Eventos em Transações

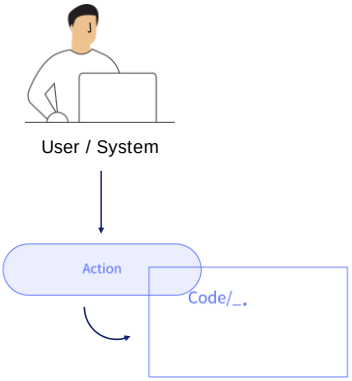
GeneXus

Vamos avançar um pouco mais no conhecimento do objeto Transação.

Já sabemos que, a partir da estrutura definida em uma transação, GeneXus cria automaticamente seu form e que, através da declaração de regras podemos definir seu comportamento.

Vamos falar agora sobre os eventos em transações.

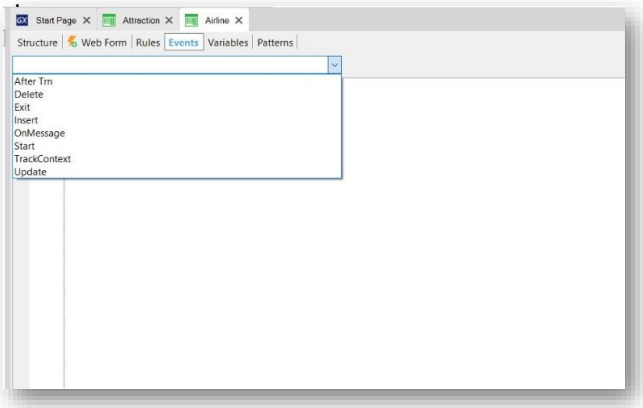
Evento: Conceito



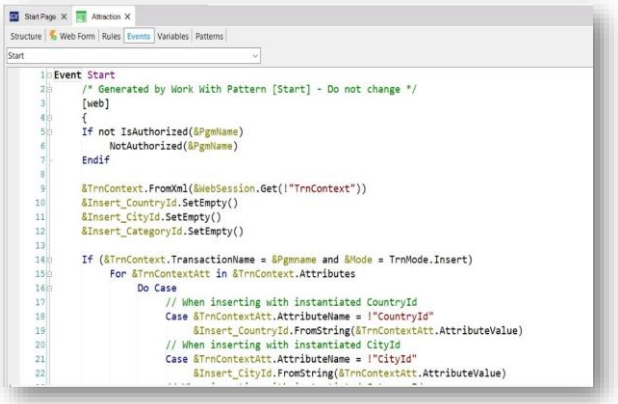
Um evento é uma ação do usuário ou do sistema que permite ativar um determinado código como resposta à referida ação.

Eventos em Transações

Em uma transação comun.



Em uma transação com o padrão Work Wtih for Web.



Observemos em primeiro lugar a transação Airline. Se formos para o setor de seus eventos, vemos que está em branco e que podemos editar e programar os eventos necessários.

Observemos que aqui sim existe código declarado e que, embora não vamos analisá-lo, é importante observar que foi gerado automaticamente pela aplicação do pattern Work With for Web.

Eventos disponiveis

Start: Define uma ação a ser executada quando é aberta e se começa a trabalhar com a transação.

Track Context: Permite programar qual ação realizar quando houver alguma alteração no contexto da execução da transação.

OnMessage: Relacionado coma as notificações web que permitem realizar ações em tempo real.

Insert, Update y Delete: Associados ao conceito de atualização das Transações dinâmicas que não veremos neste curso.

Exit: Permite indicar quais ações realizar quando for concluída a execução da transação, ou seja, quando a transação já estiver fechada.

After Trn: Permite indicar que ação realizar após cada ciclo de execução da transação, ou seja, imediatamente depois de efetuado o Commit.

Comecemos pelo Evento **Start**: Este evento define uma ação a ser executada quando é aberta e se começa a trabalhar com a transação. É um evento do sistema e geralmente é utilizado para atribuir valores a variáveis que serão então utilizadas durante a execução da transação.

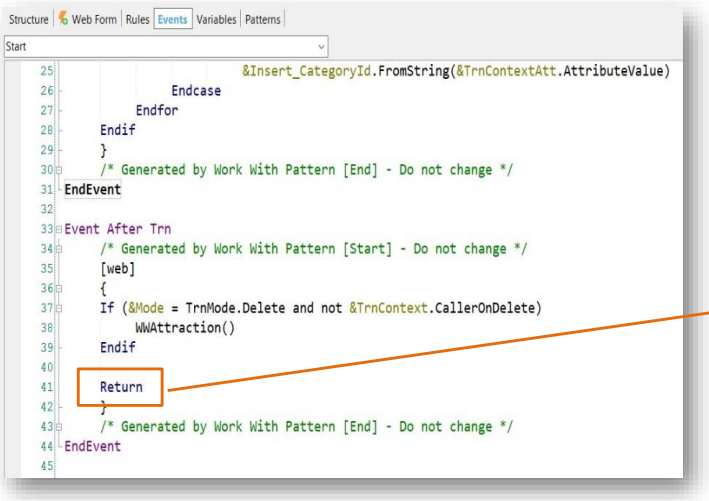
Vamos continuar com o evento **TrackContext**: Este evento permite programar qual ação realizar quando houver alguma alteração no contexto da execução da transação. Por exemplo, este evento pode estar atento à posição do cursor em um determinado controle e executar o código programado aqui.
\
O evento **OnMessagem** está relacionado coma as notificações web que permitem realizar ações em tempo real.

Os eventos **Insert, Update e Delete** estão associados ao conceito de atualização das Transações dinâmicas que não veremos neste curso. Por esse motivo, diremos apenas que estes eventos se aplicam a um caso especial de uso das transações.

O evento **Exit** permite indicar quais ações realizar quando for concluída a execução da transação, ou seja, quando a transação já estiver fechada.

Por último, vamos nos deter no evento **After Trn. Este evento permite indicar que ação realizar após cada ciclo de execução da transação, ou seja, imediatamente depois de efetuado o Commit.** Se lembrarmos do que já foi estudado sobre os momentos de disparo de regras, podemos ver que este evento AfterTrn é executado como o momento de disparo on AfterComplete.

Evento AfterTrn



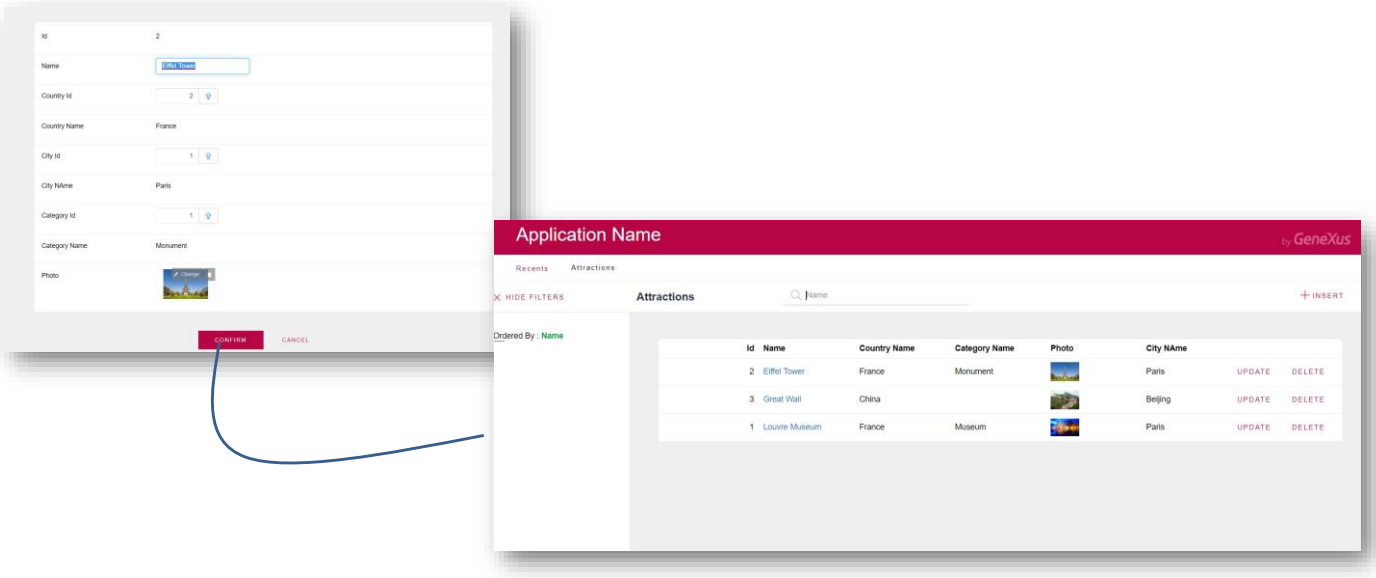
Ao inserir, excluir ou modificar uma atração turística, volte automaticamente para a tela principal Work With Attractions.

Voltemos à transação Attraction, e observemos o código declarado em seu evento AfterTrn.

Vemos que a aplicação do pattern Work With for Web adicionou código neste evento AfterTrn, adicionando em particular, o comando Return.

Isto faz com que, em execução ao inserir, excluir ou modificar uma atração turística, volte automaticamente para a tela principal Work With Attractions. Lembremos que o Commit é realizado para cada ciclo de trabalho com o form da transação e, portanto, é disparado o evento AfterTrn e será executado este comando Return.

Evento AfterTrn em execución



Suponhamos que façamos uma alteração no nome. No momento de pressionar o botão Confirm, é concluído um ciclo de trabalho com o form da transação, é disparado o Commit e, posteriormente, o evento AfterTrn.

Portanto, será executado o comando Return declarado neste evento e retornaremos à tela inicial Work With Attractions.

Talvez estejamos nos perguntando Por que não indicamos o Return condicionando-o ao momento on AfterComplete? Porque é um comando e não uma regra. Então, devemos declará-lo em um evento, e o evento que é disparado após o Commit é o evento AfterTrn.

Para finalizar, tenhamos em mente que, se precisamos resolver qualquer funcionalidade que exija a adição de código em algum evento, e vemos que esse evento já possui um código que foi gerado automaticamente pelo GeneXus, então nosso código deve ser declarado fora das marcas que indicam o código que é automaticamente mantido pelo GeneXus.

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