Eventos en Transacciones

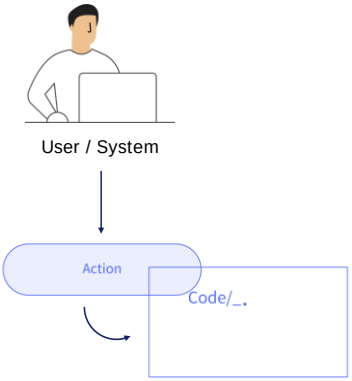
GeneXus

Avancemos un poco más en el conocimiento del objeto Transacción.

Ya sabemos que a partir de la estructura definida en una transacción GeneXus crea automáticamente su form, y que a través de la declaración de reglas podemos definir su comportamiento.

Vamos a hablar ahora de los eventos en las transacciones.

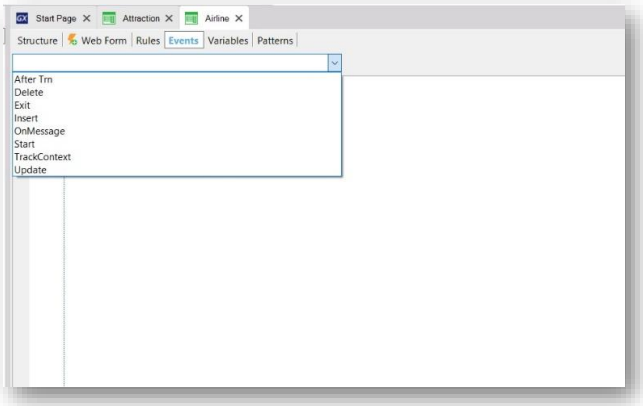
Evento: Concepto



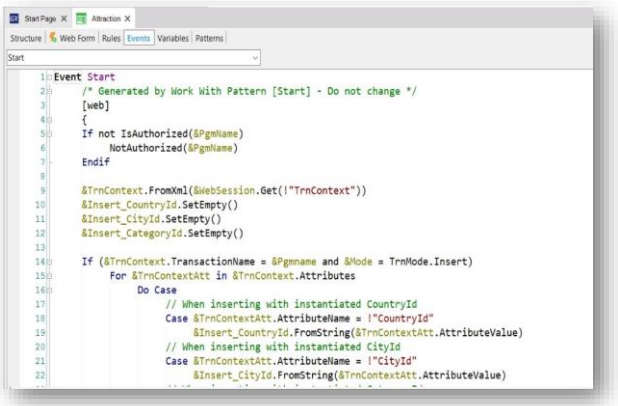
Un evento es una acción por parte del usuario, o del sistema, que permite activar un determinado código como respuesta a dicha acción.

Eventos en Transacciones

En transacción común.



En transacción con pattern WW for Web aplicado.



Observemos en primer lugar la transacción Airline. Si vamos al sector de sus eventos, vemos que está en blanco, y que podemos editar y programar los eventos que sean necesarios.

Observemos que en el sector Eventos de la transacción Attraction sí hay código declarado, y que si bien no vamos a analizarlo, es importante observar que ha sido generado automáticamente por la aplicación del pattern Work With for Web.

Eventos disponibles

Start: Define una acción a ser ejecutada cuando se abre y comienza a trabajarse con la transacción.

Track Context: Permite programar qué acción realizar cuando hay algún cambio en el contexto de la ejecución de la transacción.

OnMessage: Relacionado con las notificaciones web que permiten realizar acciones en tiempo real.

Insert, Update y Delete: Asociados al concepto de actualización de las Transacciones dinámicas Estos eventos aplican a un caso especial de uso de las transacciones.

Exit: Permite indicar qué acciones realizar cuando ya se termina la ejecución de la transacción, o sea, cuando la transacción ya se cierra.

After Trn: Permite indicar qué acción realizar luego de cada ciclo de ejecución de la transacción, o sea, inmediatamente después de efectuado el Commit.

Comencemos por el Evento **Start**: Este evento define una acción a ser ejecutada cuando se abre y comienza a trabajarse con la transacción. Es un evento del sistema, y generalmente se utiliza para asignar valores a variables que luego serán utilizadas durante la ejecución de la transacción

Sigamos con el evento **TrackContext**: Este evento permite programar qué acción realizar cuando hay algún cambio en el contexto de la ejecución de la transacción. Por ejemplo, este evento puede estar atento a la posición del cursor en un determinado control, y ejecutar entonces el código aquí programado.

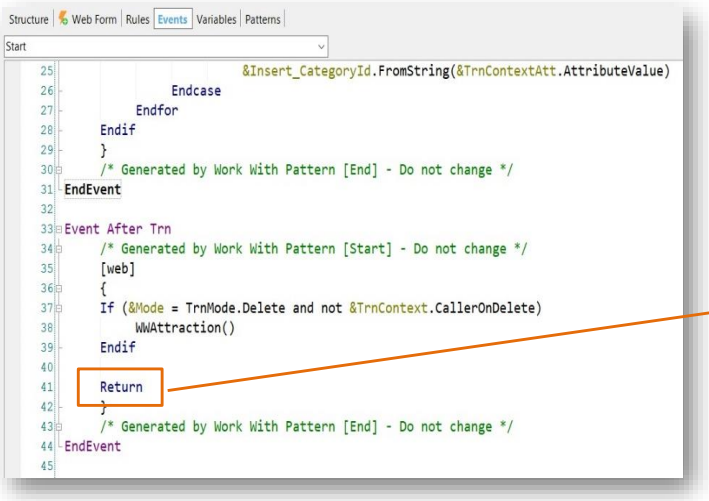
El evento **OnMessage:** está relacionado con las notificaciones web que permiten realizar acciones en tiempo real.

Los eventos **Insert, Update y Delete** están asociados al concepto de actualización de las Transacciones dinámicas que no veremos en este curso. Por tal motivo solamente diremos que estos eventos aplican a un caso especial de uso de las transacciones.

El evento **Exit** permite indicar qué acciones realizar cuando ya se termina la ejecución de la transacción, o sea, cuando la transacción ya se cierra.

Por último, vamos a detenernos en el evento **After Trn**. **Este evento permite indicar qué acción realizar luego de cada ciclo de ejecución de la transacción, o sea, inmediatamente después de efectuado el Commit.** Si recordamos lo ya estudiado sobre los momentos de disparo de reglas, podemos ver que este evento AfterTrn se ejecuta igual que el momento de disparo on AfterComplete..

Evento AfterTrn



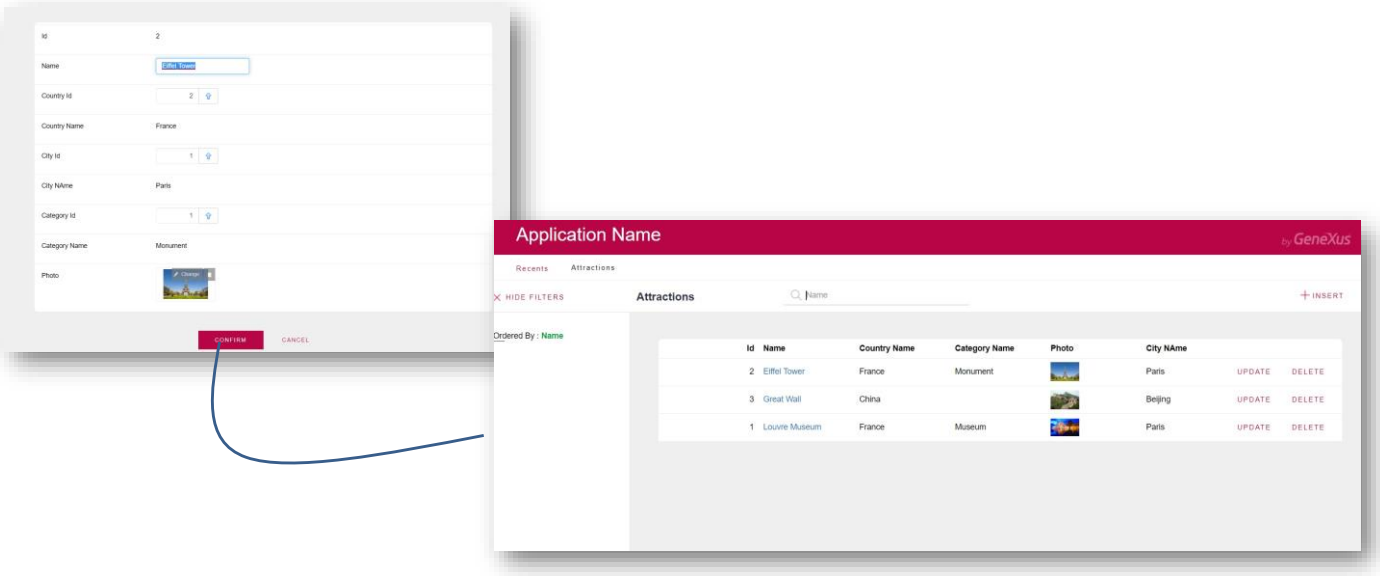
Al momento de insertar, eliminar o modificar una atracción turística, se vuelve automáticamente a la pantalla principal Work With Attractions.

Volvamos a la transacción Attraction, y observemos el código declarado en su evento AfterTrn.

Vemos que la aplicación del pattern Work With for Web agregó código en este evento AfterTrn, agregando en particular, el comando Return.

Esto hace que, en ejecución, al momento de insertar, eliminar o modificar una atracción turística, se vuelva automáticamente a la pantalla principal Work With Attractions. Recordemos que el Commit se realiza por cada ciclo de trabajo con el form de la transacción, y por lo tanto se dispara el evento AfterTrn y se ejecutará este comando Return.

Evento AfterTrn en ejecución



En ejecución, supongamos que realizamos un cambio en el nombre. Al momento de presionar el botón Confirm, se termina un ciclo de trabajo con el form de la transacción, se dispara el Commit y posteriormente el evento AfterTrn.
Por lo tanto, se ejecutará el comando Return declarado en este evento y vamos a volver a la pantalla inicial Work With Attractions.

Tal vez nos preguntemos ¿Por qué no indicamos el Return condicionándolo al momento on AfterComplete? Porque es un comando y no una regla. Entonces debemos declararlo en un evento, y el evento que se dispara luego del Commit, es el evento AfterTrn.

Para finalizar, tengamos presente, que si necesitamos resolver alguna funcionalidad que requiera agregar código en algún evento, y vemos que ese evento ya tiene código que ha sido generado automáticamente por GeneXus, entonces nuestro código deberemos declararlo fuera de las marcas que indican el código que es automáticamente mantenido por GeneXus.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications