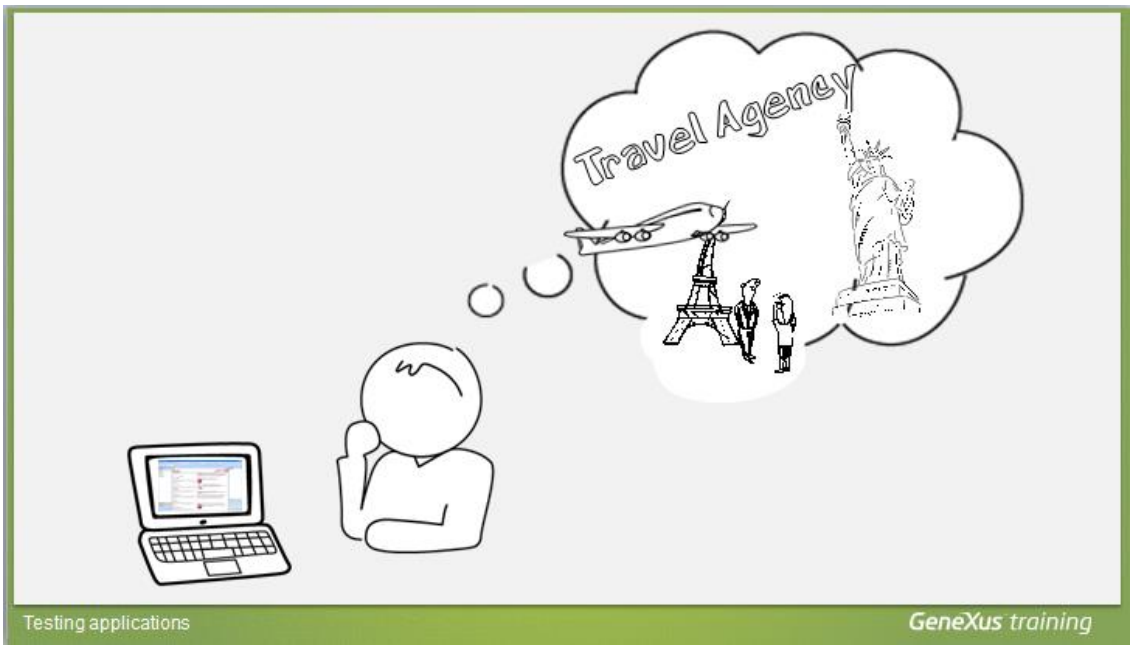
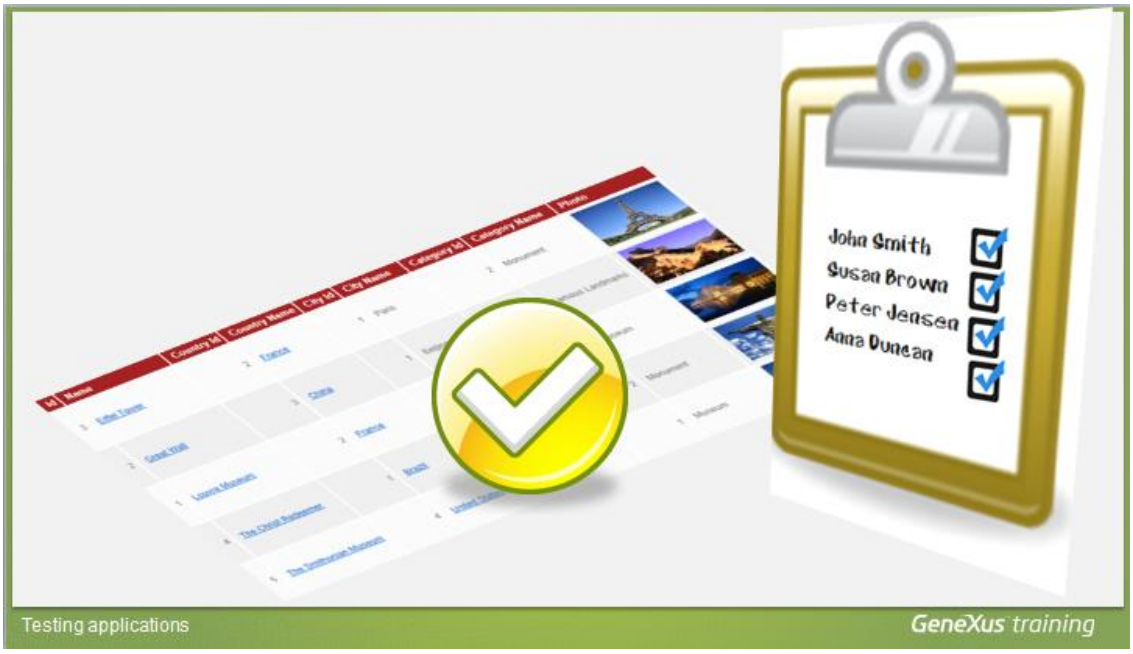
Testeando las aplicaciones (GXtest)

A medida que vamos haciendo crecer nuestra aplicación para la agencia de viajes, hemos ido agregando funcionalidades y haciendo modificaciones a cosas que habíamos implementado antes.



Sin embargo, algo que hemos omitido es volver a probar toda la aplicación luego de hacer un cambio, para asegurarnos que lo que ya teníamos funcionando, se siga comportando correctamente.



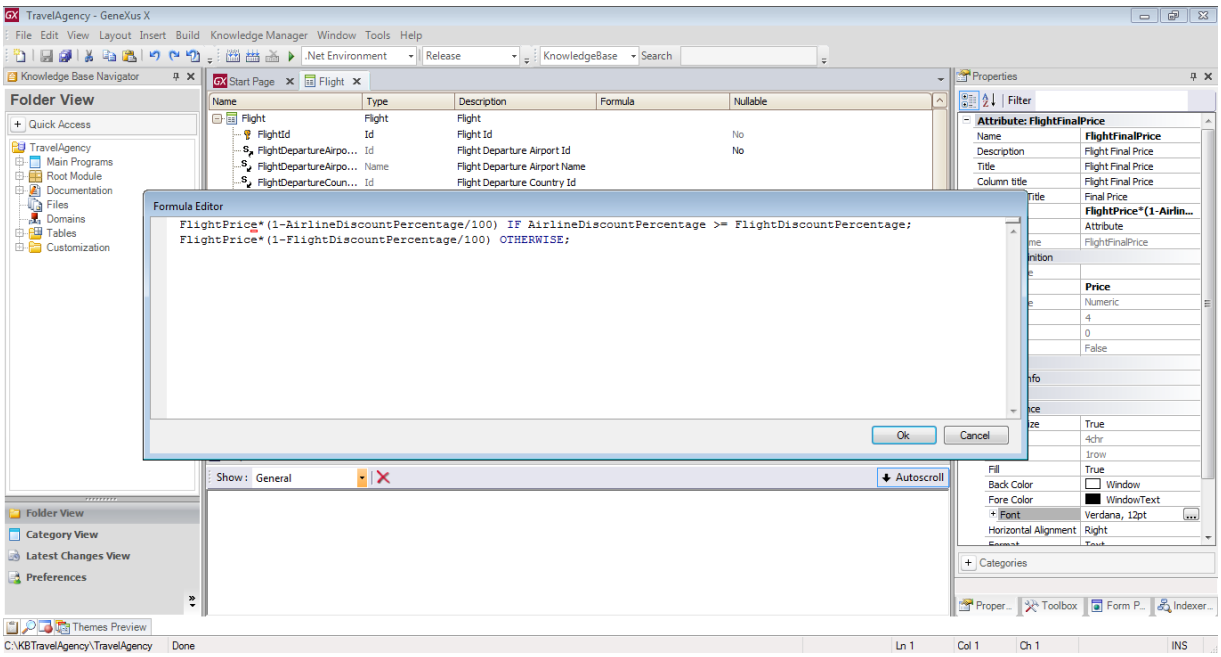
Este tipo de tarea puede volverse muy tediosa, si la aplicación crece mucho, ya que cada vez serán más las cosas a probar y sobre todo, repetir pruebas de cosas ya probadas, si bien es necesario, es bastante aburrido.

GeneXus nos ayuda en la automatización de estos test, mediante su herramienta GXtest.

GXtest nos permite grabar secuencias de operaciones para probar nuestras pantallas y luego los test se reproducen automáticamente como si fuera una persona que está ingresando datos y verificando que el sistema sigue funcionando correctamente.

Veamos un ejemplo….

Supongamos que queremos corroborar que el descuento de un vuelo se calcule correctamente. Si abrimos la transacción Flight, vemos que habíamos definido el atributo FlightFinalPrice con la siguiente fórmula…

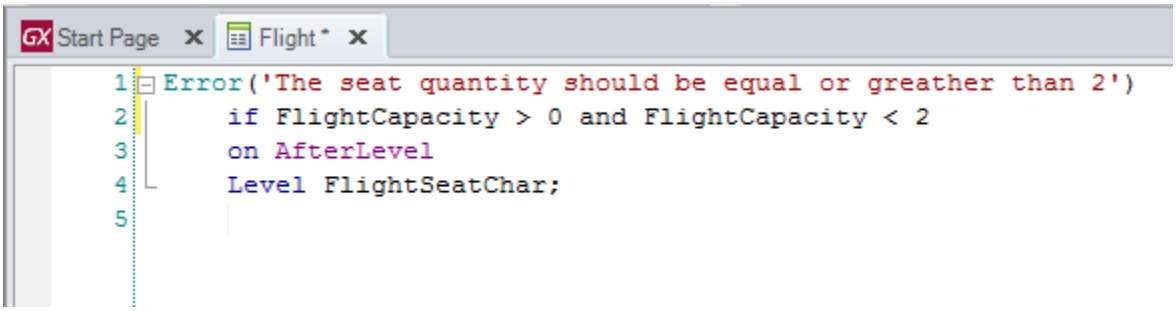


Donde establecíamos que el precio del vuelo se calcularía utilizando el mayor descuento posible, es decir, si el descuento de la aerolínea era mayor se tomaba dicho descuento y en caso contrario, se tomaba el descuento del vuelo.

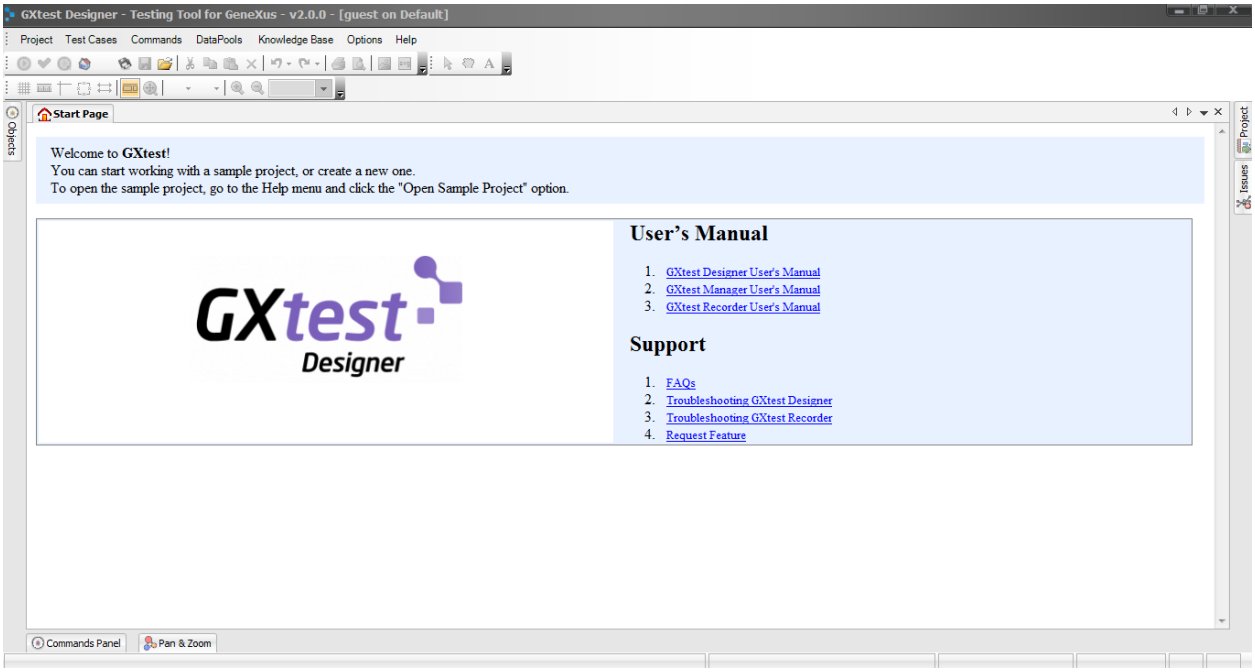
En su momento probamos esto y estuvimos de acuerdo con los resultados de los cálculos.

Vamos a probar ingresar un nuevo vuelo y aprovechemos para guardar el ingreso mediante GXtest, para repetirlo como chequeo después.

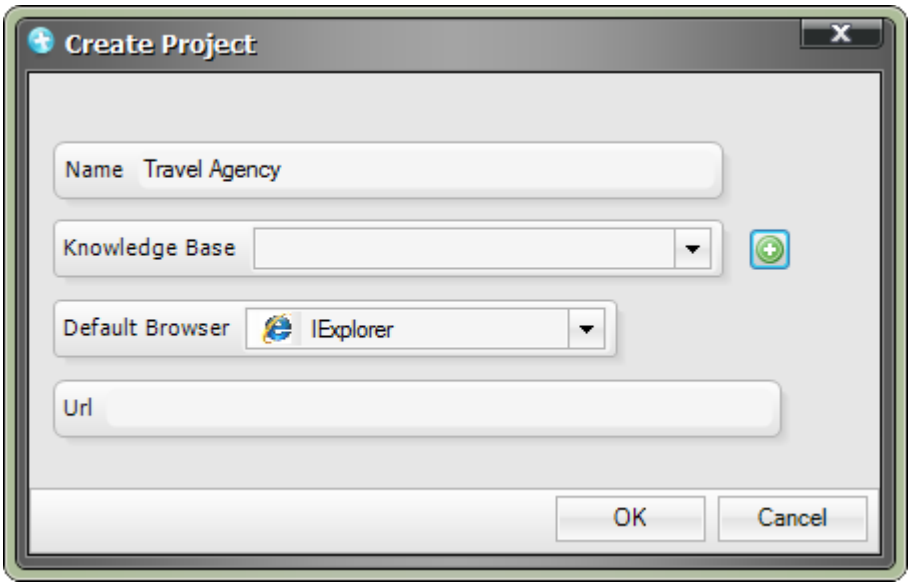
Antes, para hacer más rápida la prueba, vamos a las reglas de la transacción Flight y cambiamos este control que no nos dejaba ingresar un vuelo con menos de 8 asientos, pongamos que se pueda ingresar un vuelo con 2 asientos… y salvamos.



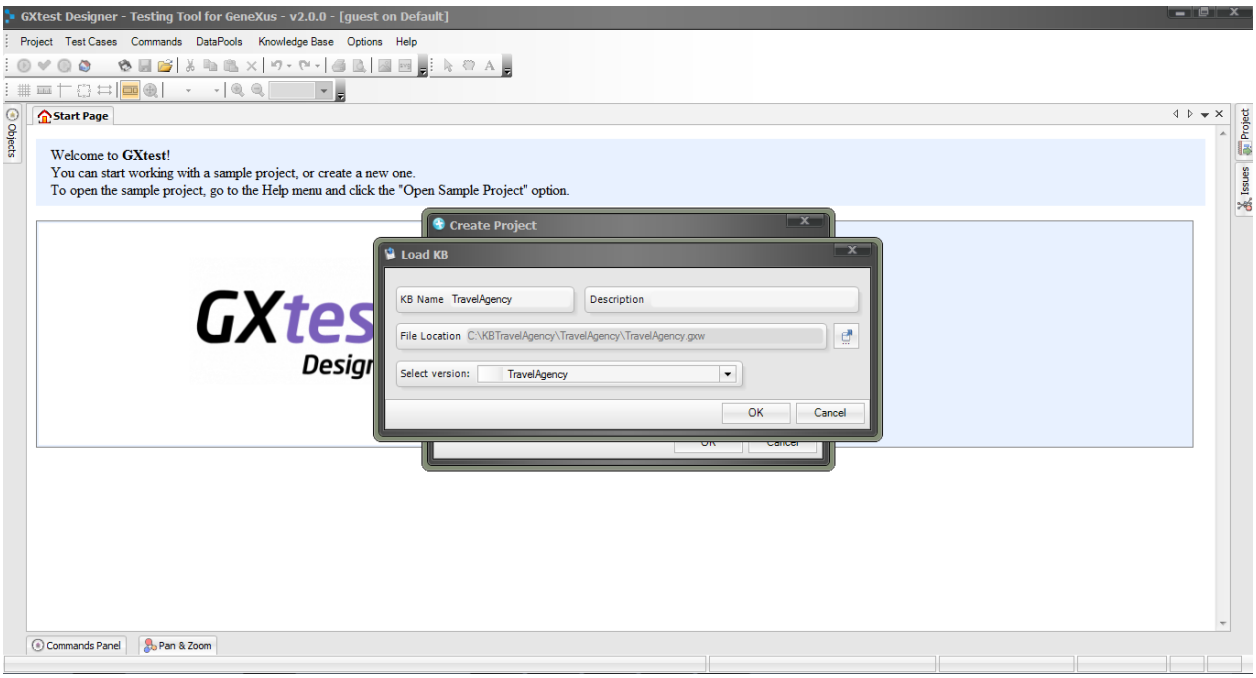
Ahora ejecutamos la herramienta Diseñador de GXtest desde nuestro acceso directo en el escritorio…



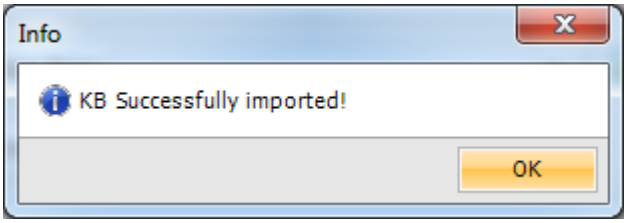
Creamos un proyecto nuevo, hacemos Project…New Project… y le damos el nombre Travel Agency.



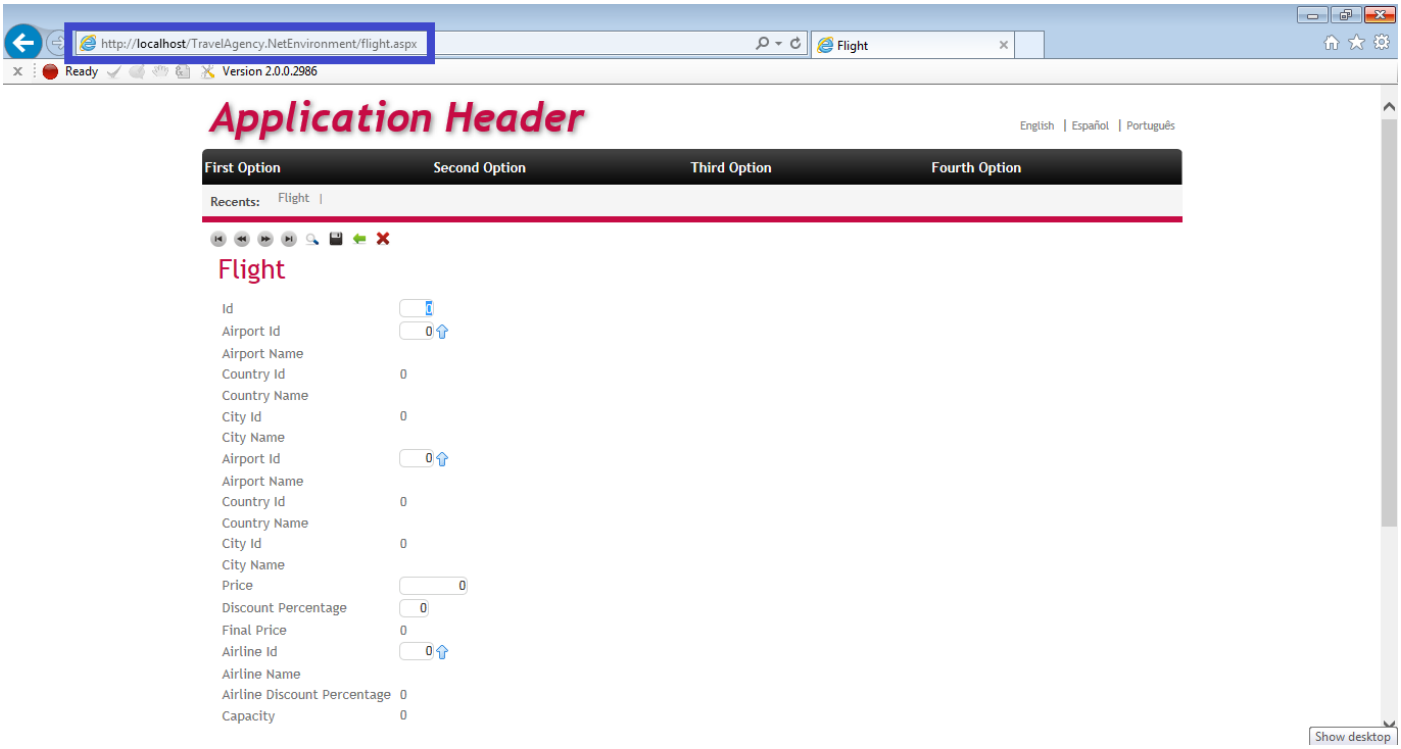
Ahora presionamos el signo de más (color verde), elegimos nuestra KB de la carpeta donde la tenemos almacenada y agregamos como descripción Travel Agency.



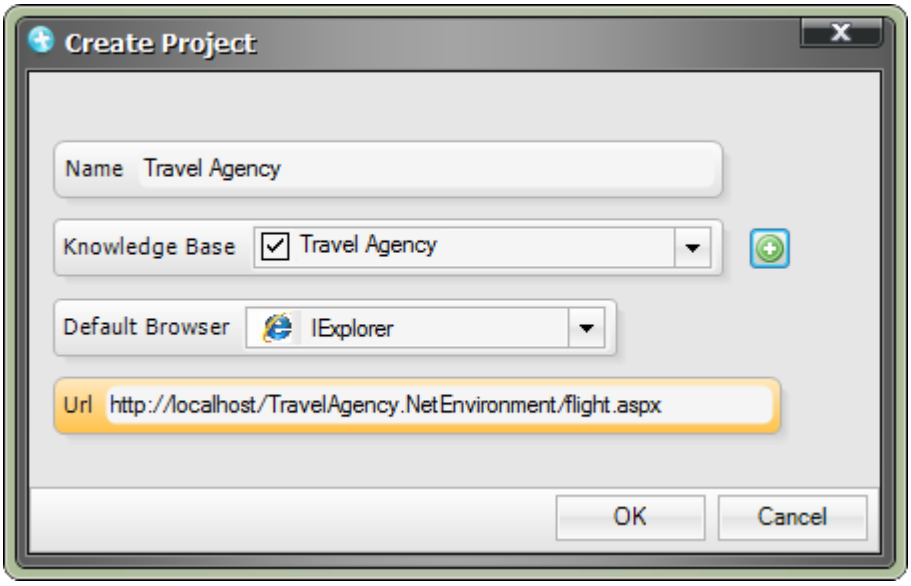
Presionamos OK y vemos que GXtest está leyendo datos que precisa de nuestra KB…Y finalmente recibimos el mensaje de que la KB se importó correctamente.



Presionamos OK… y ahora volvamos a GeneXus, damos F5 para ejecutar nuestra aplicación, abrimos la transacción Flight, copiamos la URL de misma …

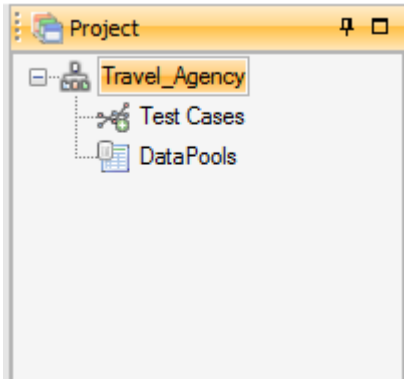


Y la pegamos en el campo URL de GXtest…

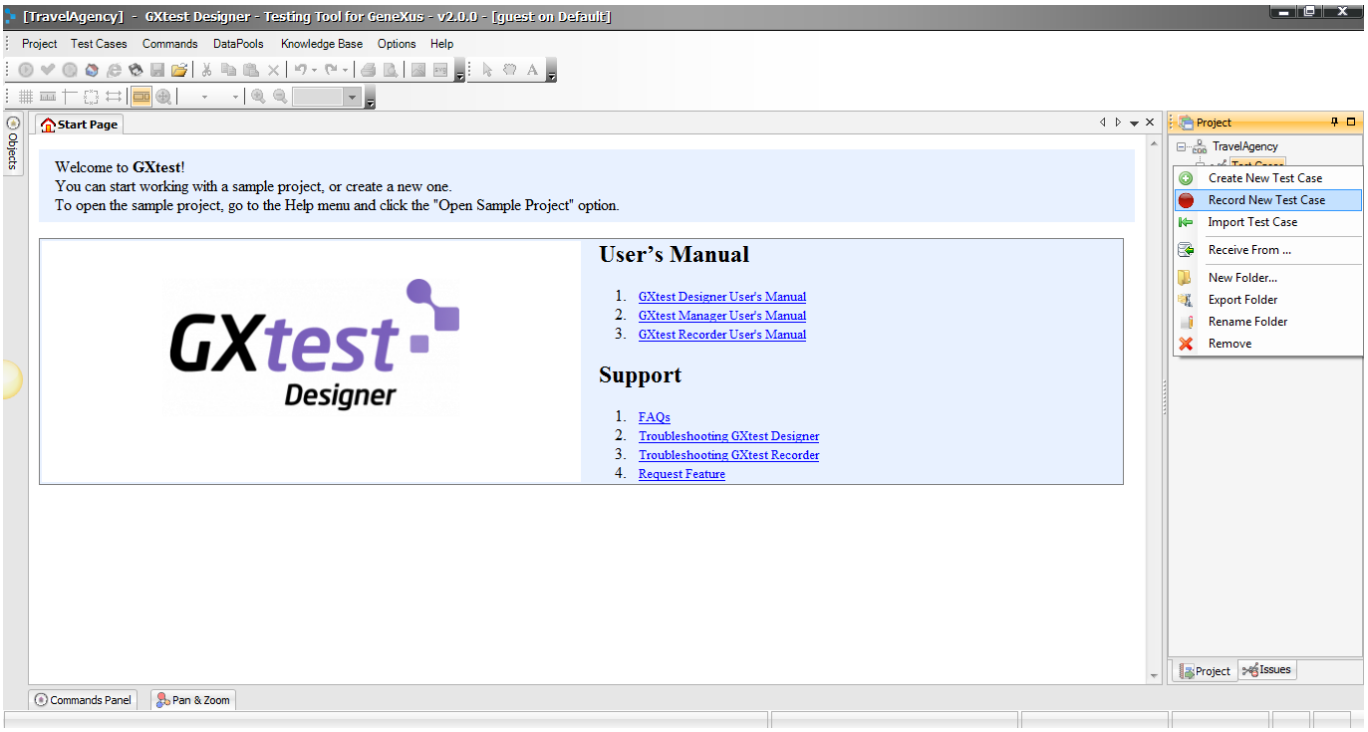


Presionamos OK.

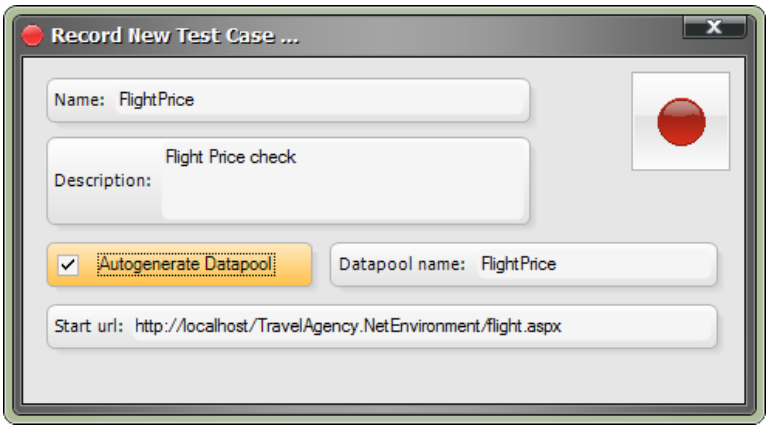
Vemos que en la parte derecha de la ventana, se creó un proyecto Travel Agency que tiene 2 partes: los casos de test y el conjunto de datos de prueba.



Damos botón derecho sobre Test Cases y elegimos Record New Test Case.



Le ponemos de nombre FlightPrice, agregamos una breve descripción…."chequear que se asigne el descuento apropiado al precio del vuelo…". Y marcamos que se autogenere el conjunto de datos.

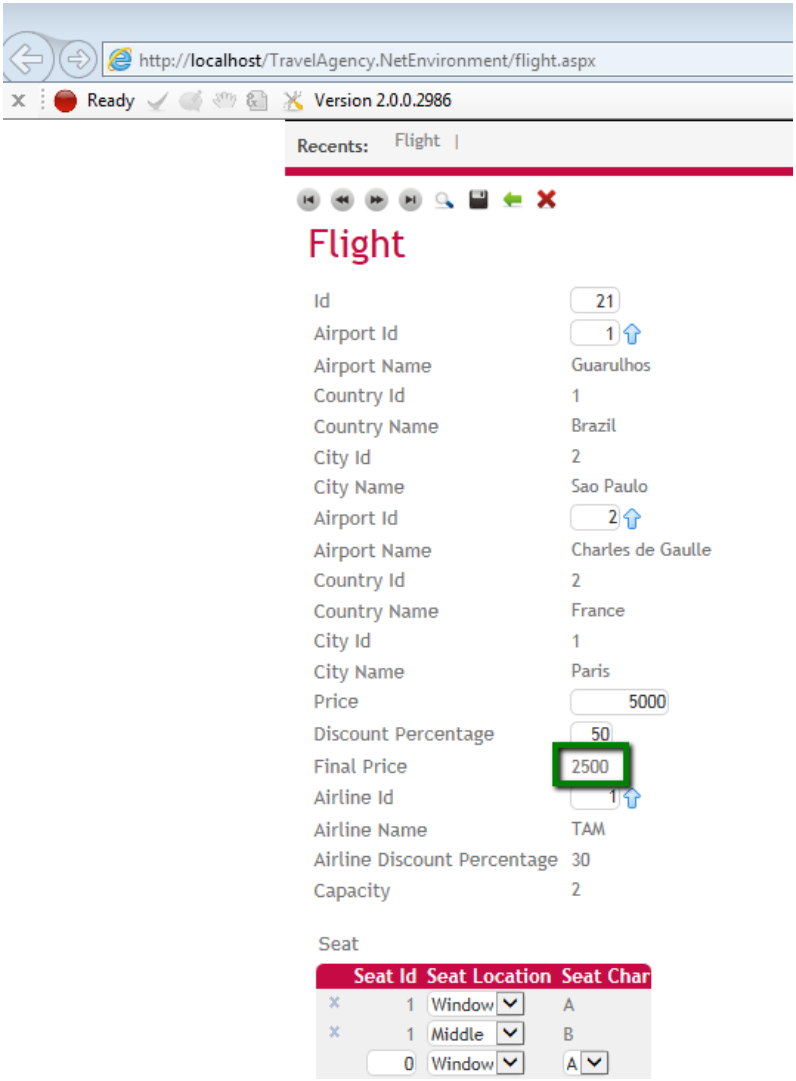


Observemos que tenemos un gran botón rojo, que vamos a presionar para comenzar la grabación de nuestro test.

Presionamos el botón… y vemos que se abre una ventana del navegador con la pantalla de la transacción Flight en ejecución.

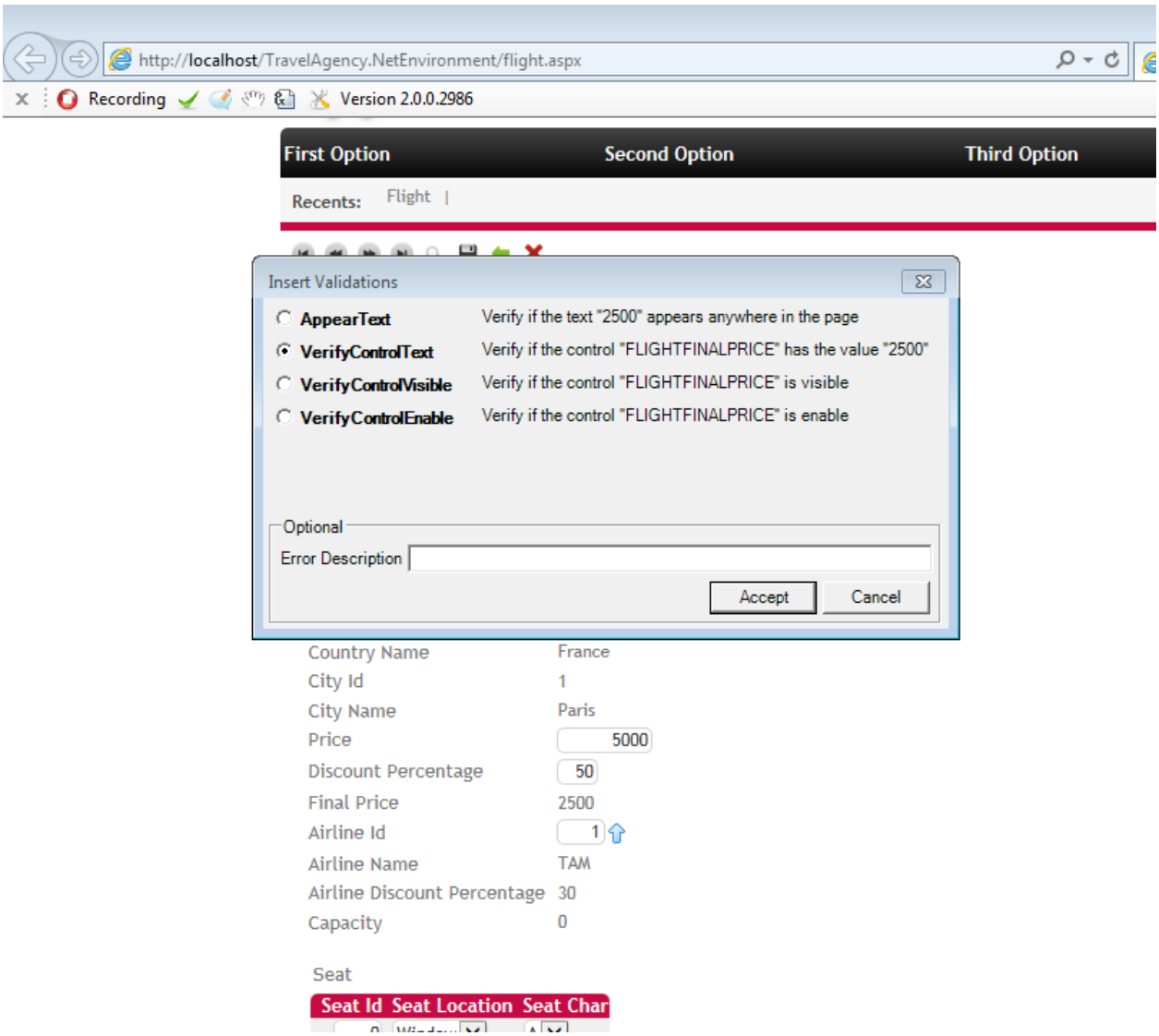
Vamos a ingresar un vuelo nuevo, así que presionamos TAB sobre el identificador ya que es autonumerado, escribimos el aeropuerto de origen: 1...Guarulhos de Sao Paulo, Brasil, y el aeropuerto de destino: 2, Charles de Gaulle en París, Francia. Ingresamos un precio de vuelo de 5000 y un descuento del vuelo del 50%... Ahora elegimos la aerolínea 1... TAM, que vemos que tiene un descuento de un 30%.

El precio del vuelo muestra 2500, por lo cual se le hizo el descuento de un 50% lo que es correcto, ya que el descuento del vuelo era mayor al descuento de la aerolínea.

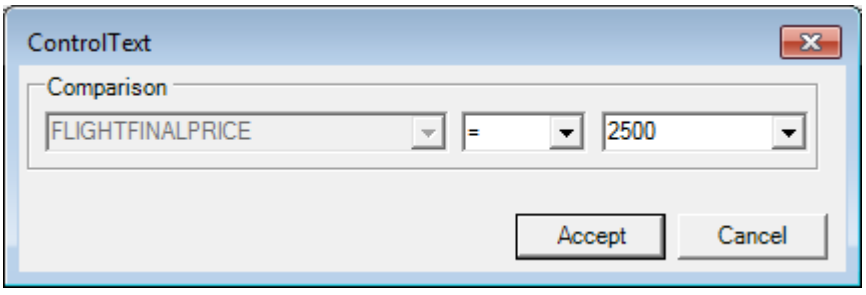


Ingresamos un par de asientos y antes de presionar Confirmar, vamos a indicarle a GXtest que para el futuro, nos ayude a verificar que el precio del vuelo se calcule correctamente.

Para eso seleccionamos el precio del vuelo y presionamos el símbolo de check, en la barra de herramientas de GXtest del navegador.

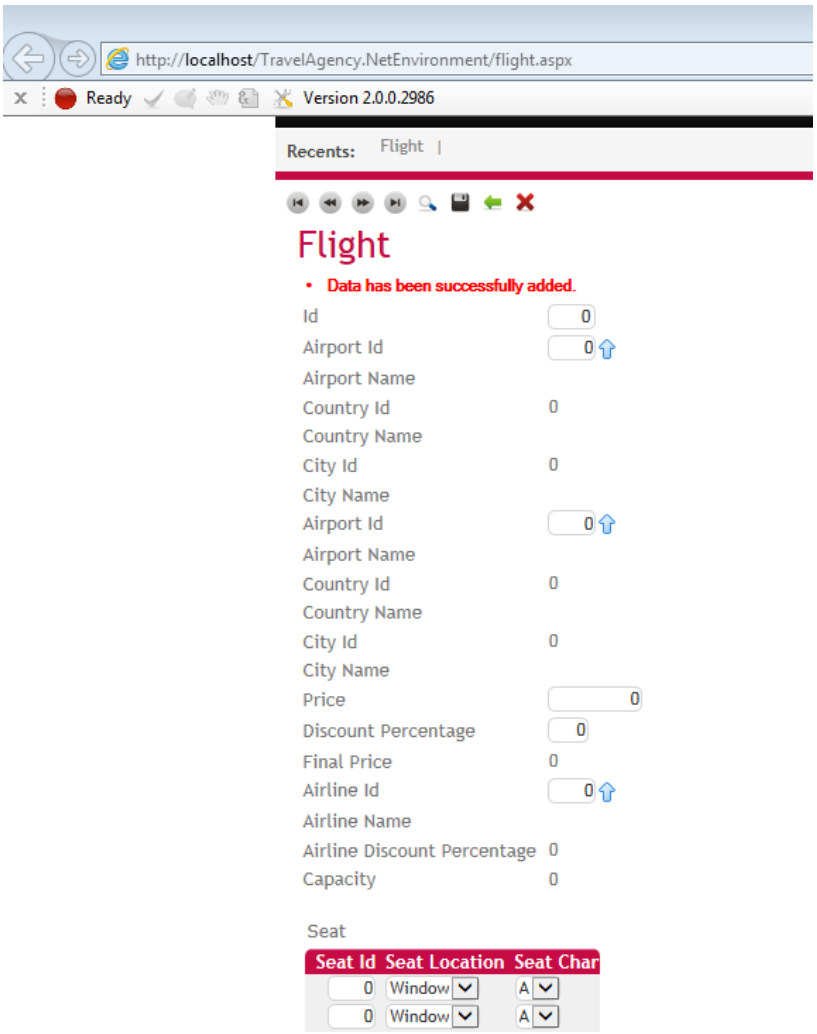


Elegimos VerifyControlText y en la descripción escribimos: “Flight Price checking”. Presionamos Aceptar y nuevamente Aceptar.



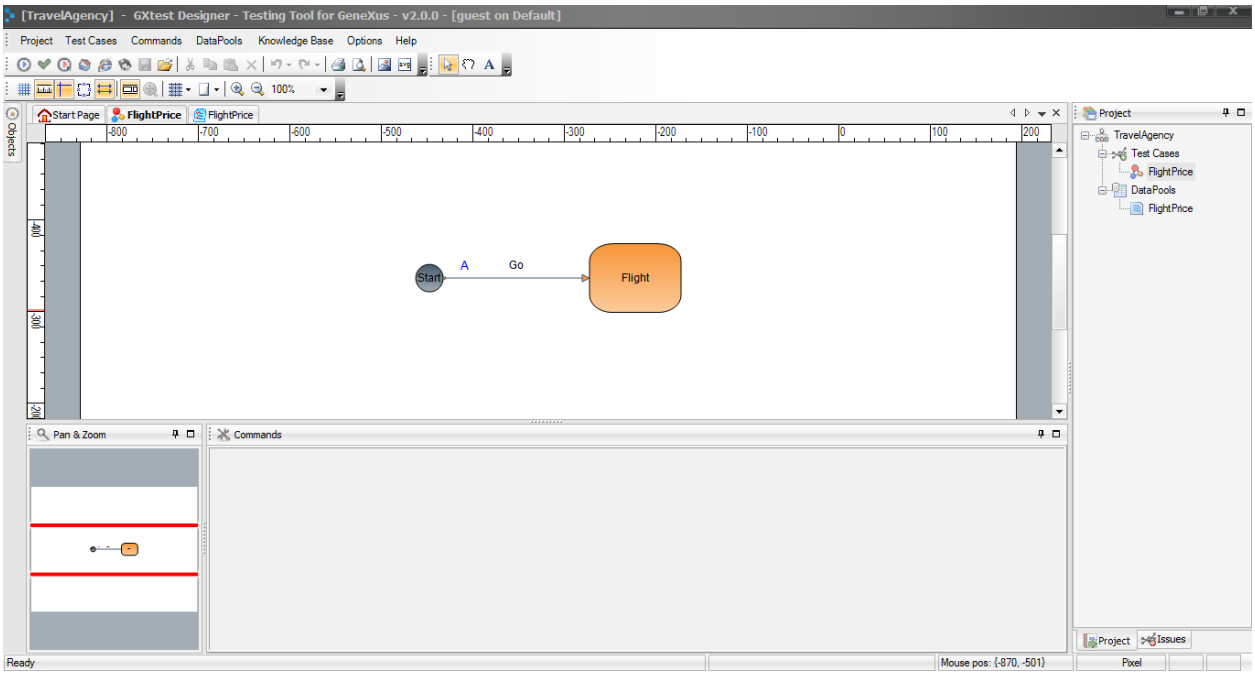
GXtest nos avisa que agregó correctamente la validación solicitada.

Presionamos OK, volvemos a la pantalla de la transacción Flight y presionamos Confirmar. Vemos que los datos se guardaron exitosamente.

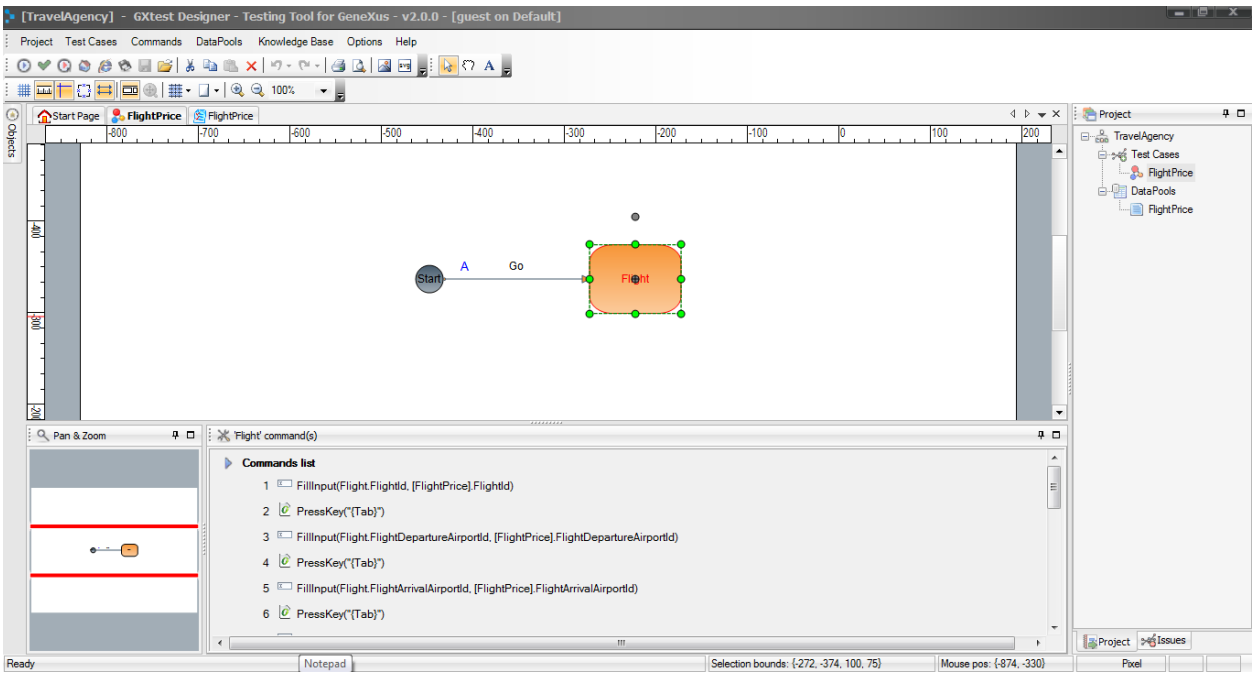


Cerramos la ventana del navegador y volvemos a GXtest.

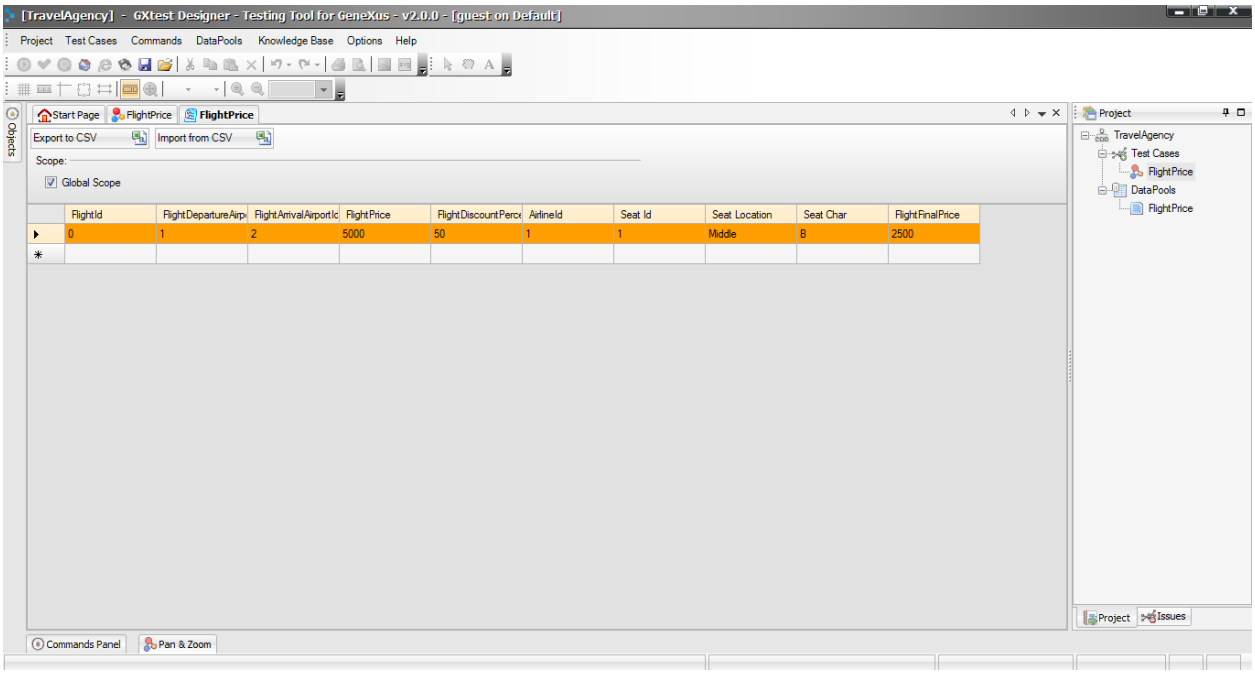
Ahora en la ventana principal de GXtest vemos un diagrama que representa nuestro caso de test ingresado.



Si seleccionamos el componente llamado Flight, vemos que en la ventana de comandos se detallan todos los pasos que fuimos haciendo para ingresar el vuelo en la transacción Flight: cuando ingresamos el identificador del vuelo, FlightId, presionamos TAB, ingresamos el identificador del aeropuerto de partida, FlightDepartureAirportId, etc….

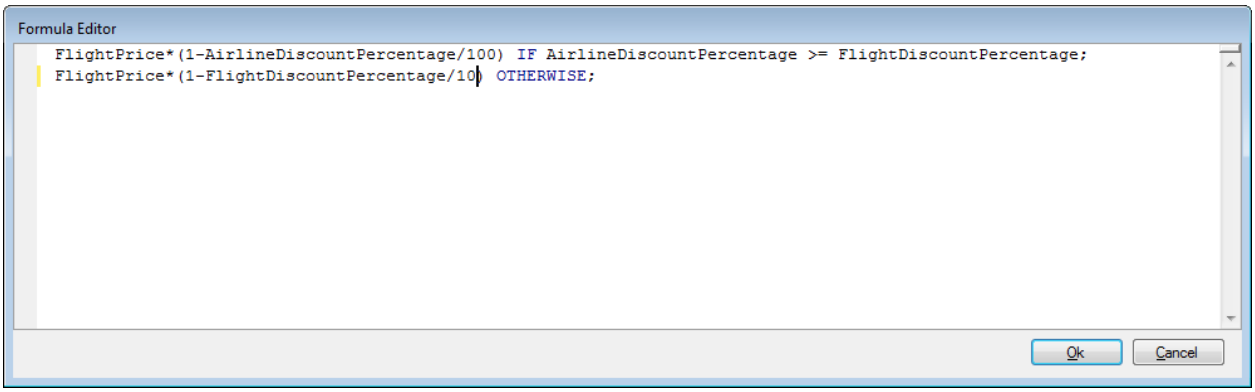


Si vamos al panel de Project, a la derecha de la pantalla y bajo DataPools hacemos clic en FlightPrice, se abre la ventana del conjunto de datos FlightPrice, que usamos para ingresar el vuelo.



Muy bien…Hasta aquí ingresamos un vuelo y almacenamos dicho ingreso como caso de prueba.

Supongamos que ahora, agregando otras funcionalidades a nuestra aplicación, sin querer hacemos modificaciones a la fórmula que calcula el precio del vuelo. Sin intención modificamos la segunda división y escribimos 10 en lugar de 100.



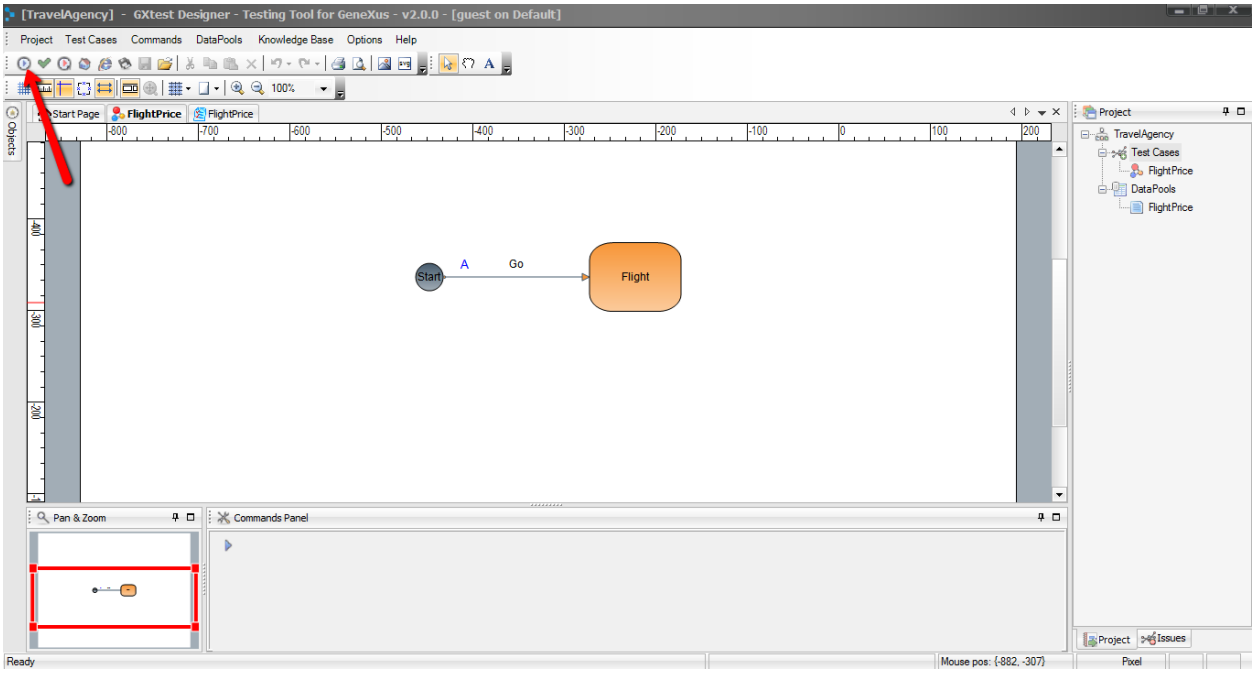
GXtest nos puede ayudar a detectar este tipo de errores!

Después de generar una nueva versión de nuestra aplicación, GXtest nos ayuda a probar que todo lo que teníamos hecho que no forma parte de los nuevos cambios, siga funcionando correctamente.

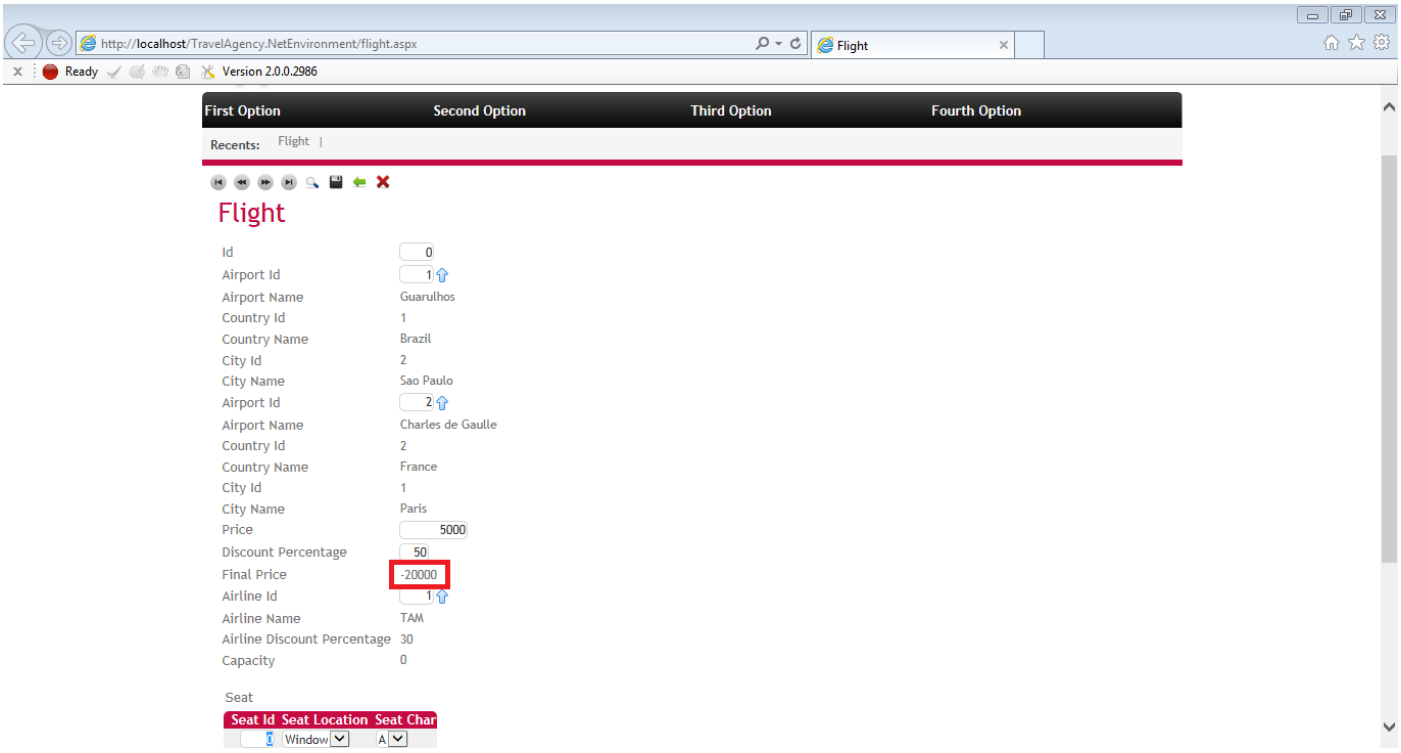
En primer lugar presionemos F5 para actualizar nuestra aplicación con el cambio…

Y a continuación, vamos a ejecutar nuevamente el caso de test que habíamos ingresado en GXtest.

Seleccionamos la solapa FlightPrice y hacemos clic en el botón de Play, que se encuentra arriba a la izquierda de la pantalla.

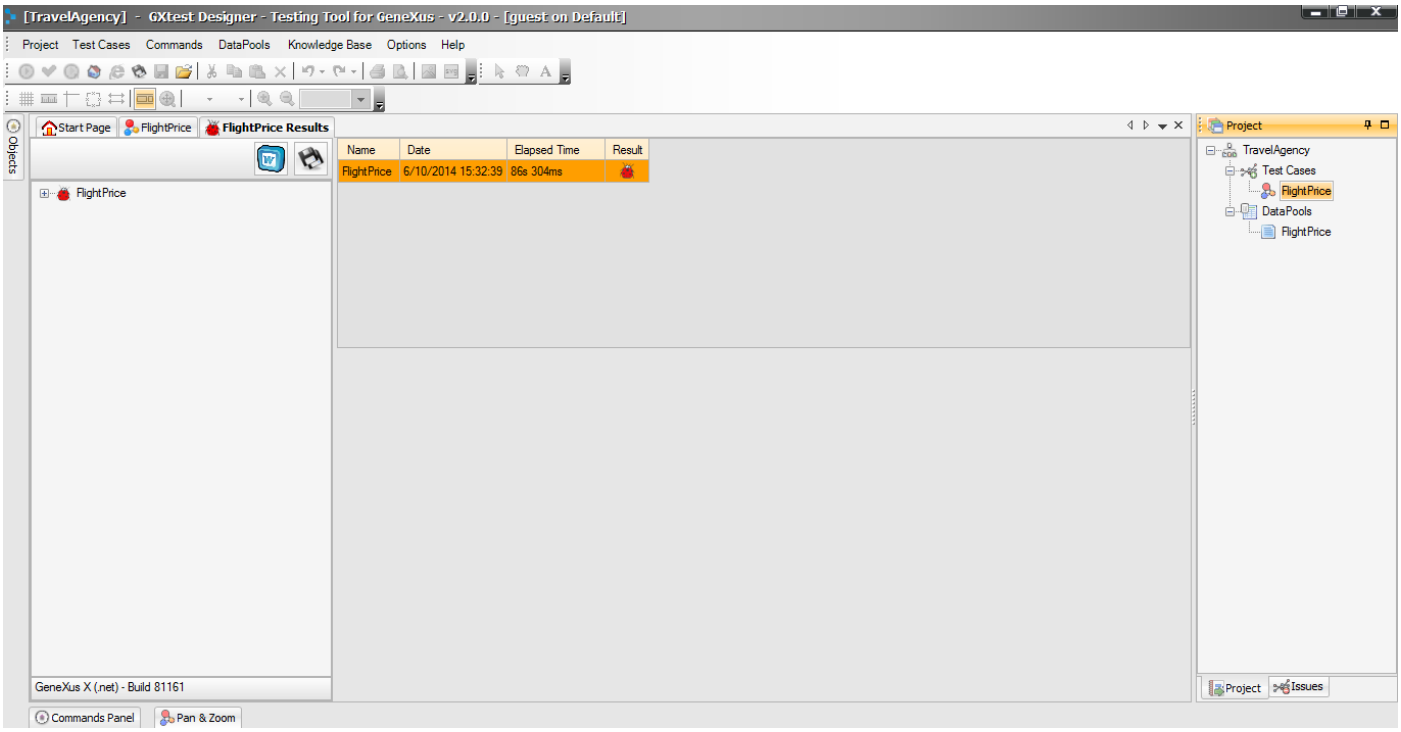


Vemos que se abre la ventana del navegador, se muestra la pantalla de la transacción Flight y empieza la ejecución automática del caso de prueba, ingresándose automáticamente los mismos datos que habíamos ingresado antes, como si fuera una persona que lo va haciendo…!



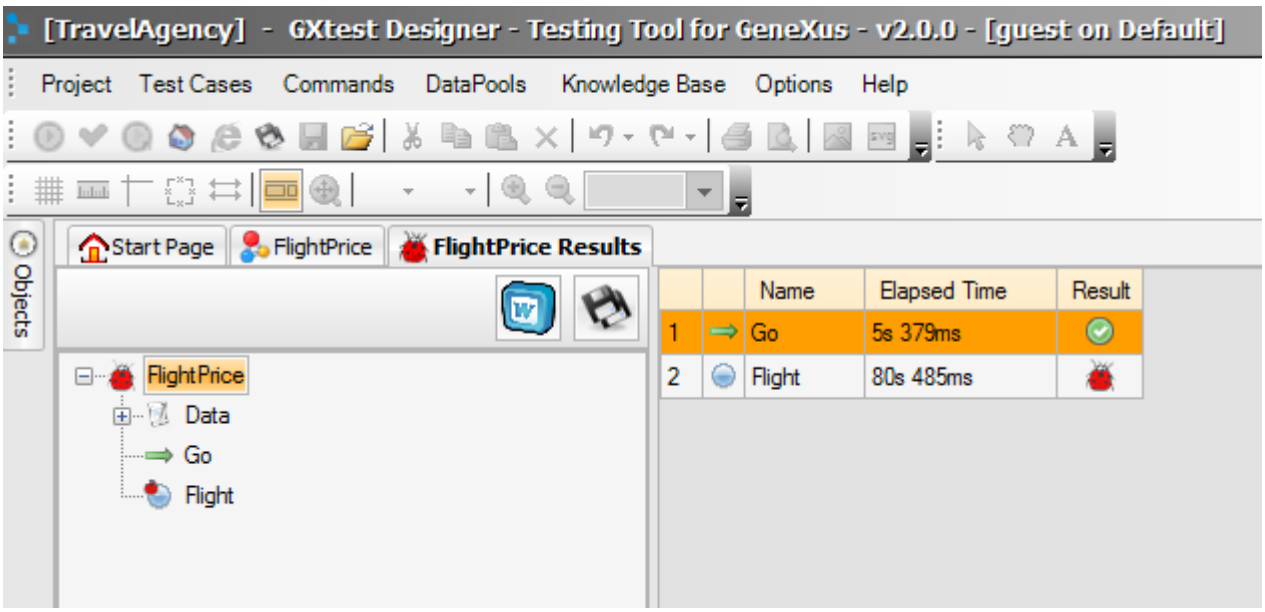
Si prestamos atención a los valores, vemos que el precio del vuelo se calculó incorrectamente y ahora muestra el valor -20000!

Cuando finaliza el caso de prueba, se activa nuevamente la ventana del Diseñador de GXtest…Hacemos clic sobre ella y vemos que se nos muestra el resultado de la prueba realizada, en una solapa llamada “FlightPrice Results”.

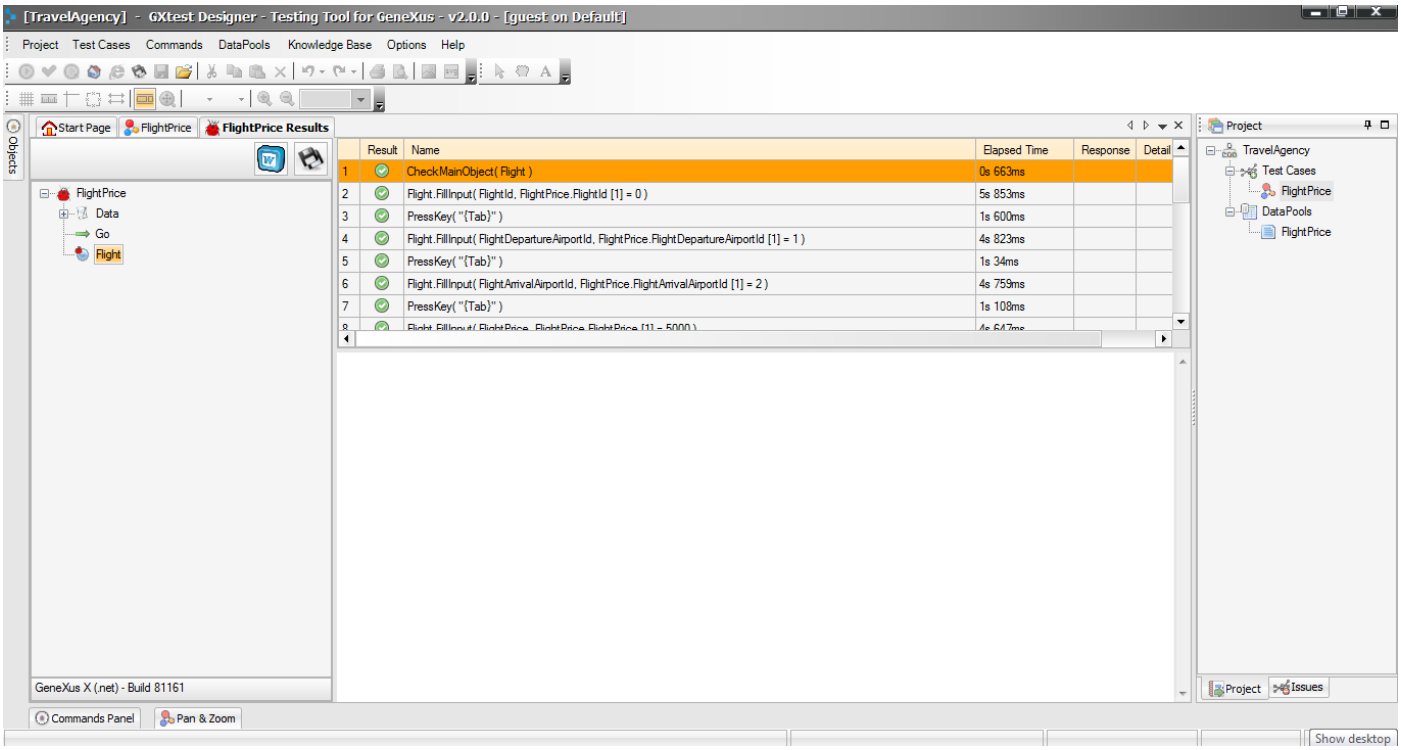


Vemos que en la columna Result hay un símbolo de un insecto, representando un “bug” encontrado.

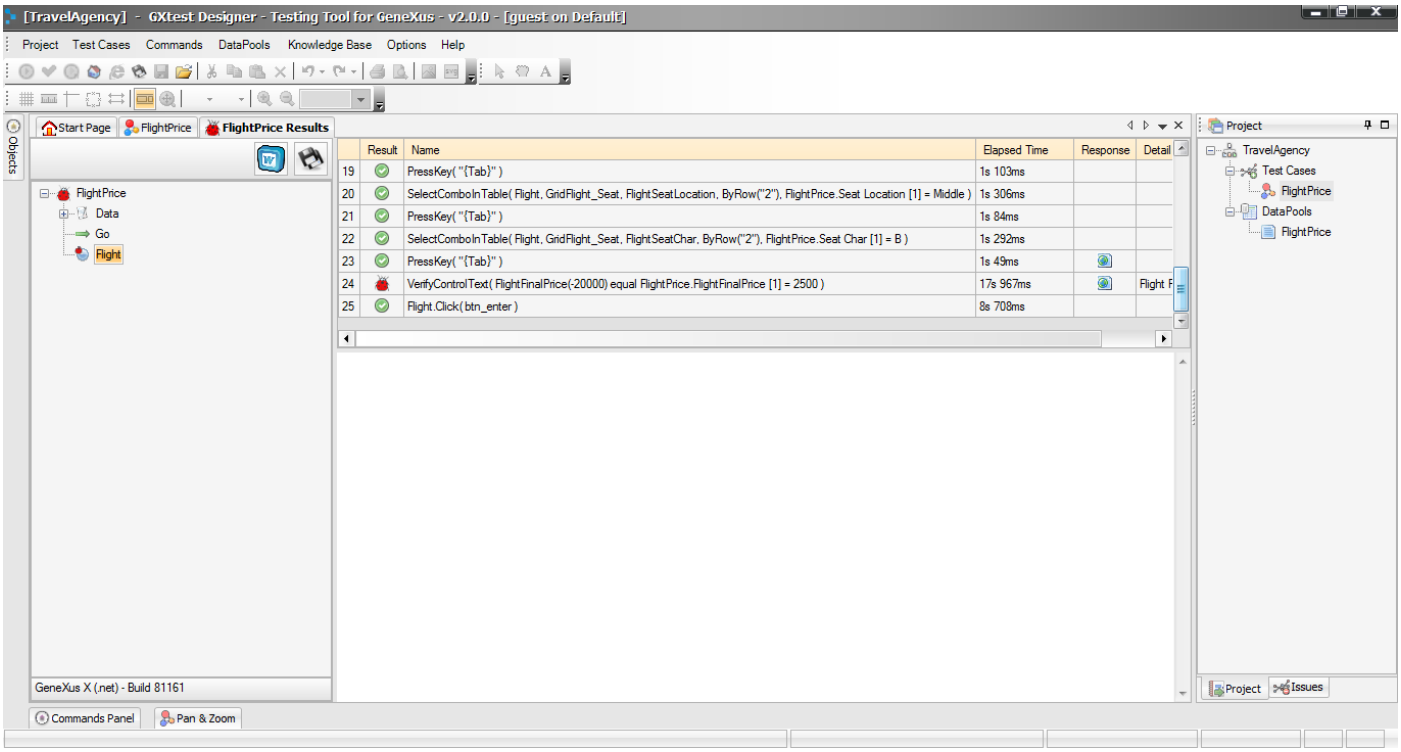
En el panel de la izquierda, hacemos click en el símbolo de + al lado de FlightPrice y vemos que se abren varias opciones y a la derecha, los test realizados.



Hacemos doble clic en Flight y GXtest nos abre una pantalla donde podemos observar la pantalla que probamos (en este caso la transacción Flight), los pasos que integran el test realizado, el tiempo que insumió cada paso y su resultado

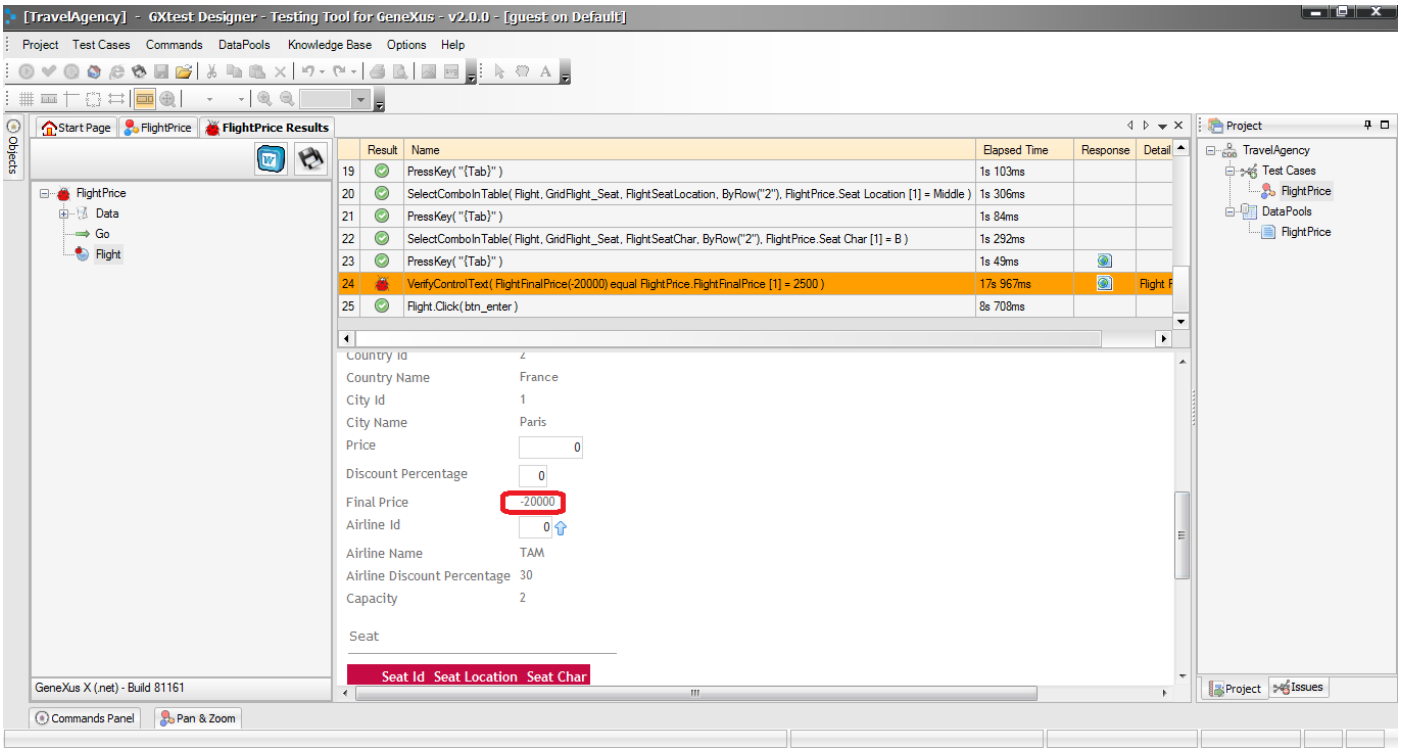


Si bajamos la barra, vemos que aparece el error de validación del precio del vuelo,



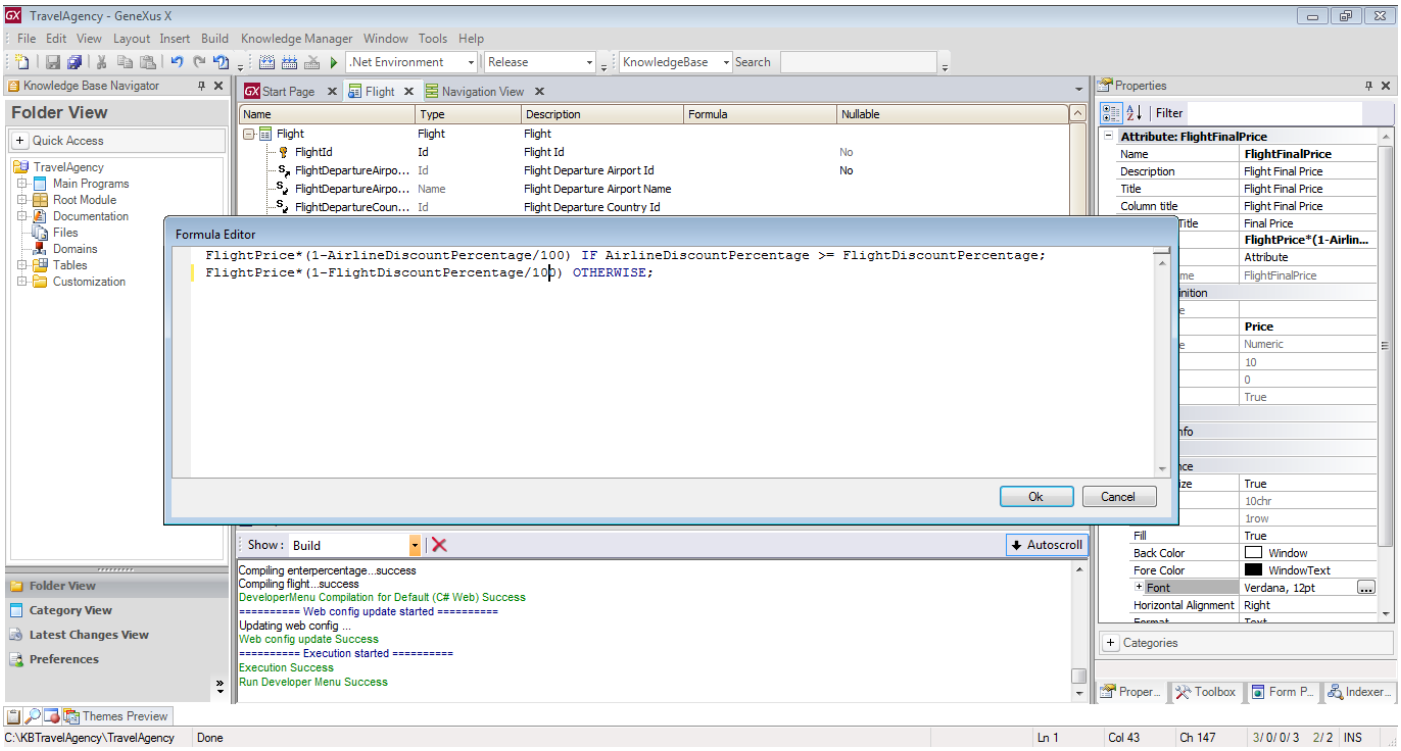
Indicándonos cuál fue el valor esperado y cuál fue el valor real obtenido.

Si hacemos clic sobre el renglón del error, abajo se despliega la pantalla del navegador, con el valor calculado incorrectamente.



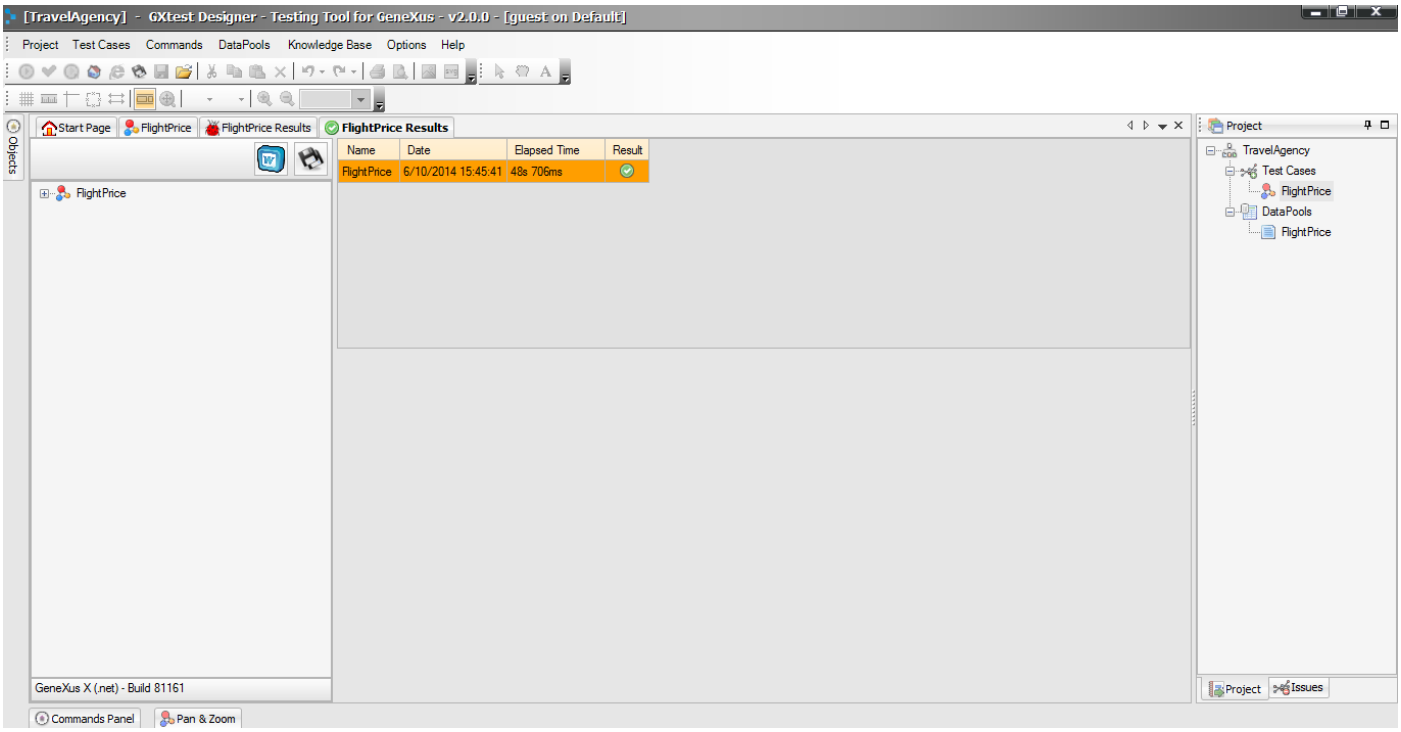
Vemos que al ejecutar este tipo de **test de regresión**, GXtest nos ayuda a verificar que ante cada cambio que hagamos al sistema, las cosas que ya estaban funcionando bien, sigan funcionando correctamente.

Este test nos dio una pista acerca del cálculo del precio del vuelo, así que vayamos a la transacción Flight para revisar nuestro código y arreglar el error.



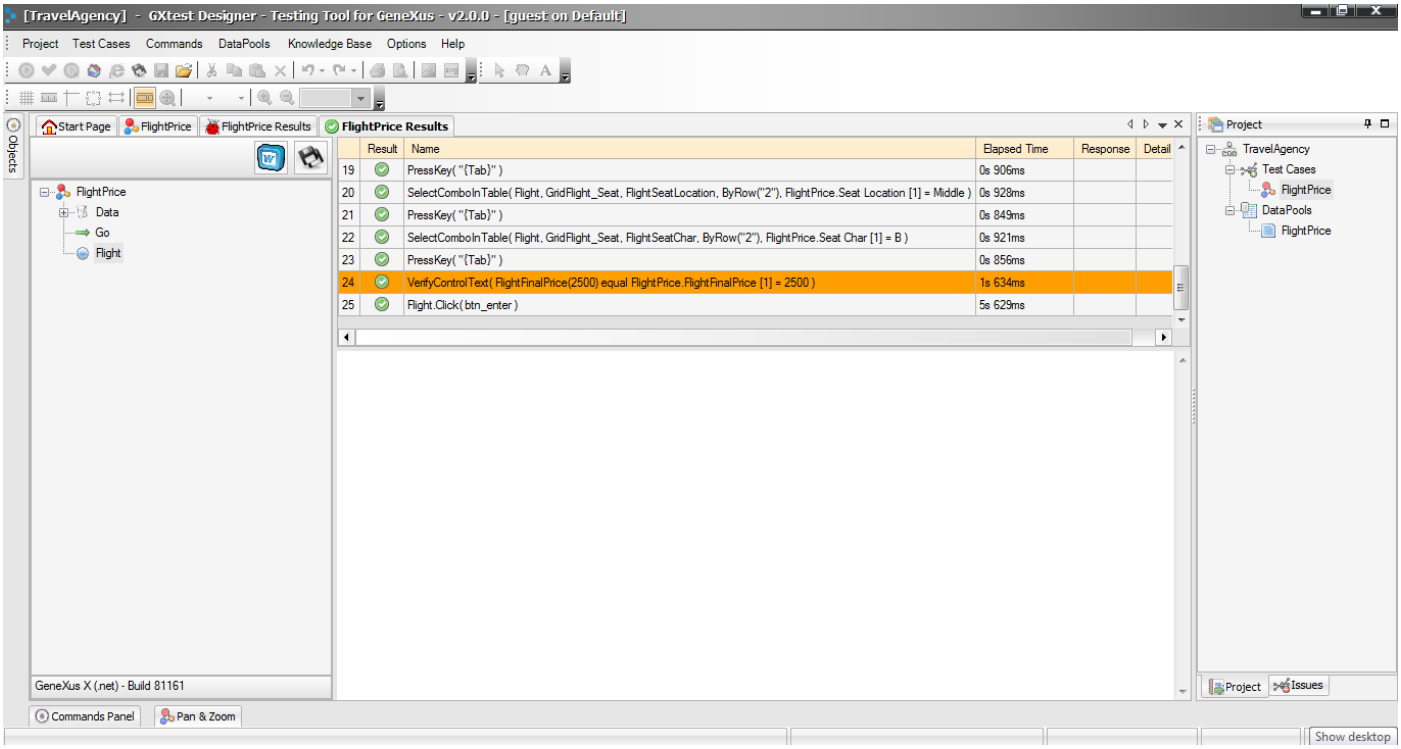
Editemos la fórmula del atributo FlightFinalPrice, pongamos nuevamente el valor del divisor en 100 y presionemos F5.

Y ahora ejecutemos nuevamente el test FlightPrice desde GXtest.

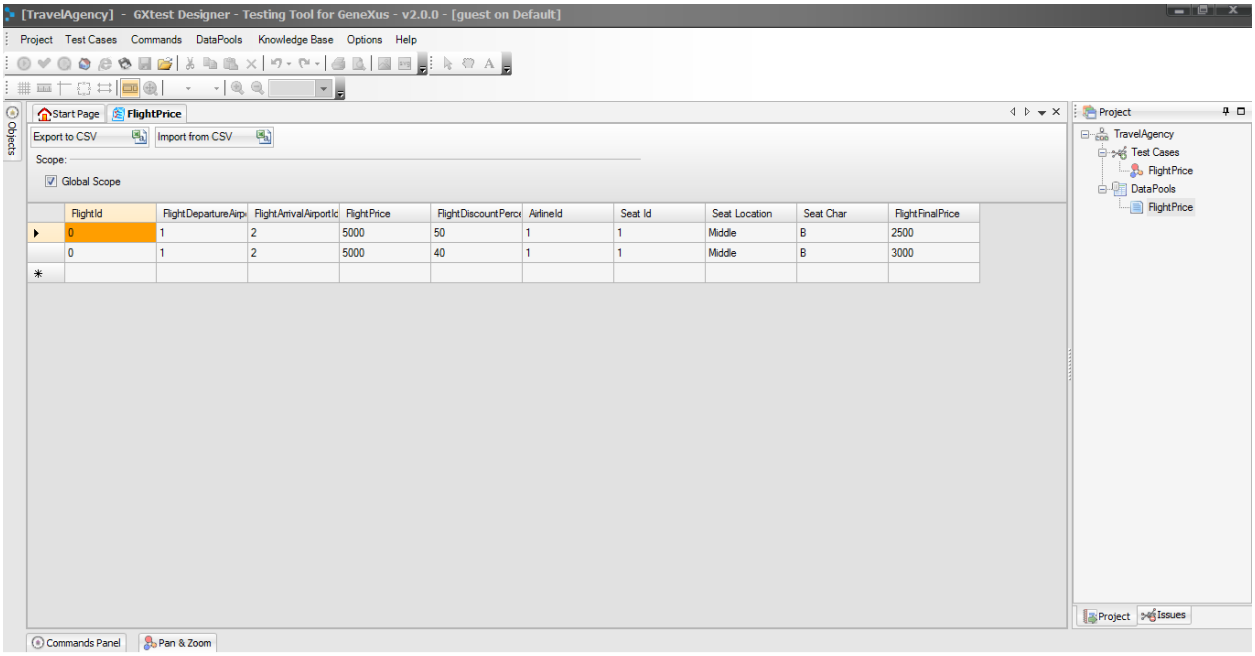


Vemos que ahora hay una nueva solapa FlightPrice Results con el resultado del nuevo test y que el resultado es el correcto.

Si abrimos el caso FlightPrice vara ver su detalle vemos que el control fue exitoso. Si seleccionamos la línea donde se verifica el precio del vuelo, como el control no falló, no vemos la pantalla del navegador ya que por defecto solamente se ve esta pantalla si se produjo un error, como en el caso anterior. Este comportamiento se configura en Options/Local Settings/Result detail.



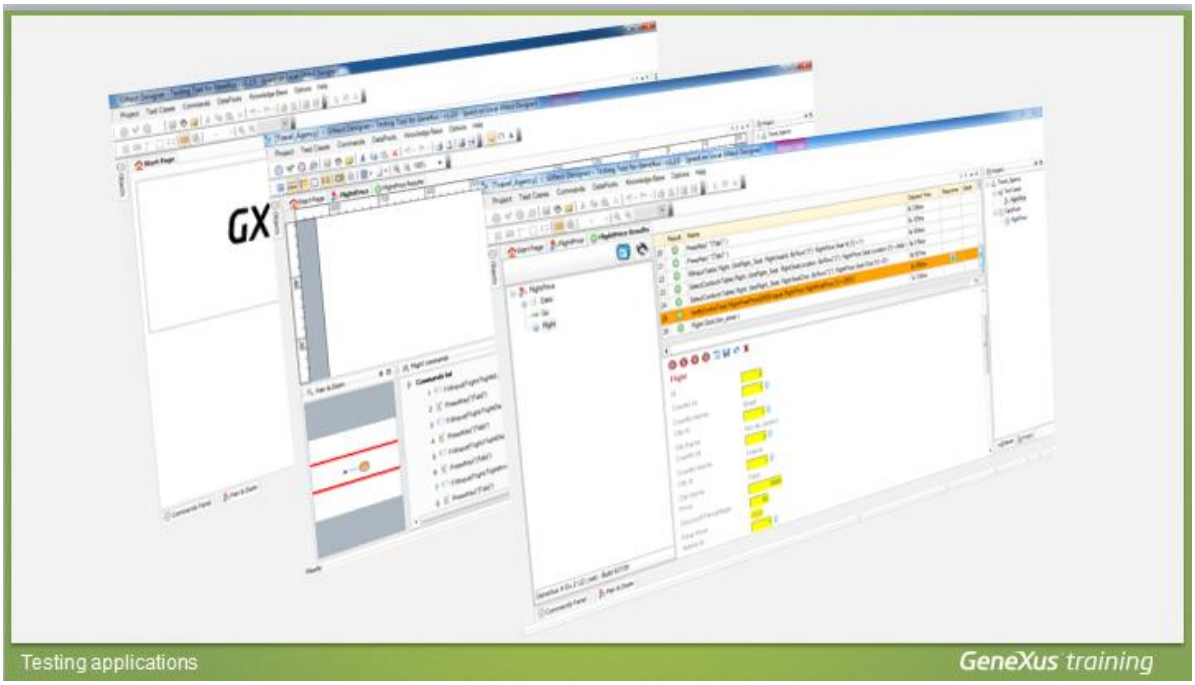
En particular, si abrimos el conjunto de datos que usamos para la prueba, haciendo doble clic FlightPrice, podríamos muy fácilmente agregar nuevos valores a la prueba



Hacemos clic y agregamos los mismos valores y vamos a un poner un descuento de un 40….y que el precio tiene que ser ahora 3000.

De este modo podemos armar nuestros test tan completos como queramos.

GXtest en un gran aliado que nos ayuda a hacer que nuestra aplicación sea confiable, reduciendo los tiempos de prueba, gracias a su facilidad para generar y ejecutar pruebas automáticas. Además nos asegura que si cambiamos la versión de GeneXus o cambiamos la plataforma, nuestra aplicación seguirá funcionando como esperamos.



Si quiere saber más, vaya a la url que se muestra en pantalla.