Phone, GeneXus	Customer Phone	
CustomerEmail	Email, GeneXus	Customer Email	

Esto significa que la transacción está siendo editada... y cuando salvamos los cambios... el asterisco desaparece... y la transacción aparece en la ventana KB Explorer:

Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerAddress	Address, GeneXus	Customer Address	
CustomerPhone	Phone, GeneXus	Customer Phone	
CustomerEmail	Email, GeneXus	Customer Email	

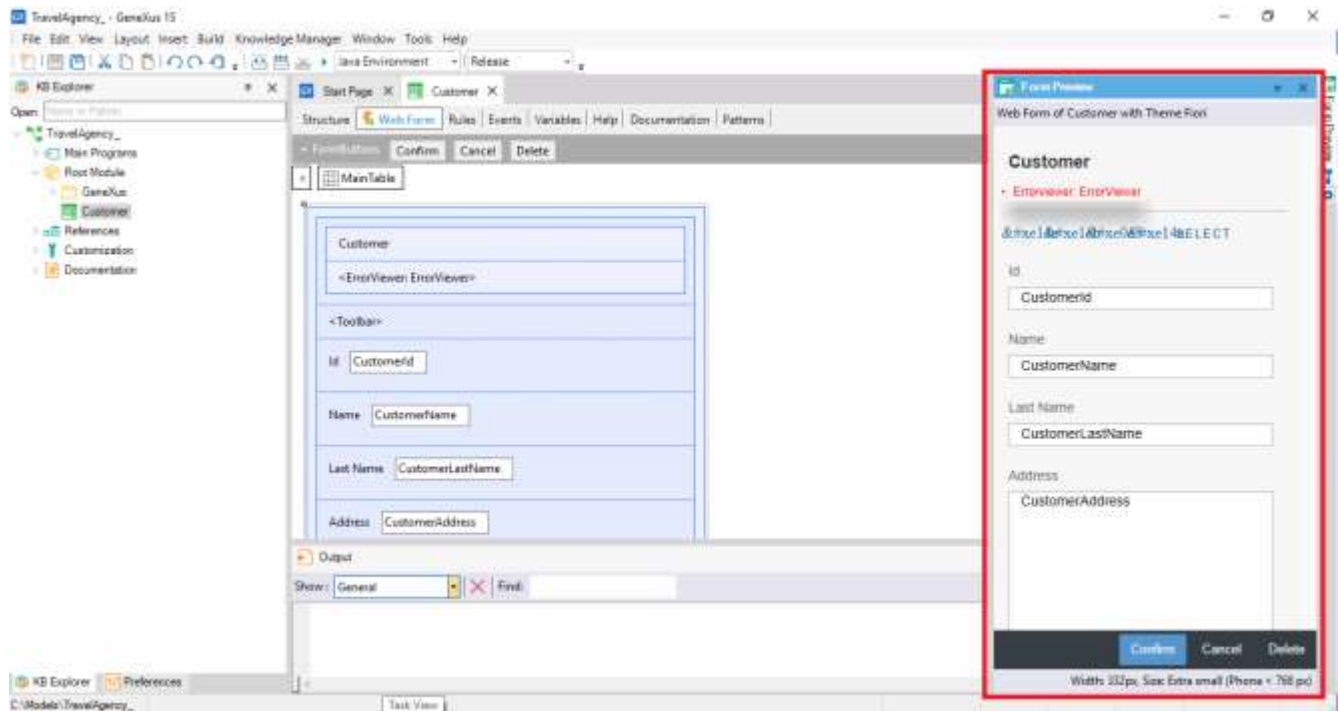
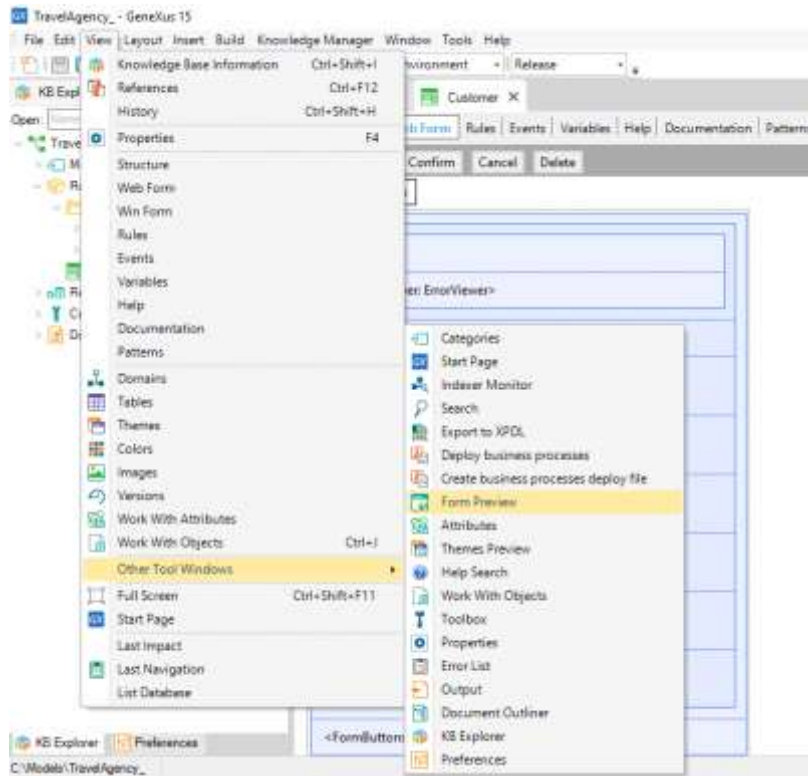
Seleccionemos ahora la sección "Web Form":

The screenshot shows the GeneXus development environment with the 'Web Form' tab selected. The interface displays a form structure for a 'Customer' entity. The form is organized into a table-like structure with the following sections:

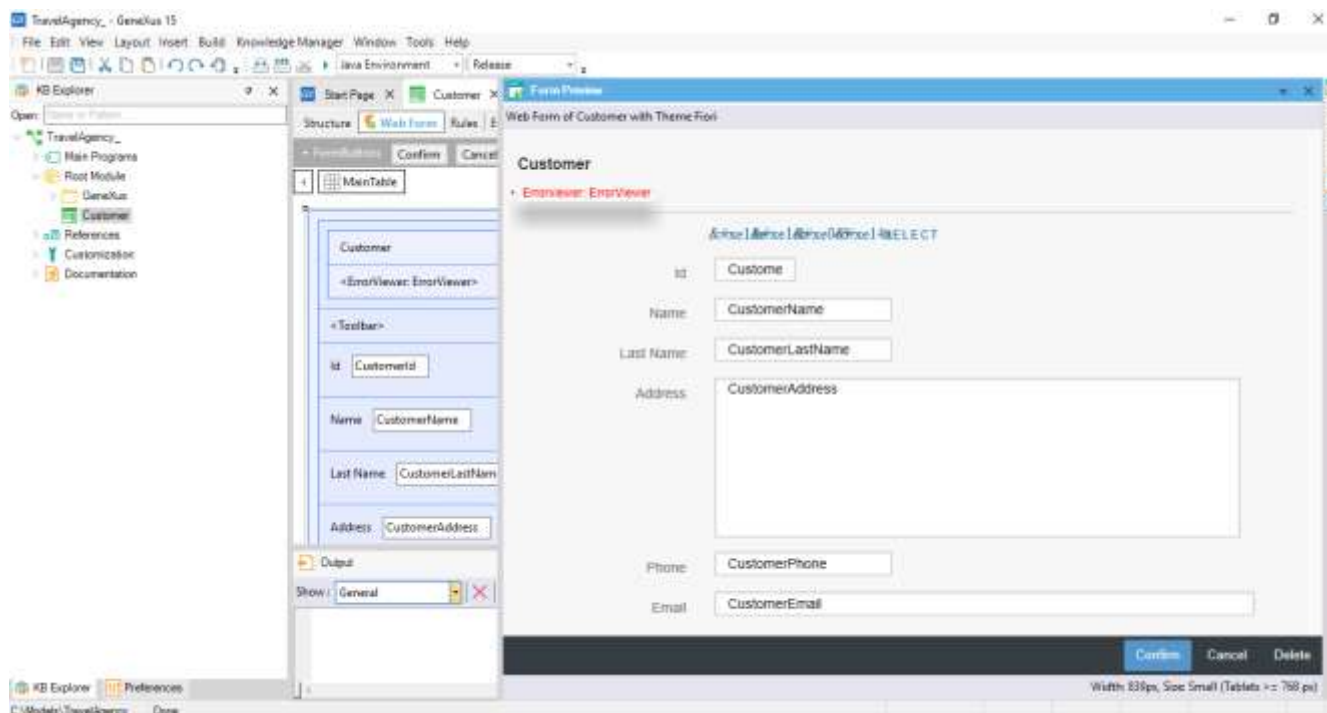
- Customer** (Header section)
- <ErrorViewer: ErrorViewer>** (Error handling section)
- <Toolbar>** (Toolbar section)
- Id** (Field: CustomerId)
- Name** (Field: CustomerName)
- Last Name** (Field: CustomerLastName)
- Address** (Field: CustomerAddress)
- Phone** (Field: CustomerPhone)
- Email** (Field: CustomerEmail)
- <FormButtons>** (Form buttons section)

Y vemos que dado que nuestra aplicación es una aplicación web, **GeneXus diseñó automáticamente acorde a la estructura definida** un formulario Web, para que, a través del mismo, los usuarios puedan realizar inserciones, modificaciones y eliminaciones de clientes.

Contamos con una herramienta para poder ver cómo lucirá el form web en ejecución de acuerdo al tamaño de la pantalla. Abrimos esa ventana a través de View / Other Tool Windows / Form Preview:



Observemos que aquí abajo nos indica el tamaño de pantalla para el que el form luzca de esta manera, donde podemos ver que todos los controles se alinean a la izquierda, con las etiquetas arriba. Se trataría de un teléfono (tamaño de pantalla extra small). Si agrandamos la pantalla para que el tamaño sea Small, como el de una tablet:



Podemos ver cómo los controles se adaptan para mostrar la información de acuerdo a ese tamaño (esta vez vemos las etiquetas a la izquierda de los campos, y con una sangría). Es decir, las aplicaciones web serán “responsivas”, o dicho de otro modo, se utiliza “responsive web design”.

Ya queremos probar esta transacción en ejecución. ¿Cómo hacemos?