

## Definiendo reglas

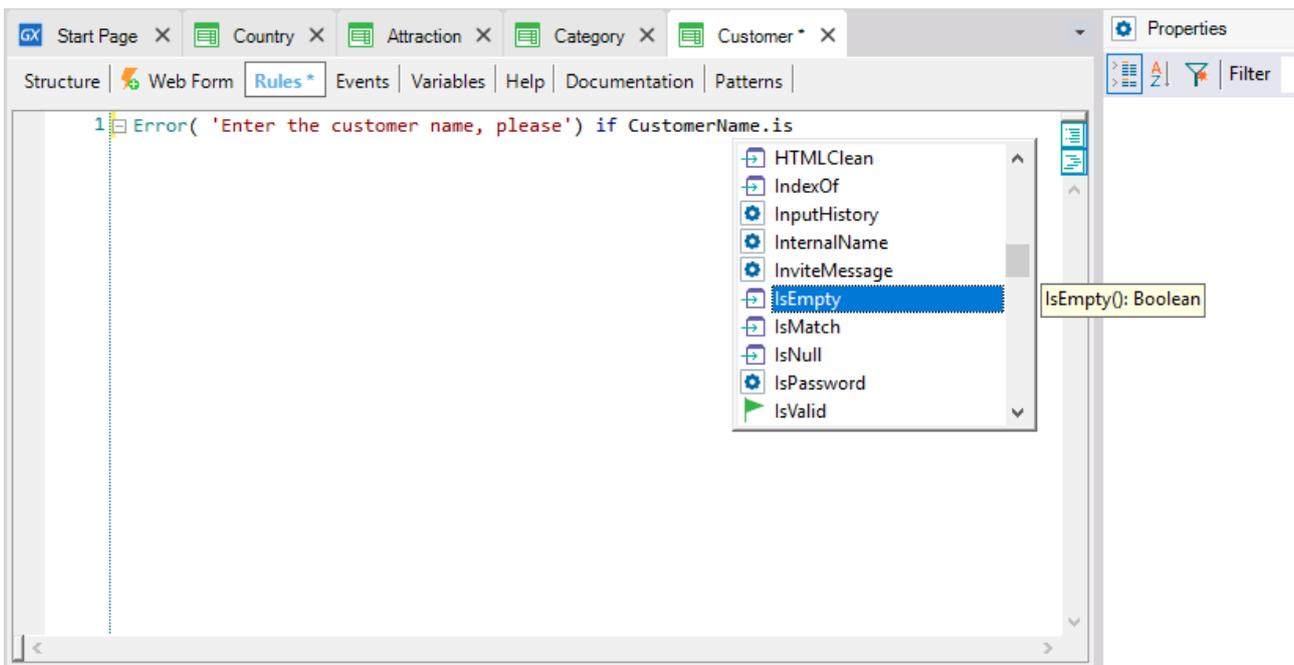
Además de todos los controles automáticos que GeneXus incluye en las aplicaciones que genera, hay ciertos controles específicos que los usuarios desean efectuar.

En las transacciones, las reglas que deben cumplirse, o los controles que nos solicitan efectuar, se definen en la sección de Rules.

Si por ejemplo un requisito es **no permitir almacenar clientes sin nombre....** contamos con una regla llamada **Error**, que nos permitirá controlar esto.

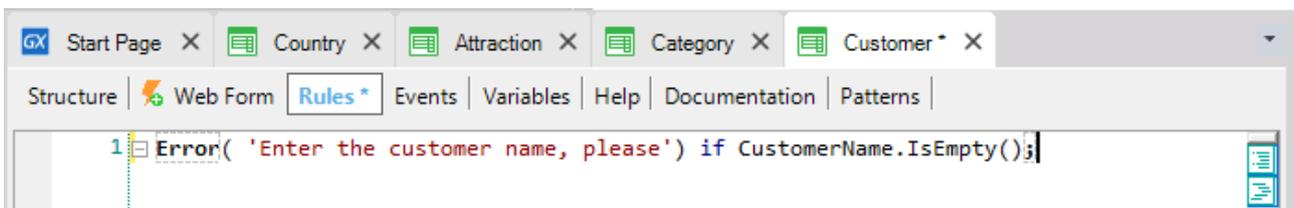
Escribimos "Error", paréntesis, y entre comillas digitamos el texto que queremos que se visualice cuando el usuario intente dejar un nombre de cliente sin completar... cerramos el paréntesis... y solamente nos está faltando indicar **cuál es la condición que se debe dar** para que se despliegue el texto.

La condición es que el atributo **CustomerName** esté vacío... entonces escribimos "if CustomerName", punto,



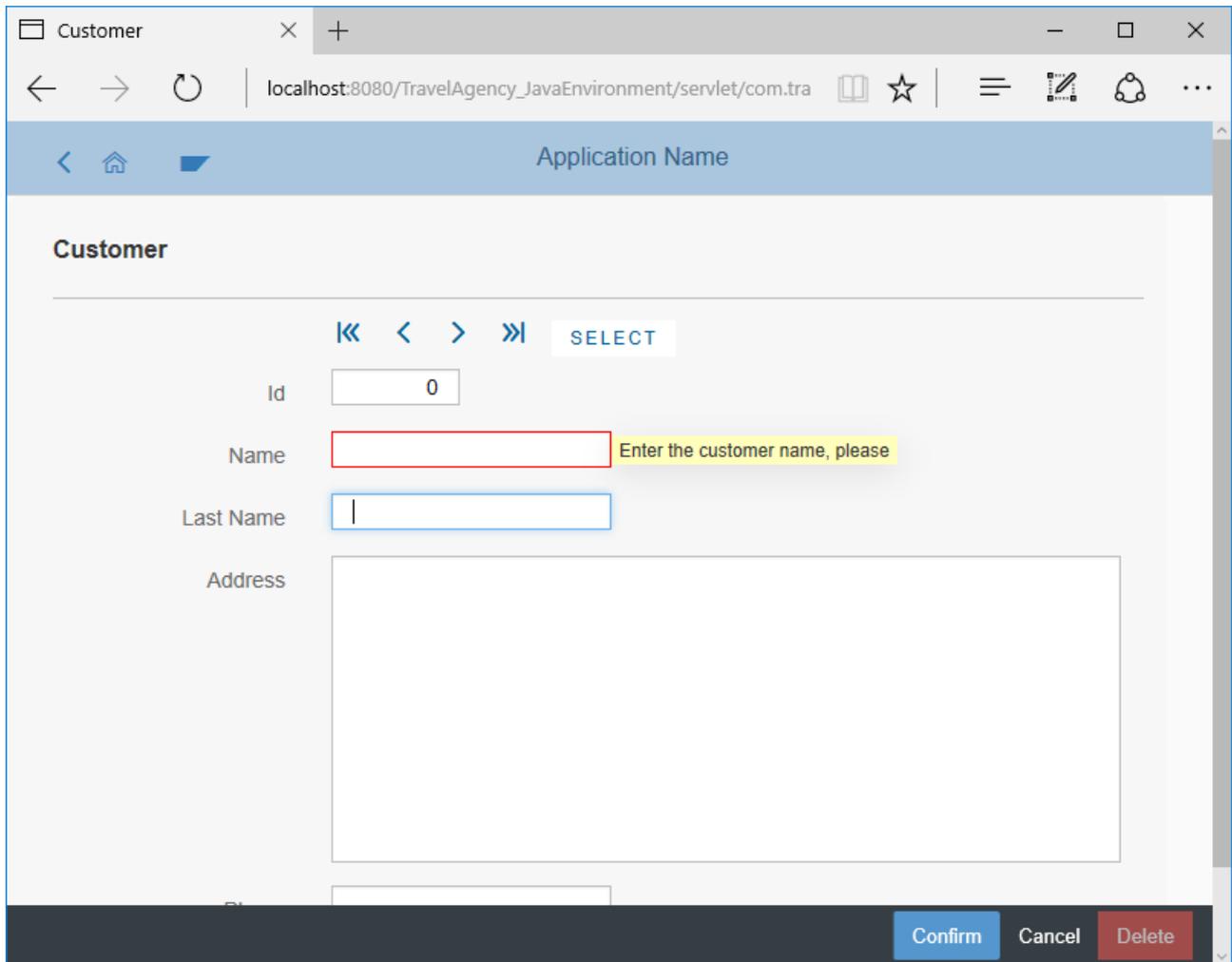
y aquí seleccionamos: IsEmpty

Toda regla que definamos debe finalizar con un punto y coma, así que lo incluimos.



Salvamos.... y presionamos F5, para ver en ejecución el funcionamiento de esta regla definida.

Ejecutamos la transacción Customer y si dejamos el nombre del cliente vacío y salimos del campo,



The screenshot shows a web browser window with the URL `localhost:8080/TravelAgency_JavaEnvironment/servlet/com.tra`. The page title is "Application Name". The main content area is titled "Customer" and contains a form with the following fields:

- Id**: A text input field containing the value "0".
- Name**: A text input field that is empty and highlighted with a red border. A yellow tooltip with the text "Enter the customer name, please" is visible next to it.
- Last Name**: A text input field that is empty.
- Address**: A large, empty text area.

At the bottom right of the form, there are three buttons: "Confirm" (blue), "Cancel" (grey), and "Delete" (red).

... aparece el texto que definimos.

La regla **Error**, aunque permite pasar al campo siguiente incluso cuando la condición se cumple, **no permitirá grabar** ... [probarlo] Este es el comportamiento por defecto. Puede modificarse para que la regla error nunca permita editar el siguiente campo. Es a través de una propiedad que no veremos aquí.

Si fuera requisito impedir que el apellido del cliente quede vacío, habría que definir una regla análoga. Así que copiemos y peguemos la definición de esta regla... modifiquemos el texto name por **lastname** y el atributo involucrado también:

```
1 Error( 'Enter the customer name, please')
2   if CustomerName.IsEmpty();
3
4 Error( 'Enter the customer last name, please')
5   if CustomerLastName.IsEmpty();
6 |
```

Pulsamos F5... ejecutamos Customer... dejamos el nombre del cliente vacío... sale el error asociado a que el nombre se dejó vacío... ingresamos Paul... intentamos dejar el apellido vacío... y sale el error asociado a que el apellido se dejó sin completar.

Customer

Id

Name

Last Name  Enter the customer last name, please

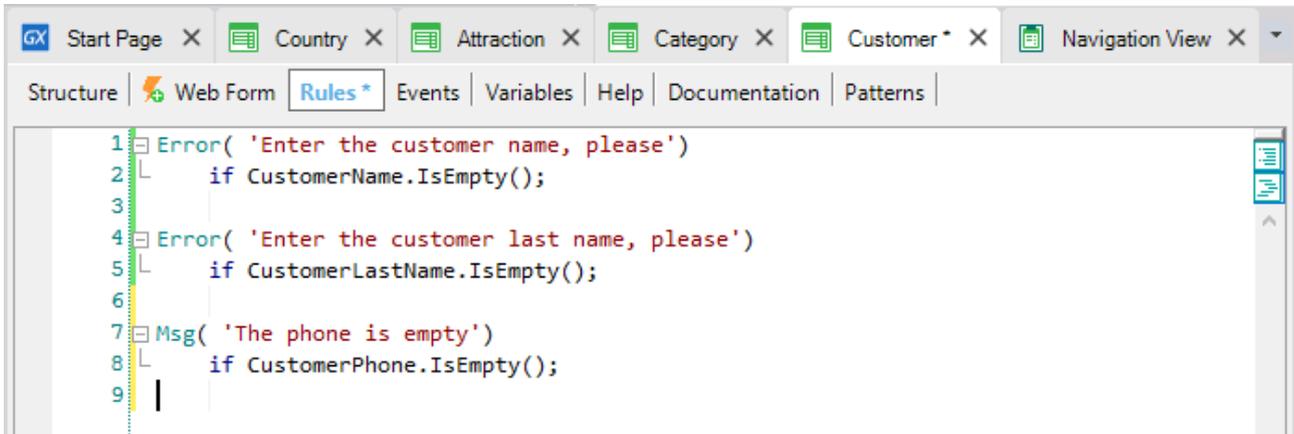
Address

Confirm Cancel Delete

Ahora bien... contamos con otra regla que tiene prácticamente la misma sintaxis que **Error**... su nombre es **Message**... y la única diferencia que presenta con respecto a **Error** es que de cumplirse la condición, sale el mensaje como aviso o advertencia, **y se podrá grabar el registro de todas formas**.

Si por ejemplo queremos avisar que han dejado sin ingresar el teléfono del cliente, pero sin obligar a ingresarlo, podemos definir una regla **Message**, paréntesis, luego el texto entre comillas simples (o dobles)... 'The phone is empty'

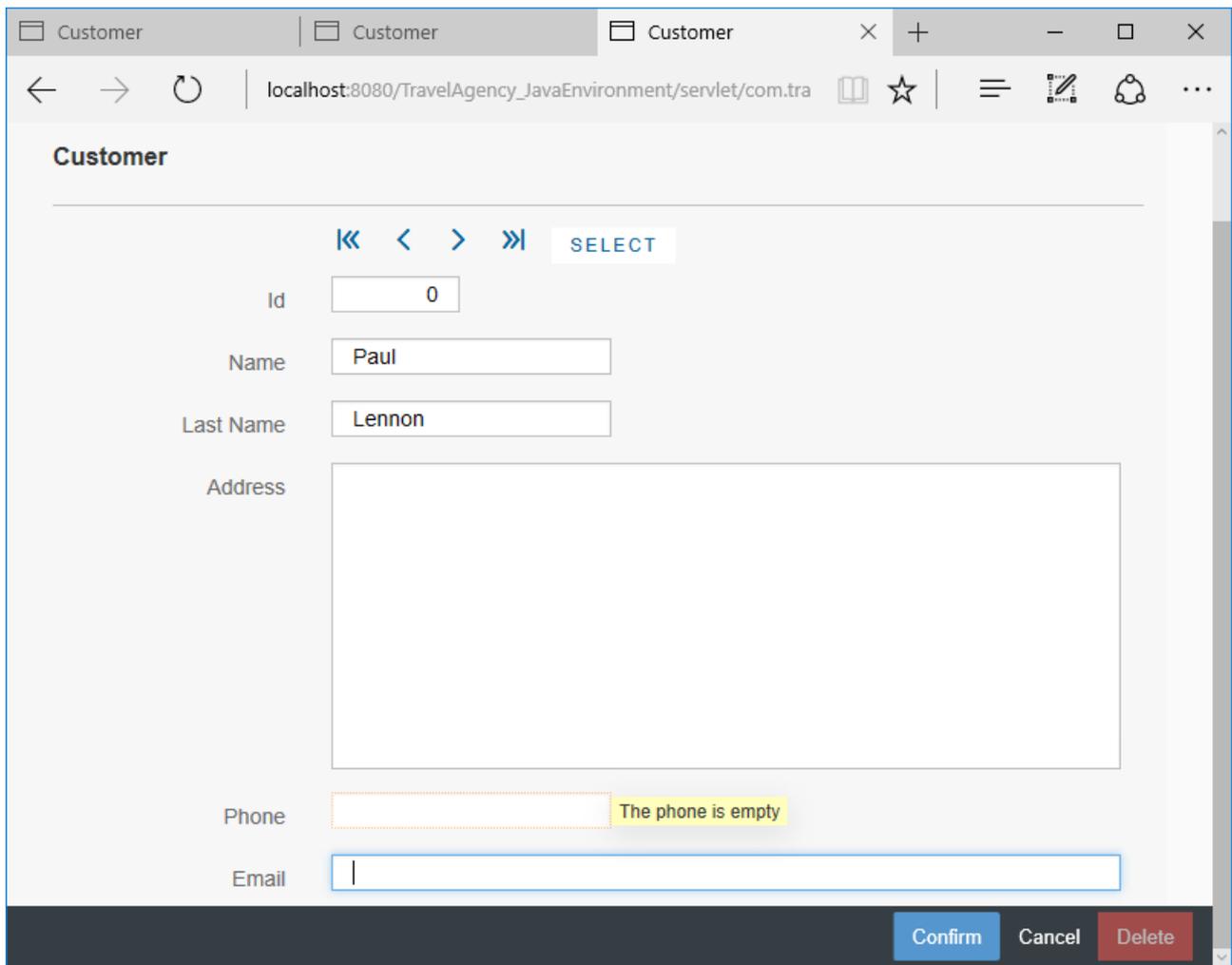
cerramos el paréntesis... y a continuación definimos la condición para que se ejecute la regla: "if CustomerPhone" ... punto... IsEmpty. Y punto y coma para finalizar la definición de la regla.



```
1 Error( 'Enter the customer name, please')
2   |   if CustomerName.IsEmpty();
3
4 Error( 'Enter the customer last name, please')
5   |   if CustomerLastName.IsEmpty();
6
7 Msg( 'The phone is empty')
8   |   if CustomerPhone.IsEmpty();
9 |
```

Presionemos F5 para probar esta funcionalidad...

Vemos que si dejamos sin ingresar el teléfono e intentamos salir del campo:



...sale el mensaje que definimos, y si ahora queremos grabar, lo conseguimos.

Este conjunto de reglas puede ser escrito en cualquier otro orden y el resultado en ejecución será exactamente el mismo, ya que GeneXus decide cuándo cada una de las reglas debe ser disparada (por ejemplo, cuando el usuario abandona cada campo involucrado, etcétera).

Por supuesto, GeneXus ofrece más reglas para definir diferentes tipos de validaciones y acciones. Cada transacción puede requerir definir sus propias reglas de comportamiento.