



Hands-On Lab 2013

Contenido

Introducción	2
Laboratorio.....	3
1. Crear el proyecto "AjaxSample"	3
2. Crear un nuevo caso de prueba llamado NewInvoice	5
3. Utilización de Datapools.....	10
4. Crear un caso de prueba desde GXtest Designer	12
5. Ejecución en FireFox.....	18
6. One Click Startup: ¡Empiece rápido y fácil con 1 clic!	19
Conclusiones	21
¿Cómo seguir?.....	21

Introducción

El objetivo de este laboratorio es descubrir las ventajas que ofrece GXtest para automatización, construyendo de manera sencilla un conjunto breve de casos de prueba automatizados.

Para la realización del taller se utilizará la aplicación Ajax Sample, la cual forma parte de las aplicaciones de ejemplo de GeneXus (<http://samples.genexus.com/ajaxsample/home.aspx>). Dicha aplicación, maneja un registro de **Cientes** (Client) y **Productos** (Product), y un conjunto de **Facturas** (Invoice) que se registran para un cliente cuando éste decide comprar ciertos productos.

Todos los archivos sobre los que se hace referencia en el laboratorio, se encuentran en el **escritorio** de la máquina, en la carpeta **Material GXtest**.

Adicionalmente, al final de este documento encontrará links de interés y documentación sobre GXtest y Abstracta.

Laboratorio

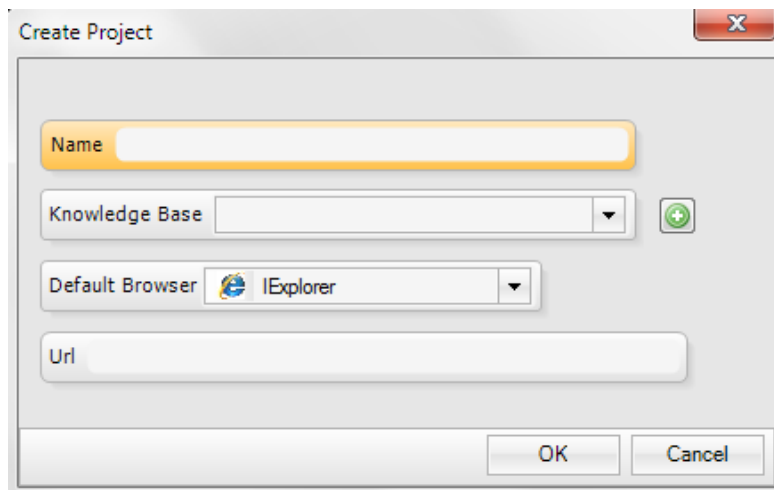
Para trabajar en GXtest, nos organizamos en proyectos. La idea es tener un proyecto de pruebas GXtest asociado a una aplicación que deseo testear.

Además del proyecto, también necesitaremos algunos datos sobre la aplicación web, sus pantallas y controles. Para esto, GXtest examina la base de conocimiento GeneXus de la aplicación y toma esta información o metadata.

1. Crear el proyecto “LabAjaxSample”

Abrir GXtest Designer (en el escritorio se puede encontrar el ícono que dice GXtest Designer).

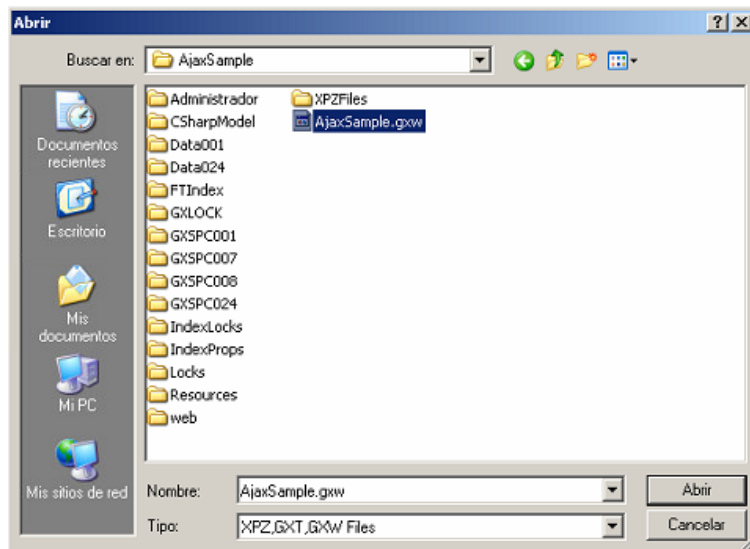
En GXtest se trabaja por proyectos, por lo que el primer paso es crear un nuevo proyecto. Para crear un proyecto ir a **Project->New Project**. Aparecerá una ventana como la siguiente:



Cuadro de dialogo de “Create Project”

Campo “Name”: Ingresar el nombre del proyecto, por ejemplo **LabAjaxSample**.

Campo “Knowledge Base”: ahora se debe asociar la Knowledge Base (KB) de la aplicación a automatizar. Para hacer esto dar clic en el botón con la figura de (+). Luego indicar el archivo llamado “**AjaxSample.gxw**” ubicado en **C:\Models\AjaxSample**

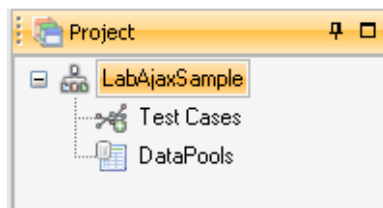


Luego de unos segundos, GXtest informará que se importó correctamente la KB.

Campo "Url": indica donde está publicada la aplicación que deseamos testear. Ingresar la siguiente url:

<http://localhost/LabGXtest/home.aspx>

Para finalizar, presionar el botón Ok y se creará el proyecto.



GXtest mantiene su propia versión de la KB de la aplicación, y estos datos se pueden importar a GXtest de varias maneras:

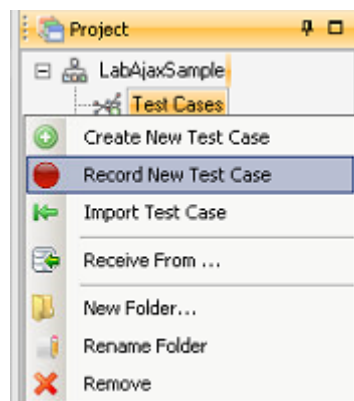
- a. Un archivo *.xpz*: Si la KB se encuentra en GeneXus 8 o 9
- b. Un archivo *.gxw*: donde se encuentra el modelo (GeneXus X Ev1 o superior).
- c. Un archivo *.gxt*: Versión reducida de la KB.

2. Crear un nuevo caso de prueba llamado NewInvoice

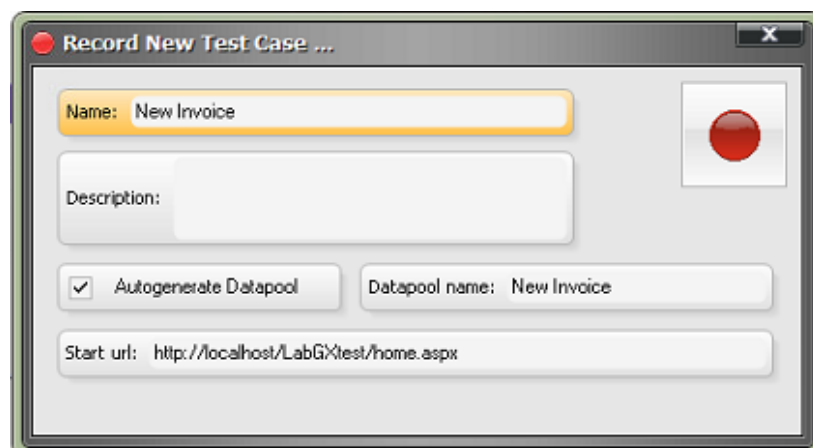
En esta sección se grabará el caso de prueba de **NewInvoice** utilizando el componente *GXtest Recorder*. Antes de comenzar, sepa que el caso de prueba consistirá en ejecutar los siguientes pasos:

1. Ingresar a la aplicación.
2. Acceder a la funcionalidad “Work With Invoices” por el menú.
3. Seleccionar el botón “+” para agregar una nueva factura.
4. Ingresar el nombre del cliente.
5. Ingresar una línea de la factura indicando:
 - a. Producto
 - b. Cantidad
6. Validar que el subtotal, los impuestos y el total sean correctos.
7. Confirmar.

Para comenzar a grabar dichos pasos en GXtest, hacer clic con el botón derecho sobre el nodo “Test Cases” que aparece en el panel derecho de GXtest Designer, luego seleccionar “**Record New Test Case**”.



Ingresar el nombre “**New Invoice**”, seleccionar el checkbox “Autogenerate Datapool” y luego presionar el botón rojo para comenzar a grabar.



GXtest abrirá el Internet Explorer para comenzar a grabar las acciones que se realicen sobre el mismo. Se podrá ver en el navegador el componente GXtest Recorder, el cuál será el encargado de grabar nuestras acciones.




Realizar los siguientes pasos en la aplicación como parte del caso de prueba que se quiere automatizar:

- a. Hacer clic en el link Work With Invoices.
En ese momento (debido a que se hizo clic en una grilla) GXtest pedirá que se le indique cuál fue el criterio para seleccionar dicha fila. Dejar la opción que viene por defecto y presionar *Accept*. Veremos en detalle estos criterios más adelante.
- b. Hacer clic el botón “+” para agregar una nueva factura.
- c. Ingresar el nombre del cliente *Carlos*. Luego ingresar una línea de la factura, indicando el producto *Xbox* y como cantidad *1*. Presionar **Tabulador**.
- d. La aplicación calculará los valores de la nueva factura ingresada. Ahora se desea validar el importe total.
Para ingresar esta validación seguir los pasos que se detallan a continuación para el campo total:

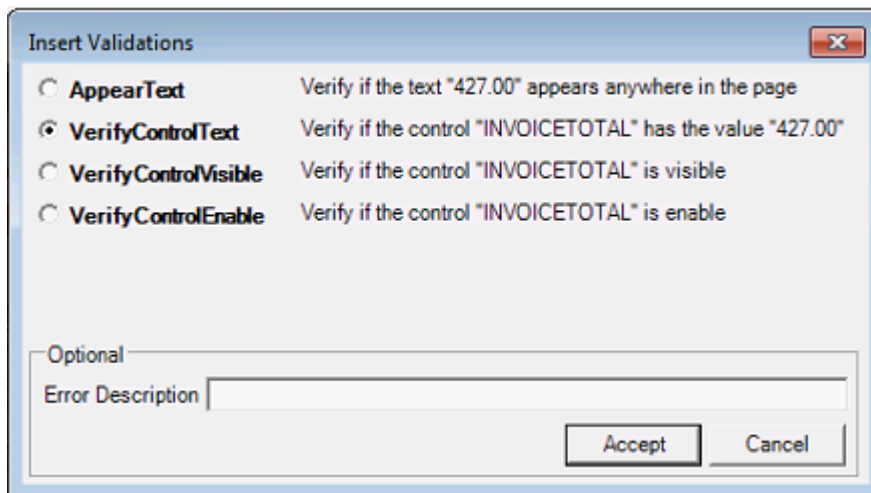
- i. seleccionar el número que muestra el valor del total en la pantalla

Invoice Line						
	Line Id	Product Name	Stock	Price	Line Quantity	Line Amount
x	1	Xbox	249	350.00	1	350.00
	0		0	0.00	0	0.00
	0		0	0.00	0	0.00
	0		0	0.00	0	0.00
	0		0	0.00	0	0.00
[New row]						
Sub Total						350.00
Taxes						77.00
Total						427.00

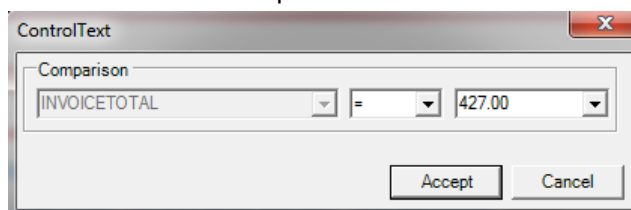
- ii. luego presionar el botón de validaciones  en GXtest Recorder.



- iii. Esto permitirá elegir el tipo de validación a agregar. En este caso seleccionar **VerifyControlText**, la cual permite comparar un valor de la pantalla con otro valor y presione *Accept*.



iv. Se abrirá otra ventana para confirmar el criterio de selección, presionar Accept.



e. Para confirmar la factura, presione el botón **Confirm**.

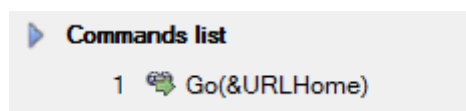
f. Por último, cerrar el navegador para terminar de grabar el caso de prueba.

De esta forma quedó grabado el caso de prueba en GXtest. Para comprender como se ven reflejadas las distintas acciones realizadas durante la grabación del caso de prueba se recomienda recorrer (situándose sobre cada página y arista) el diagrama que GXtest ha creado. El caso de prueba creado debe quedar análogo al que se muestra en la siguiente figura:



GXtest expresa las acciones que el usuario realiza en el navegador a través de comandos. Cada comando se asocia a una página (representadas por los nodos anaranjados) o a una arista (que representa las transiciones de una página a otra).

Por ejemplo, en la primera arista se encuentra el comando “Go”, el cual se puede observar en la parte inferior en la lista de comandos al seleccionar la arista inicial:

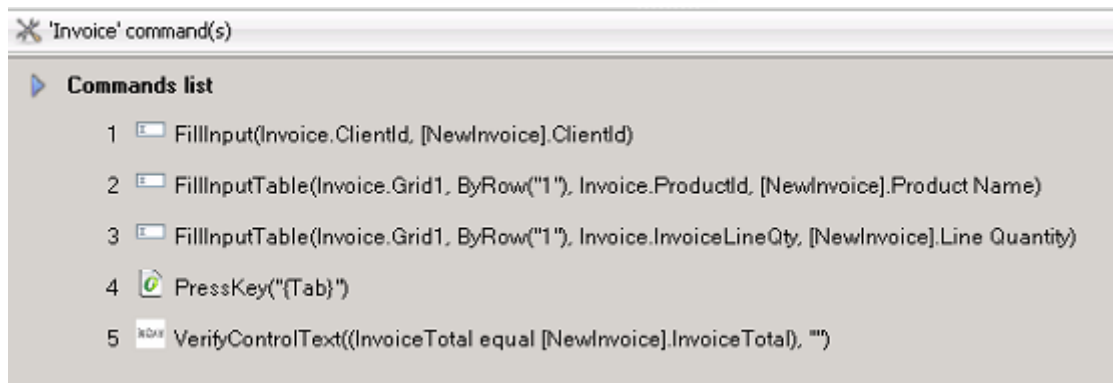
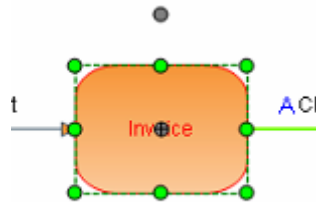


Los otros comandos que se pueden ver en el caso de prueba son:

- Click: representa un clic en un elemento.
- ClickTable: representa un clic en un elemento que se encuentra dentro de una grilla.

- FillInput: representa la acción de completar un valor en un campo.
- FillInputTable: ídem anterior, pero dentro de una grilla.
- VerifyControlText: representa la validación de que cierto control de la pantalla tenga un valor determinado.
- DPNext: Obtiene la próxima fila de datos del Datapool. Veremos más adelante qué es un DataPool.

Si seleccionamos el nodo Invoice, podemos ver los comandos asociados a los datos que ingresamos mientras dábamos de alta la factura.



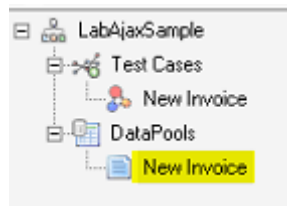
En la lista de comandos ejecutados en la página Invoice, se pueden ver cada uno de las acciones realizadas:

- Ingresar un valor en el campo ClientId (Cliente)
- Ingresar un valor en el campo ProductId, en la fila 1 de la grilla Grid1.
- Ingresar un valor en el campo InvoiceLineQty, en la fila 1 de la grilla Grid1.
- Presionar la tecla "TAB"
- Verificar el valor del control InvoiceTotal.

Pero, ¿donde están los valores ingresados durante la grabación del caso de prueba? El cliente "Carlos", el producto "Xbox", la cantidad de productos (1), y el valor "427.00" del total de la factura que quiero validar...

Si se estudia en detalle los comandos, tienen un componente [New Invoice]. Ese nombre entre paréntesis rectos indica el uso de un Datapool, en particular el Datapool New Invoice.


Abrir entonces el Datapool New Invoice, **desde el árbol del proyecto, en la sección DataPools, haciendo doble clic sobre New Invoice.**



	ClientId	Product Name	Line Quantity	InvoiceTotal
▶	Carlos	Xbox	1	427.00
*				

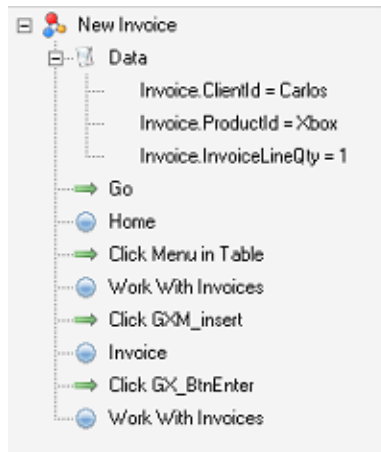
Un Datapool es una estructura tabular donde se pueden guardar los datos utilizados en el caso de prueba, de manera de separar el flujo de ejecución de los datos.

Una vez abierto el Datapool, es posible ver que aquí están los datos que se grabaron en el caso de prueba. De hecho, es posible ver que para los diferentes comandos, existe una columna para los datos de cada uno de los controles. Por ejemplo, para el nombre del cliente, se creó automáticamente la columna ClientId y se guardó allí el valor Carlos.

Volver al caso de prueba generado, y una vez que se haya podido entender el modelo creado por GXtest, **ejecutar el caso de prueba presionando el botón de ejecutar** , en la barra de herramientas de GXtest Designer, o la tecla F5 (estando el caso de prueba abierto en GXtest Designer).



Finalizada la ejecución, el resultado detallado se mostrará en un panel lateral ubicado a la izquierda dentro de GXtest Designer. Si navega en el menú de resultados, podrá observar los datos que utilizó GXtest en la ejecución y recorrer la secuencia de todos los comandos ejecutados con sus respectivos parámetros y tiempos de ejecución.



Menú de resultado, podrá recorrer cada nodo y arista que se ejecutó.

Result	Name	Elapsed Time	Response	Det.
3	PressKey("{Tab}")	0s 453ms		
4	FillInputTable(Invoice, Grid1, ProductId, ByRow("1"), New Invoice.Product Name [1] = Xbox)	2s 359ms		
5	PressKey("{Tab}")	0s 390ms		
6	FillInputTable(Invoice, Grid1, InvoiceLineQty, ByRow("1"), New Invoice.Line Quantity [1] = 1)	2s 265ms		
7	PressKey("{Tab}")	0s 468ms		
8	VerifyControlText(InvoiceTotal[427.00] equal New Invoice.InvoiceTotal [1] = 427.00)	9s 796ms		

Work With Cities
 Work With Clients
 Work With Countries
 Work With Invoices
 Work With Products

Description Inv: 10/17/13
 Client First Name Carlos
 Client Balance 4419.00
 Client Address Guarulhos

Invoice Line

Line Id	Product Name	Stock	Price	Line Quantity	Line Amount
x 1	Xbox	247	350.00	1	350.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00
[New row]					

Sub Total 350.00
 Taxes 77.00
 Total 427.00

Detalle de la ejecución de los comandos comprendidos en el nodo 'Invoice'. Se puede observar el resultado de cada comando y debajo el html (página) al momento de ejecutarse el comando.

3. Utilización de Datapools

Es importante poder variar los datos de entrada que utiliza un caso de prueba y testear así los distintos comportamientos que puede tener la aplicación. GXtest permite utilizar este enfoque mediante el concepto de Datapools, el cual almacena en forma de tablas los distintos datos utilizados en nuestros casos de prueba (tanto de entrada como resultados esperados). En este caso, se creó automáticamente el Datapool **NewInvoice**, el cual contiene **todos** los datos que ingresamos al grabar el caso de prueba.


Los datos de un Datapool se pueden ingresar de manera manual o mediante la importación de un archivo CSV (archivo de valores separados por coma, u otro carácter).

En este caso los casos de prueba se han cargado en el archivo DatosInvoice.csv (ubicado en el escritorio, en la carpeta Material GXtest).

Abrir el datapool NewInvoice e importar dicho archivo haciendo clic en el botón “Import from CSV”. También es posible copiar los datos desde una planilla Excel al datapool, (Ctrl. + C, Ctrl. + V).

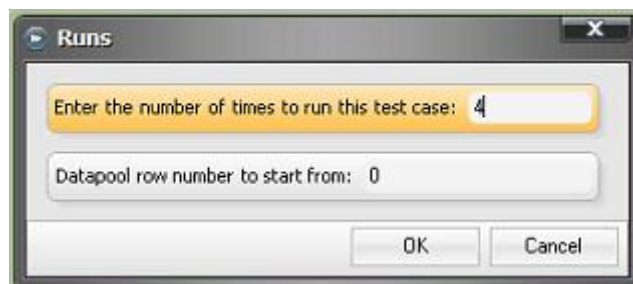
El datapool deberá quedar como se muestra a continuación:

	ClienId	Product Name	Line Quantity	InvoiceTotal
▶	Carlos	Xbox	1	427.00
	Joaquin	Mouse	1	3.66
	John	Keyboard	2	12.20
	Julio	Notebook	9000	109789020
*				

Es posible ejecutar un mismo caso de prueba repetidas veces, utilizando el botón  de la barra de herramientas de GXtest Designer, o presionando Shift + F7.



Presionar dicho botón, e ingresar el valor 4 en la ventana emergente para ejecutar el caso de prueba 4 veces.



Se abrirá el navegador y se ejecutará el caso de prueba con 4 iteraciones, tomando en cada ejecución una fila de datos distinta del Datapool.

Los resultados quedarán como se muestra a continuación:

Name	Date	Elapsed Time	Result
New Invoice	17/10/2013 19:02:05	42s 421ms	✓
New Invoice	17/10/2013 19:02:50	39s 484ms	✓
New Invoice	17/10/2013 19:03:32	39s 828ms	✓
New Invoice	17/10/2013 19:04:15	66s 906ms	✗

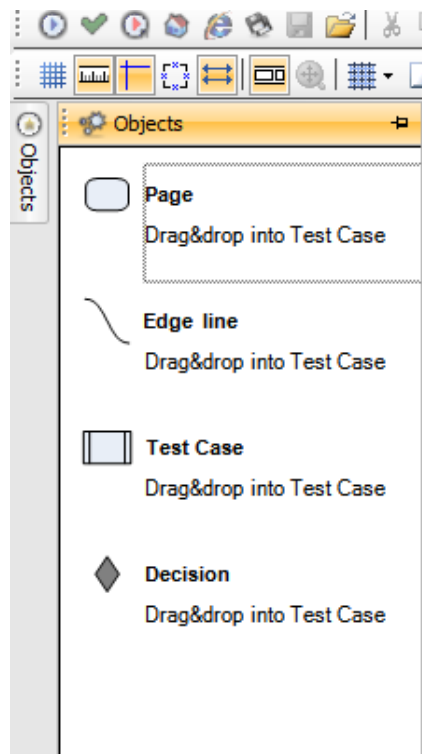
El cuarto caso dio error! Lo invitamos a investigar qué fue lo que pasó. ¿Encontramos un incidente?

Nota: Si bien automatizando podemos encontrar bugs al momento de “grabar” o generar los casos de prueba, el cometido principal de la automatización será el de asegurar que para próximas versiones, el sistema no presenta regresiones (nuevos bugs en funcionalidades que estaban funcionando correctamente)

4. Crear un caso de prueba desde GXtest Designer

Es posible generar un caso de prueba de forma manual en GXtest Designer, creando el flujo deseado arrastrando los controles que se encuentran en la barra de herramientas “Objects” ubicada a la izquierda de la pantalla.

Puede ver que los elementos que puede agregar son **páginas, aristas, decisiones**, u otros **casos de prueba**.



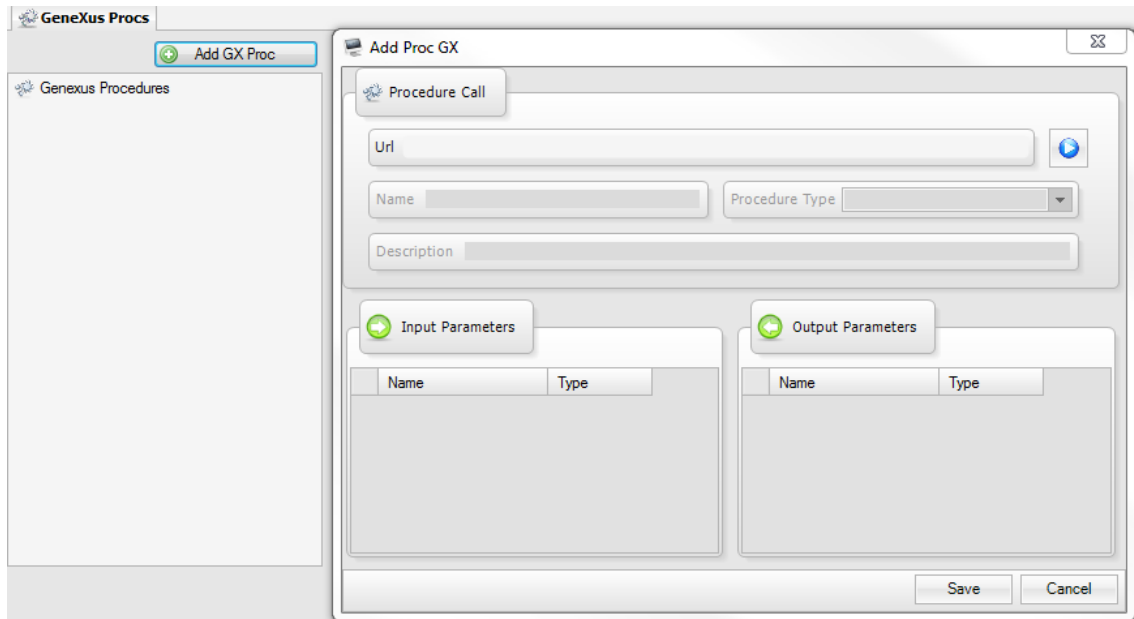
Dentro de un caso de prueba, es posible incluir otro caso de prueba como si fuera un componente más en el flujo, lo que permite reutilizar los elementos ya automatizados y crear flujos más complejos, aumentando la productividad.

Las decisiones pueden ser utilizadas para separar el flujo de ejecución del caso de prueba en dos caminos diferentes, a partir de una condición. Por ejemplo, se podría invocar a un WebService para validar si un determinado cliente no existe en la base de datos, y en ese caso darlo de alta antes de continuar con el ingreso de una factura de ese cliente.

Por otro lado, GXtest permite de manera sencilla invocar WebServices GeneXus para ser utilizados en las pruebas. Los Web Services en general son utilizados para modificar, consultar o validar valores específicos de la base de datos de la aplicación.

En esta ocasión utilizaremos el webservice ExistsClient (un webservice ya creado en la aplicación) para determinar el flujo a seguir dentro del caso de prueba: si el cliente no existe, lo daremos de alta.

Primero es necesario dar de alta el Web Service en GXtest, para luego poder utilizarlo en las pruebas. **Para esto, ir a Commands->Webservices y presionar el botón de Add GX Proc.**



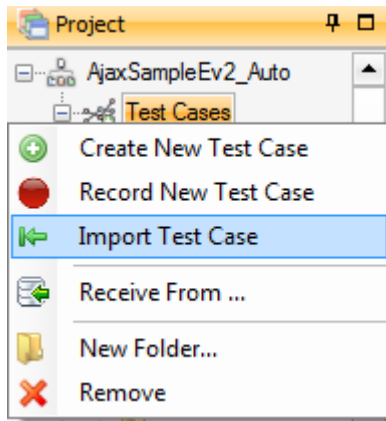
Ingresar la url <http://localhost/LabGXtest/aexistsclient.aspx?wsdl> y presionar el botón start (flecha azul). En ese momento GXtest leerá la información de los parámetros que utiliza el Web Service y las presentará. Se debe indicar el tipo de comando a agregar: validación o acción. Como lo que se quiere realizar es una validación (validar si el cliente existe o no), se debe **elegir tipo "Validation"**. Finalmente ingresar una descripción, en este caso ingresar "Verifica si un cliente existe o no en la base de datos" y **presionar Save**.

Ahora disponemos de un nuevo comando "ExistsClient" para ser utilizado desde las pruebas en GXtest como si fuese un comando nativo.

A este comando lo utilizaremos en un nuevo caso de prueba "Add or Remove Client", para preguntar si existe el cliente antes de dar de alta. La lógica será:

- Si existe el cliente, ir directamente a la baja del cliente
- Si no existe, primero hacer el alta y luego la baja de cliente.

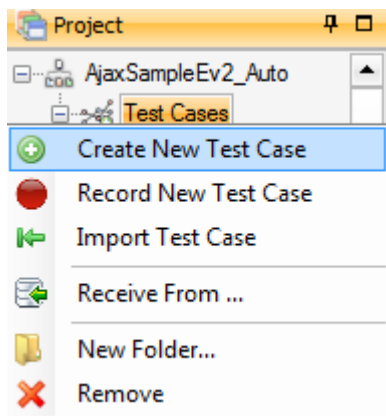
Antes de seguir, necesitamos tener entonces un caso de prueba para dar de Alta el Cliente, y otro para darlo de baja. Para esto, importaremos estos casos de prueba que ya están generados: desde el árbol del proyecto de la derecha, **hacer clic con el botón derecho sobre Test Cases**, opción **Import Test Case**, y **seleccionar el archivo AltaCliente.xml**.



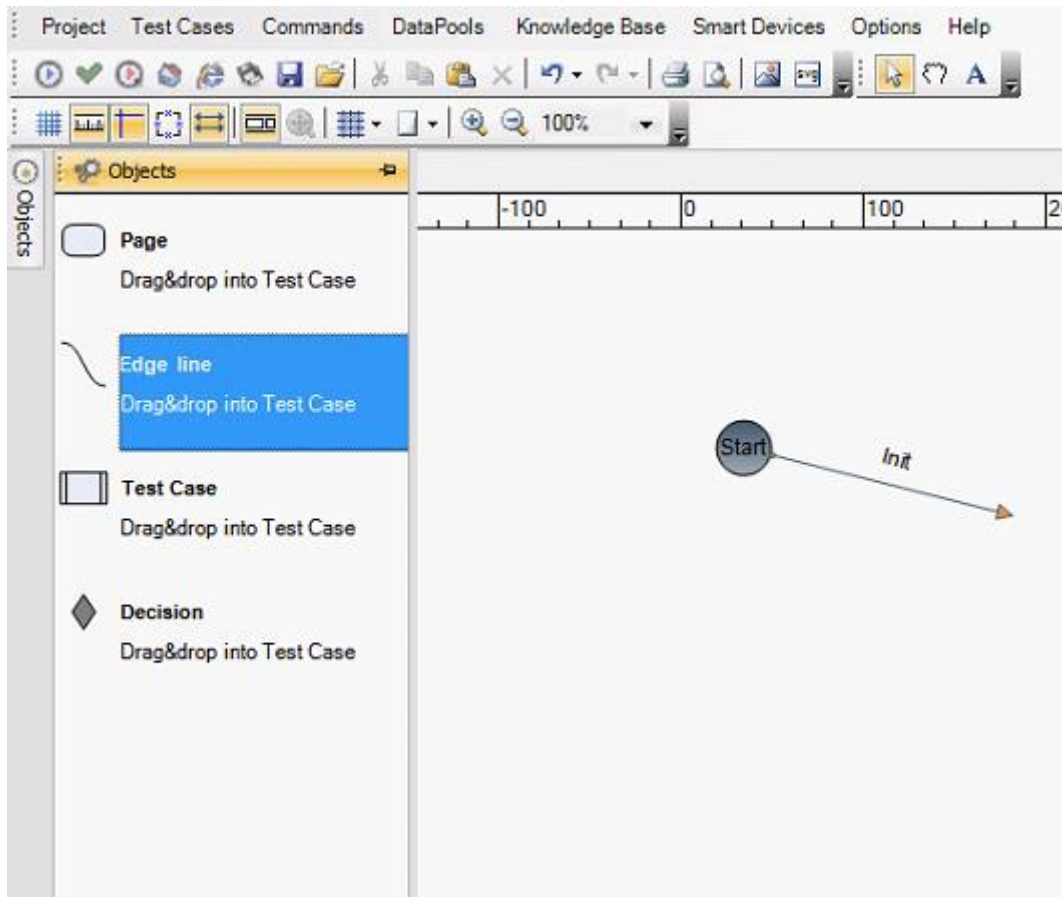
Dado que el caso de prueba tiene un datapool asociado, también se importará el datapool. **Realizar la misma acción con el caso de prueba BajaCliente (BajaCliente.xml).**

Revisar y validar los casos de prueba importados ejecutándolos.

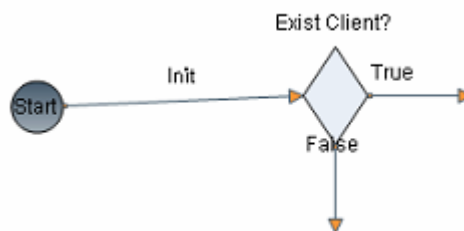
Ahora que tenemos los casos de prueba unitarios, construiremos el caso de prueba reutilizando los anteriores. Para poder expresar esto, debemos crear un nuevo caso de prueba haciendo clic con el botón derecho sobre el árbol de **Test Cases**, y seleccionar **Create New Test Case**. Al nuevo caso de prueba, darle el nombre **Add and Remove Client**



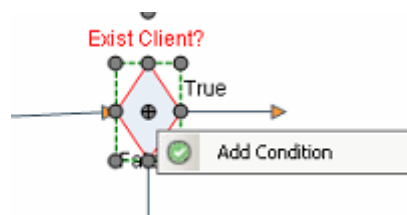
Desde la barra de herramientas **Objects**, en la izquierda de la pantalla, podemos arrastrar dentro del caso de prueba los posibles elementos que componen el flujo de prueba.



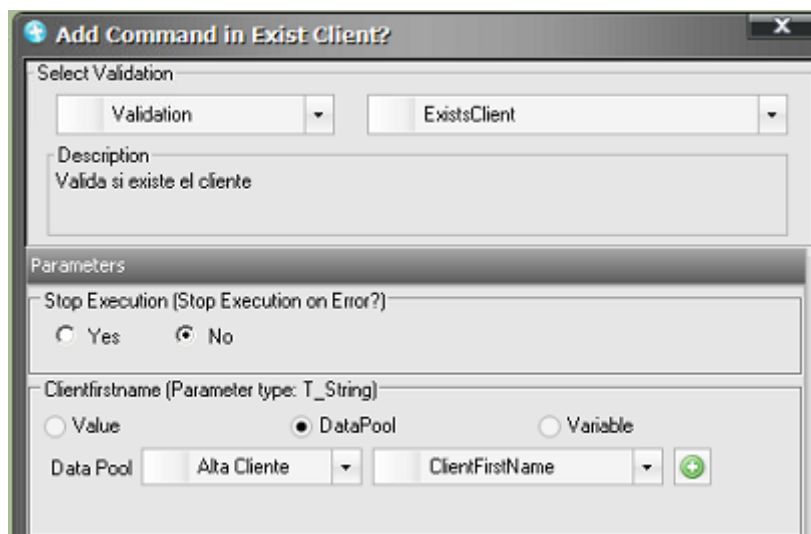
Arrastrar un nuevo elemento del tipo Decision hasta el interior del caso de prueba. **Luego unir la arista init** a la decisión agregada. A las decisiones se le pueden poner nombres intuitivos que ayuden a interpretar el caso de prueba, en este caso **hacer doble clic sobre la decisión** y ponerle el nombre **“Exist Client?”**



Ahora le debemos agregar a la Decision, la condición que hace que se siga un camino u otro. Para esto hacer **click con el botón derecho sobre la misma** y seleccionar **Add Condition**.



Una vez que se abre el dialogo de comandos, se debe seleccionar una de las validaciones disponibles, en este caso **seleccionar la validación ExistsClient** que se agregó hace unos instantes desde los Web Services.



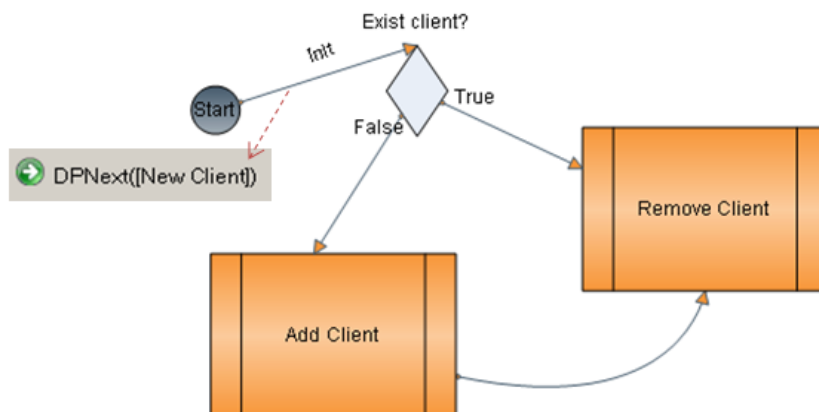
Es necesario indicarle el valor que debe utilizar para el parámetro ClientFirstName que recibe el Web Service. Para eso, **indique la columna ClientFirstName** del Datapool **Alta Cliente**. Luego **presionar Add Command** para confirmar la creación del comando.

Una vez definida la condición, incluir los casos de prueba insertados, **arrastrando desde la barra de herramientas de la izquierda un elemento Test Case**. Se abrirá una ventana para seleccionar el caso de prueba. **Elegir Alta Cliente**, y confirmar. Luego repetir el mismo proceso para agregar **Baja Cliente**.

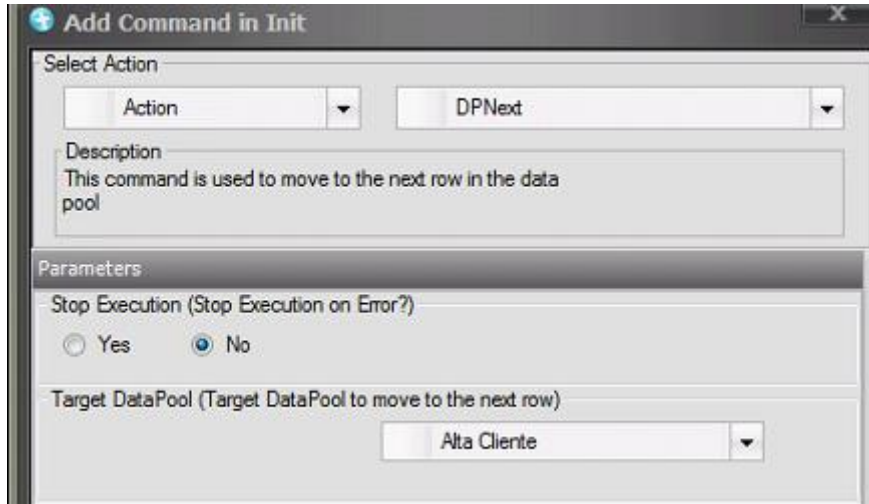
Una vez que están los casos de prueba y la decisión, **unir la arista False** de la decisión al caso de prueba **Alta Cliente** y la arista **True** al caso de prueba **Baja Cliente**.

Además, **agregar una arista que vaya desde Alta Cliente** hasta **Baja Cliente**, arrastrándola desde la barra de objetos.

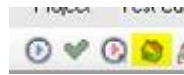
El caso de prueba resultante debería verse así:



Antes de terminar, es necesario agregar un comando DPNext en la arista Init. Esto hará que el Datapool se inicialice y quede listo para ser utilizado. Para esto, **hacer clic con el botón derecho sobre la arista Init y seleccionar Add Action**. En la ventana, **elegir la acción DPNext** en el combo, y más abajo **seleccionar el Datapool** que se quiere “mover” de fila, que en este caso es **Alta Cliente**.

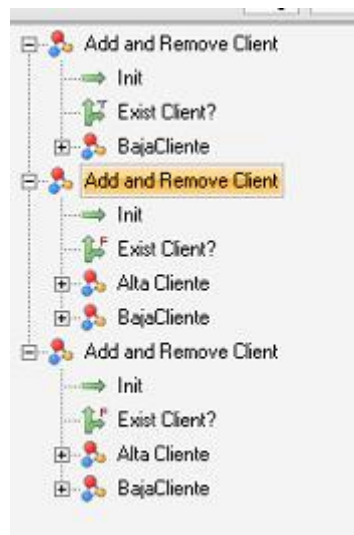


Listo, ha quedado finalizado el caso de prueba. Ahora, para testear el caso de prueba (sí, a los casos de prueba también hay que probarlos) es necesario dar de alta un cliente en la aplicación para que al ejecutarse el caso de prueba Add and Remove Client, el mismo tome otro flujo y vaya directamente a la Baja. Para abrir el navegador en la pantalla inicial de la aplicación, puede **hacer clic en el botón Home** desde GXtest Designer.



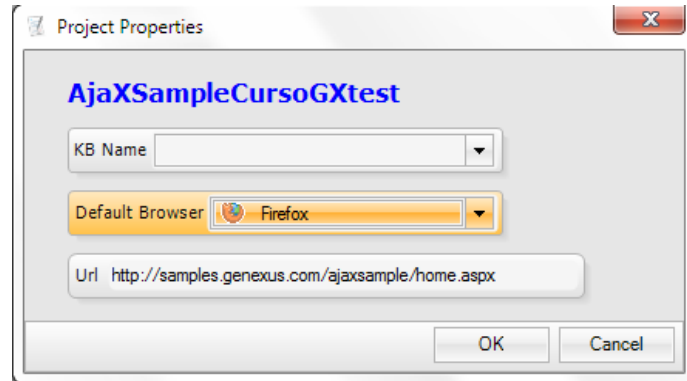
Una vez en la aplicación, ir a la pantalla de alta de clientes y agregar un nuevo cliente de nombre **Armando**.

Luego, volver a GXtest y ejecutar 3 veces el caso de prueba (desde el botón Run N times), y analizar los resultados para visualizar su correcta ejecución.



5. Ejecución en FireFox

Cambiaremos el navegador utilizado para ejecutar las pruebas. Para esto, abrir las propiedades de ejecución del proyecto: **botón derecho en el árbol de proyecto -> Properties.**



Modifique solamente el Default Browser y seleccione FireFox. Ahora ejecute nuevamente el caso de prueba anterior, podrá verificar que se abre una ventana de Firefox y allí se ejecuta el caso de prueba.

De esta manera es posible validar que el mismo caso de prueba, se ejecuta sobre los diferentes navegadores, o incluso diferentes versiones de GeneXus o plataformas sobre las que se genere la aplicación web. Esto le dará la enorme ventaja de poder validar en paralelo las múltiples variantes de sistema operativo/navegador/ambiente que antes debía probar de manera manual, una por una, consumiendo mucho tiempo y malgastando el tiempo del tester.

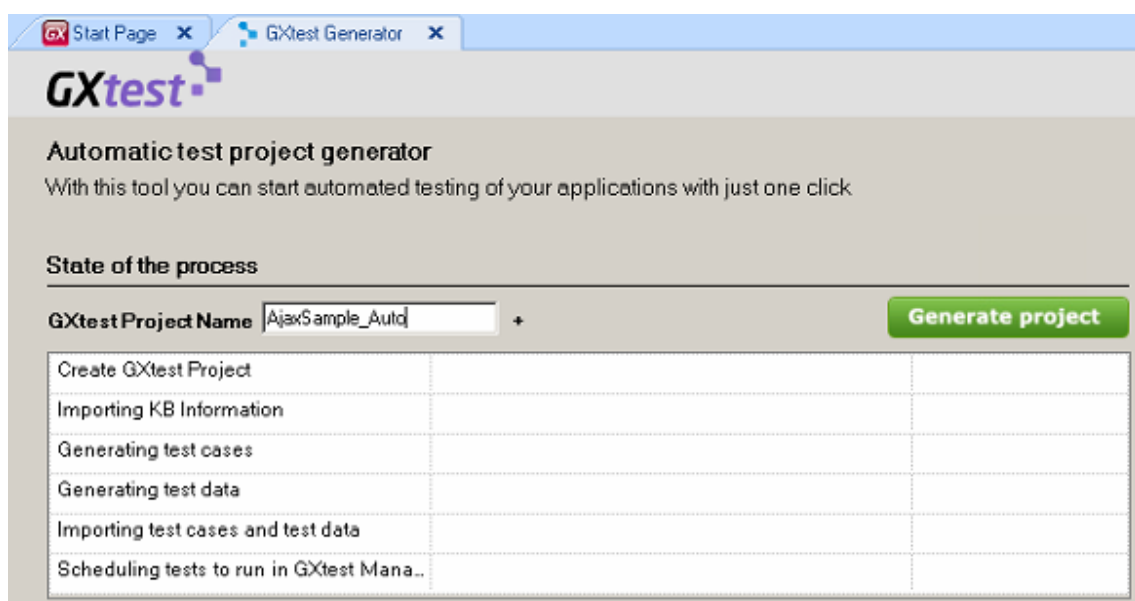
Ahora el tester puede dedicarse a automatizar nuevos casos de prueba o adaptar los existentes a las nuevas funcionalidades, mientras que las tareas tediosas y repetitivas las ejecuta GXtest.

6. One Click Startup: ¡Empiece rápido y fácil con 1 clic!

Ahora verá como GXtest lo ayuda a comenzar más rápido con la automatización de pruebas, creando y generando un proyecto con un juego de pruebas en forma automática a partir de la KB de nuestra aplicación. Esta funcionalidad se ejecuta desde GeneXus a través de la extensión de GXtest.

Para ello es necesario abrir GeneXus desde el ícono ubicado en el escritorio, y abrir la base de conocimiento asociada (AjaxSample).

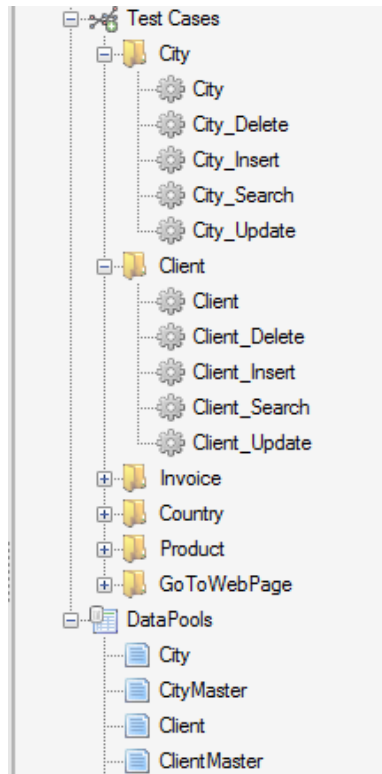
Para empezar a generar pruebas automáticamente, se debe ir al menú GXtest -> Autogenerate test cases. Esto abrirá una ventana de la extensión de GXtest para GeneXus, donde es posible iniciar el proceso desde el botón Generate Project.



Haga clic en dicho botón y observe como en la ventana se notifica paso a paso desde la creación del proyecto a la creación de los casos de prueba.

Una vez que notifica que ya fue creado e importado nuestros casos de prueba y el proceso se ha completado, puede verificar en GXtest que el proyecto fue creado e importado a nuestra herramienta de pruebas. Para eso, puede abrir GXtest Designer desde el botón "Add or Edit Test cases" que aparece en la esquina inferior izquierda de la pantalla.

Observe que se han creado a partir de nuestra aplicación (por ejemplo para AjaxSample) los siguientes casos de prueba y Datapools:



Compruebe, ejecutando un caso de prueba (por ejemplo Client, asociado a la transacción con el mismo nombre) cómo GXtest generó automáticamente a partir de la KB en GeneXus, las pruebas de Alta, Modificación, Baja y Consulta para dicha transacción, así como para cada uno de los objetos transaccionales de nuestra aplicación.

Simplemente abra el caso de prueba que desea ejecutar y presione F5.

Es importante recalcar, que la utilidad de la generación automática no es probar la aplicación completamente, sino que darle al tester una suite de pruebas con las cuales empezar a automatizar, para que pueda a partir de aquí mejorarlas, adaptarlas, o incluirlas en otras pruebas.

Conclusiones

Con esta práctica se pudo mostrar lo fácil que es generar pruebas automatizadas con GXtest y ejecutarlas con distintos juegos de datos.

Se pudo ver como diseñar un caso de prueba desde cero de forma “manual”, e incluir y componer casos de prueba simples en otros más complejos, utilizando decisiones en las cuales la ejecución pueda variar dependiendo del estado del sistema, o los datos utilizados en los Datapools.

Todo esto, lo pudo hacer sin tener que programar ni escribir ni una línea de código (como pasa con otras herramientas de automatización). Esta es una de las principales ventajas de GXtest, lo cual hace que pueda ser utilizada por especialistas del dominio, que generalmente no tienen conocimientos técnicos suficientes para programar pruebas automatizadas.

Por otro lado, con un solo clic desde GeneXus es posible generar pruebas automáticas que testean distintas transacciones y “Work With’s”, permitiendo aplicar técnicas de testing en las pruebas generadas:

- Encontrar bugs rápidamente.
- Reutilizar dichos casos de prueba para crear casos de prueba más complejos.
- Ejecutar pruebas de regresión luego de aplicar un nuevo Build de GeneXus o un cambio en la aplicación.

¿Cómo seguir?

Lo invitamos a que conozca otros componentes y funcionalidades de GXtest:

- GXtest Manager, para gestionar, agendar y ejecutar pruebas en distintos ambientes.
- GXtest Executor, para ejecutar en forma distribuida y desatendida sus suites de pruebas.
- GXtest para pruebas de performance, que le permite generar fácilmente scripts para pruebas de carga o performance sobre la aplicación.
- GXtest para pruebas de integración, que le permite validar y verificar componentes o sistemas externos a su aplicación.
- GXtest para pruebas de seguridad, que le permite generar fácilmente pruebas que verifiquen las restricciones de acceso y autorización a las diferentes pantallas de su aplicación.

A continuación encontrará información y links de utilidad sobre GXtest y Abstracta. En particular, en nuestra wiki encontrará documentación y tutoriales de interés de GXtest.

Lo invitamos a seguir descubriendo toda la suite de productos de GXtest, las opciones de capacitación, y si lo desea solicitar una versión completa de prueba por 30 días.

-  <http://genexus.com/gxtest/>
-  <http://www.abstracta.com.uy/>
-  <http://blog.abstracta.com.uy>
-  <http://gxtest.abstracta.com.uy/wiki>
-  <http://groups.google.com/group/gxtest>
-  <http://twitter.com/gxtest>
-  <http://www.youtube.com/user/AbstractaLabs>